Clemson University
TigerPrints

All Theses Theses

8-2014

Space-Capable Long and Thin Continuum Robotic Cable

Manas Milind Tonapi
Clemson University

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Part of the Engineering Commons

Recommended Citation
Tonapi, Manas Milind, "Space-Capable Long and Thin Continuum Robotic Cable" (2014). *All Theses*. 2018.
https://tigerprints.clemson.edu/all_theses/2018

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

Space-Capable Long and Thin Continuum Robotic Cable

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

by
Manas Milind Tonapi
August 2014

Accepted by:
Dr. Ian D. Walker, Committee Chair
Dr. Timothy C. Burg
Dr. Richard E. Groff

Abstract

Design of continuum robots, i.e. robots with continuous backbones, has been an active area of research in robotics for minimally invasive surgery, search and rescue, object manipulation, etc. Along the same lines, NASA developed “Tendril”, the first long and thin continuum robot of its kind, intended for in-space inspection applications.

The thesis starts with describing and discussing the key disadvantages of the current state of the art mechanical design of “Tendril” producing undesirable effects during operation. It then includes the design specifics of a novel concept for construction of a next generation long and thin, space-cable, multi-section, continuum cable-like robot, with a modified mechanical design for better performance. The new design possesses key features including controllable bending along its entire length, local compression and a compact actuation package. This new design is detailed in two versions. The first is a planar variant (suited for a 2D workspace), explaining the principle which allows the cable robot to achieve the above mentioned features. It is followed by a refined spatial version (suited for 3D workspace), where the functional characteristics are achieved within the desired aspect ratio of thin (less than 1 cm diameter) and relatively longer length (more than 100 cm) of the robotic cable.

A new forward kinematic model is then developed extending the established models for constant-curvature continuum robots, to account for the new design feature of controllable compression (in the hardware) and is validated by performing experiments with the robot in (2D) planar and (3D) spatial scenarios. This new model is found to be effective as a baseline to predict the performance of such a long and thin continuum “cable” robot.

Dedication

To my dad, Dr. Milind Tonapi for his continuous motivation to strive for excellence.

To my mom, Mrs. Madhumita Tonapi for her unending love and care for me.

To my sister, Ms. Sanchita Tonapi for being a cheerful companion.

To my grandmother, Mrs. Geeta Tonapi for her affection and support.

To all my teachers, I could not have done this without.

Last but not the least, to my fascination towards robotics.

Acknowledgments

I offer my deepest gratitude to my academic and research advisor Professor Dr. Ian D. Walker, for giving me this wonderful opportunity to work on a project[1] specifically sponsored by NASA under the National Robotics Initiative (NRI). His continuous encouragement, guidance at every juncture of the research and unceasing faith in me has been extremely helpful to reach the desired objective of the thesis. He was always available to answer my (technical as well as career related) questions whenever I needed a sense of right perspective, giving his time and knowledge liberally.

I would also like to thank (Professors) Dr Timothy C. Burg and Dr. Richard E. Groff for accepting to be my committee members and providing me with helpful suggestions for improving the thesis.

I must also thank my (Post-Doctoral Researcher) colleague Dr. Isuru S. Godage for his invaluable inputs during design and kinematic modeling stages of the robot, Mr. Tim Pruett from Machining and Technical Services for all the 3D printing services needed especially for the spacers of the robot, graduate student colleagues from architecture - Kyle Smith for laser cutting the acrylic bases used in the actuator package designs and Arash Soleimani for building the robot environment for video scenarios, summer undergraduate research students Zack Hewitt and Michael Wooten, along with Ms. Lillian Burns from Administrative Support staff at our ECE department specifically, for being patient and helpful during all the insane number of purchases liaised for the research, right from March 2013 until now. I would like to express my deepest appreciation for my friend and first year graduate student, Amith Vijaykumar, for his enormous help whenever needed, especially for setting up the actuation system for the robot and successfully completing the required experiments

[1] The research work presented in this thesis was supported in part by NASA under contract NNX12AM01G, and in part by the U.S. National Science Foundation under grant IIS-0904116.

and video scenarios in the lab during this academic year.

Finally, I thank my parents and sister who provided with the moral support, reminded me of perseverance and never say die attitude during this eventful journey as a graduate student.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The study of robotics and automation has seen considerable intrigue over the last few decades. Various kinds of robots have been designed and implemented. We can find them in action in the traditional sense of being able to do repetitive tasks or useful in an industrial environment which is potentially hazardous for humans to work (say near nuclear reactors or handling chemical wastes). Domestic or service robots are seen doing in-house cleaning and even assisting patients in certain super-specialty hospitals. Biomimetic robots is a relatively new sub-category of study, where behavior such as manipulation, locomotion or navigation of the robot, in general, is inspired from its counterpart in nature. The majority of biomimetic robots implement locomotion such as legged (like that of human, cheetah, cockroach, etc) or limbless locomotion (like that of snakes, worms, octopus, etc), climbing (like that of a gecko), jumping (like that of locusts), etc. Certain robots even imitate the morphology of humans (humanoids-human looking robots), ants (swarm robots) and even starfish (soft robots).

Rigid-link or traditional industrial robots have been commercially successful in manufacturing and general factory settings, with a highly structured and well-defined task and work-space. Outside industry, however, their utility is highly restricted due to their need for predetermined motion, limited manipulation and grasping abilities. A structure which is adaptable to its surrounding environment with a fluid like motion is possible by eliminating the hard vertebral links and replacing with a continuous and flexible backbone. Such robots are called as *continuum robots.*

Due to the absence of joints, continuum robots can perform terrain-adaptive motion [16] and full-body grasping [17] operation by means of their full-body compliant deformation. They have the ability to bend at any (selected) points, sometimes with additional extension/contraction/twisting. This gives a continuum robot a unique advantage over its rigid counterpart, enabling the robot to avoid obstacles, manipulate with adjustable stiffness over irregularly shaped objects and even adapt its body shape while maneuvering itself in an unknown and non-uniform environment.

Over the past few years various types of continuum robots have been designed. The three major design alternatives commonly used are tendon-based approaches, concentric-tube designs and locally actuated backbone designs [18]. Serially connected constant curvature sections is the underlying principle in almost all continuum robots [19].

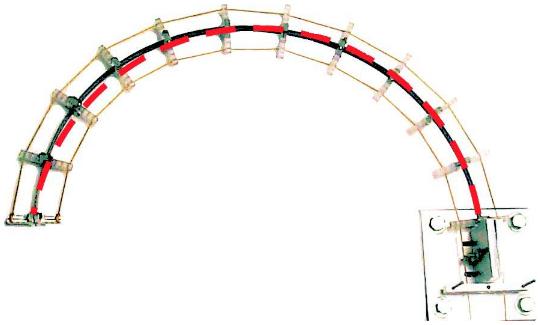

Figure 1.1: Continuum robot with constant curvature segments [1]

Minimally Invasive Surgery has been an active area of application especially for continuum concentric tube robots with small contour but restricted work environment [20–22]. Drawing inspiration, NASA developed “Tendril”, the first long and thin continuum robot of its kind, intended for in-space inspection applications [2] (see figure 1.2). Unfortunately, its operation was compromised due to major limitations in its mechanical (spring backbone elements based) construction. Hence, the need for a new and improvised design was perceived [18].

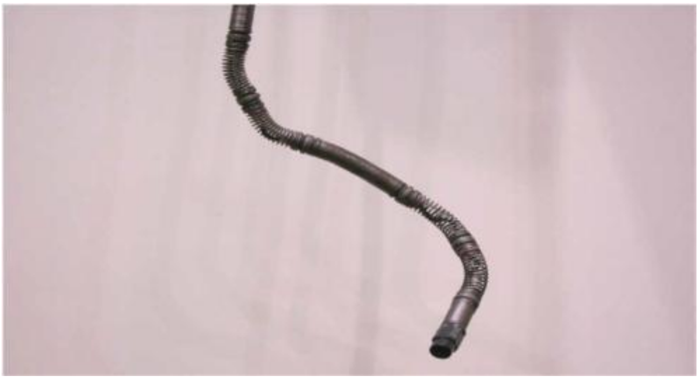

Figure 1.2: NASA's Tendril [2]

Such very long, cable like robots, intended by NASA for space applications such as inspection, remote exploration, and as a support structure, could also have potential terrestrial applications. These long continuum manipulators could be highly useful in areas of defense and security, teleoperation in hazardous environmental conditions, and even as an innovative option in traditional industrial automation [18].

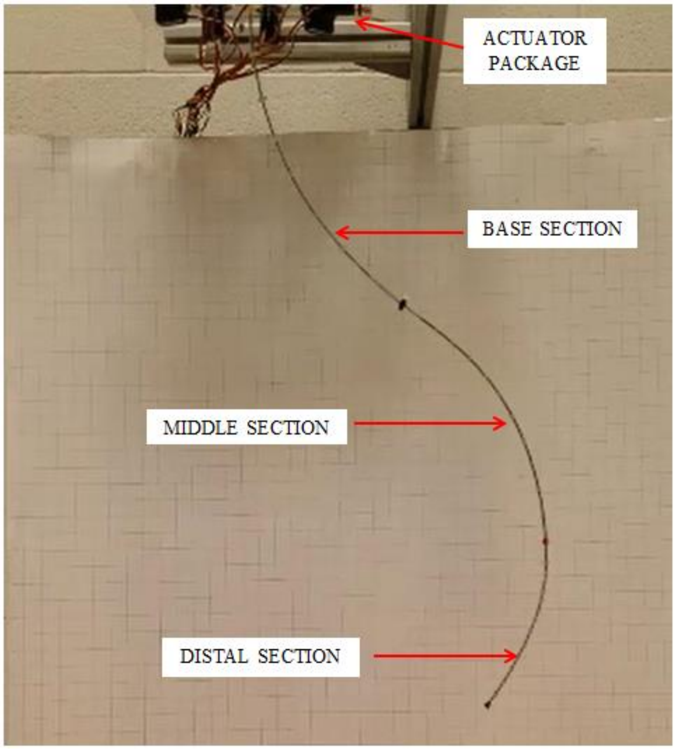

Figure 1.3: Long and slim continuum robotic cable

This thesis presents a novel design for constructing multi-section continuum robots (see figure 1.3) with a special focus on thin (less than 1 cm diameter) and relatively long length (more than 100 cm), along with its new and novel kinematic modeling and performance evaluation. The design reported in this thesis is particularly intended to be space-capable as may be perceived by NASA.

Different new alternative designs were evaluated based on an analysis and extension of the three core existing continuum robot design types [18]. Various types of continuum and continuum-style robots (viz., snake and hyper-redundant robots [1, 23–25]) have already been implemented. A few of these have certain aspects which might be applicable for development of a long thin robotic cable or rope, with suitability for employment in space, but simultaneously they have other undesired aspects preventing direct implementation of their current form. In Chapter 2, these robots are reviewed in detail.

The first generation long thin "Tendril" continuum manipulator was analyzed with respect to its mechanical design, modeling and control at Clemson under NASA-funded research [26]. "Tendril" produces undesired effects while in operation, such as buckling and torsion under load, as well as possessing the inability for any local extension or contraction and fixed bending backbone locations concentrated mainly at the tip. The new analysis presented in Chapter 3 discusses these key disadvantages due to the current mechanical design structure in detail.

A new and improved robotic cable design concept, illustrating the principle of operation, together with its planar and spatial prototypes are introduced in Chapter 4. The new spring-loaded design, which is still tendon-based like its precursor, has the favorable features of a concentric tube robot, but notably without a bulky actuator system. It overcomes most of the drawbacks of "Tendril".

Chapter 5 presents a new and novel procedure for forward kinematic modeling of the new design for operation in 2D and 3D workspace. The results from the performance evaluation of the proposed kinematics for the hardware designed are presented and discussed in Chapter 6.

Chapter 7 presents the conclusion summarizing the thesis and and provides suggestions for future work that could be done based on the lessons learned from the work in this therein.

Chapter 2

Literature Review

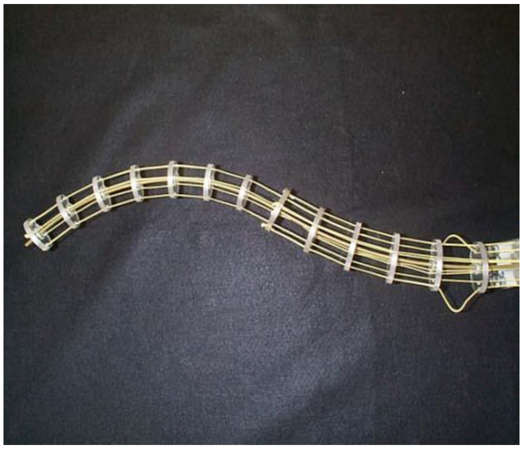

Figure 2.1: Tentacle robot with incompressible flexible backbone rod

The tentacle robot (see figure 2.1) developed previously at Clemson is a tendon based continuum robot with a flexible but incompressible backbone (garolite) rod [27]. It has spacers at fixed intervals which serve as guides for the actuating tendons. It is constructed such that each section will bend maintaining the curvature within that section approximately constant [1]. It has a slender backbone profile but lacks local extension or contraction. A modified version was developed with varying section lengths to achieve greater variety of shapes [28], but still lacked in the same area (local extension or contraction) as its predecessor. The underlying design cannot be confined to a compact form say by reeling it inside an actuator housing due to the stiff backbone rod.

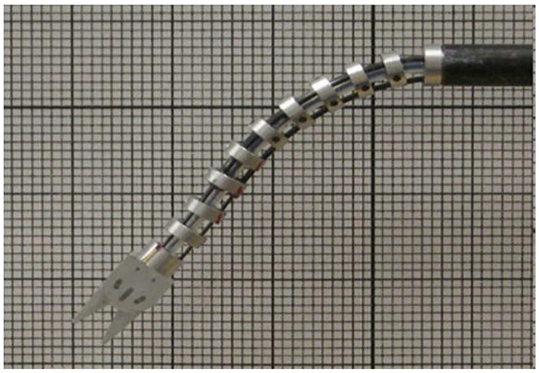

Figure 2.2: JHUs snake robot (Credit:JHU/Vanderbilt University)

A tendon-driven snake robot designed by John Hopkins University (JHU) for laryngeal surgery (see figure 2.2) has a thin profile of 4mm [29]. The robot consists of a base disk, an end disk, several spacer disks and four super-elastic tubes called the backbone tubes made from a shape memory alloy. However, it has significant complexity in degrees of freedom (DOF) and heating problems due to the use of shape memory alloy [30].

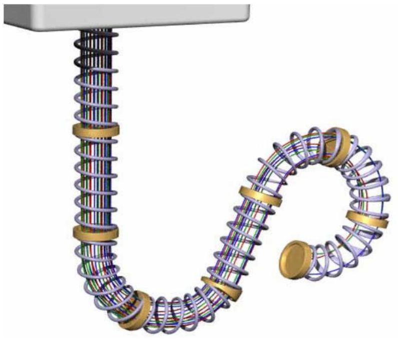

Figure 2.3: Hyper-redundant continuous robot [3]

The hyper-redundant continuous robot [3] is a 48 DOF (see figure 2.3) robot actuated by 24 motors. It is analogous to an elephant's trunk and similar to another robot with 8 DOF, developed by researchers earlier [31]. A spring is used as the backbone but with spacers in between and tendons to actuate bending. This gives the robot the ability to achieve a variety of shapes. However, both these robots lack local extension and contraction. The springy backbone in the former causes the same issues as seen in “Tendril”. Also, the higher DOF are at the cost of the large number of motors required which is undesirable for use in space applications.

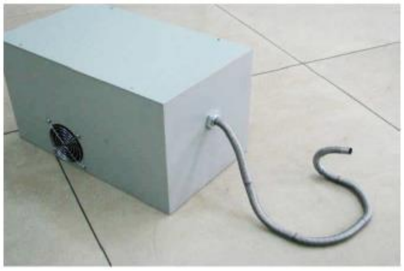

Figure 2.4: Snake-like continuum colonoscope [4]

The snake like continuum colonoscope [4] (see figure 2.4) is composed of a series of universal joints connected together by rivets with guides for carrying spring tubes. The drive cables are enclosed in the spring tubes and the entire colonoscope body is covered by an outer metallic mesh. The drawback, again, is the large number of actuators (20) needed for a relatively smaller length (60 cm) in spite of having a clean and relatively slim design.

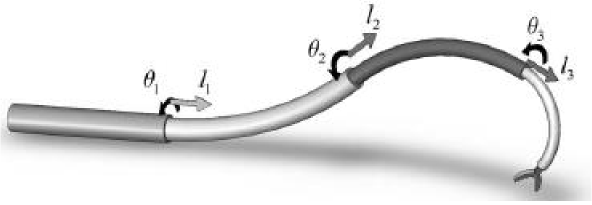

Figure 2.5: Concentric tube robot [5]

Continuum robots developed with a concentric tube structure [5] consist of pre-curved tubes sliding within each other (see figure 2.5). They are commonly used in Minimally Invasive Surgery. Some have restricted curving due to the pre-bent nature of the concentric tube design but can inherently achieve local extension and contraction as well as possessing a thin contour.

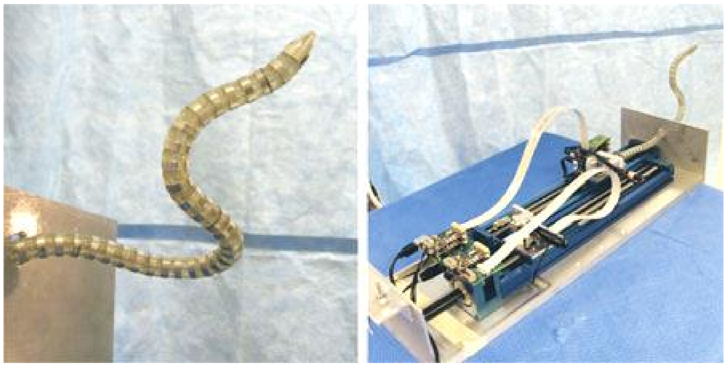

Figure 2.6: HARP snake robot (Credit: Howie Choset/CMU)

The Highly articulated Robotic Probe (HARP) introduced by Carnegie Mellon University (see figure 2.6) [30] and the multi-turn catheter by Massachusetts Institute Technology (MIT) [32] are snake robots, working on variations of the principle implemented in the ShapeLock® patent of USGI Medical [33]. They are basically concentric tubes (or one above the other adjacent tubes by MIT) having rigid cylindrical links (or modular designed beads in case of MIT) connected by a type of spherical joint (or spherical bearing surfaces by MIT). The links (or beads) are strung together by cables, three for the outer tube and one for the inner tube (or three for each of the tandem tube by MIT).

Both use a leader-follower motion mechanism. In case of the HARP, only the outer tube is bent via cables. The inner and outer tubes are alternately made rigid/limp by pulling/releasing all of their respective cables. When all the cables are pulled, friction between the links and the spherical joints make the concerned tube rigid. When the outer tube is limp it extends/bends into environment. It is then made rigid. When the inner tube is limp it catches up with the shape "locked" by the outer tube. The process is then repeated.

In case of the MIT design, each of the (adjacent) tubes can bend/extend while the other holds the shape (again by friction in between the specially designed beads). Thus, both snake robots are rigid proximally but compliant distally. Their ability to achieve multiple bends about their entire length is promising. Both however, have highly complicated mechanical designs and require a large size actuator package. The former is rather bulky and the latter, especially, needs a contact surface to enable its motion. All this is unfavorable for a robot operating in space.

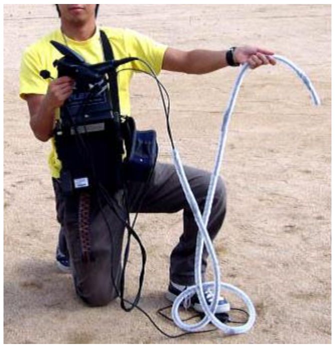

Figure 2.7: Active scope camera [6]

The Active scope camera developed (see figure 2.7), is to our knowledge, the longest type of continuum styled snake robot (8 meter) [6] to date. But its motion is only possible if its ciliary based vibration drive mechanism has a contact surface to push against. This is not promising from the point of view of in space inspection, where minimal contact is desired. Also, it lacks local backbone extension and contraction.

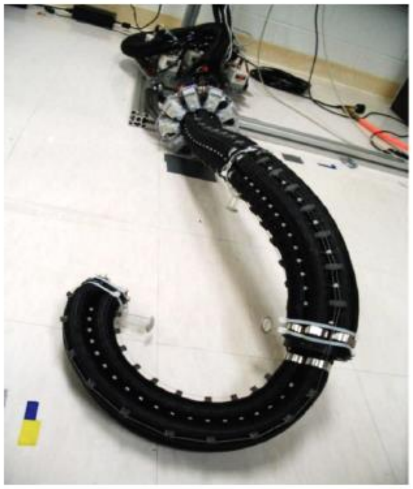

Figure 2.8: OCTARM (example of locally actuated backbone)

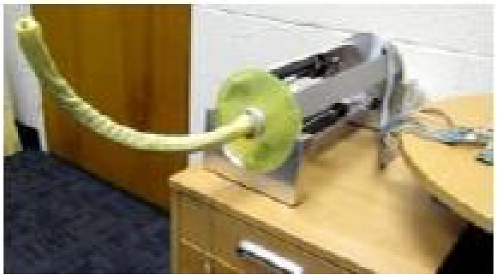

Figure 2.9: Layer Jamming snake (example of variable stiffness design) [7]

Continuum robots that work on the principle of locally actuated backbone design (see figure 2.8) and/or the variable stiffness design mostly use pneumatics (see figure 2.9) [7, 34–36] or hydraulics [37]. These have the advantage of providing both local bending and extension while in operation, but the complicated routing or valving, the requirement for fluid compression and multiple independent pressure regulation systems [19] and the resulting larger size, makes this design approach an unattractive option as a robotic cable system.

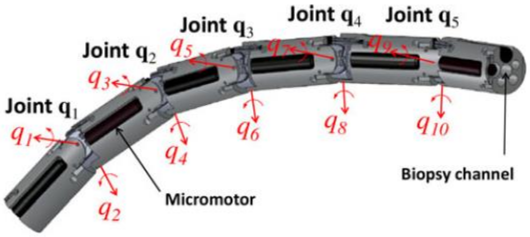

Figure 2.10: i-Snake [8]

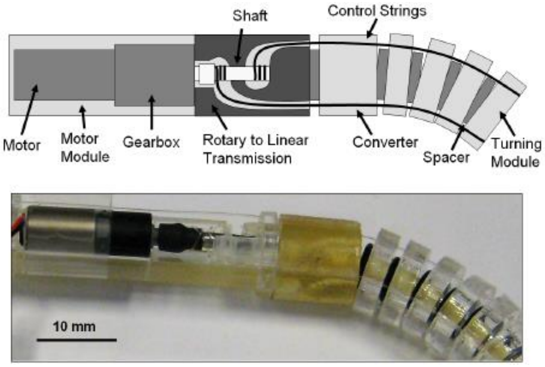

Figure 2.11: Disposable Endoscope [9]

The i-Snake (see figure 2.10) [8, 38] uses 5 universal joints having a micromotor per joint to enable bending. The disposable endoscope (see figure 2.11) incorporates motors inside the backbone to enable better steering [9] and utilizes variable stiffness (shape-lock) principle. However, this compromises the size, requires more motors for longer length and has high complexity from the design point of view.

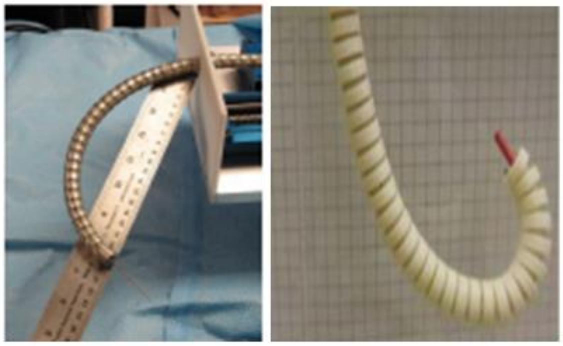

Figure 2.12: Snake-Continuum hybrid robots [10, 11]

Hybrid robots involving snake and continuum elements [10, 11] have been developed to eliminate the disadvantages each type has individually. See figure 2.12. The image on the left is the HARP design with nitinol inserted in between inner and outer probe through a channel to provide more dexterity at tip with the remaining robot being relatively stiffer. The right image is that of a hybrid robot which possesses spherical joints with an elastic backbone and tendons to control bending. However, both these structures rely strongly on the snake counterpart (i.e. segmented backbone) with respect to the mechanical design. This increases their structural complexity and decreases their ability to maintain a slender profile.

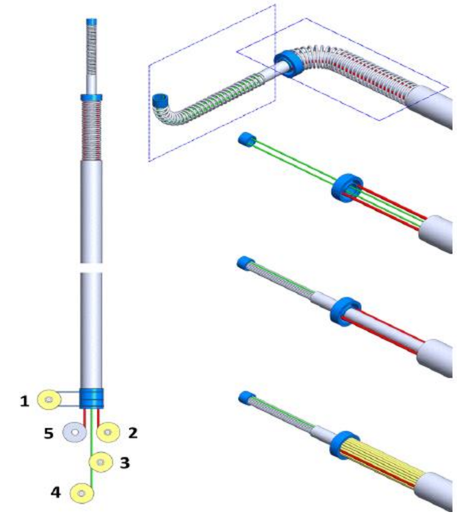

Figure 2.13: Flexible imaging probe for RAVEN [12]

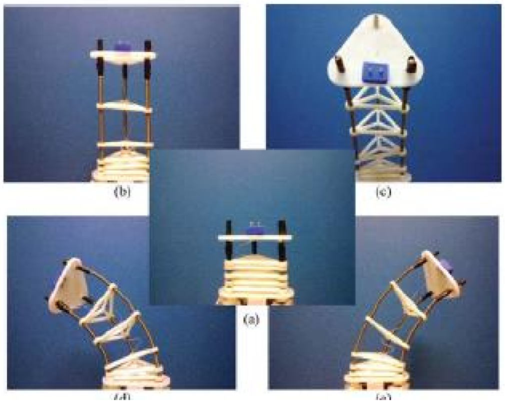

Figure 2.14: Treebot [13] (copyright John Wiley and Sons, reprinted with permission)

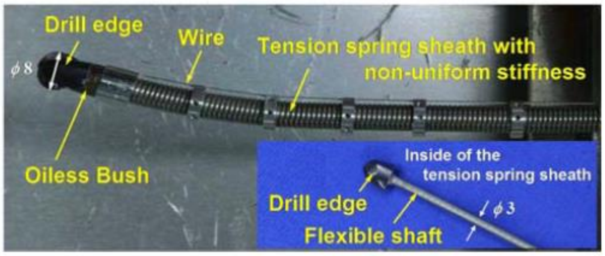

Figure 2.15: Steerable drill [14]

Springs (compression and/or extension) have been used in several designs of continuum robots. NASA's "Tendril" used a spring backbone to make it light and thin [2]. A compliant probe having a camera was integrated to RAVEN (see figure 2.13), a type of surgical robot [12]. RAVEN served as a flexible tool, again for minimally invasive surgery, comprising of compression springs for the distal section, and a spring enclosing a flexible tube for the proximal section. These springs were mainly used to achieve bending with the help of cables running through them on the inner side, and thus utilizing their capacity to compress and extend for turning. A tool adapter slides along a linear insertion axis providing translation of the probe. The "Treebot" (see figure 2.14) has a continuum section formed by triangular shaped spacers. Mechanical spring rods are used as part of a rack and pinion arrangement to achieve contraction [13]. The steerable drill (see figure 2.15) developed for ACL (Anterior Cruciate Ligament) reconstruction uses a series of variable stiffness tension springs as the backbone. It is wire controlled but has limited bending curvature [14].

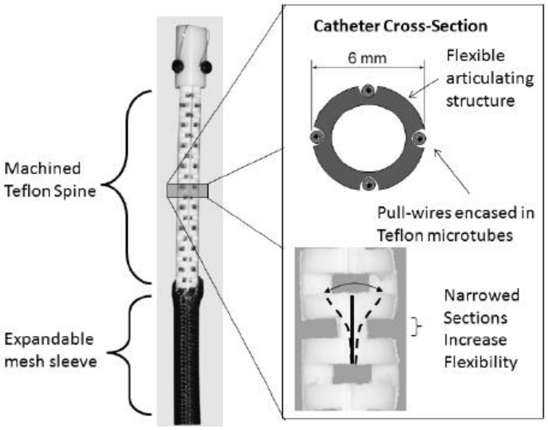
Catheter Cross-Section
6 mm
Flexible articulating structure
Machined Teflon Spine
Pull-wires encased in Teflon microtubes
Narrowed Sections Increase Flexibility
Expandable mesh sleeve

Figure 2.16: Robotic catheter [15]

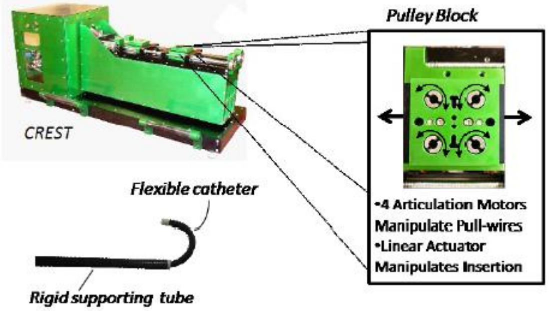
Pulley Block
CREST
Flexible catheter
Rigid supporting tube
•4 Articulation Motors Manipulate Pull-wires
•Linear Actuator Manipulates Insertion

Figure 2.17: Electromechanical sytem and flexible catheter [15]

Another catheter styled robotic system (see figure 2.16) merges the tendon based approach with a concentric tube design permitting extension as well as bending. It has a teflon spine being articulated by tendons, inside an expandable mesh sleeve. However, it is relatively short in length and larger in width. Also, the motion of the robot is only made possible with a heavy servo apparatus based testbed (see figure 2.17) [15].

The above review suggests that for a new space-robot design which is long and thin, locally actuated design approach is to be avoided due to sizing constraints. Design strategies used to build snake robots are mechanically complicated and result in a relatively heavy structure. The concentric and tendon-based designs are most suitable due to their simple and relatively light-weight hardware. However, the bulky actuation system required by almost all of the pure concentric-tube design approach is also undesirable. Thus, an optimal approach may be to have a tendon actuated design which has a concentric style arrangement of backbones. This will facilitate incorporation of the desired local compression/extension feature lacking in pure tendon-based designs. In addition, springs may also be used to provide a non-fluidic but effective local compression/extension feature without a dedicated linear actuation. A novel design concept, using these findings and eliminating the shortcomings in the construction of the current state of the art in spring-tendon designs described in Chapter 3, is explained in detail in Chapter 4

Chapter 3

Tendril Experiments

3.1 Hardware and Software Description

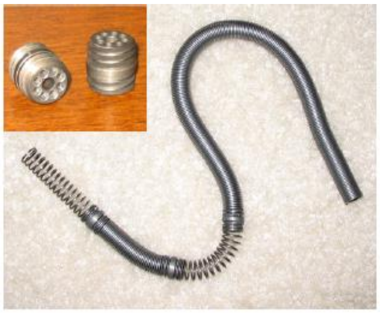

Figure 3.1: Tendril body springs connected by threaded links (inset) [2]

NASA's "Tendril" is a tendon-based design. It comprises of a series of compression and extension springs that are interconnected (see figure 3.1) using threaded links [2]. "Tendril" has only two actively controlled bending sections. The compression springs form the two short actuated sections at the distal end of the backbone and tension springs provide passive bending in the remainder of the backbone. The tendons actuating the top section are offset from the bottom section by 45° in counter-clockwise direction.

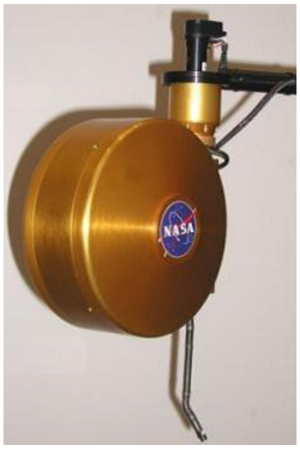

Figure 3.2: Tendril body housing (NASA) [2]

NASA's original version of "Tendril" has a body housing (see figure 3.2) with embedded actuators which can reel the backbone out (extend or retract its entire length) and rotate the entire limb. However, it lacks the ability to control any local extension or contraction. The actuator assembly subsequently developed in Clemson [26] is much simpler and designed for verifying the core features of "Tendril". This assembly is described next.

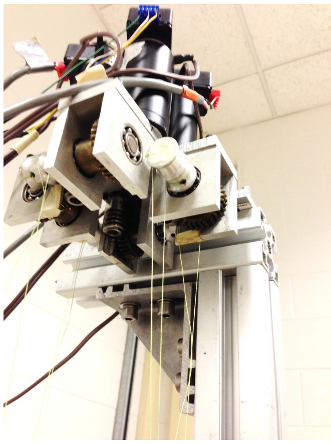

Figure 3.3: Motor and pulley assembly used in the lab

Two antagonistic pairs of tendons are attached to each section and terminated at a pulley, whose motion is achieved (see figure 3.3) by DC motors. Thus, there are four controllable DOFs. The tendons run along the entire length of the body terminating at the distal end of each section. The motors have encoders connected to them to provide position feedback. A second version of the body was constructed at Clemson from the spare parts of the original "Tendril" for testing and evaluating its capabilities. Linear amplifiers are used to provide the required power signals to the motors. The control system uses the interface Qmotor [39] to directly control the motors. A simple closed loop proportional control algorithm with an error tolerance was implemented.

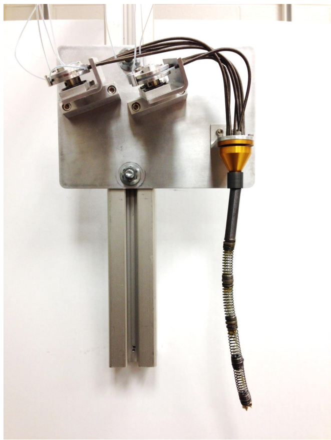

Figure 3.4: Tendril (truncated for experiments)

For the purpose of experiments the Tendril was truncated (part of the proximal backbone passive structure was removed), since some of the problematic issues identified previously [26] were due to the mechanical structure (see figure 3.4) of the actively controllable sections.

3.2 Methods and Observations

The major problem which "Tendril" exhibits is the strong coupling between the sections. When the section at the end of the robot is moved, it results in misalignment of the proximal sections. This is mainly due to the relative stiffness of different sections.

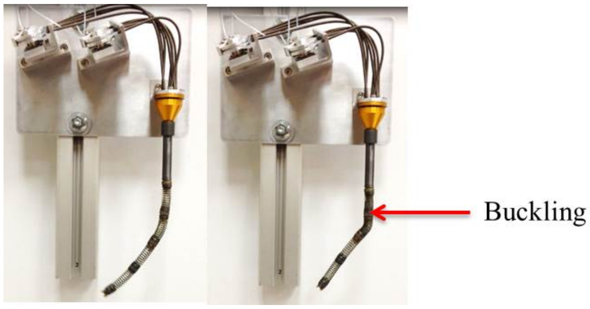

Figure 3.5: Joint buckling due to coupling of springs

When the tendon connected to the end section is pulled to achieve bending, it causes all the earlier sections (see figure 3.5) to compress/buckle (in this case only one as it is truncated to two sections).

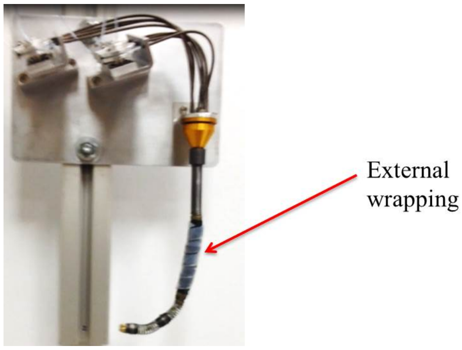

Figure 3.6: External support to prevent buckling

One potential solution to this problem is to prevent the buckling of springs by using an external wrapping (see figure 3.6) on the proximal section. A spirally cut simple cable-wrap was used for this purpose. However, this only serves to restrict its bending capability and also thicken the backbone, an undesirable feature.

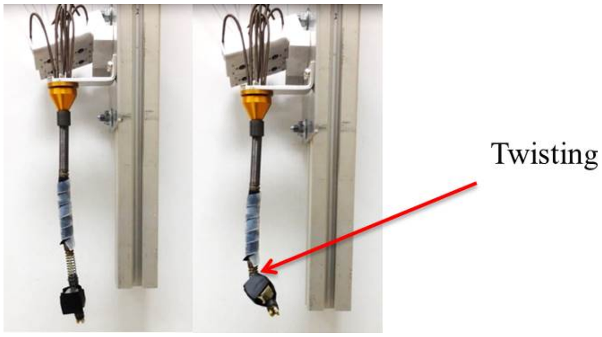

Figure 3.7: Twisting of spring due to gravity/load

Another problem exhibited by "Tendril" is twisting (see figure 3.7) of the joint springs themselves under loading. This effect is accentuated due to gravity or any load connected to the robot.

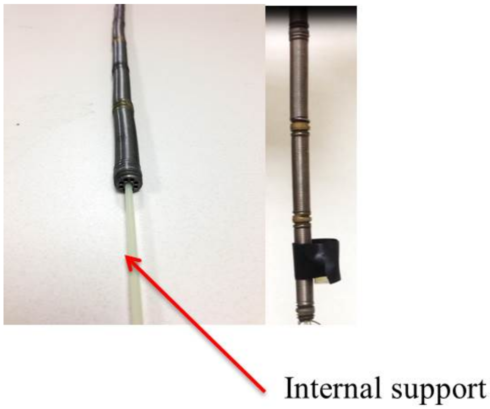

Figure 3.8: Internal support to prevent torsion

This issue can be reduced using quick-fix provision for an internal support like a bendable but stiff rod (see figure 3.8). A garolite rod was used for this purpose successfully. However this was at the cost of significantly reducing the bending ability of the backbone.

The current arrangement of (one) motor and corresponding pulley controlling two tendons causes slack in one of them. This results in the need to remove the slack by drawing in the appropriate tendon for some time before beginning the operation. Each tendon could be controlled by a single motor to prevent this issue.

Gravity causes the robot to sag a little even when it is in a vertical configuration. This effect will be more prominent if it is intended to be used in a horizontal configuration for terrestrial applications. “Tendril” is ideally supposed to follow the actively bending sections. But since the main body is only composed of passive tension springs, it is largely affected by gravity and exhibits flopping or sagging behavior.

Thus, in conclusion, “Tendril”, while possessing a compact body and actuator package (in NASA’s original), is hard to control due to the inherent uncontrollable compressibility of the backbone, torsion of joints and inability to locally extend or contract. Also, most of its length is unactuated. Kinematic model-based control has been attempted to improve its performance to some extent [26], but the undesirable effects are too strong to be used in useful applications. All the above shortcomings emphasize the need for a significantly improved structural design dedicated for long and thin robots. A new approach aimed at this need is described in Chapter 4.

Chapter 4

Cable Robot Design Concept and its Prototypes

In this chapter, we introduce a new design concept for long thin space-capable robot cables [40]. We discuss the design concept and its potential via simple prototypes. The overall performance goal specifications are given below.

4.1 Performance goals

1. Clean and simple mechanical design.
2. Thin profile but relatively long length (1:100 diameter/ length ratio).
3. Controllable bending about the entire cable length.
4. Local controllable compression and extension.
5. Absence of large and heavy actuation mechanism.

"Tendril" utilized springs to ensure a thin profile but ended up facing issues such as uncontrollable buckling of backbone elements, twisting of sections and inability to compress/extend locally [2], as discussed in Chapter 3. Since pneumatics/hydraulics which provide contraction/extension easily are to be avoided due to sizing constraints, springs are considered essential to achieve the same (non-fluidic) feature. The inspiration for the robotic "cable" was a modified version (see figure 4.1) of the tentacle robot [28] with varying section lengths. In that design tension springs were used in between the spacers, which were in turn attached to the flexible but incompressible backbone to achieve limited changes in section length. The new "cable" design, introduced in this thesis, attempts to utilize compression springs in a similar manner so as to convert the disadvantage of uncontrollable buckling into a desired feature of local controllable contraction.

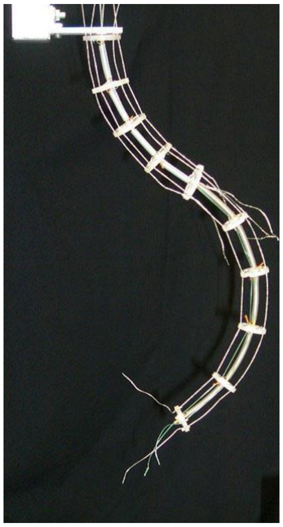

Figure 4.1: Modifed tentacle robot

The underlying concept for the robotic "cable" is to combine the most favorable properties of concentric tube and spring-loaded, tendon driven backbone designs. The core backbone of the design is made of concentric tubes. However, in extending and contracting the tubes relative to each other, instead of a bulky mechanism using linear actuators, a spring-loaded tendon based approach is adopted. The tubes "float" in and out of one another (telescopic arrangement) with springs along the backbone providing tunable resistance. Tendons are used to pull against the springs. Thus, the design concept provides both local contraction/extension and bending with a compact actuator package.

4.2 Initial prototype design

To evaluate the potential of the concept a simple prototype (see figure 4.2) was constructed.

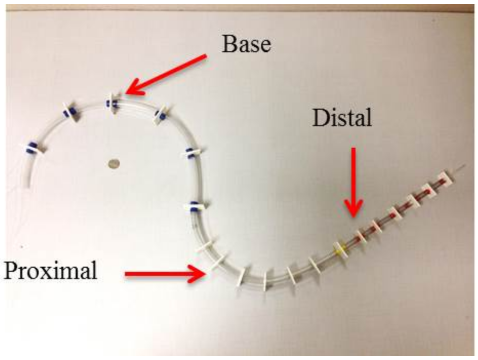

Figure 4.2: “Cable” initial prototype (alongside a US nickel)

The design features three flexible and incompressible backbone sections (see figure 4.2). PETG (Polyethylene Terephtalate Glycol-modified) was selected as the material of all three backbone sections for the initial prototype since it is readily available, bendable and machinable. The distal end is a rod with the proximal and base sections being tubes. These are arranged in a telescopic manner (see figure 4.3) as in traditional concentric tube robots [20]. The section diameters are chosen such that the distal backbone can slide freely inside the proximal adjacent one and that in turn can slide freely inside the base section.

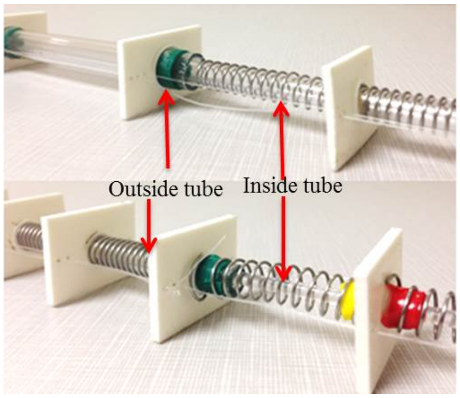

Figure 4.3: Concentric arrangement of backbone sections

The dimensions of the tubes are as specified in Table 4.1

"Cable" section	***Outer Diameter (mm)***	***Inner Diameter (mm)***	***Free Length (mm)***	***Length inside next section (mm)***
Distal	3.5	NA	450	150
Proximal	8	6	450	150
Base	12	10	600	NA

Table 4.1: Backbone dimensions for cable's (initial) prototype

3D printed spacers are used as tendon guides. They have a central circular cavity to allow the backbone to pass through them and tendons pass through peripheral circular holes. The base section has equidistant spacers fixed permanently on it. The proximal and distal sections only have a single spacer fixed permanently at the tip. The remaining spacers for both these sections can slide freely along their respective backbone sections. ABS (Acrylonitrile Butadiene Styrene) plastic was used as the material for the spacer since it has a high durability. The dimensions of the spacers are as specified in Table 4.2.

"Cable" spacers	***Spacer length (mm)***	***Spacer breadth (mm)***	***Central cavity radius (mm)***	***Tendon hole diameter (mm)***
Distal	40	25	7	1
Proximal	34	25	4.5	1
Base	28	25	2	1

Table 4.2: Spacer dimensions for for cable's (initial) prototype

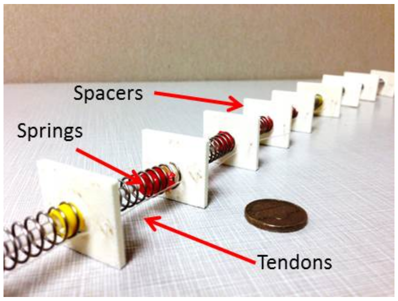

Figure 4.4: Spacers and spring arrangement (alongside a US nickel)

Compression springs (see figure 4.4) are used to hold the floating spacers together at a fixed distance with respect to each other and also provide the necessary controllable variation in the length of the backbone through active compression. All springs have the same dimensions i.e. outer diameter of 9.53 mm, inner diameter of 7 mm, wire diameter of 0.8 mm and compression rate of 0.21 lbs/in. A fishing line of 0.07 cm diameter serves as the material for tendons. There is a pair of antagonistic tendons for each section.

4.3 Principle of operation

The resulting prototype is a three section planar continuum robot. Tendon actuation produces bending in the plane and in the corresponding direction as the actuated tendon. Each section could be bent in two dimensions with the help of two tendons (2 DOF). The above design has its resulting backbone (see figure 4.5) approximating a series of connected set of constant curvature sections [19]. Hence, a constant curvature model may be assumed for modeling and analysis.

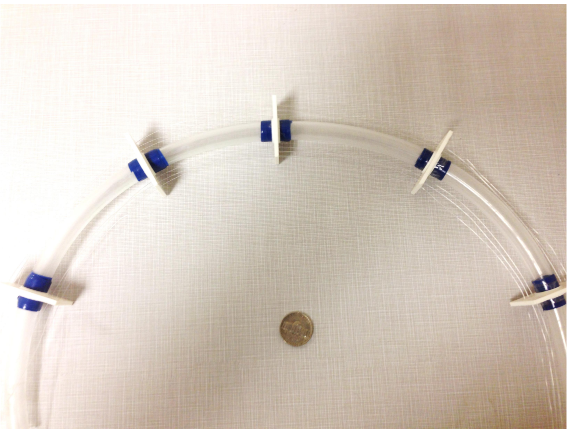

Figure 4.5: Constant curvature (alongside a US nickel)

Also since the backbones are concentric and telescoping in nature, each can achieve controlled linear motion with respect to each other. In this case compression springs are used to support the linear motion (see figure 4.6). The tendons pull the distal backbone section inside the middle and that in turn inside the base compressing each spring to its solid length. This is counteracted by the compression springs which, on releasing the tendons help to return the backbones to their original position by reassuming their relaxed state.

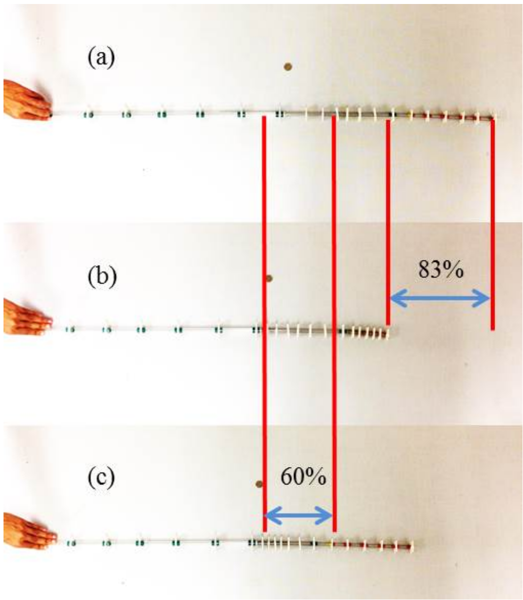

Figure 4.6: “Cable” prototype (a) Relaxed state (b) Distal section compressed (c) Middle section compressed (alongside a US nickel)

Thus, the key novelty of our design provides for limited translational motion without requiring any dedicated linear actuators. Thus, the “cable” prototype has 6 DOF in total. This design is also able to achieve a significant amount of bending (see figure 4.7). A hook shaped bend is also possible.

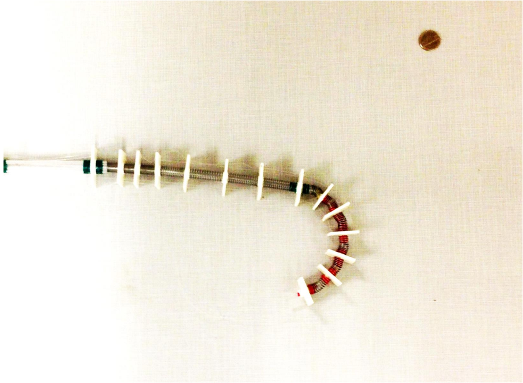

Figure 4.7: Maximum bending for distal section (alongside a US nickel)

There is some amount of coupling during compression and between the distal and proximal section, which is seen in figures 4.6 and 4.7. Coupling is also seen between the distal and proximal as well as proximal and the base section during bending. This can be sufficiently compensated for at the design phase using proper dimensions of backbones and choosing the right stiffness for springs for each section. However, the prototype is able to achieve almost 83.3% compression for the distal section (with respect to its normal length). The maximum compression achieved and bending possible in this type of prototype of the “cable” for each section, is detailed in Table 4.3.

“Cable” section	***Bending radius (mm)***	***Free length (mm)***	***Maximum compression (%)***

Distal	70	300	60(uncoupled)
			83.3(coupled)

Proximal	120	300	60(uncoupled)
Base	220	600	NA

Table 4.3: Bending and compression specifics for cable’s (initial) prototype

If the cumulative maximum compression with respect to the entire structure (length being 120 cm) is considered, it was empirically found to be 64.16% (compressed by 43 cm).

Following the promising characteristics of the first prototype of the new design, a second prototype was constructed. This prototype is closer to the desired scale (small diameter) of a space-capable robot cable and also driven by a compact actuator package. Details of the prototype are presented in the following sections.

4.4 Architecture of the First Long and Thin Prototype

The second prototype robotic cable (see figure 4.8) is a 139 cm long tendon-based three section continuum arm. About 57% of its length is 0.7 cm in diameter and the remainder is 1.4 cm in diameter. All three sections use nitinol as the main backbone element. Nitinol is an alloy composed of nickel and titanium and was found to be the most suitable material for the backbone for developing slim but long continuum cables. Nitinol can be obtained in various forms including wires, tubes and sheets with profile ranging in millimeters and length in feet. Nitinol’s property of super elasticity helps it to deform easily and regain its initial shape again. It’s oxide surface provides low frictional resistance, ensuring smooth sliding of the tubes relative to each other. It is light in weight and has good tensile strength. Such nitinol tubes can be easily bent and coiled into a reel-like mechanism simplifying its storage, like the actuator package for “Tendril” [2]. The distal section is a rod while the middle and base sections are tubes. The dimensions of the three sections are so chosen that they can be arranged in a concentric telescoping fashion, with the distal rod sliding freely inside the middle tube and the middle tube in turn sliding freely in the base tube. The dimensions of the tubes are specified in Table 4.4.

Robotic “cable” section	***Outer Diameter (cm)***	***Inner Diameter (cm)***	***Free Length (cm)***	***Length inside next section (cm)***
Distal	0.1	NA	34	15
Middle	0.17	0.13	45	7.5
Base	0.21	0.19	60	NA

Table 4.4: Nitinol backbone dimensions

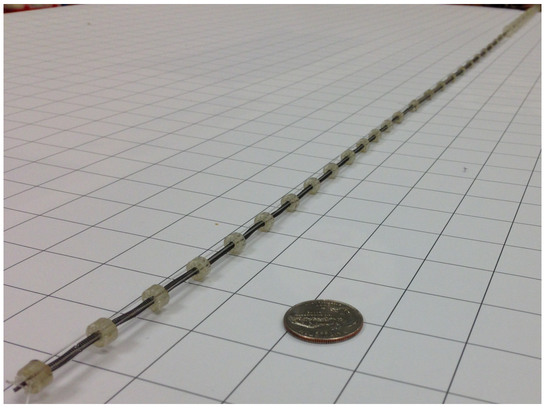

Figure 4.8: Robotic cable prototype (alongside a US quarter)

Small plastic spacers, which are 3D printed, are used as the tendon guides. The material used for the plastic is PolyJet photopolymer. They are circular in profile with a central cavity to accommodate the nitinol backbone elements. Each has three tendon guide holes spaced at 120° along the periphery of the spacer. The tendon guide hole set for each backbone element is thus spaced at 40° with respect to the adjacent section set. The dimensions of the spacers are specified in Table 4.5.

Robotic “cable” spacers	***Spacer diameter (cm)***	***Spacer width (cm)***	***Central cavity diameter (cm)***	***Tendon hole diameter (cm)***
Distal	0.7	2.5	0.12	0.1
Middle	0.7	2.5	0.2	0.1
Base	1.4	2.5	0.25	0.1

Table 4.5: Spacer dimensions

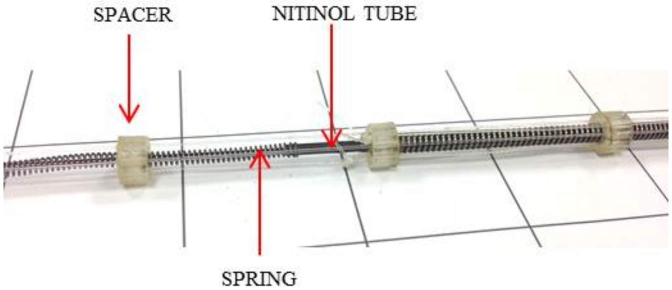

Figure 4.9: Spring supported concentric arrangement of continuum sections

Compression springs (in their relaxed state) hold the spacers in their place at a fixed distance and provide for the active local compression of their telescoping backbones as shown in figure 4.9. This translation is achieved using motor driven tendons, avoiding any additional or dedicated linear actuation mechanism. The material used for the springs is music wire. The spring parameters are as in Table 4.6. In order to minimize coupling between sections, it was ensured that the springs for the middle section have double the stiffness constant in comparison with the ones used for the distal section.

Nylon fishing lines of 0.15 cm diameter with 30 lb load carrying capacity serve as tendons for this cable robot.

Robotic“cable” section	***Outer Diameter (cm)***	***Inner Diameter (cm)***	***Spring rate (lb/in)***	***Length (mm)***
Distal	0.224	0.163	2.3	2.54
Middle	0.3	0.188	11.60	3.81

Table 4.6: Spring Parameters

4.5 Functioning of the Robotic Cable

The constructed robotic cable consists of three backbone elements, which can be approximated as a series of constant curvature sections. The actuation caused by pulling on a specific tendon produces bending in a plane containing the tendon and corresponding direction. TThe three tendons can effect bending in two dimensions and extension/contraction (limited by springs) i.e. 3 actuated DOF are provided per section. Since the robotic cable comprises of three sections, it has 9 actuated DOF in total.

The springs are normally (with the robot unactuated) in a relaxed state. However, on pulling on all three tendons simultaneously, the concentric and telescoping arrangement of the tubes enable limited controlled translational motion of the sections relative to each other. In the case of retracting all three tendons of, say, the distal section, the springs on it will contract up to their base length, thus counteracting the force, and that backbone section will slide inside the middle tube. The coupling between the sections will cause the springs on the middle section to compress as well to some extent and that in turn will move inside the base section. When the tendons are relaxed, the springs will acquire their initial relaxed state and the sections will return to their positions as before. The springs thus support finite controllable translational motion.

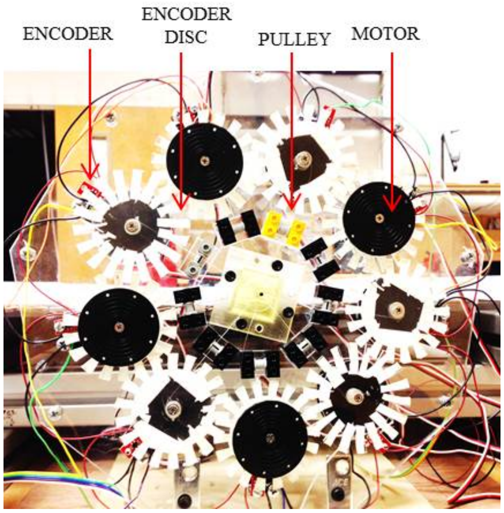

Figure 4.10: Actuation mechanism

The actuator assembly consists of 9 standard continuous rotation RC servo motors spaced at 40° which are arranged on an acrylic base plate as shown in figure 4.10. All the motors fit on a base plate in a square of area 0.1122 m. of The technical specifications of the actuators are given in Table 4.7. Each of the 9 tendons is connected to an individual servo via a pulley mechanism. Each servo is provided with an encoder (analog line sensor) to provide position feedback, with the encoder technical specifications as in Table 4.8. The servos are controlled using Arduino [41] boards (see figure 4.11). Every section of the robotic cable may have one such board.

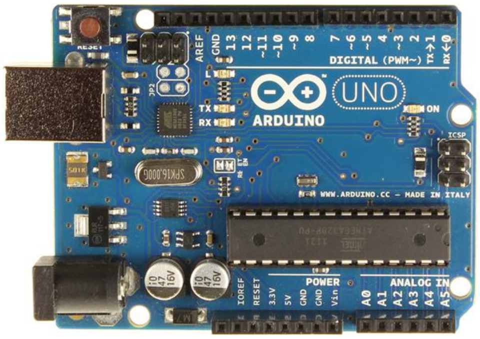

Figure 4.11: Arduino microcontroller

Robotic "Cable" section	***Working Voltage (V)***	***Stall Current(A)***	***Stall Torque (kg-cm)***	***No-load speed***
Distal	5	1.8	12.2	50 rpm
Middle	5-7.4	3 @ 8.5V	19.3 @ 7.4V	60°per 0.16 s at 7.4v
Base	5-7.4	3 @ 8.5V	19.3 @ 7.4V	60°per 0.16 s at 7.4v

Table 4.7: RC Servo parameters

Dimensions (mm×mm)	***DC Operating Voltage (V)***	***Stall Current (mA)***	***Optimal sensing distance (mm)***
7.62×13.97	5	25	3

Table 4.8: Encoder sensor parameters

The RC servos enable a simple control scheme. They have high torque capacities which actuate the robotic cable effectively. They are easily controlled using the Arduinos. The encoders give pulses or counts corresponding to the motor shaft position enabling determination of the cable tip.

In order to make such a novel continuum robot to work, the effects of compression need to be taken into account in addition to the effects resulting from the length changes of the tendons. Chapter 5 explains the need and approach to a new procedure for kinematic modeling for this cable robot in 2D and 3D arrangement.

Chapter 5

Kinematic Modeling

The novel combination of springs and concentric tubes in the design introduced in this Thesis offers significant physical advantages as discussed in the previous Chapter. However, the spring loading requires the development of new models [42] to describe the kinematics of the design. In particular, in order to achieve desired configurations (shapes), relationships between such shapes and measurable quantities (i.e. tendon lengths and/or encoder counts) must be established. Such relationships are introduced in this Chapter.

5.0.1 Modeling approach

There are a wide variety of conventions and coordinate frame choices that exist in the literature to determine kinematics of a continuum robot [43]. We choose two particular frame conventions and use it throughout the thesis - one for planar and other for spatial modeling. The details are explained in sections 5.1.2 and 5.2.2 respectively.

5.1 2D Planar Model

We first consider the situation of operating the new design of Chapter 4 in the plane. Note that the introduction of springs in the design does not alter the fundamental property of constant curvature of each section.

5.1.1 Determination of Length Change and Compression Variables for the Robotic Cable as a Function of the Encoder Counts

The initial baseline modeling for the robotic cable is based on the continuum robot forward kinematic formulas introduced in [44]. The key issue is to incorporate the effects of coupling between bending and extension/compression.

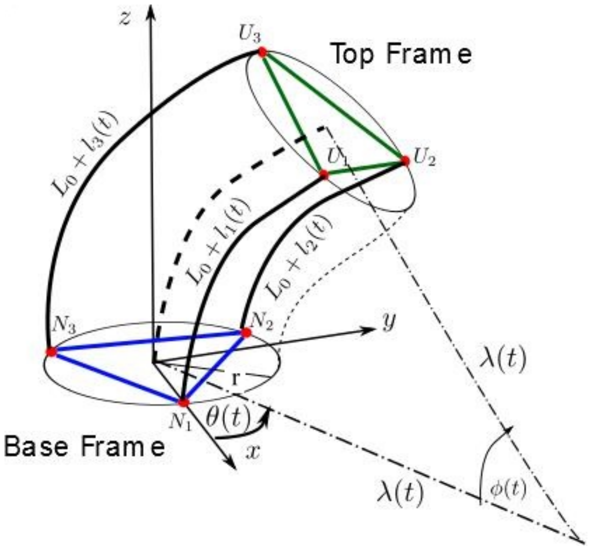

Figure 5.1: Continnum section schematic and geometrical variables

Continuum sections are approximated by constant curvature models using shape variables viz., λ, ϕ and θ as shown in figure 5.1. λ gives the radius of curvature, ϕ determines the elevation (or bending) angle and θ gives the angle of the bending plane as a function of length changes $q_j = \begin{bmatrix} l_{jl} & l_{j2} & l_{j3} \end{bmatrix}^\top$ for each section (here $j = 1, 2, 3$) with respect to the unactuated original length L_0 for each section. r_j is the distance from the center of the manipulator to the location of the tendons. The forward kinematic equations derived in [44] can be implemented as shown below for each section.

$$\lambda_j(q_j) = \frac{(3L_{j0} + l_{j1} + l_{j2} + l_{j3})r_j}{2\sqrt{l_{j1}^2 + l_{j2}^2 + l_{j3}^2 - l_{j1}l_{j2} - l_{j2}l_{j3} - l_{j1}l_{j3}}} \tag{5.1}$$

$$\phi_j(q_j) = \frac{2\sqrt{l_{j1}^2 + l_{j2}^2 + l_{j3}^2 - l_{j1}l_{j2} - l_{j2}l_{j3} - l_{j1}l_{j3}}}{3r_j} \tag{5.2}$$

$$\theta_j(q_j) = \tan^{-1}\left\{\frac{\sqrt{3}(l_{j3} - l_{j2})}{l_{j2} + l_{j3} - 2l_{j1}}\right\} \tag{5.3}$$

The arc length can then be computed as $s_j = \lambda_j \phi_j$.

$$s_j = \frac{(3L_{j0} + l_{j1} + l_{j2} + l_{j3})}{3} \tag{5.4}$$

The above formulation, however, cannot be applied directly to the new continuum robotic cable due to the added unique feature of actively controllable contraction in its middle and distal sections, which is coupled with its bending. To model this effect, we introduce a uniform compression factor $c_j \in \mathbb{R}_0{}^{-}$ in the kinematic formulation in λ and in s as below,

$$\lambda_j(q_j) = \frac{(3(L_{j0} + c_j) + l_{j1} + l_{j2} + l_{j3})r_j}{2\sqrt{l_{j1}^2 + l_{j2}^2 + l_{j3}^2 - l_{j1}l_{j2} - l_{j2}l_{j3} - l_{j1}l_{j3}}} \tag{5.5}$$

$$s_j = \frac{(3(L_{j0} + c_j) + l_{j1} + l_{j2} + l_{j3})}{3} \tag{5.6}$$

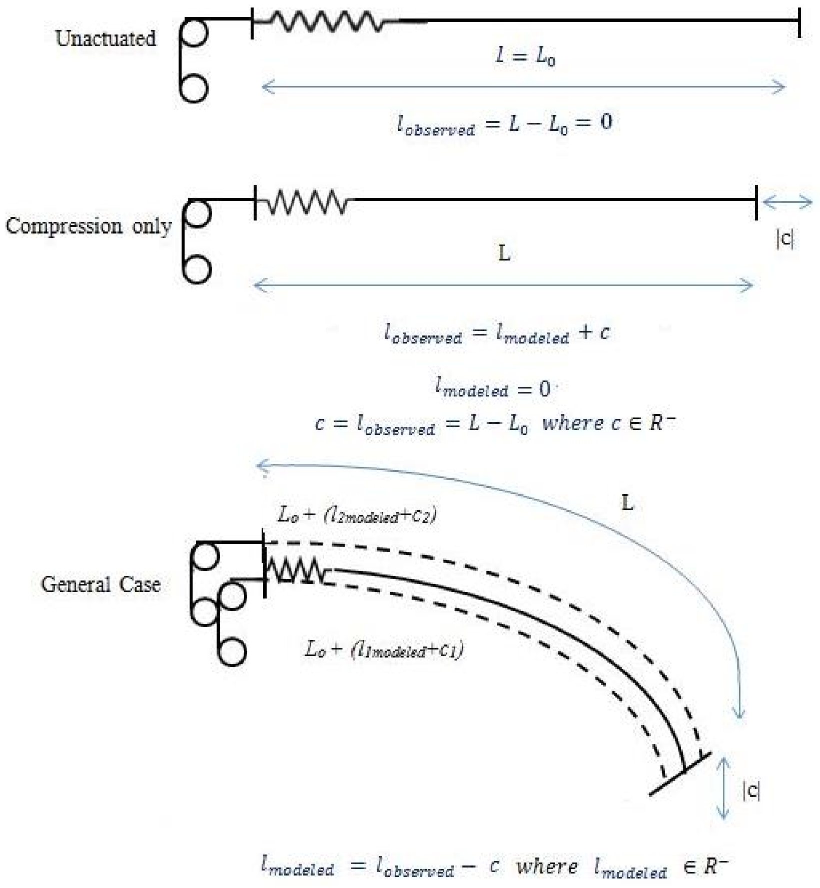
Unactuated
$l = L_0$
$l_{observed} = L - L_0 = 0$
Compression only
L
|c|
$l_{observed} = l_{modeled} + c$
$l_{modeled} = 0$
$c = l_{observed} = L - L_0$ where $c \in R^-$
L
$L_o + (l_{2modeled}+c_2)$
General Case
$L_o + (l_{1modeled}+c_1)$
|c|
$l_{modeled} = l_{observed} - c$ where $l_{modeled} \in R^-$

Figure 5.2: Modeled and apparent length changes

Figure 5.2 (top) shows a single tendon controlling a backbone of length initial length L_0, and is connected to a motor through a pulley. When the backbone is in relaxed state length change $l_{observed} = 0$. When all the tendons are pulled equally as in figure 5.2 (middle), the spring-loaded backbone compresses and $l_{observed}$ becomes $L - L_0$. This is the length change observed (or in this case same as compression c_j) at the actuator (or encoder). In a generic case as in figure 5.2 (bottom) however, both compression and bending occur simultaneously and length change $l_observed$ contains both compression and bending effects.

Hence, to use the kinematic models, the new effective backbone length change will be incorporated as in (5.5) and (5.6). This length change $l_{modeled}$ will be used in developing a kinematic model for this design and is given as $l_{modeled} = l_{observed} - c_j$. This terminology *observed* length change (length change without modeling spring-induced compression) and *modeled* length change (length change with compression subtracted from base section length) will be used frequently in the following subsections of the Chapter. The length changes used in the new kinematic formulation will be the modeled length changes.

The compression cannot be distinguished from the observed length change, as a separate entity, when the tendons are pulled. It is incorporated in all the observed length changes itself. In order to justify and validate the introduction of compression in the modeling, experiments were performed to obtain the compression term separately from the observed length changes when the tendons are pulled to achieve bending.

The modeled change in length q_j in every case, however, is now not with respect to the initial length L_0 but with the modified initial length $(L_{j0} + c_j)$ where $c_j \in {\mathbb{R}_0}^-$ reflects compression (due to the springs) in addition to the modeled length changes to account for the coupling. A method to compute c_j and relate the model to measurable encoder counts is given next.

To simplify the initial analysis and reduce the effect of gravity, the cable was placed on a table to analyze the 2D planar performance of its middle section and distal sections (as shown in figure 4.8), since these are the sections having springs in their design wherein compression has to be taken into account in the kinematics. Let $l_{ji}(i = 1, 2, 3)$ indicate the modeled length changes q_j of the three tendons for given j^{th} section of the robot. Similarly $\sigma_{ji}(i = 1, 2, 3)$ are the encoder counts to the corresponding length changes. Since we consider 2D operation, only l_{j1} and l_{j2} will be considered in this section of the Chapter. The index j can be either 2 or 3 depending on whether the analysis is for the middle or distal section of the cable respectively.

Note that the models (5.2) and (5.6) provide key information regarding the modeled length changes corresponding to given shape ϕ and s of the distal section. Note also that in the plane, θ can be neglected. Actuating tendon 1 for the distal section (i.e. modeled length change is l_{31}) for the planar case, (5.2) simplifies to (5.7). Similarly, equation (5.4) simplifies to (5.8).

$$\phi_3(q_3) = \frac{2l_{31}}{3r_3} \tag{5.7}$$

$$s_3 = \frac{3(L_{30} + c_{31}) + l_{31}}{3} \tag{5.8}$$

Equation (5.7) shows that a linear relation exists between bending angle ϕ and modeled length change l_{31} (since r_3 is constant), which can be represented as,

$$\phi_3(q_3) = p_3 l_{31} \tag{5.9}$$

where $p_3 = \dfrac{2}{3r_3}$ is the slope constant of the line.

We next relate ϕ to encoder counts σ_{31}. Table 5.1 shows experimental data obtained by actuating tendon 1 for the distal section. A general procedure for obtaining the data is explained in detail in 5.2.1. The table illustrates the mapping relating the encoder pulse counts (σ) corresponding to given shape variables of the distal section.

Encoder Counts	***Angle ϕ***	***Radius λ***	***Arc Length s=λ ϕ***
σ	***(rad)***	***(cm)***	***(cm)***
0	0	Inf	34
14	0.4794	69.89	33.52
31	0.6365	51.76	32.95
57	0.9070	35.6	32.29
81	1.0948	28.05	30.71
107	1.3175	22.46	29.59
145	1.5167	18.612	28.23

Table 5.1: 2D experimental data for distal section (average values)

The data obtained from the experiments suggested that line fit approximations could be used to establish a relationship between ϕ and encoder pulses σ i.e.

$$\phi_3(q_3) = q_3 \sigma_{31} \tag{5.10}$$

where q_3 is slope constant of the line.

Knowing r_3 and ϕ_3 from the experimental data, l_{31} can be computed by equation (5.9) and a relationship can be established between modeled length change l_{31} and the corresponding encoder count (σ_{31}) by equating equations (5.9) and (5.10). The resulting linear model relating modeled length changes and encoder counts for the distal section is hence given by,

$$l_{3i} = m_3 \sigma_{3i} \tag{5.11}$$

where $m_3 = \dfrac{q_3}{p_3}$. Note that there is no shape of interest when the encoder count σ_i is zero (and hence when length change l_i is zero).

On repeating the experiment several times the slope constant average m_3 was computed to be $m_3 = -3.648 \times 10^{-5}$(m).

Next knowing L_{30}, taking s_3 from experimental data and l_{31} computed in (5.11), the compression factor c_{31} can be determined from (5.8) as

$$c_{31} = s_3 - \frac{l_{31}}{3} - L_{30} \tag{5.12}$$

Note that while bending the robot, when there is no modeled length change l_{31}, the arc length s is equal to the initial default length L_{30} and hence compression c_{31} is zero.

The coupling compression factor c_{31} was then plotted with respect to the corresponding encoder counts σ_{31}. The plot showed a linear correlation between them. The resulting linear model relating compression and encoder counts for the distal section for a general tendon is hence given by,

$$c_{3i} = n_3 \sigma_{3i} \tag{5.13}$$

After multiple tests, the slope constant n_3 was computed to be $n_3 = -4.12 \times 10^{-4}$(m).

Encoder Counts	***Angle*** ϕ	***Radius*** λ	***Arc Length s=***λ ϕ
σ	*(rad)*	*(cm)*	*(cm)*
0	0	Inf	34
37	0.17	260.64	44.31
55	0.2754	158.43	43.63
91	0.3752	113.89	42.73
118	0.475	88.9	42.25
126	0.5365	78.28	41.99
151	0.6032	68.67	41.42

Table 5.2: 2D experimental data for middle section (average values)

The above procedure was repeated for the middle section. Table 5.2 shows the experimental data obtained by actuating one tendon of the middle section. After several experiments, the slope constant averages m_2 and n_2 were determined to be $m_2 = -1.329 \times 10^{-5}$(m) and $n_2 = -2.36 \times$

10^{-4}(m) using the same procedure described above for the distal section.

Using these derived changes in modeled lengths and coupled compressions as a function of encoder counts in the forward kinematic equations (5.5), (5.2) and (5.3), the kinematic performance of the robotic cable can be predicted effectively (see Chapter 6). The unactuated original length L_{j0} is now augmented with additional compression values c_{ji} for the middle and distal sections. The modified formulations, using (5.11) and (5.13) for describing the shape configuration of the continuum cable are expressed in (5.14), (5.15) and (5.16).

$$\lambda_j(\sigma_{ji}) = \frac{(3(L_{j0} + n_j \sum_{i=1}^{3} \sigma_{ji}) + m_j \sum_{i=1}^{3} \sigma_{ji})r_j}{2m_j\sqrt{\sigma_{j1}^2 + \sigma_{j2}^2 + \sigma_{j3}^2 - \sigma_{j1}\sigma_{j2} - \sigma_{j2}\sigma_{j3} - \sigma_{j1}\sigma_{j3}}} \tag{5.14}$$

$$\phi_j(\sigma_{ji}) = \frac{2m_j\sqrt{\sigma_{j1}^2 + \sigma_{j2}^2 + \sigma_{j3}^2 - \sigma_{j1}\sigma_{j2} - \sigma_{j2}\sigma_{j3} - \sigma_{j1}\sigma_{j3}}}{3r_j} \tag{5.15}$$

Since the robot is analyzed in a planar orientation thus far, the angle of bending plane θ is zero. In general, it will be given as,

$$\theta_j(\sigma_{ji}) = \tan^{-1}\left\{\frac{\sqrt{3}(\sigma_{j3} - \sigma_{j2})}{\sigma_{j2} + \sigma_{j3} - 2\sigma_{j1}}\right\} \tag{5.16}$$

5.1.2 Direct Kinematic Modeling

The convention for assigning frames for the purpose of planar modeling is as shown in figure 5.3. A local coordinate frame is assigned at the base of the constant curvature section of the continuum robot under consideration such that the origin of the frame coincides with the center of the manipulator. We align the positive y-axis along the tangent to the section. The frame can in general rotate about the backbone through angle θ, but since the model is planar, in this section of the Chapter we assume that the frame can only translate along frame axis in the bending plane. Thus, with θ being zero, a positive bending angle ϕ will produce bending about the positive z-axis, such that when ϕ becomes 180° the tip of the section will touch the positive x-axis. The plane of each section is, thus, modeled in the local xy-plane.

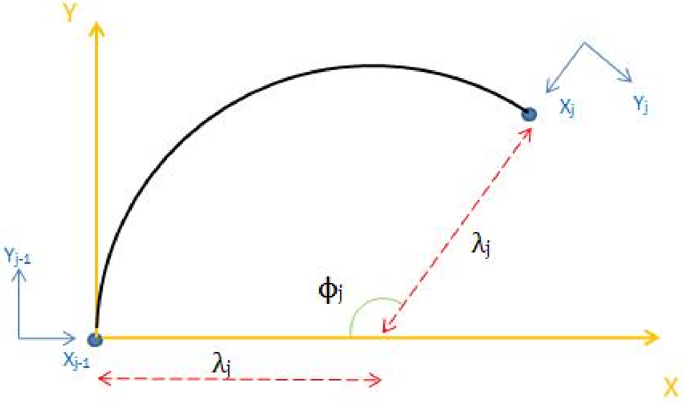

Figure 5.3: Transformations used to get HTM for planar motion

The homogenous transformation matrix (HTM) for the j^{th} section [44] can now use the shape variables derived in equations (5.14) and (5.15) as function of the encoder counts. The HTM is derived by using following sequence of transformations to model planar bending.

$$ {}^{j-1}_{j}T = P_x(\lambda_j) R_z^T(\phi_j) P_x(-\lambda_j) \tag{5.17}$$

The coordinate frame assigned to the base of the j^{th} section of the continuum cable robot is first translated through distance λ about the x-axis, rotated clockwise through angle ϕ about z- axis and finally translated through distance λ along the current negative x-axis to reach the tip frame of the robot section.

Since the shape variables λ_j and ϕ_j have been shown in the previous section as functions of the measurable encoder counts (σ_{ji}) in equations (5.14) and (5.15), the HTM will be expressed as,

$$ {}^{j-1}_{j}T = \begin{bmatrix} \cos(\phi_j(\sigma_{ji})) & \sin(\phi_j(\sigma_{ji})) & 0 & \lambda_j(\sigma_{ji})(1-\cos(\phi_j(\sigma_{ji}))) \\ -\sin(\phi_j(\sigma_{ji})) & \cos(\phi_j(\sigma_{ji})) & 0 & \lambda_j(\sigma_{ji})\sin(\phi_j(\sigma_{ji})) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.18}$$

Considering that the robotic cable is a 3 section continuum arm, the complete HTM will be given as,

$$ {}^0_3T = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \tag{5.19} $$

Since there are no springs in the base section, ${}^0_1T$ can be computed using the direct kinematic formulas as in (5.1), (5.2) and (5.3). However, it is still necessary to relate its shape variables to measurable variables. The method is the same as described in the previous section for the middle and distal sections, except that compression term is absent and only the linear relationship between length changes and the encoder counts needs to be used for computation of ${}^0_1T$.

This completes the forward kinematic model for 2D operation of the long and slim continuum robotic cable.

5.2 3D Spatial Model

We next generalize the kinematic modeling approach of section 5.1 to the spatial case.

5.2.1 Determination of Compression and Length Change Variables for the Robotic Cable as a Function of the Encoder Counts

In case of the general spatial arrangement, the procedure of computing modeled length change directly from ϕ as in the planar case cannot be used since angle ϕ is now dependent on more than one modeled length change. Similarly, λ is also dependent on compression and modeled length changes caused by multiple tendons. The situation becomes a case of an underdetermined system of equations with more unknowns (four - compression and three modeled length changes) than the number of independent equations (three - kinematic equations (5.5), (5.2) and (5.3)). Thus, for a given shape of a a section of the cable, a simple linear mapping of modeled length changes and compression with the encoder counts, as in the 2D case, cannot be used. Hence, a different approach is needed.

As noted earlier, the compression is inherent in all the three (observed) length changes visible when the tendons are actuated. Physically when tendons are actuated, it is seen that compression occurs first in most cases (since it is easier than bending). Bending will also begin slowly as it is coupled with compression. Once maximum compression is reached pure bending will be seen.

Let us assume that compression is equally distributed in the three length changes. Hence, by logical intuition, let the maximum of the three observed length changes be the compression c. If that compression is subtracted from the three observed length changes the residual modeled length change will cause the effective bending.

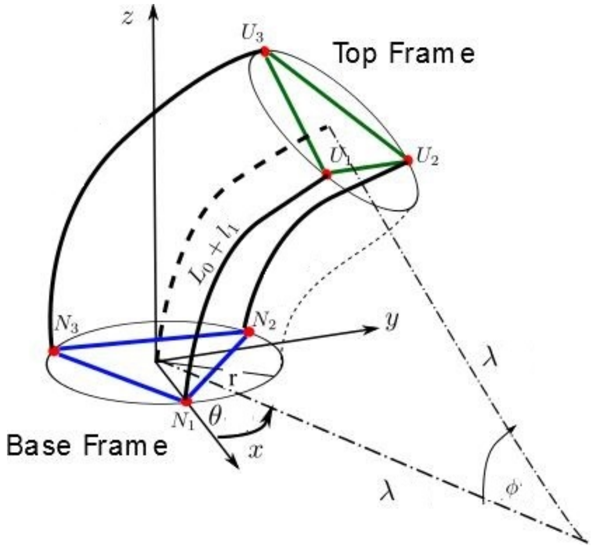

Figure 5.4: Continnum section schematic and geometrical relationship

Knowing λ, ϕ and θ for a given configuration of a section, the observed length change for a tendon ($l_{1_{observed}}$) can be computed as follows from basic geometry.

The total length of the tendon will be equal to the arc length subtended by it. The backbone arc length (say L_1) is $L_1 = \lambda\phi$, where λ is the radius of the arc and ϕ is angle subtended by it. Note we assume that the tendons curve to match the backbone curvature, a reasonable assumption for the cable robot hardware. However, since the section is bending in a plane at an angle θ with respect to the x-axis of the reference frame (as shown in figure 5.4), a projection (equal to $rcos\theta$) from the reference to the bending plane, needs to be subtracted from the radius λ. The parameter r, as specified earlier, is the distance from center of the manipulator to the location of the tendon. This is illustrated in figure 5.4. Thus, the arc length is $L_1 = \phi(\lambda - rcos\theta)$. Since we are interested in observed length change $l_{1_{observed}}$ of the tendon, this will be $(L_1 - L_0)$, where L_0 is the tendon's default (or relaxed state) length. Thus, the observed length change for the j^{th} section can be generalized as below:

$$l_{j1_{observed}} = \phi_j(\lambda_j - r_j cos\theta_j) - L_{j0} \tag{5.20}$$

The observed length changes for the remaining two tendons can be computed by moving θ through an angle of 120° and 240° respectively (see figure 5.1).

$$l_{j2_{observed}} = \phi_j(\lambda_j - r_j cos\left(\frac{2\pi}{3} - \theta_j\right)) - L_{j0} \tag{5.21}$$

$$l_{j3_{observed}} = \phi_j(\lambda_j - r_j cos\left(\frac{4\pi}{3} - \theta_j\right)) - L_{j0} \tag{5.22}$$

Since, compression c_j is assumed to be the maximum length change of the three observed ones computed in (5.20) to (5.22), the modeled length changes can be then computed as follows,

$$l_{j1} = l_{1_{observed}} - c_j \tag{5.23}$$

$$l_{j2} = l_{2_{observed}} - c_j \tag{5.24}$$

$$l_{j3} = l_{3_{observed}} - c_j \tag{5.25}$$

We observe from equations (5.23) to (5.25) that compression and the modeled length changes can be expressed using the kinematic models as functions of shape variables as defined below,

$$c_j = g_c(\lambda_j, \phi_j, \theta_j) \tag{5.26}$$

$$l_{j1} = g_{l_{j1}}(\lambda_j, \phi_j, \theta_j) \tag{5.27}$$

$$l_{j2} = g_{l_{j2}}(\lambda_j, \phi_j, \theta_j) \tag{5.28}$$

$$l_{j3} = g_{l_{j3}}(\lambda_j, \phi_j, \theta_j) \tag{5.29}$$

We next empirically determine the relationship between the shape variables and the encoder counts. Experiments were performed with the cable across the achievable range of spatial configurations, to compute the shape variables λ, ϕ and θ for the distal and middle sections with reference to the measurable encoder counts. Approximately 200 data points were collected for the distal section and about 130 for the middle section. The details of the data is provided in Appendix A and B respectively, for purpose of brevity. It provides a set of data points sampling the following relationships,

$$\lambda_j = h_\lambda(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.30}$$

$$\phi_j = h_\phi(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.31}$$

$$\theta_j = h_\theta(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.32}$$

These data points obtained from experiments were substituted in equations (5.20) to (5.22) to obtain a corresponding set of data points for observed length changes given encoder counts. Then using the assumption that maximum of the observed length changes is compression, equations (5.23) to (5.25) were used to obtain modeled length changes.

In order to verify whether the this assumption for compression was valid, the modeled length changes and the compression computed for a given configuration were substituted in the kinematic formulations (5.5), (5.2) and (5.3). The shape parameters matched the experimental values of λ, ϕ and θ, validating the (logical) intuitive assumption.

Thus, substituting a finite sampled data set for equations (5.30) to (5.32) in (5.26) to (5.29), we obtain a finite sampling of the functional relationship between the controllable kinematic variables compression and modeled length changes (which define the shape configuration) given the actual measurable encoder readings:

$$c_j = f_{c_j}(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.33}$$

$$l_{j1} = f_{l_{j1}}(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.34}$$

$$l_{j2} = f_{l_{j2}}(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.35}$$

$$l_{j3} = f_{l_{j3}}(\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) \tag{5.36}$$

Once the nature of these four functions is known, the kinematic modeling will be complete. Since we have a large amount of data, we can approximate these functions with the resulting interpolation functions to then be able to provide the controllable variables from arbitrary encoder readings as inputs. The first step would ideally be to plot the dependent and independent variables. However, since the length changes and compression (dependent variables) are a function of three encoder values (independent variables), the plotting will be a four dimensional case and the output not make sense visually. Hence, a polynomial regression method was adopted.

Polynomial regression establishes a nonlinear relationship between the value of the independent variable and the conditional mean of the dependent variable. As it is considered as a statistical estimation problem the regression is linear in-spite of fitting a non-linear model to the data. In other words, the regression function is linear in the unknown coefficients that are estimated from the data. Polynomial models are useful in predicting and estimating the nature of responses for given values of input variables. The approach gives a sense of how the input variables can cause the output variables to respond to a particular value and direction. It one of the most common models used for analysis of designed experiments. Second-order polynomial models are mostly utilized in engineering applications [45].

The generalized second order polynomial used herein for the spatial case of the robotic cable is as given below,

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \tag{5.37}$$

where Y is the dependent variable (one of (5.33) to (5.36)), x is the independent variable (encoder values) and $\alpha_i (i = 0, 1, 2)$ are the model coefficients. However, since here the dependent variables (being compression and length changes) each being dependent on three independent parameters (encoder values), multiple regression needs to be used [45]. Hence, the complete polynomial model for a given section j will be given as in (5.38) where Y_j is either compression c_j or any of length changes l_{j1}, l_{j2} and l_{j3}, since all are (independent) polynomial functions of the encoder counts $\sigma_{ji} (i = 1, 2, 3)$.

$$Y_j = (\alpha_{j0} + \alpha_{j1}\sigma_{j1} + \alpha_{j2}\sigma_{j1}^2)(\alpha_{j3} + \alpha_{j4}\sigma_{j2} + \alpha_{j5}\sigma_{j2}^2)(\alpha_{j6} + \alpha_{j7}\sigma_{j3} + \alpha_{j8}\sigma_{j3}^2) \tag{5.38}$$

There are 9 independent coefficients in this multiple polynomial regression model. The total number of model coefficients α_{jk} (including the dependent ones generated in the multiple regression process) becomes 27 ($k = 1, 2, .., 27$). In total, four such polynomial models are implemented (one each for c_j, l_{j1}, l_{j2} and l_{j3}). The model coefficients were determined as follows:

1. A data set of shape variables λ, ϕ, θ and the corresponding measured encoder counts σ is collected. This was achieved using the following procedure (used in measurements for 2D model as well),

   (a) Corresponding to each (set of three) encoder data point, for a particular bending plane angle θ (in case of measurements used for the planar model θ is not needed), we take images of the articulating section such that the bending angle ϕ will increase from 0° to desired angle in arbitrary steps in the plane.

   (b) A grid having squares of 1 inch was used as the backdrop plane for measurement reference.

   (c) For every curved position of the section for a given configuration of θ, width and height of the curved section was measured in Paint.NET (free image and photo editing software). Paint.NET provides the number of pixels corresponding to the figure (in this case width and height measuring lines)drawn.

   (d) Using the grid as reference, these length measurements were converted into actual widths and height of the section for given orientation.

   (e) The radius λ and bending angle ϕ is then computed using geometry (see figure 5.5),

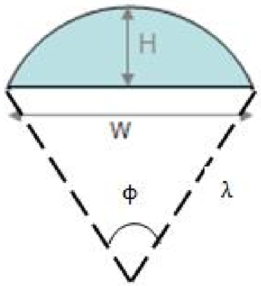

Figure 5.5: Measurement of radius and bending angle of an arc

$$\lambda = \frac{H}{2} + \frac{W^2}{8H} \tag{5.39}$$

$$\phi = 2sin^{-1}\frac{W/2}{\lambda} \tag{5.40}$$

2. Equations (5.20) to (5.22) are used to compute the observed length changes for every case.

3. Maximum of the three observed length changes is selected as compression in each case.

4. Equations (5.23) to (5.25) are used to compute the modeled length changes.

5. Then the resulting large set of $\{\sigma_{ji} | c_j, l_{ji}, l_{j2}, l_{j3}\}$ data points are interpolated in MATLAB® using structure (5.38) to find four interpolated functions of (5.33) to (5.36).

All the coefficients for each of the four polynomial model are listed in the Appendix C.

Using these derived function approximations for changes in lengths and coupled compressions as a function of three encoder counts in the forward kinematic equations (5.2) to (5.3) and (5.5), the performance of the robotic cable can be predicted. The modified kinematic formulation for describing the shape configuration of the continuum cable in spatial arrangement is as expressed in (5.41), (5.42) and (5.43). In these expressions, $c_{j_{est}}$ is the estimated compression from the polynomial function (5.33) while $l_{j1_{est}}$, $l_{j2_{est}}$ and $l_{j3_{est}}$ are estimated length changes from the polynomial functions (5.34) to (5.36) respectively. All these functions are of the form (5.38).

$$\lambda_j(\sigma_{ji}) = \frac{(3(L_{j0} + c_{j_{est}}) + l_{j1_{est}} + l_{j2_{est}} + l_{j3_{est}})r_j}{2\sqrt{l^2_{j1_{est}} + l^2_{j2_{est}} + l^2_{j3_{est}} - l_{j1_{est}} l_{j2_{est}} - l_{j2_{est}} l_{j3_{est}} - l_{1_{est}} l_{3_{est}}}} \tag{5.41}$$

$$\phi_j(\sigma_{ji}) = \frac{2\sqrt{l^2_{j1_{est}} + l^2_{j2_{est}} + l^2_{j3_{est}} - l_{j1_{est}} l_{j2_{est}} - l_{j2_{est}} l_{j3_{est}} - l_{1_{est}} l_{3_{est}}}}{3r_j} \tag{5.42}$$

$$\theta_j(\sigma_{ji}) = \tan^{-1}\left\{ \frac{\sqrt{3}(l_{j3_{est}} - l_{j2_{est}})}{l_{j3_{est}} + l_{j2_{est}} - 2l_{j1_{est}}} \right\} \tag{5.43}$$

5.2.2 Direct Kinematic Modeling

The convention for assigning frames for the purpose of spatial modeling is as shown in figure 5.6. A local coordinate frame is assigned at the base of the constant curvature section of the continuum robot under consideration such that the origin of the frame coincides with the center of the manipulator. We align the positive z-axis along the tangent to the section. The frame rotates about the backbone through angle θ (selecting the plane in which bending is to occur i.e. bending plane) and translates along the frame axis through λ in the bending plane. At a given bending plane angle θ, a positive bending angle ϕ will produce bending about the positive y-axis, such that when the ϕ becomes 180° the tip of the section will touch the positive x-axis of the corresponding bending plane (i.e. translation by λ_j about local x then rotate by θ_j about z to put frame at section tip). The plane of each section is, thus, modeled in the local xz-plane.

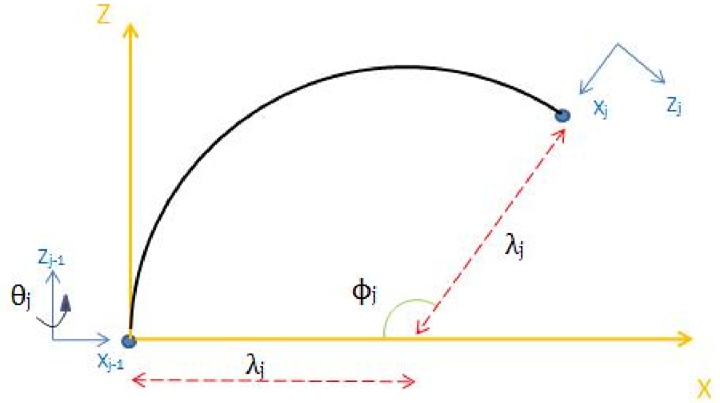

Figure 5.6: Transformations used to get HTM for spatial motion

The homogenous transformation matrix (HTM) for a j^{th} section [44] can now use the shape parameters derived in equations (5.41), (5.42) and (5.43) as function of the encoder counts. The HTM is derived by using following sequence of transformations to achieve spatial bending.

$$T_j = R_z(\theta_j)P_x(\lambda_j)R_y(\phi_j)P_x(-\lambda_j)R_z^T(\theta_j) \tag{5.44}$$

The coordinate frame assigned to the base of j^{th} the section of the continuum cable robot is first rotated through angle θ about z-axis, then translated through distance λ about the x-axis, then rotated through angle ϕ about y-axis, then translated through distance λ along the negative x-axis and finally rotated clockwise through angle θ about z-axis to reach the tip frame of the robot.

Let $\cos(\phi_j(\sigma_{ji})) = cp_{ji}$, $\cos(\theta_j(\sigma_{ji})) = ct_{ji}$, $\sin(\phi_j(\sigma_{ji})) = sp_{ji}$ and $\sin(\theta_j(\sigma_{ji})) = st_{ji}$. As in 2D, since the shape variables λ_j and ϕ_j have been shown in the previous section as functions of the measurable encoder counts (σ_{ji}) in equations (5.41), (5.42) and (5.43) then the HTM obtained from (5.44) is,

$$
{}^{j-1}_{j}T = \begin{bmatrix} cp_{ji}ct_{ji}^2 + st_{ji}^2 & ct_{ji}st_{ji}(cp_{ji}-1) & ct_{ji}sp_{ji} & \lambda_{ji}ct_{ji}(1-cp_{ji}) \\ ct_{ji}st_{ji}(cp_{ji}-1) & ct_{ji}^2 + cp_{ji}st_{ji}^2 & st_{ji}sp_{ji} & \lambda_{ji}st_{ji}(1-cp_{ji}) \\ -ct_{ji}sp_{ji} & -st_{ji}sp_{ji} & cp_{ji} & \lambda_{ji}sp_{ji} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.45}
$$

Considering that the robotic cable is a 3 section continuum arm, the complete HTM will be the product of the three HTM for $j = 1, 2, 3$. Again, as in the 2D case, since there are no springs in the base section, ${}^{0}_{1}T$ can be computed using the direct kinematic formulas as in (5.1), (5.2) and (5.3). However, it is still necessary to relate its shape to measurable parameters. As in the 2D case, this can be achieved by using the same method described in previous section for the middle and distal sections, except that compression term is absent and only the polynomial regression function approximations for length changes (in terms of the encoder counts) need to be used for computation of ${}^{0}_{1}T$. Also, an additional $R^{4\times4}$ homogeneous rotation matrix, given for our particular hardware arrangement as $R_z^T(\frac{2\pi}{9})$, has to be taken into consideration to account for the relative arrangement of tendons in the adjacent sections (eg: starting from an i^{th} tendon for base section a 40° clockwise shift will lead to the i^{th} tendon of middle section and another 40° clockwise shift will lead to the i^{th} tendon of distal section). Hence, the complete HTM for the spatial operation of the cable will be given as,

$$
{}^{0}_{3}T = {}^{0}_{1}T \cdot R_z^T(\frac{2\pi}{9}) \cdot {}^{1}_{2}T \cdot R_z^T(\frac{2\pi}{9}) \cdot {}^{2}_{3}T \tag{5.46}
$$

This completes the forward kinematic model for the general 3D case of the long and slim continuum robotic cable.

Chapter 6

Results and Discussion

In this Chapter we compare results generated using the kinematic model of Chapter 5 with those measured using the cable robot hardware of Chapter 4. The objective is to assess the ability of the kinematic model to predict the shape (configuration) of the hardware.

6.1 2D Planar Model

The modified direct kinematic equations derived in subsection 5.1.1 of Chapter 5 were implemented in MATLAB® providing the encoder counts as inputs and computing the various shape variables (viz., λ, ϕ and s) of the robot. The results were then compared with the corresponding results with the cable robot hardware of Chapter 4 to consider the effectiveness of the new modeling procedure on the performance of the robot in 2D space.

Considering the cable as a single section continuum cable consisting of the distal section alone, the results obtained were as in figures 6.1-6.6. The general procedure for obtaining the data in the plot is described in detail in section 5.2.1.

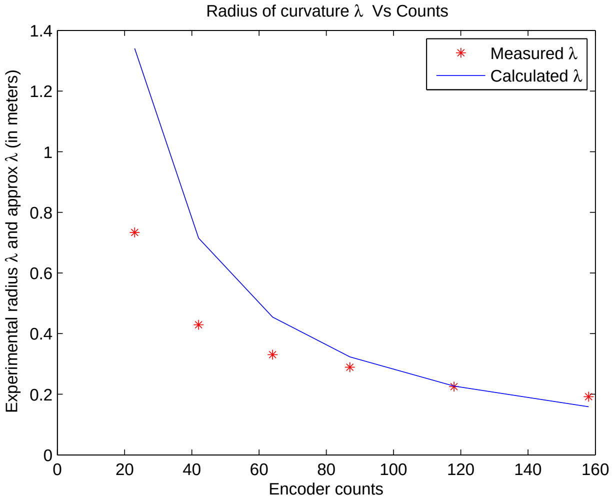

Figure 6.1: Actual and calculated radius of curvature λ relative to encoder counts for the distal section bending left

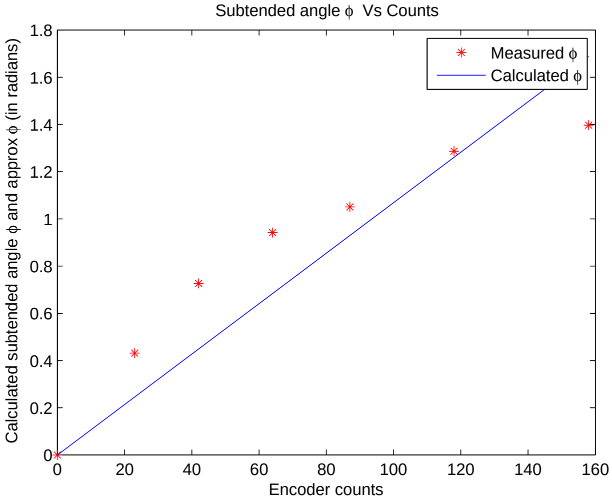

Figure 6.2: Actual and calculated bending angle ϕ relative to encoder counts for the distal section bending left

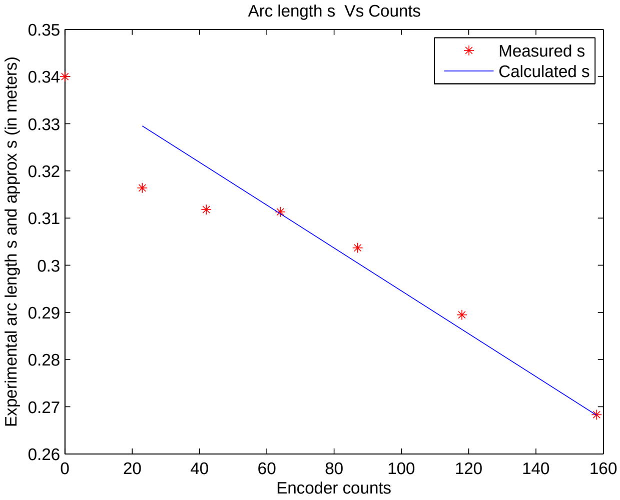

Figure 6.3: Actual and calculated arc s length relative to encoder counts for the distal section bending left

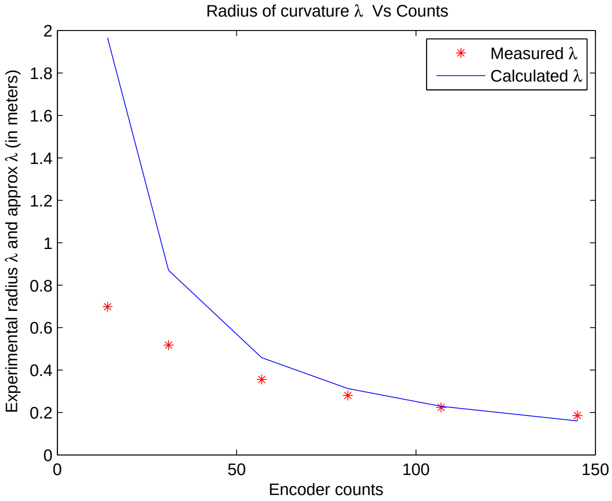

Figure 6.4: Actual and calculated radius of curvature λ relative to encoder counts for the distal section bending right

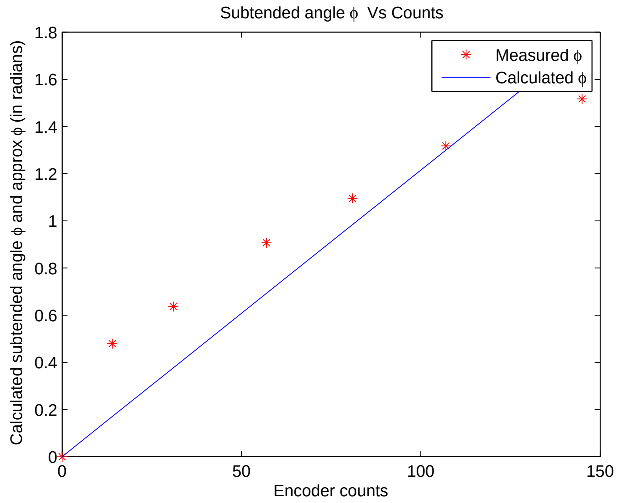

Figure 6.5: Actual and calculated bending angle ϕ relative to encoder counts for the distal section bending right

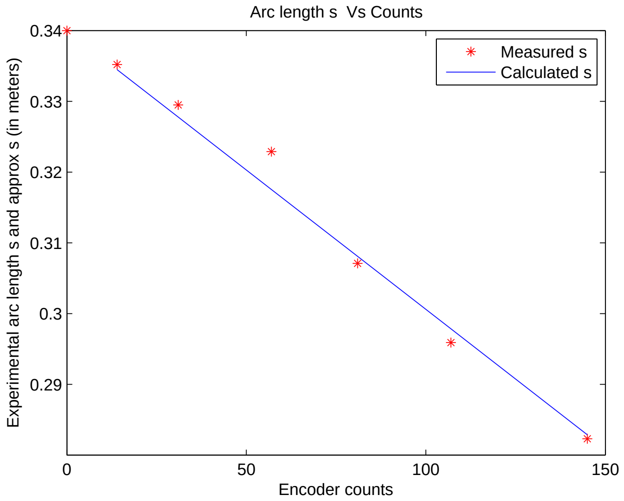

Figure 6.6: Actual and calculated arc length s relative to encoder counts for the distal section bending right

For the distal case, the results for λ are unimpressive in comparison to the middle section results to follow next. This can be attributed to the fact that the compression caused by the springs used in this section is heavily coupled with its bending. The orientation of tendons for the section in this particular 2D arrangement also affects its accuracy (explained in more detail after figure 6.11).

The procedure was then repeated for the middle section, and figures 6.7-6.9 show the outcome.

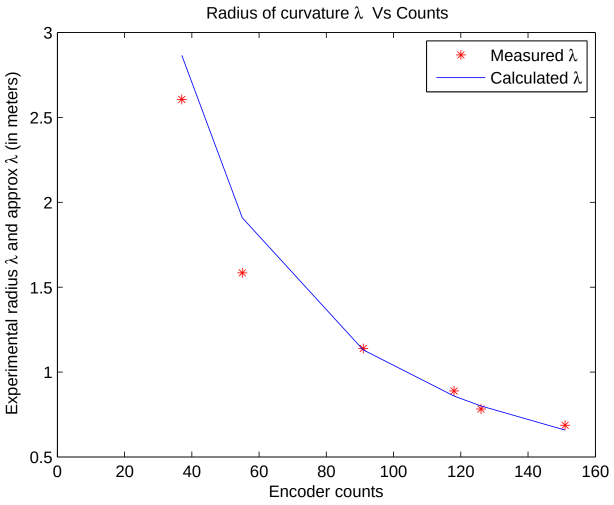

Figure 6.7: Actual and calculated radius of curvature relative λ to encoder counts for the middle section

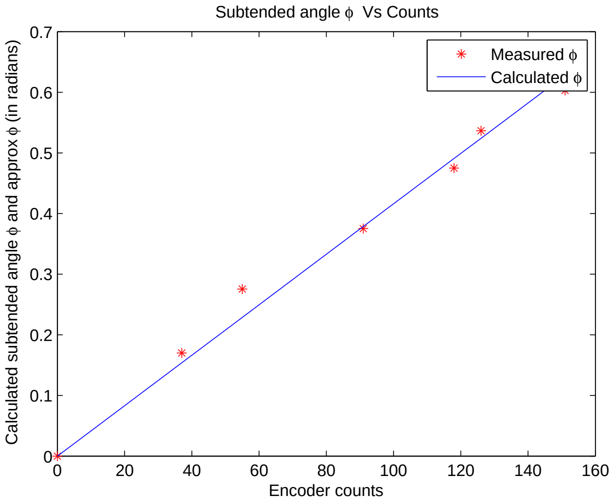

Figure 6.8: Actual and calculated bending angle ϕ relative to encoder counts for the middle section

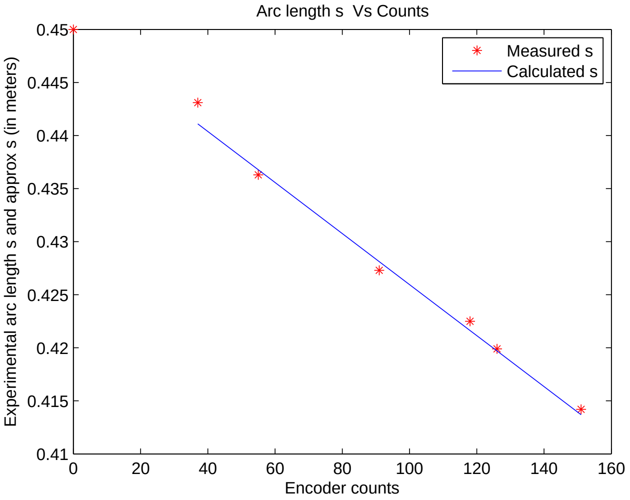

Figure 6.9: Actual and calculated arc length s relative to encoder counts for the middle section

The computed kinematic variables using the linear models are generally in good agreement with the experimental kinematic variables. This can be attributed to the fact that the springs used in this section are relatively stiffer in comparison to the distal section. The orientation of tendons for the section in this particular 2D arrangement also plays a major role (which is explained after figure 6.13) in getting superior results.

6.1.1 2D Forward Kinematics Result

These shape variables were then fed into the direct kinematic model obtained in subsection 5.1.2 of Chapter 5 to obtain the position variables (x, y) of the cable tip.

The x and y coordinates indicating the position of the tip of the robotic cable were computed first considering the cable as a single section continuum cable consisting of the distal section alone. The coordinates are plotted with reference to their corresponding encoder counts provided as inputs.

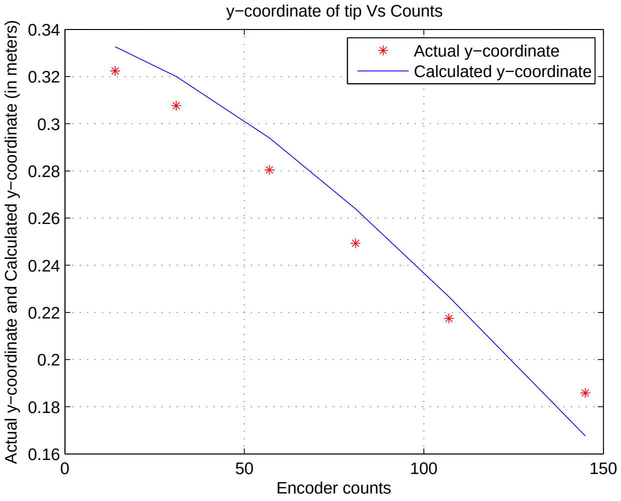

Figure 6.10: Y-coordinate of distal section tip relative to encoder counts

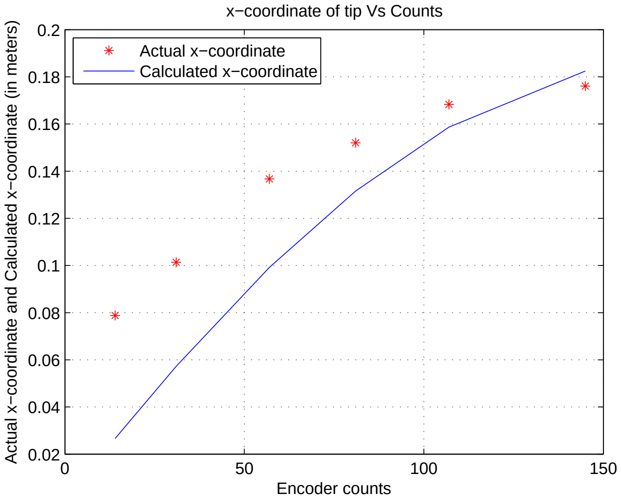

Figure 6.11: X-coordinate of distal section tip relative to encoder counts

Figures 6.10-6.11 show the output given by the kinematic model taking as input the calculated shape variables value in comparison with the actual values obtained from the hardware of Chapter 4. The actual coordinates were computed by substituting experimental shape variables in 2D HTM obtained previously in (5.17), while the model coordinates were computed by substituting the modeled shape variables in the 2D HTM obtained in (5.18). The results are reasonable for the y-coordinate of the tip position. For the x-coordinate, however, the results are not as satisfactory.

The main reason for this may be attributed to various unmodeled effects. One major factor is the inherent friction present on the 2D experimental surface. Also, the robot is designed for operation in 3D space but is arranged in a planar manner. Thus, due to the orientation of tendons (120°apart), only a component of the force actuating the tendon intended to cause bending, deflects it in the horizontal plane in this section. The tendon tries to deflect in a plane angular to the horizontal surface, thereby thrusting the section on the surface. This results in more frictional forces. Thus, friction and tendon orientation affects the results obtained significantly. Another factor can be the unmodeled effects of the motor and sensor systems.

The same procedure was repeated for the middle section. Figures 6.12-6.13 depict the comparative results.

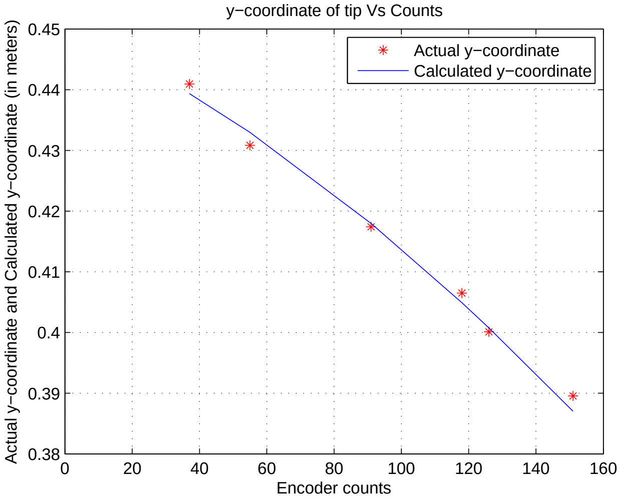

Figure 6.12: Y-coordinate of middle section tip relative to encoder counts

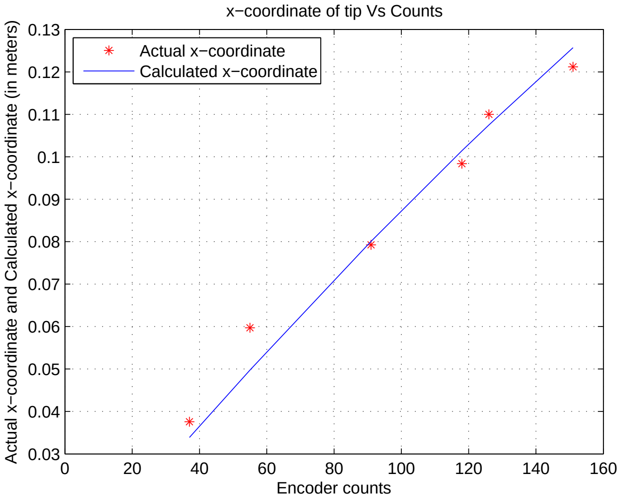

Figure 6.13: X-coordinate of middle section tip relative to encoder counts

The results agree quite well. They imply that the linear approximations incorporated in the modeling to account for the effect of the spring-loaded tendon design are justified. Also, the middle section is less compressible as compared to the distal section and there is absence of compressive coupling with the base section (which lacks springs). The actuating tendon for the middle section was also oriented in the horizontal plane where bending is actually intended. All this helps the direct transfer of the total forces to achieve bending.

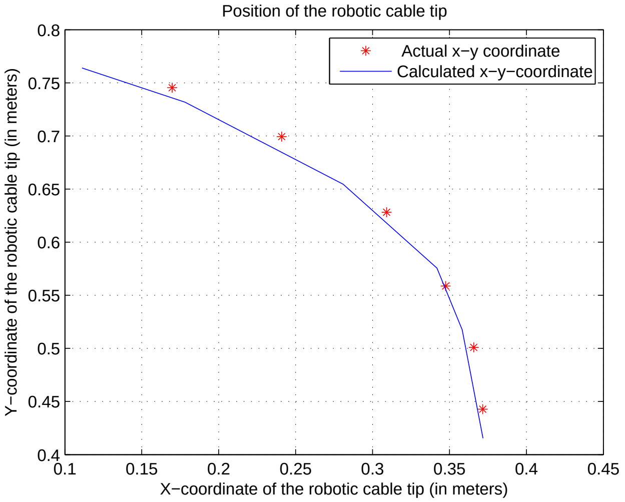

Figure 6.14: XY-coordinates of multi-section cable tip relative to encoder counts

Finally, the robotic cable was operated as a two section continuum manipulator cable (middle and distal sections operating since those incorporated the spring-loaded concentric tube aspect of the design) and the 2D HTM for the multi-section version, as a function of the encoder inputs, as in (5.19), was computed. The resulting position of the end of the robotic cable provided by the kinematic model was plotted in comparison with the actual position. Figure 6.14 shows excellent results for this two section version of the cable.

Thus, the linear regression method is quite effective in the kinematic modeling of such multi-section long and slim continuum robotic cables having local compression feature using spring driven tubular backbone elements.

6.2 3D Spatial Model

The modified direct kinematic equations derived in subsection 5.2.1 of Chapter 5 were implemented in MATLAB® providing the encoder counts as inputs and computing the various shape variables (viz., λ, ϕ and θ) of the robot. The results were then compared with the corresponding results with the cable robot hardware of Chapter 4 to consider the effectiveness of the new modeling procedure on the performance of the robot in 3D space.

Considering the robot as a single section continuum cable consisting of the distal section alone, the results obtained were as in figures 6.15-6.17.

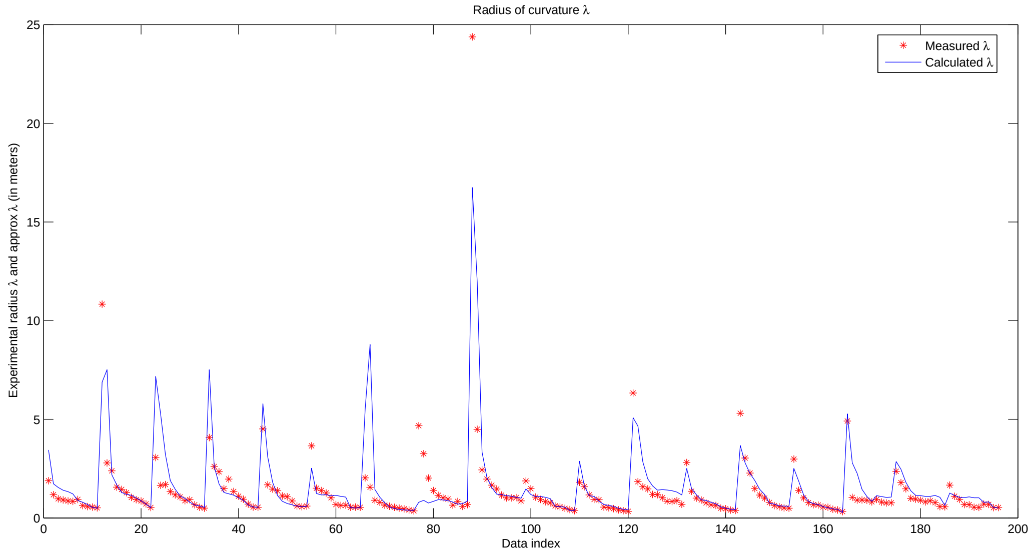

Figure 6.15: Actual and calculated radius of curvature λ for the distal section in 3D space

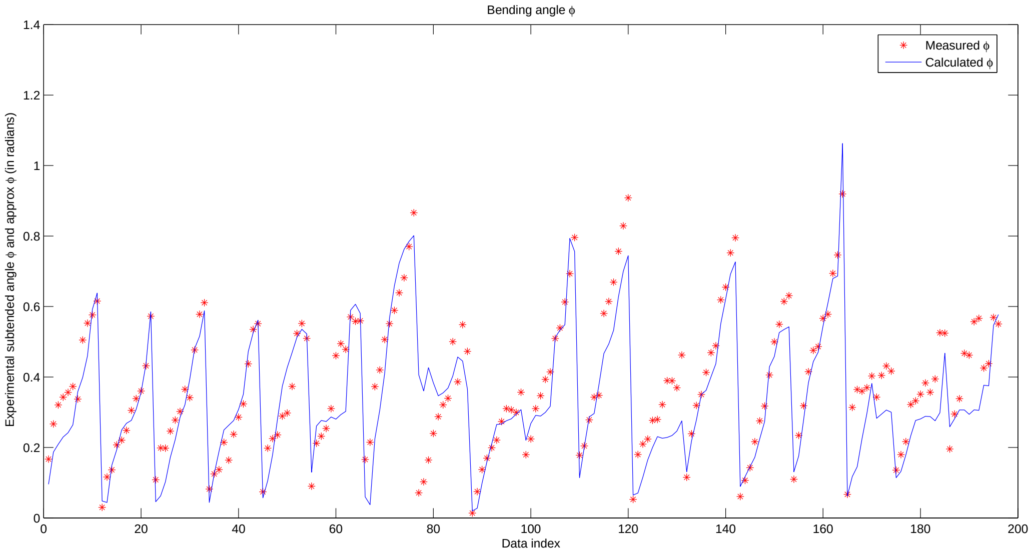

Figure 6.16: Actual and calculated bending angle ϕ for the distal section in 3D space

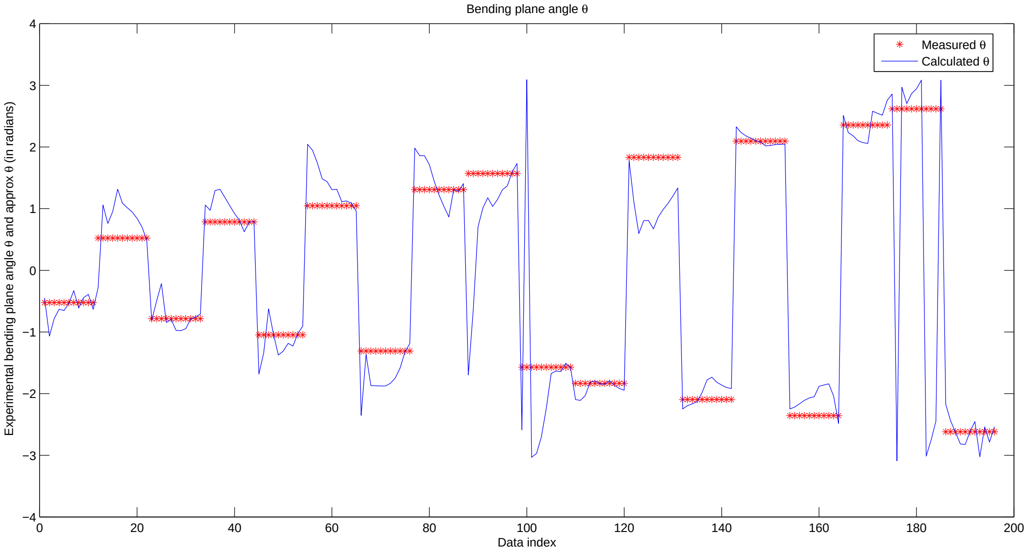

Figure 6.17: Actual and calculated bending plane angle θ for the distal section in 3D space

The procedure was then repeated for the middle section, and figures 6.18-6.20 show the outcome.

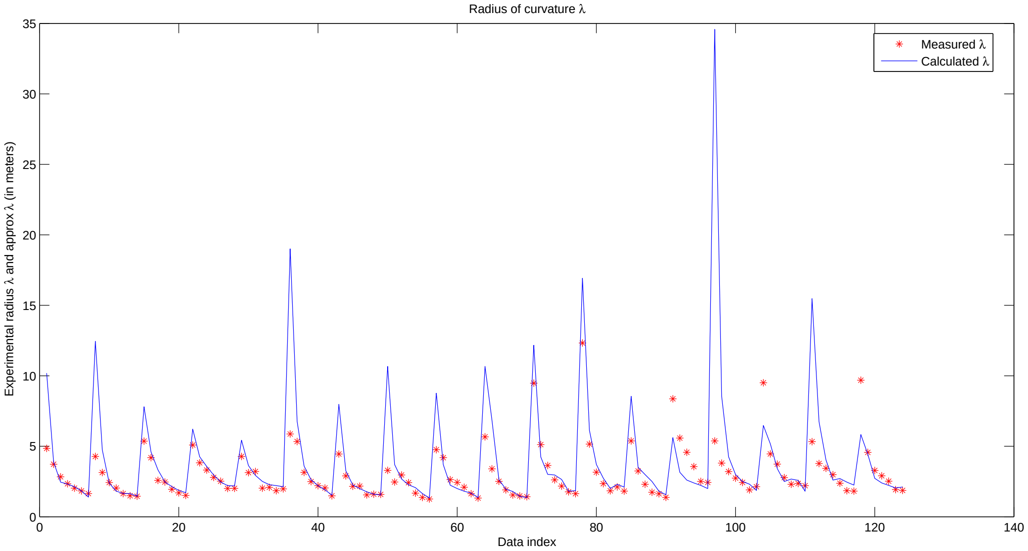

Figure 6.18: Actual and calculated radius of curvature λ for the middle section in 3D space

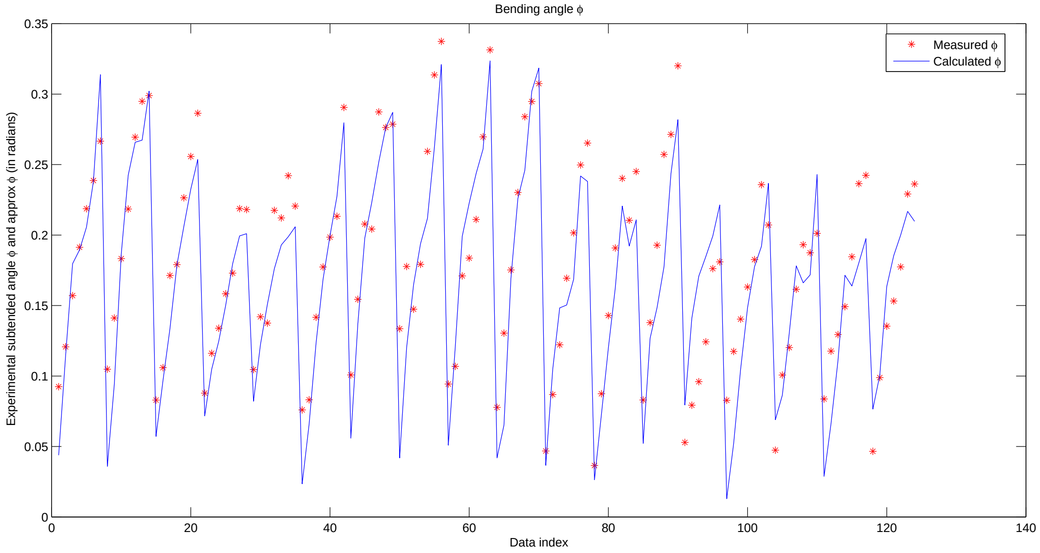

Figure 6.19: Actual and calculated bending angle ϕ for the middle section in 3D space

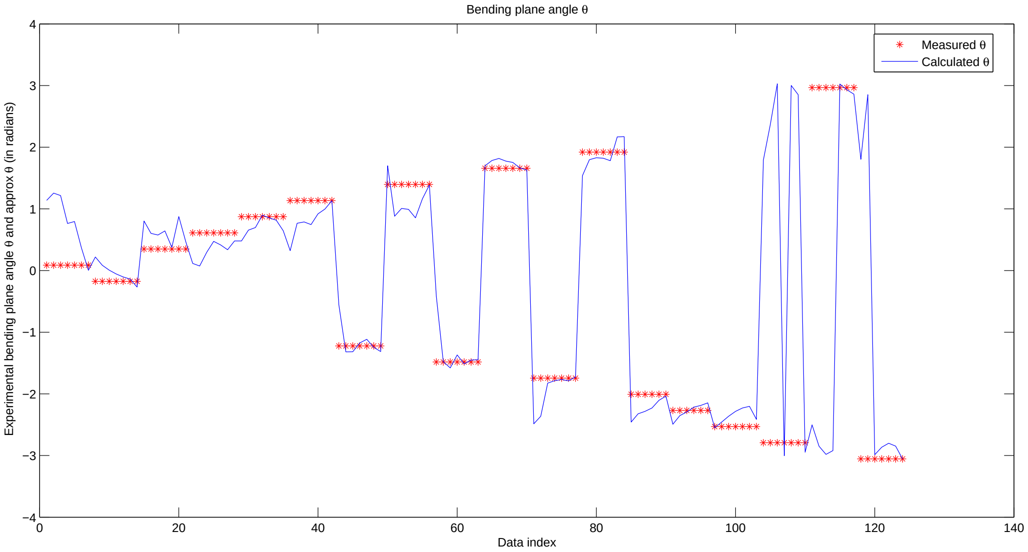

Figure 6.20: Actual and calculated bending plane angle θ for the middle section in 3D space

In general, the results for each shape variable for both the distal and middle sections operated individually as cable robots are good. The deviation in the estimation can be attributed to various unmodeled factors like effect of gravity opposing the bending motion (deviating the hardware from constant curvature) to some extent, encoder noise, slippage at the pulley, the minimal free rotational DOF due to the concentric arrangement of backbone elements, friction between tendons sliding through holes etc. The most prominent is the free rotational DOF. Since the surface of nitinol is very smooth (almost frictionless) it enhances the relative sliding between the backbone sections, but simultaneously results in limited free rotation about the point where the last tendon spacers and spring (for that backbone section) connects to the adjacent backbone section when multiple tendons try to bend the section. However, the rudimentary multiple polynomial regression method is able to track the kinematic shape estimation reasonably well when the only input provided are the encoder values. This implies that the analytic approach used to compute the modeled length changes and the compression variables incorporated in the kinematics to account for the effect of the springs-loaded tendon design is justified.

6.2.1 3D Forward Kinematics Result

The shape variables were then fed into the direct kinematic model obtained in section 5.2.2 of Chapter 5 to obtain the position variables (x, y, z) of the cable tip.

The x,y and z coordinates indicating the position of the tip of the robotic cable were computed, considering the robot as a single section continuum cable consisting of the distal section alone. Figures 6.21-6.23 depict the results. As in 2D, the actual coordinates were computed by substituting experimental shape variables in 3D HTM obtained previously in (5.44), while the model coordinates were computed by substituting the modeled shape variables in the 3D HTM obtained in (5.45).

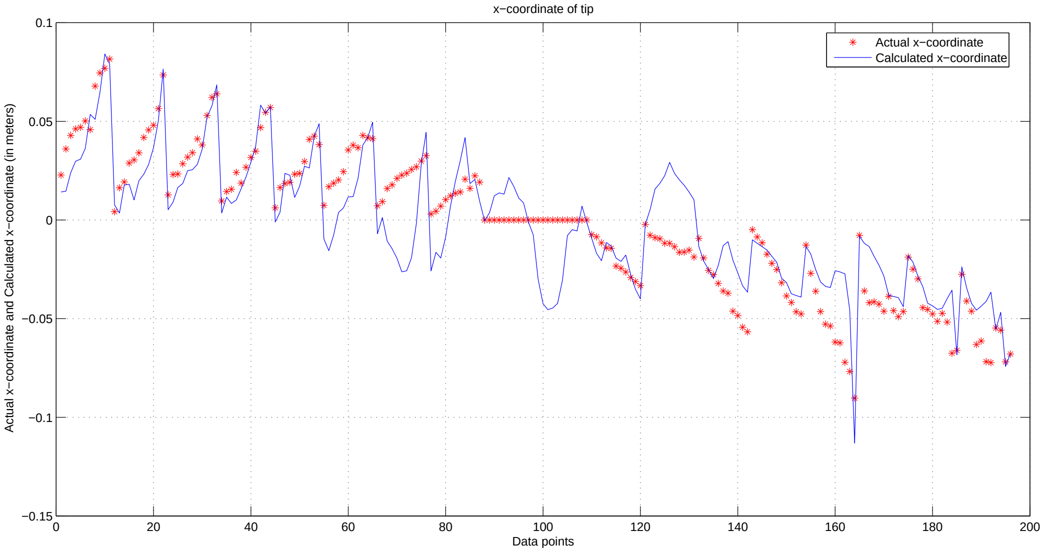

Figure 6.21: X-coordinate of distal section tip in 3D space

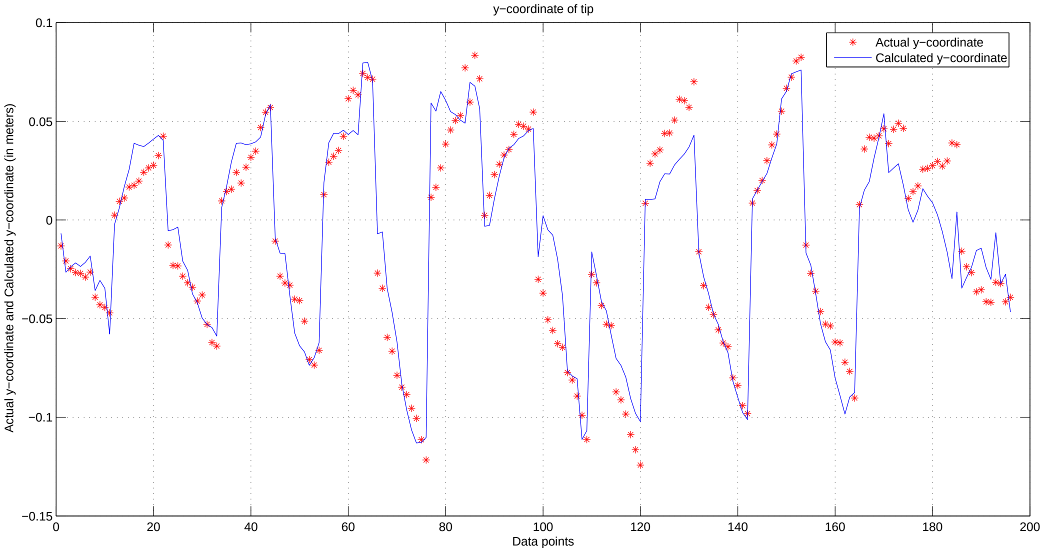

Figure 6.22: Y-coordinate of distal section tip in 3D space

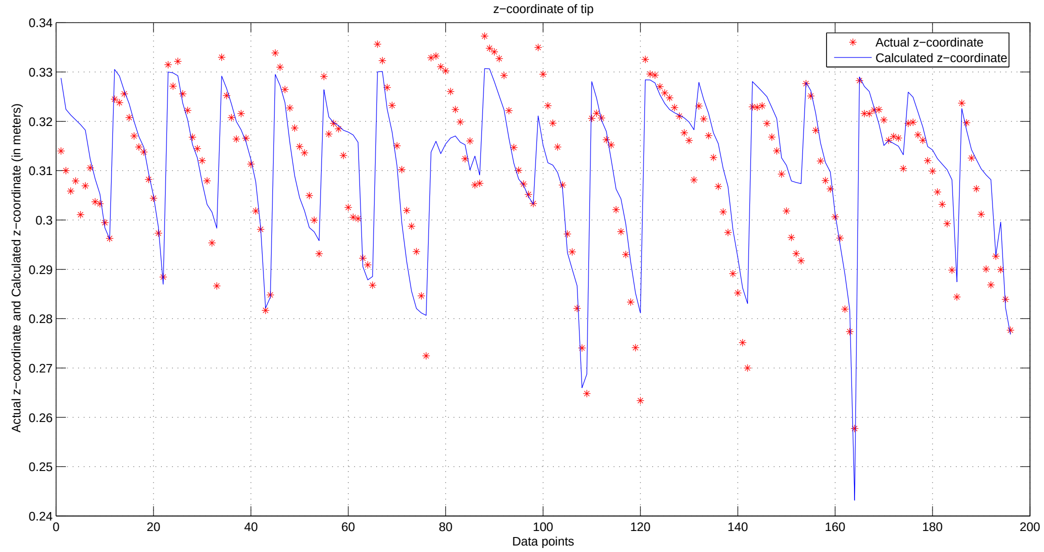

Figure 6.23: Z-coordinate of distal section tip in 3D space

The same procedure was repeated for the middle section. Figures 6.24-6.26 depict the results.

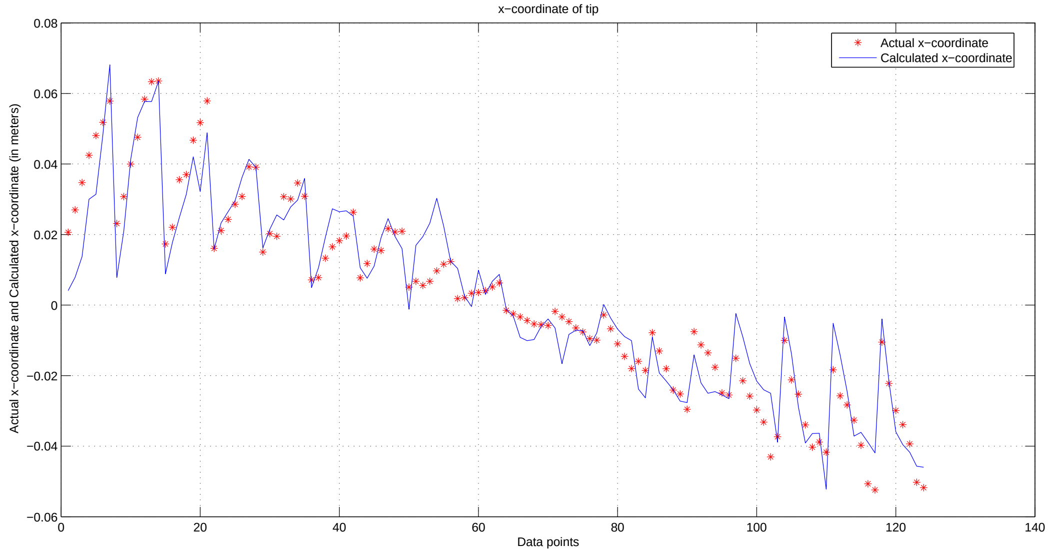

Figure 6.24: X-coordinate of middle section tip in 3D space

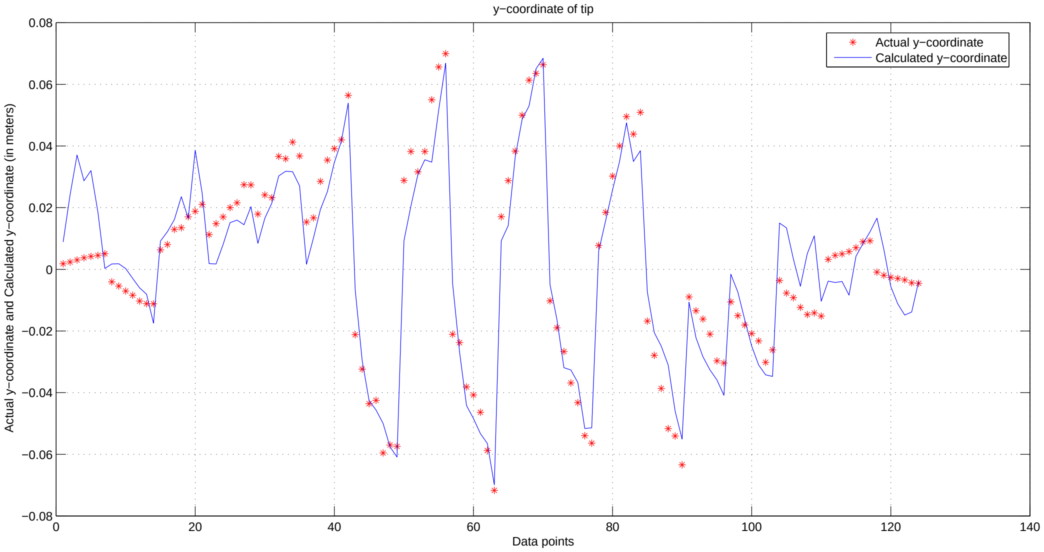

Figure 6.25: Y-coordinate of middle section tip in 3D space

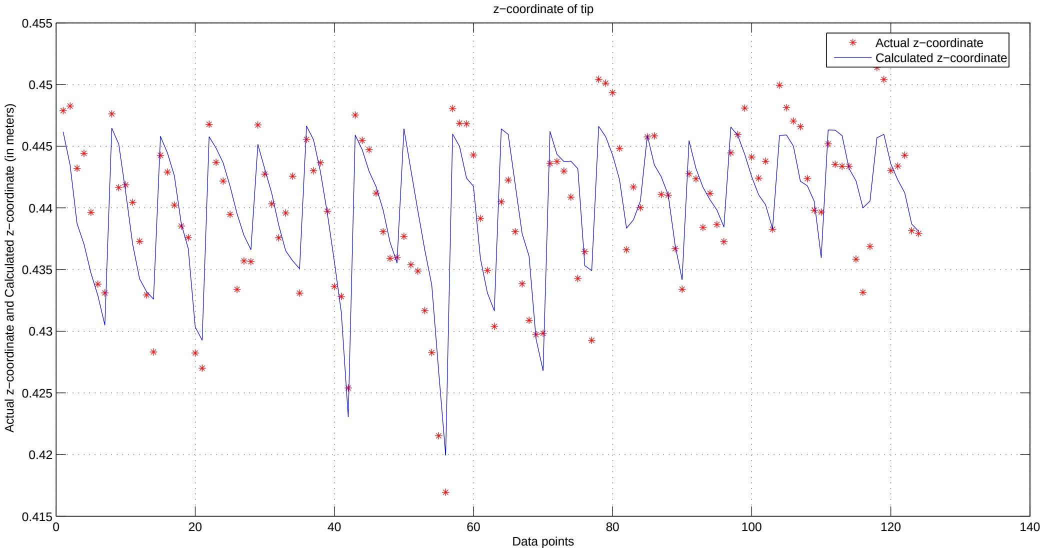

Figure 6.26: Z-coordinate of middle section tip in 3D space

The model predicts the location of all three coordinates for the cable tip quite well, for each individual section under consideration. It can be used to make the robot achieve an orientation agreeably close to the desired orientation.

As in the 2D case, the robotic cable was finally operated as a two section continuum manipulator cable and the 3D HTM for the multi-section version, as a function of the encoder inputs, as in (5.46), was computed. Figures 6.27-6.29 show the results for this two section version of the cable, which demonstrates very good correspondence between model and hardware.

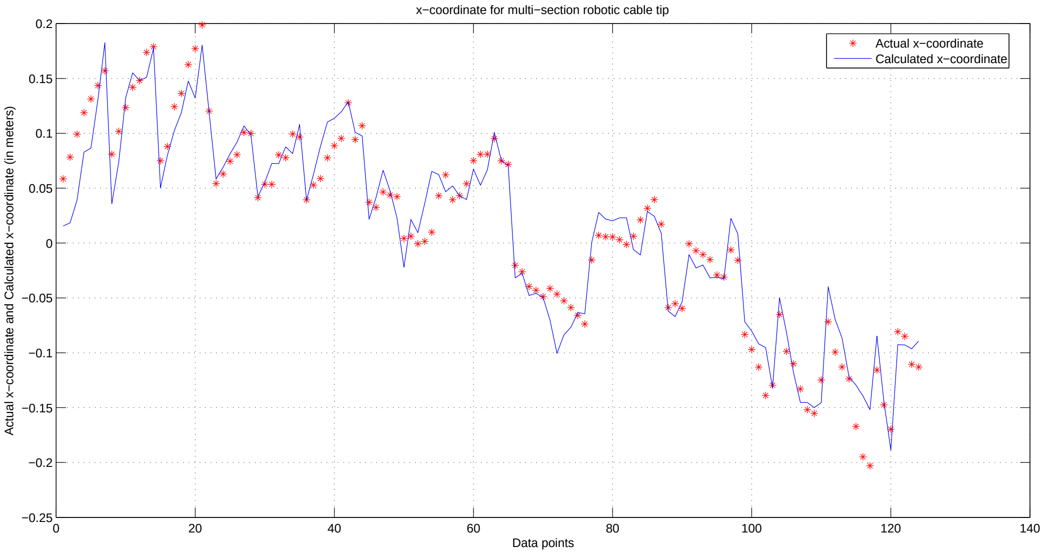

Figure 6.27: X-coordinate of multi-section cable tip in 3D space

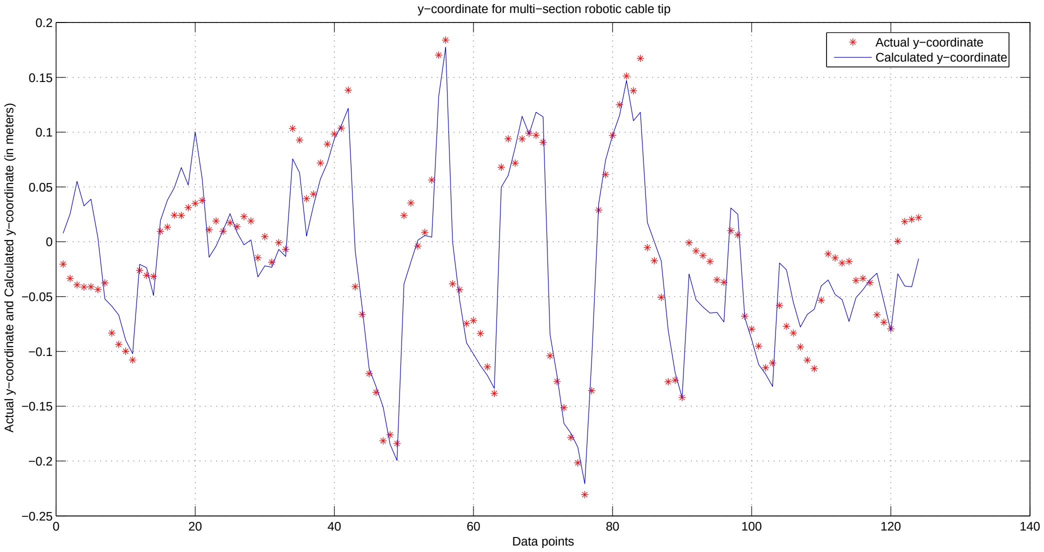

Figure 6.28: Y-coordinate of multi-section cable tip in 3D space

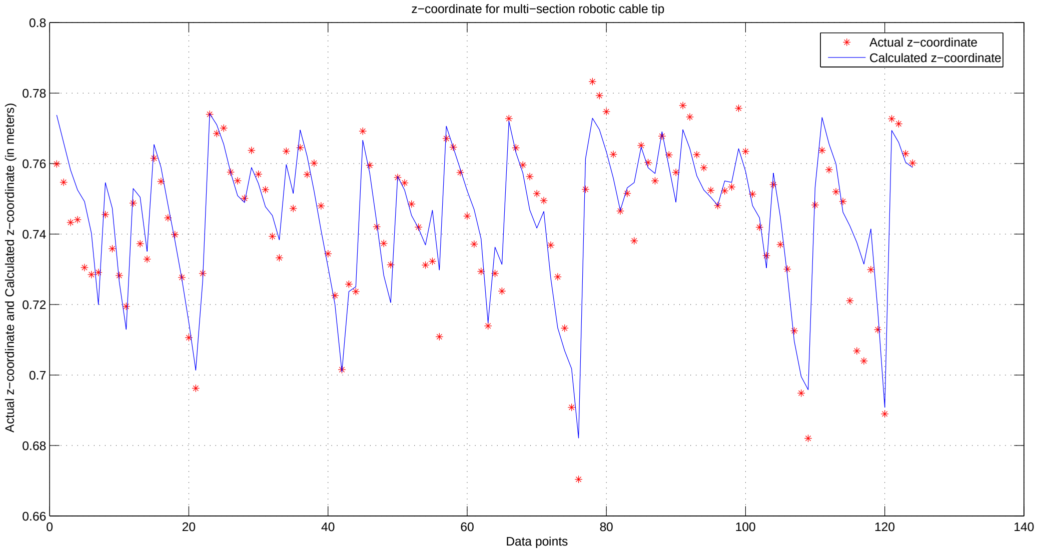

Figure 6.29: Z-coordinate of multi-section cable tip in 3D space

In general, for all the cases depicted in this subsection of this Chapter, the errors in the model can be attributed to those resulting from the assumptions made in the shape variable estimation, as mentioned earlier and discussed further in Chapter 7. Using these coordinates as starting point references, another mode of sensory feedback (say vision) can be employed and a suitable control system could be designed to drive the cable to a more precise position, if desired.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

The objectives of the research presented in this thesis were to design a novel long (more than 100 cm) and thin (less than 1 cm diameter) continuum robotic cable suitable for NASA's needs, the construction of its first prototype and it's actuation assembly and finally validating a new kinematic model derived to predict its performance in both planar and spatial work-space operation. The successful attainment of all these goals were discussed in detail in the preceding Chapters. In this Chapter, we summarize the results and provide suggestions to augment this research.

Chapter 2 provides an extensive review of the present state of the art in the field of continuum, snake and corresponding hybrid robots from the point of view of suitability for a new design with space application and sizing constraints in mind. This review helps to narrow and shortlist the most satisfactory and feasible design approach for an improved space-capable cable robot. The findings of the review led to our selection of a spring-loaded, tendon actuated design with a concentric style arrangement of backbone elements.

In Chapter 3, the disadvantages of the "Tendril" design, which is NASA's current (and only) version of such a long and thin continuum robot, especially with respect to its mechanical structure were identified and analyzed through simple experiments. A considerable portion of the "Tendril" being passive, uncontrollable compressibility of the backbone, torsion of joints and inability to locally extend or contract were the most prominent drawbacks inferred from the experimental analysis. The analysis gives insight into the approach for a new structural design.

Avoiding the existing disadvantages in the current "Tendril" design analyzed in Chapter 3 and exploiting the possibilities for a potential optimal approach for a new hardware from Chapter 2, we implement simple functional prototypes to verify the feasibility of a new design concept for the cable. This is discussed in detail in Chapter 4. The design consists of three backbone elements arranged in a telescopic concentric arrangement. The distal and middle sections of the cable robot have compression springs held at uniform distance by spacers serving as guides for tendons. Each section has its end spacer attached permanently to the corresponding backbone section. The base section is devoid of springs and only has tendon spacers at a fixed distance (all attached permanently to the section). When the tendons are pulled for a particular section, the design produces (constant-curvature) bending of that section in a plane corresponding to the direction of the actuated tendon. The compression springs and telescoping nature of the backbone enable limited but controlled linear motion of the (distal and middle) backbone sections relative to each other. The first prototype analyzed in Chapter 4 is a planar version of the cable to validate the concept's prospect with manual actuation. The refined prototype is a spatial version incorporating the new features at desired sizing and developing an actuation mechanism to drive the cable robot. The actuation assembly consists of nine RC servos, nine encoders and an Arduino microcontroller. These two prototypes are the first of this type of continuum manipulator. The design features a compact actuator package ($5.05\times10^{-3}m^3$) in comparison with most of the current state of the art continuum robots used for Minimally Invasive Surgery (eg: CREST [46] requires a space of about 0.1026 m^3). This novel design concept and its implementation with its space-saving actuation system is one of the main contributions of this thesis to the construction of multi-section, long and thin continuum cable-like robots.

Chapter 5 evaluates the need for a new kinematic modeling method arising out of the spring loading and concentric nature of the new robotic cable design. A new kinematic model is introduced. The first step involves introduction of compression (c) as a variable in the established formulation of continuum variables (λ, ϕ and θ) and its computation. This is augmented by the frequent use of *observed* and *modeled* length change terminology introduced in this Chapter. Experiments are performed with the refined cable prototype collecting data related to the shape configuration of the robot's middle and distal sections to operate in both 2D and 3D workspace. For the 2D arrangement, computation of compression and length change variables is straightforward as it reduces to a simple case and results in a deterministic set of equations representing its forward kinematic variables. In the 2D kinematics, linear regression is used to model the length change and compression as a function of encoder inputs using the corresponding 2D experimental data collected using the prototype. For the 3D arrangement, however, compression and length change variables depend on multiple independent variables leading to an underdetermined system of equations. Hence, an analytic approach based on a logical intuitive assumption is used (i.e. compression is equally distributed in the three length changes and the maximum of the three observed length changes is the compression c). This assumption is later verified to be reasonable when comparing the resulting model to empirical data from the prototype. In the 3D kinematics introduced in Chapter 5, multiple polynomial regression is used to model the length changes and compression as a function of encoder inputs using the corresponding 3D experimental data collected using the prototype. The modeled shape variables are substituted in the 2D and 3D direct kinematic equations so as to predict the position coordinates of the robotic cable tip and thus validate the (new) planar and spatial forward kinematic model's performances corresponding to the cable hardware's operation in 2D and 3D workspace. This new modified kinematic formulation for 2D and 3D operation is the other significant contribution of this thesis for modeling of such long and thin continuum cable-like robots.

In Chapter 6, the results of the new 2D and 3D kinematic formulations for determining continuum (shape) variables are validated by implementing them in in MATLAB®, once using the model (linear and polynomial regression respectively) generated shape variables and then with shape variables extracted from the experimental data. Note that the model generated shape variables were functional approximations of the encoder inputs only. The same procedure was repeated for the forward kinematic equations for 2D and 3D workspace. The results agree well, in general, for both 2D and 3D case. The deviation in the estimation in certain cases in 2D operation, is due to the orientation of tendons, fiction on the experimental surface and the unmodeled effects of motor and sensor systems. In 3D the estimation is less accurate at certain data points due to the effect of gravity opposing the bending motion (deviating the hardware from constant curvature) to some extent, the minimal free rotational DOF due to the concentric arrangement of backbone elements, friction between tendons sliding through guides, etc. Thus, introduction of the new compression variable in the kinematic formulation and linear and polynomial regression methods to estimate a given configuration as a function of encoder counts gives effective baseline results to evaluate the performance of this new kind of robotic cable.

7.2 Suggestions for Future Work

The research work proposed in this thesis is an initial investigation into space-capable continuum cable robots. This section provides ideas and suggestions which could be implemented to extend and improve different aspects the work. The potential improvement areas include construction of refined prototypes of the robotic cable using this design concept, refined actuation module designs with additional features, incorporation of a vision system and dedicated modeling and control algorithms.

The nitinol backbone elements used in the current prototype were procured by trial and error since except size, we found no information about their physical and material properties as a reference. This was also the case with the compression springs. Judicious selection of both nitinol and springs helped to provide reasonable bending as well as local compression/extension in the refined prototype. However, more research would be helpful in selection of nitinol rods/tubes in terms of their material, size and physical properties. Nitinol has a convenient combination of super-elastic bending and strength exists and is available commercially. A more systematic selection of compression springs having more optimal stiffness and size can play a significant role in superior performance of the cable robot. The current prototype can serve as a reference for future selection of nitinol and springs. A relatively significant improvement to the current design should be the method used to fix the plastic spacers onto the nitinol. Adhesive (a combination of Epoxy and Superglue) was used for the current refined prototype of the cable due to the sizing restraint. This adhesive is only just sufficient to hold the spacer with nitinol, when the tendons are actuated. However, it restricts the evaluation of the maximum bending/breaking capacity of the nitinol, since at high bending the spacer can seperate. An alternative super-strong adhesive (matched to the two surfaces in contact) or another approach needs to be developed without increasing the size of the cable robot. Nylon fishing lines used as tendons herein, though thin and strong, show signs of stretching after continuous and prolonged use. This can affect the accuracy of the encoder readings for a given configuration. An alternative for the tendon material will also be helpful. The selection of plastic used for spacers can be investigated more to reduce friction (as they slide along with the springs on the backbone as well) between tendons and their guide holes on the periphery of the spacers.

A suitable soft covering sheath can be added to enclose the entire length of cable and thereby protect the tendons, spacers and the nitinol backbone from inadvertent damage. An actuator unit consisting of a coiling mechanism on the lines of the “Tendril” housing [2] can be constructed. Since the nitinol can be easily bent and coiled, such a system will be useful to simplify the storage and mobility of the robotic cable for practical use in space or terrestrial applications in future. It will also enable rotational base motion of the entire cable as in the original “Tendril”. Two additional motors (brushless DC preferred for their efficient operation) would be required, one for coiling and other for the rotation of the entire cable base. Once the maximum bending capacity of the nitinol (and/or breaking capacity of tendons/spacers) is estimated for each section, motors with higher torques can be employed for more bending of each section. Simple line sensors (as encoders) for sensing position were used herein due to their compact size. However, better and more accurate position sensors can also be employed. String encoders can also be used to obtain the observed lengths directly instead of relying on encoder counts. This would avoid the analytic (assumption) method used for estimating compression as then the system will no longer remain under-determined.

As the cable is to be potentially used for inspection applications in space, a vision system (say a CCD as in “Tendril” [2]) may be incorporated at the robotic cable’s tip. Firstly, such a system will be useful from practical point of view for example, in remote explorations. It may also serve as an additional sensory feedback made to be used as an input in model and control algorithms to be developed in the future to achieve desired shape configuration and navigate more precisely.

The forward kinematic model developed in Chapter 5 is one possible method of modeling. Other alternative approaches (for forward as well as inverse kinematics) may also be verified to find the most optimal/accurate method. Dynamics and models of torsion have not been utilized in the current strategy. In fact, a torsion model may be useful to minimize the error due to the free rotational DOF in every section. The effect of gravity can also be added in the model for superior spatial performance. A suitable control algorithm is needed to use feedback (from the position and/or vision sensors) to ensure precise navigation and manipulation through a given unstructured environment.

Considering the suggestions enumerated above, the design strategy for this new class of multi-section continuum cable robot and its novel kinematic modeling introduced in this thesis to predict its performance can be used as a baseline to ultimately realize a “rope-like” robotic cable with huge potential for deployment in real-life space and terrestrial scenarios of need in the near future.

Appendices

Appendix A 3D Experimental Data for Distal Section

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
6	0	2	9.572918331	188.8002913	-30
11	0	10	15.27775374	117.6496909	-30
14	0	10	18.37657928	97.01948276	-30
16	0	10	19.63947292	91.61207997	-30
17	0	11	20.42112682	86.28957374	-30
19	0	11	21.38982709	84.15602047	-30
27	0	13	19.32584763	93.84543672	-30
31	0	21	28.93697296	62.76525066	-30
36	0	21	31.67482397	57.75802124	-30
45	0	26	33.01057556	54.96868194	-30
50	0	39	35.25664668	51.32523778	-30
3	0	0	1.716550902	1083.280135	30
3	2	0	6.661866151	279.1046435	30
9	5	0	7.822772318	239.2012672	30
12	9	0	11.86827588	155.9625872	30
14	18	0	12.64843765	144.7885805	30
23	26	0	14.2396521	127.9737978	30
28	30	0	17.48583867	104.4263132	30
36	40	0	19.39949737	92.79638634	30
40	48	0	20.63941915	86.35969513	30
44	57	0	24.74456212	71.03096484	30
51	73	0	32.79439503	53.25448686	30
3	0	1	6.212192082	306.3270573	-45
4	0	1	11.38482351	165.7209861	-45
6	0	1	11.33634942	168.9767936	-45
11	0	8	14.11980841	133.4587542	-45
15	0	11	15.92156894	117.4605561	-45
19	0	17	17.31321751	106.4485417	-45
22	0	20	20.92507415	88.04889787	-45
29	0	26	19.55640991	93.21723819	-45
38	0	31	27.32293549	67.08729964	-45
41	0	33	33.12310254	54.04967839	-45
47	0	38	35.00168103	49.97079651	-45
3	2	0	4.688951325	407.3401767	45

Encoder	***Encoder***	***Encoder***	***Angle ϕ***	***Radius λ***	***Angle θ***
Count σ_1	***Count σ_2***	***Count σ_3***	***(deg)***	***(cm)***	***(deg)***
7	5	0	7.162976676	260.8145729	45
9	10	0	7.856279323	234.6429811	45
14	18	0	12.28707899	148.678111	45
19	23	0	9.399669457	196.8908631	45
25	29	0	13.60912272	134.5444382	45
30	37	0	16.38678327	110.3555303	45
35	44	0	18.54919407	94.87509469	45
40	55	0	25.05025753	70.40546849	45
45	68	21	30.66461396	55.23219287	45
45	71	25	31.58757679	54.37274032	45
2	0	2	4.241893733	451.3663596	-60
5	0	5	11.34809364	168.2033399	-60
12	0	7	12.90917665	146.128836	-60
18	0	17	13.49780301	138.2568449	-60
20	0	26	16.55611399	111.8233176	-60
27	0	32	17.05903708	107.3353504	-60
35	0	37	21.38890729	85.99287324	-60
44	0	47	29.9912073	61.00707695	-60
51	0	51	31.62339602	57.20555028	-60
57	0	57	29.18747135	60.1151254	-60
1	5	0	5.167453799	365.3804616	60
2	11	0	12.14636651	150.8556604	60
6	15	0	13.29191887	139.0028658	60
12	20	0	14.5438233	126.8243097	60
14	24	0	17.78135849	102.5189811	60
17	26	0	26.39046881	68.06722873	60
17	29	0	28.32731933	63.33983368	60
21	34	0	27.41142135	65.22344274	60
22	46	12	32.68152635	54.12382131	60
22	48	17	31.97723667	54.92927117	60
22	50	23	32.05475492	54.03326594	60
0	0	1	9.500796766	203.3435502	-75
2	0	1	12.31246029	155.8333141	-75
4	0	11	21.35818095	89.74673139	-75
5	0	16	24.0549603	79.29619206	-75
6	0	23	29.00964746	64.9648111	-75

Encoder	***Encoder***	***Encoder***	***Angle ϕ***	***Radius λ***	***Angle θ***
Count σ_1	***Count σ_2***	***Count σ_3***	***(deg)***	***(cm)***	***(deg)***
6	0	35	31.59388331	59.21244157	-75
6	0	45	33.74639597	54.350938	-75
6	0	56	36.6057628	50.0980929	-75
6	0	70	39.04561411	46.60560021	-75
6	0	85	44.11534934	40.88852801	-75
6	0	91	49.61499516	35.76844313	-75
2	19	0	4.083643229	467.4551858	75
5	20	0	5.877146471	325.4430574	75
6	30	0	9.42334651	202.204482	75
9	33	0	13.73736471	139.0601744	75
13	38	0	16.47008375	115.000693	75
14	45	0	18.38500531	102.2200135	75
16	47	0	19.45064431	96.05636044	75
16	52	0	28.65704011	65.14470388	75
17	53	13	22.13496544	83.87309925	75
17	54	13	31.41728019	58.91480057	75
17	54	29	27.08543361	67.5213872	75
1	0	0	0.792530485	2438.345197	90
2	0	0	4.272200334	449.445488	90
6	3	0	7.872854113	243.8946621	90
9	7	0	9.707299682	197.3187676	90
12	12	0	11.40620246	166.5123557	90
25	25	0	12.64979662	147.117701	90
37	29	0	15.68319397	116.4129147	90
42	37	0	17.77501101	101.56726	90
44	40	0	17.57875203	101.7465238	90
48	44	0	17.15541711	103.4601473	90
51	49	0	20.43187748	86.89213907	90
0	4	11	10.27204916	187.8505776	-90
0	14	16	12.84678936	148.2187178	-90
0	22	25	17.7788094	105.8395297	-90
0	24	27	19.86938563	94.03779379	-90
0	26	35	22.53133732	82.15220518	-90
0	35	47	23.74429103	76.26430188	-90
0	47	54	29.17816978	60.9508224	-90
0	52	68	30.89768409	57.16428355	-90

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
0	60	81	35.11862922	49.03281199	-90
0	69	87	39.72232599	42.88224715	-90
0	76	96	45.59240736	37.07024028	-90
1	0	4	10.19073903	181.2015355	-105
1	0	8	11.7118342	158.4494703	-105
2	0	13	15.94351349	116.7428344	-105
6	0	16	19.63248836	94.12893757	-105
8	0	22	19.93824636	92.43436271	-105
8	0	28	33.255695	55.08757977	-105
8	0	30	35.20220974	51.62970773	-105
14	0	36	38.33152679	47.23957907	-105
14	0	44	43.36205545	41.27001276	-105
14	0	51	47.49209569	37.18345287	-105
14	0	56	52.05329882	33.40024478	-105
2	3	0	3.010318955	633.2511052	105
4	3	0	10.32058641	183.9642799	105
7	3	0	12.00998008	158.2800371	105
10	6	0	12.82571785	147.3159801	105
13	8	0	15.86565113	119.1572788	105
16	8	0	16.00672009	117.7705987	105
16	11	0	18.43186128	102.0895992	105
16	13	0	22.30912519	84.5763522	105
16	15	0	22.2967314	83.72932686	105
16	18	0	21.16138559	87.56671639	105
16	24	0	26.48853356	69.08263064	105
0	0	4	6.599567502	281.1378426	-120
0	0	8	13.68176027	135.5006438	-120
0	0	11	18.29798414	100.9969205	-120
0	0	15	20.07715008	91.07659764	-120
3	0	18	23.667583	76.42372974	-120
9	0	24	26.8668609	66.74558888	-120
12	0	28	28.00178852	63.35904588	-120
13	0	37	35.46536407	49.83041205	-120
13	0	43	37.52096066	46.82711906	-120
13	0	50	43.10863496	40.25993693	-120
13	0	54	45.55151063	37.82175148	-120

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
0	3	0	3.489443488	530.5843199	120
0	4	0	6.101767897	303.7123757	120
0	5	0	8.167803114	227.442945	120
0	6	0	12.397808	148.8436173	120
0	8	0	15.78140245	116.4835647	120
0	10	0	18.19075526	100.5812996	120
1	19	0	23.24481175	78.37137538	120
1	21	0	28.63080166	62.98731648	120
1	27	0	31.49066481	56.75476689	120
1	28	0	35.20168696	50.85928981	120
1	29	0	36.13911977	49.46815045	120
0	0	4	6.293608856	298.8681524	-135
0	0	6	13.42549391	140.0385598	-135
0	0	11	18.2393267	101.6588336	-135
0	0	17	23.771265	77.39014887	-135
0	0	21	27.24608278	67.27567708	-135
0	0	23	27.86879263	65.52334525	-135
6	0	33	32.44624058	56.03801925	-135
8	0	40	33.1154681	54.23908896	-135
8	0	48	39.76625985	44.0752094	-135
33	0	57	42.76338588	40.85208226	-135
33	0	74	52.68296927	32.40554657	-135
0	2	0	3.830161146	491.4263302	135
0	4	0	17.97851538	104.1787153	135
0	5	0	20.88559249	90.19497026	135
0	8	0	20.61920204	91.51908314	135
0	11	0	21.18205176	89.21792207	135
0	15	0	23.07174705	81.73140307	135
0	15	9	19.63648618	94.07682114	135
0	16	9	23.17659809	80.52710371	135
0	17	9	24.70133751	75.76391742	135
0	21	15	23.88197706	76.67685061	135
0	4	3	7.76921872	236.4007468	150
0	4	5	10.29214025	178.9917256	150
0	7	7	12.40708502	147.6644265	150
0	11	8	18.44321692	99.93862394	150

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
0	16	14	19.04079734	95.64118304	150
0	17	16	20.11668483	90.10269807	150
0	21	21	21.94902348	81.78177391	150
0	24	26	20.43056351	86.85110391	150
0	30	31	22.59420414	77.88927203	150
0	32	40	30.11364701	57.77118013	150
3	37	61	30.04789494	56.79755533	150
0	0	10	11.20648511	166.5458184	-150
0	4	15	16.88770583	110.0567501	-150
0	9	21	19.38668319	94.15931485	-150
0	15	25	26.76175223	68.03389706	-150
0	25	31	26.46529243	67.57515899	-150
0	26	39	31.89938081	54.88955618	-150
0	30	45	32.45075595	53.45914051	-150
6	51	46	24.37619923	70.90913425	-150
6	56	50	25.08498893	68.38883803	-150
6	61	71	32.6039347	52.69251483	-150
6	73	77	31.54470738	53.07262239	-150

Table 1: 3D experimental data for distal section

Appendix B 3D Experimental Data for Middle Section

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
2	8	0	485.1021058	5.297555948	5
8	24	0	372.2719004	6.915793638	5
19	39	0	283.3675928	8.998545633	5
32	42	0	233.6706889	10.96389813	5
38	54	0	202.6516109	12.52959634	5
50	66	0	183.4643688	13.67778996	5
60	76	0	164.3954805	15.27510179	5
2	0	0	427.8097595	6.005923759	-10
7	0	0	313.9071212	8.087836563	-10
17	0	0	242.4445803	10.50118795	-10
26	0	0	203.1957787	12.51893279	-10
32	0	0	164.2467859	15.44082351	-10
41	11	0	148.9466213	16.89784412	-10
50	25	0	145.4218802	17.1293702	-10
4	8	0	536.1889958	4.752706434	20
9	11	0	418.8624857	6.069745587	20
15	16	0	258.1720476	9.817764524	20
28	32	0	246.2455709	10.25831265	20
39	41	0	194.9180406	12.97346192	20
45	71	0	169.3005687	14.65181877	20
55	80	0	151.1223821	16.41278528	20
5	0	0	509.1047902	5.034588209	35
8	0	0	382.7687964	6.656494716	35
12	7	0	331.2020017	7.672263057	35
18	16	0	278.6404777	9.074594029	35
26	22	0	251.7410043	9.913224182	35
34	30	0	200.7451319	12.53533103	35
38	43	0	201.3181418	12.49730019	35
7	7	0	427.813758	5.993711714	50
13	16	0	312.7918735	8.137111331	50
19	23	0	321.1665913	7.88010358	50
25	36	0	202.7312755	12.46471309	50
32	45	0	208.7313123	12.15759828	50
35	49	0	184.5641388	13.87436702	50

Encoder	***Encoder***	***Encoder***	***Angle ϕ***	***Radius λ***	***Angle θ***
Count σ_1	***Count σ_2***	***Count σ_3***	***(deg)***	***(cm)***	***(deg)***
40	53	0	197.9092466	12.64039458	50
1	0	0	587.2136323	4.351478968	65
5	9	0	533.1787039	4.766106213	65
13	19	0	314.1799191	8.117951896	65
24	30	0	249.1954265	10.16362018	65
33	49	0	219.8369391	11.37616062	65
40	65	0	204.428342	12.22320128	65
49	87	0	148.4724656	16.6493694	65
7	0	5	444.7136202	5.775595323	-70
17	0	15	289.6062797	8.848342036	-70
28	0	20	215.5638419	11.9059615	-70
37	0	20	217.4911852	11.70398231	-70
49	0	23	154.5718289	16.46384156	-70
60	0	36	159.7635045	15.83356719	-70
62	0	40	158.5080965	15.96508312	-70
0	9	0	328.6186621	7.654062735	80
12	20	0	246.3093664	10.18128507	80
20	33	0	296.0734277	8.446366891	80
29	45	0	242.170122	10.26798497	80
39	58	0	166.9968487	14.85950046	80
43	77	0	136.636992	17.96872814	80
43	87	0	125.9423806	19.33303597	80
6	0	4	475.5983745	5.405817681	-85
13	0	15	419.0909198	6.12090619	-85
22	0	24	262.5417499	9.798722469	-85
33	0	24	243.221167	10.5252305	-85
41	0	39	209.5706818	12.09571719	-85
53	0	44	163.2516716	15.45109346	-85
66	0	46	132.2881998	18.9863162	-85
0	9	0	567.0417927	4.455382026	95
0	15	0	339.9964335	7.474006506	95
0	45	0	251.2706708	10.04040126	95
0	65	0	190.1134218	13.19134799	95
0	73	0	153.8125409	16.26837985	95
0	99	0	147.9295582	16.8873613	95
0	108	0	142.0194495	17.61738909	95

Encoder	***Encoder***	***Encoder***	***Angle*** ϕ	***Radius*** λ	***Angle*** θ
Count σ_1	***Count*** σ_2	***Count*** σ_3	***(deg)***	***(cm)***	***(deg)***
0	0	6	947.4871847	2.683489314	-100
0	0	15	511.5868097	4.976122036	-100
10	0	20	363.6099095	6.997708855	-100
11	0	20	261.4846842	9.706695784	-100
13	0	22	216.8853175	11.55065502	-100
21	0	39	176.6123126	14.30742331	-100
26	0	40	163.7605233	15.19621755	-100
0	5	0	1232.789589	2.093906499	110
0	17	0	515.5855652	5.008379982	110
0	30	0	315.5371753	8.187070347	110
0	43	0	234.5062058	10.93444478	110
0	63	0	183.5129217	13.7634705	110
0	70	22	211.3945533	12.06032217	110
0	75	29	181.3497741	14.04205524	110
0	0	1	885.8656049	2.893367645	-115
0	0	8	537.4459126	4.757584227	-115
0	0	18	324.3553636	7.900435221	-115
0	0	21	230.2120112	11.04627374	-115
0	0	25	173.3872055	14.73583474	-115
0	0	34	162.8880259	15.55090887	-115
0	0	39	137.7522094	18.33839323	-115
0	0	2	1022.030671	2.486365771	-130
0	2	11	836.7270472	3.033279917	-130
0	2	19	558.1515612	4.545704286	-130
0	2	23	457.4714525	5.499241612	-130
0	0	26	356.0253137	7.118244568	-130
0	0	28	250.2610752	10.09483334	-130
0	0	31	242.955839	10.36821787	-130
0	0	3	537.4717852	4.743464117	-145
0	0	8	380.6408523	6.728073173	-145
0	0	15	320.3305367	8.041204575	-145
0	0	21	273.4237715	9.34787576	-145
0	0	25	243.6408987	10.46201127	-145
0	0	27	190.0296419	13.50527926	-145
0	15	27	213.0857711	11.86927931	-145
0	16	0	950.1364609	2.714325726	-160

Encoder	***Encoder***	***Encoder***	***Angle ϕ***	***Radius λ***	***Angle θ***
Count σ_1	***Count σ_2***	***Count σ_3***	***(deg)***	***(cm)***	***(deg)***
0	25	4	445.8922219	5.767955505	-160
0	34	12	372.892918	6.885372984	-160
0	41	20	277.6709035	9.25516225	-160
0	50	20	230.421757	11.06849107	-160
0	58	24	236.0164766	10.73968992	-160
0	58	39	219.9961214	11.5281216	-160
0	0	5	532.4412328	4.796280274	170
0	8	8	377.6982193	6.743870842	170
0	20	11	343.5486026	7.415123662	170
0	34	18	298.2106935	8.550195347	170
0	48	19	237.4263642	10.57761521	170
0	59	27	184.8871272	13.54888859	170
0	66	34	182.0037666	13.88848166	170
0	18	0	968.610191	2.671026384	-175
0	27	8	456.2574583	5.665370363	-175
0	35	17	328.31044	7.755246805	-175
0	35	20	290.5998764	8.776361217	-175
0	35	22	251.691583	10.16674844	-175
0	44	27	192.8699096	13.13052749	-175
0	56	32	187.0950408	13.53647604	-175

Table 2: 3D experimental data for middle section

Appendix C Polynomial Coefficients used for 3D Modeling

$$F_{distal} = \begin{bmatrix} f_{c_3} & f_{l_{31}} & f_{l_{32}} & f_{l_{33}} \end{bmatrix} = \begin{bmatrix}
-0.0091563 & -6.89E-05 & -0.00011641 & -0.00019856 \\
-0.00045909 & 8.32E-07 & -8.36E-06 & -8.40E-05 \\
3.65E-06 & -4.30E-07 & 2.06E-07 & 7.37E-07 \\
-0.000773 & 2.15E-06 & -9.08E-05 & 2.14E-05 \\
2.07E-05 & 4.28E-07 & 1.63E-06 & 1.59E-06 \\
-8.06E-08 & 1.03E-08 & -2.57E-08 & -1.25E-08 \\
1.79E-05 & 2.74E-08 & 1.14E-06 & -4.39E-07 \\
-6.59E-07 & -2.26E-08 & 8.51E-09 & -2.42E-08 \\
3.81E-09 & 8.26E-11 & 1.33E-11 & 1.90E-10 \\
3.48E-05 & -7.70E-05 & 5.48E-08 & 9.41E-06 \\
-1.65E-05 & 2.22E-06 & 1.79E-06 & 2.31E-06 \\
9.62E-08 & 9.75E-09 & -3.88E-08 & -5.02E-08 \\
2.77E-05 & 2.45E-06 & 3.72E-06 & -1.16E-06 \\
-2.08E-05 & 5.58E-07 & -5.23E-07 & 4.43E-07 \\
2.50E-07 & -9.31E-09 & 4.52E-09 & -6.81E-09 \\
-9.24E-07 & -6.91E-08 & -5.44E-08 & 2.56E-08 \\
4.24E-07 & -7.24E-09 & 5.30E-09 & -6.51E-09 \\
-5.15E-09 & 1.48E-10 & -5.24E-11 & 1.11E-10 \\
-1.24E-05 & 6.45E-07 & -6.08E-07 & -1.01E-07 \\
9.59E-07 & -6.18E-08 & -1.44E-09 & -2.84E-08 \\
-1.03E-08 & 4.10E-10 & 3.32E-10 & 7.48E-10 \\
-9.96E-08 & -5.40E-09 & -3.23E-08 & 1.38E-08 \\
5.78E-07 & -4.06E-08 & 5.28E-09 & -1.69E-08 \\
-1.06E-08 & 7.15E-10 & 3.05E-10 & 3.14E-10 \\
9.58E-09 & 7.60E-10 & 5.98E-10 & -3.24E-10 \\
-1.13E-08 & 6.37E-10 & -6.67E-11 & 2.61E-10 \\
1.98E-10 & -1.32E-11 & -3.72E-12 & -5.26E-12
\end{bmatrix}$$

$$F_{middle} = \begin{bmatrix} f_{c_2} & f_{l_{21}} & f_{l_{22}} & f_{l_{23}} \end{bmatrix} = \begin{bmatrix} -0.0030884 & -0.00014095 & -0.00012441 & -9.85E-05 \\ -6.72E-05 & 1.38E-05 & 1.60E-07 & -1.67E-05 \\ -2.57E-06 & -3.49E-07 & 2.99E-08 & -1.65E-07 \\ -1.83E-05 & 1.07E-06 & -1.17E-05 & 2.73E-06 \\ 2.00E-05 & 2.71E-07 & 2.55E-07 & -1.02E-06 \\ -7.65E-07 & -5.97E-09 & -7.58E-09 & 1.30E-08 \\ -8.78E-07 & -4.38E-08 & 1.80E-08 & -1.76E-08 \\ -2.92E-07 & -5.63E-09 & -2.56E-09 & 1.75E-08 \\ 1.27E-08 & 1.70E-10 & 8.70E-11 & -2.12E-10 \\ -0.0001025 & -3.56E-05 & 5.69E-06 & 7.49E-06 \\ 2.33E-05 & 1.00E-06 & 3.68E-07 & -1.48E-06 \\ -5.61E-07 & -5.64E-09 & -1.05E-08 & 3.90E-08 \\ -1.18E-06 & 5.80E-07 & -3.25E-07 & -3.01E-08 \\ 0.067439 & 0.067439 & 0.067439 & 0.067439 \\ 0.016864 & 0.016864 & 0.016864 & 0.016864 \\ -6.05E-08 & -3.09E-09 & -2.71E-09 & -1.03E-09 \\ 0.085192 & 0.085192 & 0.085192 & 0.085192 \\ 0.083521 & 0.083521 & 0.083521 & 0.083521 \\ -3.73E-06 & 3.03E-07 & -1.31E-07 & -3.11E-07 \\ -3.98E-08 & -1.82E-08 & -3.41E-09 & 2.81E-08 \\ 4.36E-09 & 2.13E-10 & 1.40E-10 & -6.15E-10 \\ 1.97E-07 & -8.74E-09 & 1.81E-08 & 5.09E-09 \\ 0.074185 & 0.074185 & 0.074185 & 0.074185 \\ 0.0018875 & 0.0018875 & 0.0018875 & 0.0018875 \\ -5.07E-10 & 5.70E-11 & -7.19E-11 & -1.08E-11 \\ 0.019249 & 0.019249 & 0.019249 & 0.019249 \\ 0.018885 & 0.018885 & 0.018885 & 0.018885 \end{bmatrix}$$

Appendix D MATLAB® and Arduino Programs for the Robotic Cable

D.1 MATLAB® code for 2D results of λ, ϕ and s (a): Distal section bending left

```
clc
% Bend left
% data=[counts, subtended angle, lambda, s]
data=[0 0 Inf 34;
    23 24.72 73.34 31.64;
    42 41.62 42.92 31.18;
    64 53.98 33.05 31.13;
    87 60.21 28.905 30.37;
    118 73.73 22.5 28.95;
    158 80.06 19.2 26.83];
data(:,2:4)=[pi/180.*(data(:,2)) .01*data(:,3:4)]
r=.00213; % radius of cross-section
L0=.34; % relaxed unactuated length of tendon
% // Here we derive two length changes, dl: length change
% and c: compression of the spring from the
% measurements made above and fit a*x type fits.

% effective length change, dl for phi is derived
% from the definition of phi as
dl=-3*data(:,2)*r/2
lengthChangeModel=fit(data(:,1),dl,'a*x')
A1= -3.415e-05; A2=0; % linear fit model
% plot(data(:,1),dl,data(:,1),A1*data(:,1)+A2)
% title('Relationship between encoder counts
% and length changes & corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Change in length')
% figure
% % now calculate compression from the definition of S
compression=(3*data(:,4)-3*L0-dl)/3
compressionModel=fit(data(:,1),compression,'a*x')
B1= -0.000443; B2= 0; % linear fit model
% plot(data(:,1),compression,data(:,1),B1*data(:,1)+B2)
% title('Relationship between encoder counts and
compression & corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Compression')
% figure
% % now try to compute lambda from above values
calcLambda=(3*(L0+compression)+dl)*r./(2*dl)
% % plot to see if it matches and
% % from the figure it shows that it matches very well.
% plot(data(:,3),abs(calcLambda))
% title('Radius of curvature comparison \lambda')
% xlabel('Experimental \lambda')
% ylabel('Calculated \lambda')
% figure
% based on these fit models, phi can be given as
approx_phi=abs(2*(A1*data(:,1)+A2)/(3*r))
plot(data(:,1),data(:,2),'r',data(:,1),approx_phi)
title('Subtended angle \phi  Vs Counts')
xlabel('Encoder counts')
ylabel('Calculated subtended angle \phi and approx \phi')
legend('Measured \phi','Calculated \phi','Location','NorthEast')
approx_Lambda=abs((3*(L0+(B1*data(:,1)+B2))+(A1*data(:,1)+A2))*r
./(2*(A1*data(:,1)+A2)))
figure
plot(data(:,1),data(:,3),'r',data(:,1),approx_Lambda)
legend('Measured \lambda','Calculated \lambda','Location','NorthEast')
title('Radius of curvature \lambda  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental radius \lambda and approx \lambda')
figure
approx_s=approx_phi.*approx_Lambda
plot(data(:,1),data(:,4),'r',data(:,1),approx_s)
title('Arc length s  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental arc length s and approx s')
legend('Measured s','Calculated s','Location','NorthEast')
```

D.2 MATLAB® code for 2D results of λ, ϕ and s (b): Distal section bending right

```
clc
% Bend right
% data=[counts, subtended angle, lambda, s]
data=[0 0 Inf 34;
    14 27.47 69.89 33.52;
    31 36.47 51.76 32.95;
    57 51.97 35.6 32.29;
    81 62.73 28.05 30.71;
    107 75.49 22.46 29.59;
    145 86.9 18.612 28.23];
data(:,2:4)=[pi/180.*(data(:,2)) .01*data(:,3:4)]
r=.00213; % radius of cross-section
L0=.34; % relaxed unactuated length of tendon

% // Here we derive two length changes, dl: length change
% and c: compression of the spring from the
% measurements made above and fit a*x type fits.
% effective length change, dl for phi is derived
% from the definition of phi as
dl=-3*data(:,2)*r/2
lengthChangeModel=fit(data(:,1),dl,'a*x')
A1= -3.881e-05; A2=0; % linear fit model
% plot(data(:,1),dl,data(:,1),A1*data(:,1)+A2)
% title('Relationship between encoder counts
% and length changes & corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Change in length')
% figure
% now calculate compression from the definition of S
compression=(3*data(:,4)-3*L0-dl)/3
%compressionModel=fit(data(:,1),compression,'a*x')
B1= -0.000381; B2= 0; % linear fit model
% plot(data(:,1),compression,data(:,1),B1*data(:,1)+B2)
% title('Relationship between encoder counts and compression &
% corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Compression')
% based on these fit models, phi can be given as
approx_phi=abs(2*(A1*data(:,1)+A2)/(3*r))
plot(data(:,1),data(:,2),'r',data(:,1),approx_phi)
title('Subtended angle \phi  Vs Counts')
xlabel('Encoder counts')
ylabel('Calculated subtended angle \phi and approx \phi')
legend('Measured \phi','Calculated \phi','Location','NorthEast')
approx_Lambda=abs((3*(L0+(B1*data(:,1)+B2))+(A1*data(:,1)+A2))*r
./(2*(A1*data(:,1)+A2)))
figure
plot(data(:,1),data(:,3),'r',data(:,1),approx_Lambda)
legend('Measured \lambda','Calculated \lambda','Location','NorthEast')
title('Radius of curvature \lambda  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental radius \lambda and approx \lambda')
figure
approx_s=approx_phi.*approx_Lambda
plot(data(:,1),data(:,4),'r',data(:,1),approx_s)
title('Arc length s  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental arc length s and approx s')
legend('Measured s','Calculated s','Location','NorthEast')
```

D.3 MATLAB® code for 2D results of λ, ϕ and s (c): Middle section bending right

```
clc
%  middle
% data=[counts, subtended angle, lambda, s]
% Bend right
data=[0 0 Inf 45;
    37 9.74 260.64 44.31
    55 15.78 158.43 43.63;
    91 21.5 113.89 42.73;
    118 27.21 88.9 42.25;
    126 30.74 78.28 41.99;
    151 34.56 68.67 41.42];
% Bend left
% data=[0 0 Inf 45;
%     74 18.51 130.55 42.17;
%     128 30.74 78.28 41.99;
%     178 43.09 55.82 41.98];
data(:,2:4)=[pi/180.*(data(:,2)) .01*data(:,3:4)]
r=.00213; % radius of cross-section
L0=.45; % relaxed unactuated length of tendon
% // Here we derive two length changes, dl: length change
% and c: compression of the spring from the
% measurements made above and fit a*x type fits.

% effective length change, dl for phi is derived
% from the definition of phi as
dl=-3*data(:,2)*r/2
lengthChangeModel=fit(data(:,1),dl,'a*x')
A1= -1.329e-05; A2=0; % linear fit model right bend
%A1= -1.351e-05; A2=0; % linear fit model left bend
% plot(data(:,1),dl,data(:,1),A1*data(:,1)+A2)
% title('Relationship between encoder counts
and length changes & corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Change in length')
% figure
% % now calculate compression from the definition of S
compression=(3*data(:,4)-3*L0-dl)/3
compressionModel=fit(data(:,1),compression,'a*x')
B1= -0.000236; B2= 0; % linear fit model right bend
%B1= -0.000207; B2= 0; % linear fit model left bend
% plot(data(:,1),compression,data(:,1),B1*data(:,1)+B2)
% title('Relationship between encoder counts and compression &
% corresponding best fitting line')
% xlabel('Encoder counts')
% ylabel('Compression')
% figure
% % now try to compute lambda from above values
calcLambda=(3*(L0+compression)+dl)*r./(2*dl)
% % plot to see if it matches and
% % from the figure it shows that it matches very well.
% plot(data(:,3),abs(calcLambda))
% title('Radius of curvature comparison \lambda')
% xlabel('Experimental \lambda')
% ylabel('Calculated \lambda')
% figure
% based on these fit models, phi can be given as
approx_phi=abs(2*(A1*data(:,1)+A2)/(3*r))
plot(data(:,1),data(:,2),'r',data(:,1),approx_phi)
title('Subtended angle \phi  Vs Counts')
xlabel('Encoder counts')
ylabel('Calculated subtended angle \phi and approx \phi')
legend('Measured \phi','Calculated \phi','Location','NorthEast')
approx_Lambda=abs((3*(L0+(B1*data(:,1)+B2))+(A1*data(:,1)+A2))*r
./(2*(A1*data(:,1)+A2)))
figure
plot(data(:,1),data(:,3),'r',data(:,1),approx_Lambda)
legend('Measured \lambda','Calculated \lambda','Location','NorthEast')
title('Radius of curvature \lambda  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental radius \lambda and approx \lambda')
figure
approx_s=approx_phi.*approx_Lambda
plot(data(:,1),data(:,4),'r',data(:,1),approx_s)
title('Arc length s  Vs Counts')
xlabel('Encoder counts')
ylabel('Experimental arc length s and approx s')
legend('Measured s','Calculated s','Location','NorthEast')
```

D.4 MATLAB® code for 2D results of forward kinematics (d) Distal and Middle sections considered individually

```
clc
%distal right
data=[0 0 Inf 34;
    14 27.47 69.89 33.52;
    31 36.47 51.76 32.95;
    57 51.97 35.6 32.29;
    81 62.73 28.05 30.71;
    107 75.49 22.46 29.59;
    145 86.9 18.612 28.23];
%distal left
% data=[0 0 Inf 34;
%     23 24.72 73.34 31.64;
%     42 41.62 42.92 31.18;
%     64 53.98 33.05 31.13;
%     87 60.21 28.905 30.37;
%     118 73.73 22.5 28.95;
%     158 80.06 19.2 26.83];
%middle right
% data=[0 0 Inf 45;
%     37 9.74 260.64 44.31
%     55 15.78 158.43 43.63;
%     91 21.5 113.89 42.73;
%     118 27.21 88.9 42.25;
%     126 30.74 78.28 41.99;
%     151 34.56 68.67 41.42];
data(:,2:4)=[pi/180.*(data(:,2)) .01*data(:,3:4)]
r=.00213; %cross-section radius
L0=.34; %distal (relaxed unactuated length of tendon)
%L0=.45; %middle (relaxed unactuated length of tendon)
A1= -3.648e-5; A2=0; % linear fit model distal
B1= -0.000412; B2= 0; % linear fit model distal
%A1= -1.329e-05; A2=0; % linear fit model middle
%B1= -0.000236; B2= 0; % linear fit model middle
approx_Lambda=abs((3*(L0+(B1*data(:,1)+B2))+(A1*data(:,1)+A2))*r
./(2*(A1*data(:,1)+A2)))
approx_phi=abs(2*(A1*data(:,1)+A2)/(3*r))
%computed x-coordinate from 2D forward kinematics
approx_x=approx_Lambda.*(1-cos(approx_phi))
%computed y-coordinate from 2D forward kinematics
approx_y=approx_Lambda.*sin(approx_phi)
%actual x-coordinate from 2D forward kinematics
actual_x=data(:,3).*(1-cos(data(:,2)))
%actual y-coordinate from 2D forward kinematics
actual_y=data(:,3).*sin(data(:,2))
plot(data(:,1),actual_x,’r*-’,data(:,1),approx_x,’bv-’)
grid on
xlabel(’Encoder counts’)
ylabel(’Actual x-coordinate and Calculated x-coordinate’)
legend(’Actual x-coordinate’,’Calculated x-coordinate’,’Location’,’NorthWest’)
title(’x-coordinate of tip Vs Counts’)
figure
plot(data(:,1),actual_y,’r*-’,data(:,1),approx_y,’bv-’)
grid on
xlabel(’Encoder counts’)
ylabel(’Actual y-coordinate and Calculated y-coordinate’)
legend(’Actual y-coordinate’,’Calculated y-coordinate’,’Location’,’NorthEast’)
title(’y-coordinate of tip Vs Counts’)
```

D.5 MATLAB® code for 2D results of forward kinematics (e) Distal and Middle sections combined

```
clc
%distal right
data_d=[0 0 Inf 34;
    14 27.47 69.89 33.52;
    31 36.47 51.76 32.95;
    57 51.97 35.6 32.29;
    81 62.73 28.05 30.71;
    107 75.49 22.46 29.59;
    145 86.9 18.612 28.23];
%distal left
```

```
% data=[0 0 Inf 34;
%     23 24.72 73.34 31.64;
%     42 41.62 42.92 31.18;
%     64 53.98 33.05 31.13;
%     87 60.21 28.905 30.37;
%     118 73.73 22.5 28.95;
%     158 80.06 19.2 26.83];
%middle right
data_m=[0 0 Inf 45;
    37 9.74 260.64 44.31
    55 15.78 158.43 43.63;
    91 21.5 113.89 42.73;
    118 27.21 88.9 42.25;
    126 30.74 78.28 41.99;
    151 34.56 68.67 41.42];
data_d(:,2:4)=[pi/180.*(data_d(:,2)) .01*data_d(:,3:4)]
data_m(:,2:4)=[pi/180.*(data_m(:,2)) .01*data_m(:,3:4)]
r=.00213; %cross-section radius
L0_d=.34; %distal (relaxed unactuated length of tendon)
L0_m=.45; %middle (relaxed unactuated length of tendon)
A1_d= -3.648e-5; A2_d=0; % linear fit model distal
B1_d= -0.000412; B2_d= 0; % linear fit model distal
A1_m= -1.329e-05; A2_m=0; % linear fit model middle
B1_m= -0.000236; B2_m= 0; % linear fit model middle
%lambda for distal from linear approximation
approx_Lambda_d=abs((3*(L0_d+(B1_d*data_d(:,1)+B2_d))+
(A1_d*data_d(:,1)+A2_d))*r./(2*(A1_d*data_d(:,1)+A2_d)))
%phi for distal from linear approximation
approx_phi_d=abs(2*(A1_d*data_d(:,1)+A2_d)/(3*r))
%lambda for middle from linear approximation
approx_Lambda_m=abs((3*(L0_m+(B1_m*data_m(:,1)+B2_m))+
(A1_m*data_m(:,1)+A2_m))*r./(2*(A1_m*data_m(:,1)+A2_m)))
%phi for middle from linear approximation
approx_phi_m=abs(2*(A1_m*data_m(:,1)+A2_m)/(3*r))
%computed x-coordinate from 2D forward kinematics
approx_x=(cos(approx_phi_m).*approx_Lambda_d.*(1-cos(approx_phi_d)))+
(approx_Lambda_d.*sin(approx_phi_m).*sin(approx_phi_d))+
(approx_Lambda_m.*(1-cos(approx_phi_m)))
%computed y-coordinate from 2D forward kinematics
approx_y= -(sin(approx_phi_m).*approx_Lambda_d.*(1-cos(approx_phi_d)))
 +(approx_Lambda_d.*cos(approx_phi_m).*sin(approx_phi_d))+
 (approx_Lambda_m.*sin(approx_phi_m))
%actual x-coordinate from 2D forward kinematics
actual_x=(cos(data_m(:,2)).*data_d(:,3).*(1-cos(data_d(:,2))))+
(data_d(:,3).*sin(data_m(:,2)).*sin(data_d(:,2)))+
(data_m(:,3).*(1-cos(data_m(:,2))))
%actual y-coordinate from 2D forward kinematics
actual_y=-(sin(data_m(:,2)).*data_d(:,3).*(1-cos(data_d(:,2)))) +
(data_d(:,3).*cos(data_m(:,2)).*sin(data_d(:,2)))+(data_m(:,3).*sin(data_m(:,2)))
plot(actual_x,actual_y,’r*-’,approx_x,approx_y,’bv-’)
grid on
```

```
xlabel('X-coordinate of the robotic cable tip')
ylabel('Y-coordinate of the robotic cable tip')
legend(' Actual x-y coordinate','Calculated x-y-coordinate','Location','NorthEast')
title('Position of the robotic cable tip')
```

D.6 MATLAB® code for 3D results of λ, ϕ and θ and forward kinematics (f) Distal and Middle sections considered individually

```
 % distal section
r=.00213; %cross-section radius
L0=.34; %distal (relaxed unactuated length of tendon)
%L0=.45; %middle (relaxed unactuated length of tendon)
 %random initialize coefficients for model
 a = 0.001 + 0.09.*rand(27,1) %model constant used later
 b=a % initialize model_1 coefficients used later
 c=b % initialize model_2 coefficients used later
 d=c % initialize model_3 coefficients used later
 % % %lambda, phi, theta
 Data=[lambda, phi, theta]
 %Data into standard units
 Data=[Data(:,1)*.01 (pi/180)*(Data(:,2:3))];
 l1=(Data(:,1)-r*cos(Data(:,3))).*Data(:,2)-L0;
 l2=(Data(:,1)-r*cos(2*pi/3-Data(:,3))).*Data(:,2)-L0;
 l3=(Data(:,1)-r*cos(4*pi/3-Data(:,3))).*Data(:,2)-L0;
 LC=[l1 l2 l3]
 % compression is maximum of observed lengths
 C=max(LC,[],2)
 %modeled lengths dL
 dL=LC-C*ones(1,3)
 % evaluate shape variables from modeled lengths
 den=(2*(sqrt((dL(:,1).^2)+(dL(:,2).^2)+(dL(:,3).^2)-
 (dL(:,1).*dL(:,2))-(dL(:,2).*dL(:,3))-(dL(:,1).*dL(:,3)))));
 %lambda
 calclambda=[abs((3*(L0+C)+dL(:,1)+dL(:,2)+dL(:,3))*r./den)]
 %phi
 calcphi=[2*sqrt(dL(:,3).^2+dL(:,1).^2+dL(:,2).^2-dL(:,1).*dL(:,2)
 -dL(:,1).*dL(:,3)-dL(:,3).*dL(:,2))/(3*r)]
 %theta
 calctheta =[atan2(sqrt(3)*(dL(:,3)-dL(:,2)),dL(:,2)+dL(:,3)-2*dL(:,1))]
 % shape variables from modeled lengths matches with data shape variables.
 %Counts
 enc=[enc, enc2, enc3]

 %mapping function between counts and compression

 modelfun=@(a,enc)(a(1)+(a(2)*enc(:,3))+(a(3)*(enc(:,3).^2))+
(a(4)*enc(:,2))+(a(5)*enc(:,2).*enc(:,3))+
(a(6)*enc(:,2).*(enc(:,3).^2))+(a(7)*(enc(:,2).^2))+
(a(8)*enc(:,3).*(enc(:,2).^2))+(a(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(a(10)*enc(:,1))+(a(11)*enc(:,1).*enc(:,3))+
(a(12)*enc(:,1).*(enc(:,3).^2))+(a(13)*enc(:,1).*enc(:,2))+
(a(14)*enc(:,1).*enc(:,3).*enc(:,2))+
(a(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(a(16)*enc(:,1).*(enc(:,2).^2))+
(a(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(a(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+
(a(19)*(enc(:,1).^2))+
(a(20)*enc(:,3).*(enc(:,1).^2))+
(a(21)*(enc(:,1).^2).*(enc(:,3).^2))+(a(22)*(enc(:,1).^2).*enc(:,2))+
(a(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(a(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(a(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(a(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(a(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 beta = nlinfit(enc,C,modelfun,a)
 a=beta
 format short g
 giva=a % coefficients  of compression model
 %Compression computed from model
 Computedcompr=(a(1)+(a(2)*enc(:,3))+(a(3)*(enc(:,3).^2))+
(a(4)*enc(:,2))+(a(5)*enc(:,2).*enc(:,3))+(a(6)*enc(:,2).*(enc(:,3).^2))+
(a(7)*(enc(:,2).^2))+(a(8)*enc(:,3).*(enc(:,2).^2))+
(a(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(a(10)*enc(:,1))+(a(11)*enc(:,1).*enc(:,3))+(a(12)*enc(:,1).*(enc(:,3).^2))+
(a(13)*enc(:,1).*enc(:,2))+
(a(14)*enc(:,1).*enc(:,3).*enc(:,2))+(a(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(a(16)*enc(:,1).*(enc(:,2).^2))+(a(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(a(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+(a(19)*(enc(:,1).^2))+
(a(20)*enc(:,3).*(enc(:,1).^2))+(a(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(a(22)*(enc(:,1).^2).*enc(:,2))+(a(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(a(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(a(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(a(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(a(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 Compare=[Computedcompr C]
 % plot(C,'r')
 % hold on;
 % plot(Computedcompr)
 % hold on;
 % err=((Computedcompr-C)./C).*100
 % figure
 % plot (err)
 modelfun_1=@(b,enc)(b(1)+(b(2)*enc(:,3))+(b(3)*(enc(:,3).^2))+
(b(4)*enc(:,2))+(b(5)*enc(:,2).*enc(:,3))+
(b(6)*enc(:,2).*(enc(:,3).^2))+(b(7)*(enc(:,2).^2))+
(b(8)*enc(:,3).*(enc(:,2).^2))+(b(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(b(10)*enc(:,1))+(b(11)*enc(:,1).*enc(:,3))+
(b(12)*enc(:,1).*(enc(:,3).^2))+(b(13)*enc(:,1).*enc(:,2))+
(b(14)*enc(:,1).*enc(:,3).*enc(:,2))+
(b(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(b(16)*enc(:,1).*(enc(:,2).^2))+
(b(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(b(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+
(b(19)*(enc(:,1).^2))+
(b(20)*enc(:,3).*(enc(:,1).^2))+
(b(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(b(22)*(enc(:,1).^2).*enc(:,2))+
(b(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(b(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(b(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(b(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(b(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 %format short g
 beta = nlinfit(enc,dL(:,1),modelfun_1,b)
 b=beta
 givb=b % coefficients  of length model 1
 a=b
 %Modeled length change 1 computed from model
 ComputedL_1=(a(1)+(a(2)*enc(:,3))+(a(3)*(enc(:,3).^2))+
(a(4)*enc(:,2))+(a(5)*enc(:,2).*enc(:,3))+(a(6)*enc(:,2).*(enc(:,3).^2))+
(a(7)*(enc(:,2).^2))+(a(8)*enc(:,3).*(enc(:,2).^2))+
(a(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(a(10)*enc(:,1))+(a(11)*enc(:,1).*enc(:,3))+(a(12)*enc(:,1).*(enc(:,3).^2))+
(a(13)*enc(:,1).*enc(:,2))+
(a(14)*enc(:,1).*enc(:,3).*enc(:,2))+(a(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(a(16)*enc(:,1).*(enc(:,2).^2))+(a(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(a(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+(a(19)*(enc(:,1).^2))+
(a(20)*enc(:,3).*(enc(:,1).^2))+(a(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(a(22)*(enc(:,1).^2).*enc(:,2))+(a(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(a(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(a(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(a(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(a(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 Compare_dl1=[ComputedL_1 dL(:,1)]
 % figure
 % plot(dL(:,1),'r')
 % hold on
 % plot(ComputedL_1)
 % hold on
 %
 modelfun_2=@(c,enc)(c(1)+(c(2)*enc(:,3))+(c(3)*(enc(:,3).^2))+
(c(4)*enc(:,2))+(c(5)*enc(:,2).*enc(:,3))+
(c(6)*enc(:,2).*(enc(:,3).^2))+(c(7)*(enc(:,2).^2))+
(c(8)*enc(:,3).*(enc(:,2).^2))+(c(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(c(10)*enc(:,1))+(c(11)*enc(:,1).*enc(:,3))+
(c(12)*enc(:,1).*(enc(:,3).^2))+(c(13)*enc(:,1).*enc(:,2))+
(c(14)*enc(:,1).*enc(:,3).*enc(:,2))+
(c(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(c(16)*enc(:,1).*(enc(:,2).^2))+
(c(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(c(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+
(c(19)*(enc(:,1).^2))+
(c(20)*enc(:,3).*(enc(:,1).^2))+
(c(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(c(22)*(enc(:,1).^2).*enc(:,2))+
(c(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(c(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(c(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(c(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(c(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 beta = nlinfit(enc,dL(:,2),modelfun_2,c)
 c=beta
 givc=c % coefficients  of length model 2
 a=c
  %Modeled length change 2 computed from model
 ComputedL_2=(a(1)+(a(2)*enc(:,3))+(a(3)*(enc(:,3).^2))+
(a(4)*enc(:,2))+(a(5)*enc(:,2).*enc(:,3))+(a(6)*enc(:,2).*(enc(:,3).^2))+
(a(7)*(enc(:,2).^2))+(a(8)*enc(:,3).*(enc(:,2).^2))+
(a(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(a(10)*enc(:,1))+(a(11)*enc(:,1).*enc(:,3))+(a(12)*enc(:,1).*(enc(:,3).^2))+
(a(13)*enc(:,1).*enc(:,2))+
(a(14)*enc(:,1).*enc(:,3).*enc(:,2))+(a(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(a(16)*enc(:,1).*(enc(:,2).^2))+(a(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(a(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+(a(19)*(enc(:,1).^2))+
(a(20)*enc(:,3).*(enc(:,1).^2))+(a(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(a(22)*(enc(:,1).^2).*enc(:,2))+(a(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(a(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(a(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(a(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(a(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 Compare_dl2=[ComputedL_2 dL(:,2)]
 % figure
 % plot(dL(:,2),'r')
 % hold on
 % plot(ComputedL_2)
 % hold on
 %
 modelfun_3=@(d,enc)(d(1)+(d(2)*enc(:,3))+(d(3)*(enc(:,3).^2))+
(d(4)*enc(:,2))+(d(5)*enc(:,2).*enc(:,3))+
(d(6)*enc(:,2).*(enc(:,3).^2))+(d(7)*(enc(:,2).^2))+
(d(8)*enc(:,3).*(enc(:,2).^2))+(d(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(d(10)*enc(:,1))+(d(11)*enc(:,1).*enc(:,3))+
(d(12)*enc(:,1).*(enc(:,3).^2))+(d(13)*enc(:,1).*enc(:,2))+
(d(14)*enc(:,1).*enc(:,3).*enc(:,2))+
(d(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(d(16)*enc(:,1).*(enc(:,2).^2))+
(d(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(d(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+
(d(19)*(enc(:,1).^2))+
(d(20)*enc(:,3).*(enc(:,1).^2))+
(d(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(d(22)*(enc(:,1).^2).*enc(:,2))+
(d(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(d(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(d(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(d(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(d(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 beta = nlinfit(enc,dL(:,3),modelfun_3,d)
 d=beta
 givd=d % coefficients  of length model 3
 a=d
  %Modeled length change 3 computed from model
 ComputedL_3=(a(1)+(a(2)*enc(:,3))+(a(3)*(enc(:,3).^2))+
(a(4)*enc(:,2))+(a(5)*enc(:,2).*enc(:,3))+(a(6)*enc(:,2).*(enc(:,3).^2))+
(a(7)*(enc(:,2).^2))+(a(8)*enc(:,3).*(enc(:,2).^2))+
(a(9)*(enc(:,3).^2).*(enc(:,2).^2))+
(a(10)*enc(:,1))+(a(11)*enc(:,1).*enc(:,3))+(a(12)*enc(:,1).*(enc(:,3).^2))+
(a(13)*enc(:,1).*enc(:,2))+
(a(14)*enc(:,1).*enc(:,3).*enc(:,2))+(a(15)*(enc(:,1)).*(enc(:,3).^2).*enc(:,2))+
(a(16)*enc(:,1).*(enc(:,2).^2))+(a(17)*enc(:,1).*(enc(:,2).^2).*enc(:,3))+
(a(18)*enc(:,1).*(enc(:,2).^2).*(enc(:,3).^2))+(a(19)*(enc(:,1).^2))+
(a(20)*enc(:,3).*(enc(:,1).^2))+(a(21)*(enc(:,1).^2).*(enc(:,3).^2))+
(a(22)*(enc(:,1).^2).*enc(:,2))+(a(23)*(enc(:,1).^2).*enc(:,2).*enc(:,3))+
(a(24).*(enc(:,1).^2).*enc(:,2).*(enc(:,3).^2))+
(a(25)*(enc(:,1).^2).*(enc(:,2).^2))+
(a(26)*enc(:,3).*(enc(:,1).^2).*(enc(:,2).^2))+
(a(27)*(enc(:,1).^2).*(enc(:,2).^2).*(enc(:,3).^2)))
 Compare_dl3=[ComputedL_3 dL(:,3)]
 % figure
 % plot(dL(:,3),'r')
 % hold on
 % plot(ComputedL_3)
 % hold on

 %modeled lengths
 dL_computed=[ComputedL_1 ComputedL_2 ComputedL_3]

 Computed shape variables
 den_cmprcomp=(2*(sqrt((dL_computed(:,1).^2)+(dL_computed(:,2).^2)+
(dL_computed(:,3).^2)-(dL_computed(:,1).*dL_computed(:,2))-
(dL_computed(:,2).*dL_computed(:,3))-(dL_computed(:,1).*dL_computed(:,3)))));
 % %lambda
 lambda_cmprcomp=[abs((3*(L0+Computedcompr)+dL_computed(:,1)
+dL_computed(:,2)+dL_computed(:,3))*r./den_cmprcomp)]
 index=1:size(lambda_cmprcomp);
 figure
 plot(calclambda,'r')
 hold on
 plot(lambda_cmprcomp)
 xlabel('Data index')
 ylabel('Experimental radius \lambda and approx \lambda')
 legend('Measured \lambda','Calculated \lambda','Location','NorthEast')
 title('Radius of curvature \lambda')
```

```
% %phi
 calcphi_cmprcomp=[2*sqrt(dL_computed(:,3).^2+dL_computed(:,1).^2+
dL_computed(:,2).^2-dL_computed(:,1).*dL_computed(:,2)
-dL_computed(:,1).*dL_computed(:,3)-dL_computed(:,3).*dL_computed(:,2))/(3*r)]
 figure
 plot(calcphi,’r’)
 hold on
 plot (calcphi_cmprcomp)
 xlabel(’Data index’)
 ylabel(’Experimental subtended angle \phi and approx \phi’)
 legend(’Measured \phi’,’Calculated \phi’,’Location’,’NorthEast’)
 title(’Bending angle \phi’)
 % %theta
 %calctheta_cmprcomp =[atan2(sqrt(3)*(dL_computed(:,3)-
 dL_computed(:,2)),dL_computed(:,2)
+dL_computed(:,3)-2*dL_computed(:,1)) Data(:,3)]
 theta
 calctheta_cmprcomp =[atan2(sqrt(3)*(dL_computed(:,3)-
 dL_computed(:,2)),dL_computed(:,2)
+dL_computed(:,3)-2*dL_computed(:,1))]
 figure
 plot(calctheta,’r’)
 hold on
 plot (calctheta_cmprcomp)
 xlabel(’Data index’)
 ylabel(’Experimental bending plane angle \theta and approx \theta’)
 legend(’Measured \theta’,’Calculated \theta’,’Location’,’NorthEast’)
 title(’Bending plane angle \theta’)
% computed x-coordinate from 3D forward kinematics
 approx_x= lambda_cmprcomp.*cos(calctheta_cmprcomp).*(1-cos(calcphi_cmprcomp))
% computed y-coordinate from 3D forward kinematics
 approx_y= lambda_cmprcomp.*sin(calctheta_cmprcomp).*(1-cos(calcphi_cmprcomp))
% computed z-coordinate from 3D forward kinematics
 approx_z= lambda_cmprcomp.*sin(calcphi_cmprcomp)
% actual x-coordinate from 3D forward kinematics
 actual_x= -Data(:,1).*cos(Data(:,3)).*(cos(Data(:,2))-1)
% actual y-coordinate from 3D forward kinematics
 actual_y= -Data(:,1).*sin(Data(:,3)).*(cos(Data(:,2))-1)
% actual z-coordinate from 3D forward kinematics
 actual_z= Data(:,1).*sin(Data(:,2))
 compare_x=[approx_x actual_x]
 compare_y=[approx_y actual_y]
 compare_z=[approx_z actual_z]
 figure
 plot(index,actual_x,’r*-’,index,approx_x,’bv-’)
 grid on
 xlabel(’Data points’)
 ylabel(’Actual x-coordinate and Calculated x-coordinate’)
 legend(’Actual x-coordinate’,’Calculated x-coordinate’,’Location’,’NorthEast’)
 title(’x-coordinate of tip’)
 figure
 plot(index,actual_y,'r*-',index,approx_y,'bv-')
 grid on
 xlabel('Data points')
 ylabel('Actual y-coordinate and Calculated y-coordinate')
 legend('Actual y-coordinate','Calculated y-coordinate','Location','NorthEast')
 title('y-coordinate of tip')
 figure
 plot(index,actual_z,'r*-',index,approx_z,'bv-')
 grid on
 xlabel('Data points')
 ylabel('Actual z-coordinate and Calculated z-coordinate')
 legend('Actual z-coordinate','Calculated z-coordinate','Location','NorthEast')
 title('z-coordinate of tip')
 giv=[giva givb givc givd] % all coefficents of the 4 models
```

D.7 MATLAB® code for 3D HTM formula (g) Distal and Middle sections

```
clc
syms ct_m st_m l_m cp_m sp_m ct_d st_d l_d cp_d sp_d
% matrices for producing 3D HTM of middle section
A_m=[ct_m -st_m 0 0;st_m ct_m 0 0;0 0 1 0; 0 0 0 1]
B_m=[1 0 0 l_m;0 1 0 0; 0 0 1 0;0 0 0 1]
C_m=[cp_m 0 sp_m 0;0 1 0 0;-sp_m 0 cp_m 0; 0 0 0 1]
D_m=[1 0 0 -l_m;0 1 0 0;0 0 1 0;0 0 0 1]
E_m=[ct_m st_m 0 0;-st_m ct_m 0 0; 0 0 1 0;0 0 0 1]
T_m=A_m*B_m*C_m*D_m*E_m;
%middle section HTM
T_m=simplify(T_m)
%4X4 homogenous rotational matrix
R=[cos(2*pi/9) -sin(2*pi/9) 0 0;sin(2*pi/9) cos(2*pi/9) 0 0;0 0 1 0; 0 0 0 1]'
% matrices for producing 3D HTM of distal section
A_d=[ct_d -st_d 0 0;st_d ct_d 0 0;0 0 1 0; 0 0 0 1]
B_d=[1 0 0 l_d;0 1 0 0; 0 0 1 0;0 0 0 1]
C_d=[cp_d 0 sp_d 0;0 1 0 0;-sp_d 0 cp_d 0; 0 0 0 1]
D_d=[1 0 0 -l_d;0 1 0 0;0 0 1 0;0 0 0 1]
E_d=[ct_d st_d 0 0;-st_d ct_d 0 0; 0 0 1 0;0 0 0 1]
T_d=A_d*B_d*C_d*D_d*E_d;
%distal section HTM
T_d=simplify(T_d)
%Combined HTM 3D
HTM=T_m*R*T_d
simplify(HTM)
v=HTM(:,4) % x-y-z coordinates formula
```

D.8 MATLAB® code for 3D results of λ, ϕ and θ and forward kinematics (h) Distal and Middle sections combined

```
clc
%Choose any random but same number of data points from distal and middle section
%data shown earlier for 3D (In this case 124 data points were taken)
% repeat the 3D code for computing shape variables for each section
%for that many number of data.
% Subscript m (_m) denotes data, parameters and shape variables
for middle section and subscript d (_d) denotes all that for distal section

%trignometric values of computed shape variables
ct_m = cos(calctheta_m_cmprcomp);
st_m = sin(calctheta_m_cmprcomp);
l_m = lambda_m_cmprcomp;
cp_m = cos(calcphi_m_cmprcomp);
sp_m = sin(calcphi_m_cmprcomp);
ct_d = cos(calctheta_d_cmprcomp);
st_d = sin(calctheta_d_cmprcomp);
l_d = lambda_d_cmprcomp;
cp_d = cos(calcphi_d_cmprcomp);
sp_d = sin(calcphi_d_cmprcomp);
% computed x-coordinate of combined section from 3D combined forward kinematics
approx_x=ct_m.*l_d.*sp_d.*sp_m - ct_m.*l_m.*(cp_m - 1) -
 ct_d.*l_d.*(cp_d - 1).*((0.7660.*cp_m.*ct_m.^2) + (0.7660.*st_m.^2)
  - (0.6428.*ct_m.*st_m.*(cp_m - 1))) -
l_d.*st_d.*(cp_d - 1).*((0.6428.*cp_m.*ct_m.^2) +
(0.6428.*st_m.^2) + (0.7660.*ct_m.*st_m.*(cp_m - 1)));
% computed y-coordinate of combined section from 3D combined forward kinematics
approx_y=l_d.*sp_d.*sp_m.*st_m - l_m.*st_m.*(cp_m - 1) -
l_d.*st_d.*(cp_d - 1).*((0.7660.*cp_m.*st_m.^2) +
(0.7660.*ct_m.^2) + (0.6428.*ct_m.*st_m.*(cp_m - 1))) +
ct_d.*l_d.*(cp_d - 1).*((0.6428.*cp_m.*st_m.^2) + (0.6428.*ct_m.^2)
- (0.7660.*ct_m.*st_m.*(cp_m - 1)));
% computed z-coordinate of combined section from 3D combined forward kinematics
approx_z=l_m.*sp_m + cp_m.*l_d.*sp_d + ct_d.*l_d.*((0.7660.*ct_m.*sp_m)
 - (0.6428.*sp_m.*st_m)).*(cp_d - 1) + l_d.*st_d.*((0.6428.*ct_m.*sp_m)
  + (0.7660.*sp_m.*st_m)).*(cp_d - 1);
%trignometric values of actual shape variables
ct_m = cos(Data_m(:,3));
st_m = sin(Data_m(:,3));
l_m = Data_m(:,1);
cp_m = cos(Data_m(:,2));
sp_m = sin(Data_m(:,2));
ct_d = cos(Data_d(:,3));
st_d = sin(Data_d(:,3));
l_d = Data_d(:,1);
cp_d = cos(Data_d(:,2));
sp_d = sin(Data_d(:,2));
% actual x-coordinate of combined section from 3D combined forward kinematics
```

```
actual_x=ct_m.*l_d.*sp_d.*sp_m - ct_m.*l_m.*(cp_m - 1) -
 ct_d.*l_d.*(cp_d - 1).*((0.7660.*cp_m.*ct_m.^2) + (0.7660.*st_m.^2) -
(0.6428.*ct_m.*st_m.*(cp_m - 1))) -
l_d.*st_d.*(cp_d-1).*((0.6428.*cp_m.*ct_m.^2) +
(0.6428.*st_m.^2) + (0.7660.*ct_m.*st_m.*(cp_m - 1)));
% actual y-coordinate of combined section from 3D combined forward kinematics
actual_y=l_d.*sp_d.*sp_m.*st_m - l_m.*st_m.*(cp_m - 1) -
 l_d.*st_d.*(cp_d - 1).*((0.7660.*cp_m.*st_m.^2) +
 (0.7660.*ct_m.^2) + (0.6428.*ct_m.*st_m.*(cp_m - 1))) +
  ct_d.*l_d.*(cp_d - 1).*((0.6428.*cp_m.*st_m.^2) + (0.6428.*ct_m.^2)
  - (0.7660.*ct_m.*st_m.*(cp_m - 1)));
% actual z-coordinate of combined section from 3D combined forward kinematics
actual_z=l_m.*sp_m + cp_m.*l_d.*sp_d + ct_d.*l_d.*((0.7660.*ct_m.*sp_m)
 - (0.6428.*sp_m.*st_m)).*(cp_d - 1) + l_d.*st_d.*((0.6428.*ct_m.*sp_m)
  + (0.7660.*sp_m.*st_m)).*(cp_d - 1);

Comparex=[approx_x actual_x]
Comparey=[approx_y actual_y]
Comparez=[approx_z actual_z]

index=1:size(actual_x);
figure
plot(index,actual_x,’r*-’,index,approx_x,’bv-’)
grid on
xlabel(’Data points’)
ylabel(’Actual x-coordinate and Calculated x-coordinate ’)
legend(’Actual x-coordinate’,’Calculated x-coordinate’,’Location’,’NorthEast’)
title(’x-coordinate for multi-section robotic cable tip ’)
figure
plot(index,actual_y,’r*-’,index,approx_y,’bv-’)
grid on
xlabel(’Data points’)
ylabel(’Actual y-coordinate and Calculated y-coordinate’)
legend(’Actual y-coordinate’,’Calculated y-coordinate’,’Location’,’NorthEast’)
title(’y-coordinate for multi-section robotic cable tip’)
figure
plot(index,actual_z,’r*-’,index,approx_z,’bv-’)
grid on
xlabel(’Data points’)
ylabel(’Actual z-coordinate and Calculated z-coordinate’)
legend(’Actual z-coordinate’,’Calculated z-coordinate’,’Location’,’NorthEast’)
title(’z-coordinate for multi-section robotic cable tip’)
```

D.9 Arduino code

This program is used in actuating the cable robot for obtaining various shape configurations for a given section, used in data collection process by taking images of the shapes obtained.

```
#include <Servo.h>
#include <PinChangeInt.h>
#include <PinChangeIntConfig.h>

#include < avr/io.h >
// ISR interrupt service routine
#include < avr/interrupt.h >

Servo servo1_unwind;  // create servo object to control a servo
Servo servo1_wind;

Servo servo2_unwind;  // create servo object to control a servo
Servo servo2_wind;

Servo servo3_unwind;  // create servo object to control a servo
Servo servo3_wind;

volatile int count1 = 0;
volatile int count2 = 0;
volatile int count3 = 0;
int servotime=200;

//middle servos stop at given
int servo1stop=94;
int servo2stop=91;
int servo3stop=93;

//all distal servos stop at 85
//int servo1stop=85;
//int servo2stop=85;
//int servo3stop=85;

int servowind1=91;
int servounwind1=97;
int servowind2=87;
int servounwind2=94;
int servowind3=90;
int servounwind3=96;
// Pins generated by software for taking input from  2 encoders
#define PIN2 4
#define PIN3 7


void setup()
{
 // ready to be made as interrupts through software
  pinMode(PIN2, INPUT);
  digitalWrite(PIN2, HIGH); // pull up resistor
  pinMode(2, INPUT);
  digitalWrite(2, HIGH); // pull up resistor
  pinMode(PIN3, INPUT);
  digitalWrite(PIN3, HIGH); // pull up resistor

 // servo control pins
  servo1_wind.attach(3);
  servo2_wind.attach(5);
  servo3_wind.attach(6);
  // software interrupt for 2 encoders
  PCintPort::attachInterrupt(PIN2, servocount2,CHANGE);
  PCintPort::attachInterrupt(PIN3, servocount3,CHANGE);
// hardware interrupt for 1 encoder
  attachInterrupt(0, servocount1, CHANGE);

  Serial.begin(9600);

}

void loop()
{
  byte servo;
 // stops all servos
 servo1_wind.write(servo1stop);
 servo2_wind.write(servo2stop);
 servo3_wind.write(servo3stop);

if (Serial.available())
{

servo = Serial.read();

// keyboard letter a and z control servo 1
if(servo==’a’)
{
 servo1_wind.write(servounwind1);
 delay(servotime);
 servo1_wind.write(servo1stop);

}

else if(servo==’z’)
{
 servo1_wind.write(servowind1);
 delay(servotime);
 servo1_wind.write(servo1stop);

}
// keyboard letter s and x control servo 2
else if(servo==’s’)
{
 servo2_wind.write(servounwind2);
 delay(servotime);
 servo2_wind.write(servo2stop);

}

else if(servo==’x’)
{
 servo2_wind.write(servowind2);
 delay(servotime);
 servo2_wind.write(servo2stop);

}
// keyboard letter d and c control servo 3
else if(servo==’d’)
{
 servo3_wind.write(servounwind3);
 delay(servotime);
 servo3_wind.write(servo3stop);

}

else if(servo==’c’)
{
 servo3_wind.write(servowind3);
 delay(servotime);
 servo3_wind.write(servo3stop);

}
// keyboard number 7 stops all servos
else if(servo==’7’)
{
 servo1_wind.write(servo1stop);
 servo2_wind.write(servo2stop);
 servo3_wind.write(servo3stop);
 count1=0;
 count2=0;
 count3=0;
}

}

// encoder readings
Serial.print(count1);
Serial.print(" ");

Serial.print(count2);
Serial.print(" ");

Serial.print(count3);
Serial.println(" ");

}

// functions for counting encoder signals
void servocount1()
{
  count1++;
}

void servocount2()
{
  count2++;
}

void servocount3()
{
  count3++;
}
```

References

[1] 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.

[2] J. Mehling, M. Diftler, M. Chu, and M. Valvo, "A minimally invasive tendril robot for in-space inspection," in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pp. 690–695, Feb 2006.

[3] J. Yang, P. Jason, and K. Abdel-Malek, "A hyper-redundant continuous robot," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1854–1859, May 2006.

[4] H. Hu, P. Wang, B. Zhao, M. Li, and L. Sun, "Design of a novel snake-like robotic colonoscope," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pp. 1957–1961, Dec 2009.

[5] R. Webster, J. Romano, and N. Cowan, "Mechanics of precurved-tube continuum robots," *Robotics, IEEE Transactions on*, vol. 25, pp. 67–78, Feb 2009.

[6] K. Hatazaki, M. Konyo, K. Isaki, S. Tadokoro, and F. Takemura, "Active scope camera for urban search and rescue," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2596–2602, Oct 2007.

[7] Y.-J. Kim, S. Cheng, S. Kim, and K. Iagnemma, "Design of a tubular snake-like manipulator with stiffening capability by layer jamming," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4251–4256, Oct 2012.

[8] K.-W. Kwok, K. H. Tsoi, V. Vitiello, J. Clark, G. Chow, W. Luk, and G.-Z. Yang, "Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints," *Robotics, IEEE Transactions on*, vol. 29, pp. 15–31, Feb 2013.

[9] Y. Chen, S. Tanaka, and I. Hunter, "Disposable endoscope tip actuation design and robotic platform," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 2279–2282, Aug 2010.

[10] Z. Li and R. Du, "Design and analysis of a bio-inspired wire-driven multi-section flexible robot.," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[11] M. Mahvash and M. Zenati, "Toward a hybrid snake robot for single-port surgery," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pp. 5372–5375, Aug 2011.

[12] C. Velasquez, H. King, B. Hannaford, and W. Yoon, "Development of a flexible imaging probe integrated to a surgical telerobot system: Preliminary remote control test and probe design," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, pp. 894–898, June 2012.

[13] T. L. Lam and Y. Xu, "Biologically inspired tree-climbing robot with continuum maneuvering mechanism," *Journal of Field Robotics*, vol. 29, no. 6, pp. 843–860, 2012.

[14] H. Watanabe, K. Kanou, Y. Kobayashi, and M. G. Fujie, "Development of a "steerable drill" for acl reconstruction to create the arbitrary trajectory of a bone tunnel," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 955–960, IEEE, 2011.

[15] J. Jung, R. S. Penning, N. J. Ferrier, and M. Zinn, "A modeling approach for continuum robotic manipulators: Effects of nonlinear internal device friction," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 5139–5146, Sept 2011.

[16] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4, pp. 2849–2854 vol.4, 1999.

[17] Q. Zhao and F. Gao, "Design and analysis of a kind of biomimetic continuum robot," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pp. 1316–1320, Dec 2010.

[18] I. Walker, "Robot strings: Long, thin continuum robots," in *Aerospace Conference, 2013 IEEE*, pp. 1–12, March 2013.

[19] I. D. Walker, "Continuous backbone continuum robot manipulators," *ISRN Robotics*, vol. 2013, 2013.

[20] P. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *Robotics, IEEE Transactions on*, vol. 26, pp. 209–225, April 2010.

[21] R. S. Penning, J. Jung, N. J. Ferrier, and M. Zinn, "An evaluation of closed-loop control options for continuum manipulators," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 5392–5397, May 2012.

[22] L. Torres, R. Webster, and R. Alterovitz, "Task-oriented design of concentric tube robots using mechanics-based models," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4449–4455, Oct 2012.

[23] T. Aoki, A. Ochiai, and S. Hirose, "Study on slime robot: development of the mobile robot prototype model using bridle bellows," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 3, pp. 2808–2813 Vol.3, April 2004.

[24] H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 2, pp. 707–715 vol.2, 2001.

[25] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: an artificial elephant's nose with maneuverability for rescue operation," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, pp. 2454–2459 vol.3, 2001.

[26] L. S. Cowan, "Analysis and experiments for tendril-type robots," Master's thesis, Clemson University, 2008.

[27] I. Gravagne, C. Rahn, and I. Walker, "Large deflection dynamics and control for planar continuum robots," *Mechatronics, IEEE/ASME Transactions on*, vol. 8, pp. 299–307, June 2003.

[28] M. Blessing and I. D. Walker, “Novel continuum robots with variable-length sections,” *Mechatronic Systems 2004*, vol. 1, 2005.

[29] A. Kapoor, N. Simaan, and R. H. Taylor, “Suturing in confined spaces: constrained motion control of a hybrid 8-dof robot,” in *Advanced Robotics, 2005. ICAR’05. Proceedings., 12th International Conference on*, pp. 452–459, IEEE, 2005.

[30] A. Degani, H. Choset, B. Zubiate, T. Ota, and M. Zenati, “Highly articulated robotic probe for minimally invasive surgery,” in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 3273–3276, Aug 2008.

[31] M. W. Hannan and I. D. Walker, “Analysis and experiments with an elephant’s trunk robot,” *Advanced Robotics*, vol. 15, no. 8, pp. 847–858, 2001.

[32] Y. Chen, J. Chang, A. Greenlee, K. Cheung, A. Slocum, and R. Gupta, “Multi-turn, tension-stiffening catheter navigation system,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5570–5575, May 2010.

[33] T. Maahs, V. Saadat, C. Rothe, and T. Le, “Disposable shapelocking system,” Sept. 28 2005. US Patent App. 11/238,298.

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

[35] W. McMahan, B. Jones, and I. Walker, “Design and implementation of a multi-section continuum robot: Air-octor,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2578–2585, Aug 2005.

[36] Y. Bailly and Y. Amirat, “Modeling and control of a hybrid continuum active catheter for aortic aneurysm treatment,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 924–929, April 2005.

[37] Credit, “European Commission Research and Innovation.” `http://www.stiff-flop.eu/`, 2012. [Online; accessed 23-March-2014].

[38] D. Noonan, V. Vitiello, J. Shang, C. Payne, and G.-Z. Yang, “A modular, mechatronic joint design for a flexible access platform for mis,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 949–954, Sept 2011.

[39] M. S. Loffler, N. P. Costescu, and D. M. Dawson, “Qmotor 3.0 and the qmotor robotic toolkit: a pc-based control platform,” *Control Systems, IEEE*, vol. 22, no. 3, pp. 12–26, 2002.

[40] M. M. Tonapi, I. S. Godage, and I. D. Walker, “Next-generation rope like robot for in-space inspection,” in *IEEE Aerospace Conference*, March 2014.

[41] `http://www.arduino.cc/`, 2014. [Online; accessed 25-April-2014].

[42] M. M. Tonapi, I. S. Godage, and I. D. Walker, “Design, modeling and performance evaluation of a long and slim continuum robotic cable (to appear),” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, IEEE, September 2014.

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

[44] I. Godage, E. Guglielmino, D. Branson, G. Medrano-Cerda, and D. Caldwell, “Novel modal approach for kinematics of multisection continuum arms,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1093–1098, Sept 2011.

[45] N. I. of Standards and T. (NIST), “Engineering Statistics Handbook.” `http://http://www.itl.nist.gov/div898/handbook/ppc/section4/ppc431.htm/`, 2003. [Online; accessed 18-April-2014].

[46] D. F. Brunner, *Development of a continuum robotics electromechanical testbed.* University of Wisconsin–Madison, 2009.