Real-time and Customizable Inverse Kinematics for Extensible Multisection Continuum Robots: SLInKi and AMoRPH Algorithms

Shou-Shan Chiang

A dissertation presented for the degree of
Doctor of Philosophy

Faculty Advisor: Cagdas Onal, Ph.D.
Worcester Polytechnic Institute
Worcester MA
Dec 2023

Dissertation Committee

Cagdas Onal, Ph.D. *advisor*	Department of Robotics Engineering, Worcester Polytechnic Institute
Jing Xiao , Ph.D.	Department of Robotics Engineering, Worcester Polytechnic Institute
Berk Calli, Ph.D.	Department of Robotics Engineering, Worcester Polytechnic Institute
Yunus Telliel, Ph.D.	Department of Humanities & Arts, Worcester Polytechnic Institute

In Memory of

Ms. S.-H. Pan

A beloved mother and mother-in-law
whose kindness and wisdom brightened
our lives and will be forever missed.

Abstract

Extensible multisection continuum robots are characterized by their continuous and compliant structures, which provide them with superior adaptability and flexibility compared to traditional robots. Nonetheless, the complexity of their kinematics and inverse kinematics presents significant challenges. This study addresses these challenges by introducing two innovative algorithms for inverse kinematics: SLInKi and AMoRPH. SLInKi is a hybrid approach that combines state lattice searching algorithms, exFABRIKc, and analytical solvers. Its main focus is on real-time solution and optimization of shape and motion smoothness, allowing for customization. On the other hand, AMoRPH is a novel analytical solver that utilizes a virtual linkage model in order to efficiently solve the 5 degree-of-freedom inverse kinematics problem and find balanced and smooth solutions. When faced with non-uniform cases and obstacles, segment addition and geometry adjustments are employed. This algorithm effectively addresses the inverse kinematics problem with speed, accuracy, and versatility. The validity of both algorithms is demonstrated through simulations and experiments, which showcase their effectiveness in real-time control, precise tracking, and obstacle avoidance for continuum robots.

Acknowledgments

I would first like to thank my beloved wife, Sandra, for her endless love and support during the most difficult times. Your strength has inspired me daily.

Sincerest thanks to my delightful dogs, Wowo and Hana, for tolerating my irregular walking schedules and accompanying me during countless late nights of study. Your playful presence kept my spirit afloat.

I wish to express my heartfelt gratitude to my family for their unwavering support throughout this journey. You were my refuge.

Special thanks to my advisor, Dr. Cagdas Onal, for his guidance, wisdom, and continuous encouragement. Under your mentorship, I have grown both academically and personally.

I am also deeply grateful to my laboratory colleagues, whose collaboration and camaraderie made the research experience enriching and enjoyable. Our teamwork and friendship have left me with fond memories.

This material is based upon work partially supported by the National Science Foundation (NSF) under Grant No. CMMI-1752195 and DGE-1922761. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid progress in the field of robotics and artificial intelligence has ushered in a new era of technological possibilities. However, it is crucial to acknowledge the inherent risks associated with these machines, particularly in their interaction with humans. As robots become increasingly prevalent, it is imperative to approach coexistence with these entities thoughtfully and responsibly, with an emphasis on human safety and reliability in their design and operation.

To navigate this complex landscape, it is essential to establish rigorous ethical and safety protocols to govern the development of robots. These protocols form the basis for ensuring predictable and controllable behavior of robots, thereby promoting a harmonious coexistence between humans and machines. Simultaneously, the implementation of robust regulatory systems is of utmost importance, guiding the trajectory of robotics progress in line with societal interests and minimizing potential harm.

In science fiction narratives centered around robots, continuum arms find application in various instances. Noteworthy examples include the tentacles of Dr. Octopus, the aerial appendages of Laputa's guardian robot, and the sophisticated robots depicted in Naoki Urasawa's Pluto. These imaginative depictions, while showcasing the potential of continuum arms, underscore the existing disparity with the reality of the field of robotics.

While these imaginative depictions showcase the potential of continuum arms, the reality in the field of robotics presents a different picture. Research on continuum arms lags behind that of traditional articulated robot arms. In the broader field of robotics, mechanics, dynamics, and control have already become foundational subjects. Yet, when it comes to continuum arms, there is still no solution robust enough for real-world application.

This study embarks on a visionary pursuit, systematically unraveling the challenges, identifying gaps on the path to answers, and conscientiously bridging each one. The aspiration is to push the boundaries of human intelligence even further. Moreover, the figures accompanying this study illustrate specific examples relevant to the discourse, providing visual context for the concepts discussed.

1.1 Motivation

To fully unlock the potential benefits of robots, two fundamental attributes are indispensable:

1. A reliable mechanism ensuring human safety, preventing any potential harm or injury.
2. Responsive capabilities for executing real-time assistive actions promptly and effectively.

Addressing these needs calls for a focus on soft continuum robots, equipped with inherent compliance to minimize potential harm. Additionally, the deployment of precise and expeditious inverse kinematics algorithms ensures the necessary real-time responsiveness of continuum robots.

1.2 Problem Statement and Scope

The aim of this study is to create a fast and accurate Inverse Kinematics (IK) algorithm that can effectively ascertain the position and orientation of a multisection extensible continuum robot. Emphasis is placed on developing a practical solution that accounts for the constraints and capabilities of real-world robotic systems.

This study focuses on the computation of inverse kinematics (IK), assuming the presence of a few obstacles in the environment. While hardware workspace constraints will be taken into account, any discrepancies stemming from mechanical assembly errors and system control inaccuracies will not be addressed. In the computation of errors, the results of the IK calculations will be incorporated into forward kinematics (FK) simulations, rather than accounting for errors in the physical robot.

1.3 Contributions

This research addresses the aforementioned problems by introducing two Inverse Kinematics (IK) algorithms:

SLInKi:

- A fast and accurate IK solution designed for extensible continuum robots.
- Extension of FABRIKc (exFABRIKc) to make it applicable to extensible continuum robots.
- Introduction of a geometric analytical solution.
- Utilization of State Lattice Planner for IK calculations.
- Adaptability to various cost functions for posture adjustments in different scenarios.
- Proposal of a method to integrate two different IK algorithms seamlessly.

- Implementation of obstacle avoidance strategies.

AMoRPH:

- Rapid, accurate, and smooth IK solution.
- Introduction of a geometric analytical solution.
- Ensuring smooth shape and motion transitions.
- Implementation of a specialized segmentation method to ensure uniform length for each section.
- Incorporation of whole-body collision detection mechanisms.
- Integration of obstacle avoidance strategies.

Chapter 2

Related Work

Continuum robots bend continuously along their length instead of at discrete joints [13]. Their compliance, safety, and ability to negotiate constrained environments have led to renewed attention over the past decade. However, inverse kinematics (IK) calculations are challenging for continuum robots due to these properties.

2.1 Continuum Robots

Biology has facilitated significant progress in the field of robotics, specifically in the realm of continuum robots. Engineering marvels, such as the serpentine locomotion of snakes and the agile capabilities demonstrated by elephant trunks, tongues, and octopus tentacles in their interactions with their surroundings, have naturally served as inspiration for researchers aiming to replicate these exceptional functionalities in electromechanical devices. In this burgeoning research area, numerous novel continuum robots are published each year, and many review papers employ various classification methods to analyze and compare these continuum robots.

These classification criteria encompass several aspects, namely: continuous or discrete, extensible, number of sections, actuators per section, degrees of freedom per section, actuator spacing, actuation method, multi-section coupling, application domain,

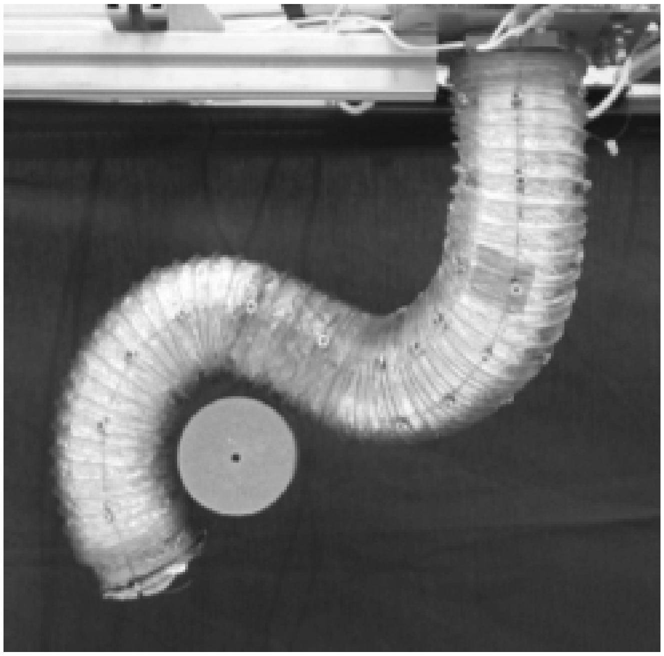


Figure 2.1: Air-OCTOR multisection continuum robot [1]

bio-inspiration, robot structure, material composition, type of sensors, and design principles. [9, 14, 15] In the following paragraphs, we will present an introduction and classification discussion on several representative continuum robots.

The Air-OCTOR [1] and OctArm [2] designs, as illustrated in Fig. 2.1 and Fig. 2.2 respectively, both employ a similar design concept. These mechanisms are characterized by their continuous and extensible nature, featuring three actuators per section and providing three degrees of freedom per section. Inspired by the functionality of an elephant trunk, both systems are designed for general-purpose applications. However, there are a few distinctions between them. Firstly, the number of sections differs, with the Air-OCTOR employing two sections while the OctArm utilizes three sections. Additionally, the actuation mechanisms of these systems also vary, with the Air-OCTOR being tendon-driven and the OctArm utilizing pneumatics.

Festo's award-winning Bionic Handling Assistant (BHA) [3], depicted in Fig. 2.3, also takes inspiration from the design of an elephant trunk. This robot features a 3-segment pneumatics continuum robot that utilizes cables to detect its position. In addition,

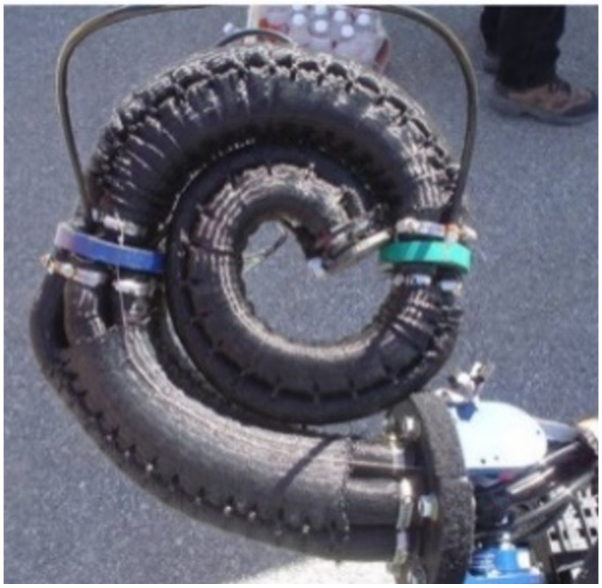


Figure 2.2: OctArm V-Continuum manipulator [2]

it boasts a high level of integration, an independent wrist degree of freedom, and an Adaptive Gripper based on the patented Fin Ray Effect®. The design of this robot is aimed at facilitating safe interaction between humans and machines.

Marchese et al. [4] have put forth a proposition for a fully soft silicone rubber arm with modular design and fabrication. This arm (Fig. 2.4) consists of multiple modules or segments, each of which is equipped with four fluidic actuators that allow for movement in three dimensions. Noteworthy aspects of this manipulator include a hollow cylindrical interior for fluid transmission lines and soft end plates for the segments. The presence of a hollow interior aids in the assembly of modules, while markers on the end plates facilitate the capture of segment endpoint positions by an external vision system, which is crucial for an autonomous manipulation system. It is also important to mention that the arm's exterior shape, which tapers longitudinally and resembles a clover leaf, has a significant influence on its dynamics.

In their study, Renda et al. [5] introduced a tendon-driven manipulator (Fig. 2.5) that drew inspiration from the structure of an octopus arm. The robotic arm consisted of a singular conical piece of silicone, which represented a Cosserat beam. Multiple

Figure 2.3: Festo's BHA [3]

Figure 2.4: Soft fluidic elastomer manipulator gently taking a small ball from the palm of its human collaborator. [4]

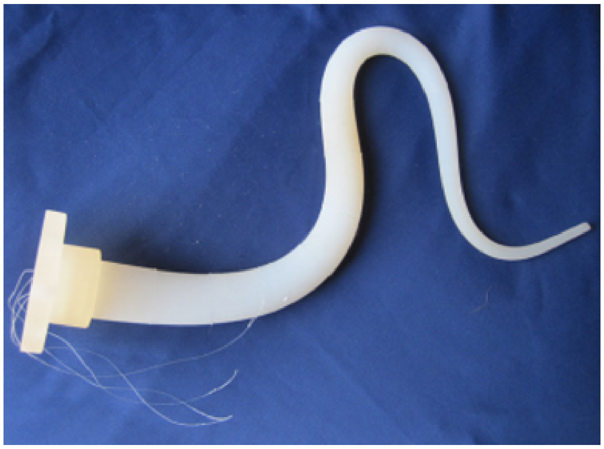


Figure 2.5: Octopus-like arm prototype. [5]

cables were situated within the body of the arm and connected to a rigid plastic disc that was integrated into the arm at various distances from its base. These discs were incorporated into the silicone during the manufacturing process and could be positioned at any location within the arm.

Qi and the team [6] employed a 3D printer to create a continuum-style robot that is notable for its unique design consisting of numerous layers of compliant planar modules connected in a series (Fig. 2.6). What sets this robot apart is that its flexibility is not dependent on the inherent compliance of a continuous central component or soft skeletal elements, but rather it makes use of the inherent compliance found in each structured planar module. The utilization of compliant planar modules offers several significant benefits, such as the ability to generate linear output motion and the avoidance of friction among the joints.

Zhang and colleagues introduced a novel type of extensible continuum robots (Fig. 2.7), drawing inspiration from origami. [7] They achieved this by integrating an origami structure as a skeleton and utilizing a helical spring as a backbone. Initially, they outlined the design process of a three-degree-of-freedom origami parallel structure, which is formed from a two-dimensional crease pattern on a flat sheet. Subsequently, by incorporating a helical spring into the origami parallel structure, they created a compliant parallel

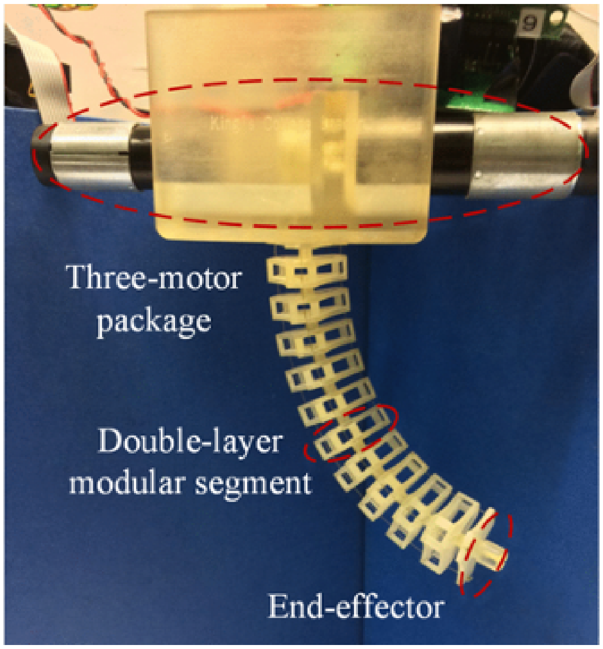


Figure 2.6: 3-D printed prototype of the proposed continuum manipulator concept with multilayer deformable planar springs in series and articulated by tendons which are actuated by motors in the periphery. [6]

module. This innovative approach resulted in a continuum robot with six successive and interconnected compliant modules. They then proceeded to investigate the virtual rotational axis and orientation of the upper platform of the compliant parallel module utilizing geometry-based kinematic principles.

Irrespective of the variations in the designs of continuum manipulators, their kinematics can often be described using a piece-wise constant curvature (PCC) model. This perspective was presented in a comprehensive review conducted by Webster and Jones [9] on the topic of continuum robots. In their review, Webster and Jones discuss different approaches to kinematic modeling and demonstrate that these approaches yield similar results when the PCC modeling assumption is applied. This assumption implies that each segment of a multisection arm deforms uniformly with a constant curvature. We will further examine this constant curvature assumption in Sec. 3.2.1.

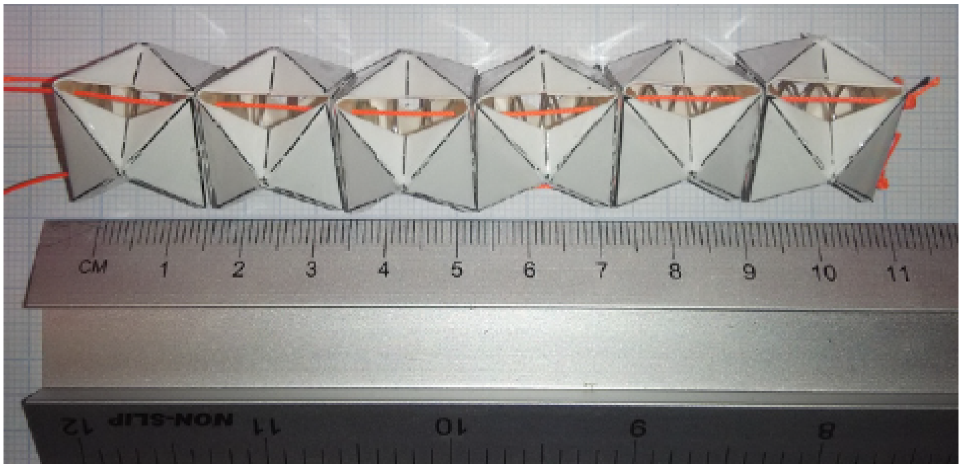


Figure 2.7: A prototype of the proposed extensible continuum robot with six parallel modules. [7]

2.2 Inverse Kinematics for Continuum Robots

Just like all serial robots, and especially in the case of redundant robots, inverse kinematics poses a challenge. In order to address this challenge for a constant-curvature continuum robot, one typically begins by computing the inverse mapping between the task space and the configuration space, known as the robot-independent inverse mapping. This computation yields the arc parameters for all sections of the robot, corresponding to the desired tip pose. However, solving this problem for a continuum robot with a large number of sections is not a trivial task. Nevertheless, there are several approaches available for inverse kinematics.

A closed-form geometric approach is presented in Neppalli et al. [16] for both single-section and multiple-section cases. This approach involves modeling each section using a spherical joint and a straight rigid link, and subsequently applying an analytical process to solve the inverse kinematics for this model. The resulting solution is then converted back into arc parameters. One of the advantages of this approach is that it provides a straightforward means of accessing the complete solution space of the rigid-link robot, and it can readily be extended to an n-link robot. However, it is worth noting that this approach can only calculate the position but not the orientation, and this formulation does not yet take into account physical actuation limitations such as constraints on actuator lengths, forces, or locations.

Traditional numerical methods for inverse kinematics of rigid robots, such as Jacobian-based techniques, are effective. The Jacobian approach offers a robot-independent IK solutions. By driving a virtual copy of the robot from any initial estimate, including its current configuration, to the desired configuration, it finds a unique solution. [17] In this strategy, actuator limits can be incorporated into the control law to ensure physical realizability of the robot's trajectory. Nonetheless, when applied to continuum robots, these methods require substantial computational resources. Furthermore, they do not guarantee an accurate solution. [18]

In their study, Jones and Walker [19] effectively converted a single section of a continuum module into a serial linkage robot of equivalent characteristics, utilizing the constant curvature assumption. The forward kinematics (FK) of the serial linkage robot were derived using the Denavit-Hartenberg (D-H) convention, enabling calculation of the corresponding cable length and establishment of connections between the work space, configuration space, and actuator space. Calculating the Jacobian of these transformations allows for obtaining the inverse kinematics (IK) equation of the entire module.

Santoso and Onal [8] are based on Jones' method, but simplify the process by calculating the Jacobian directly from the FK equation. In addition, Santoso also uses the cable length of the actuator space directly with the FK equation and uses optimization to find the solution.

Because continuum robot bodies bend continuously along their length, they possess infinite passive degrees of freedom (DoF). Most researchers make the common assumption that each finite section of the continuum body follows a constant curvature arc, meaning that a bending direction, bending curvature, and arc length are sufficient to represent their state [8]. Using these parameters, end-effector poses can be calculated via forward kinematics [20]. In contrast, the inverse problem-using desired tip poses to calculate arc profiles-is much more difficult for continuum manipulators, since they do not always possess closed-form solutions [21, 22, 23, 16].

Zhang et al. [10] based on the algorithm FABRIK [24] which is used to handle the IK of serial linkage robots, importing an additional layer of configuration space: virtual link/joint allows the algorithm to solve the continuum robot problem. This method is very fast and can take into account position and direction. Subsequent successors like [25], [26] have modified the computation process to be faster or more accurate. CRRIK of Wu et al. uses a similar approach [27].

CIDGIKc [28] is an inverse kinematics solver based on the distance-geometric approach. The distance-geometric formulation is commonly employed for the computation of complex spatial geometric problems, such as those encountered in chemical structure analysis. This method has the capability to determine the IK solution for extensible multisection continuum robots operating in environments with numerous obstacles. However, it should be noted that this method exhibits a slow computational speed. In experimental scenarios, the time required to obtain a solution ranges from 3 to 50 seconds.

Xiao and Vatcha [29] present an online motion planner for planar continuous manipulators. This method plans the motion of the entire continuous manipulator in real time to grasp a specific object in an environment where the movement of other objects is uncertain. The calculated pose enables the continuous manipulator to avoid obstacles while reaching the target location. However, this method is limited to calculating motion in two-dimensional space.

Li and Xiao [30] presented a comprehensive formulation for the problem and a method for achieving automatic planning in mobile continuum manipulation, taking into account various task and environmental constraints. This approach not only ensures collision avoidance but also ensures the fulfillment of specific position and orientation constraints as required by the task. In cases where there is a collision on the concave side, a new

Method	Fast	Accurate in 3D	Solution Gauranteed	Customizable	Obstacle Avoidance
Jacobian-based		X			
Optimization-based		X			
FABRIKc	X	X	X	X	
Analytical Geometry	X		X	X	
CIDGIKc		X	X	X	X
Extended RAMP	X		X	X	X

Table 2.1: IK Solver Functionality Requirements

configuration of the section is generated by increasing the radius of the arc.

In consideration of the requirements mentioned in the problem statement (Sec. 1.2), it is evident that while various algorithms exist that are suitable for different purposes, none of them, as of now, fully meet all the specified requirements in our application context. A comparative analysis of these algorithms is provided in the table 2.1.

Chapter 3

Experimental Platform and Modeling

We utilize the origami continuum arm as our primary platform for the modeling, designing, and testing of the inverse kinematics (IK) algorithms. The lightweight feature of the robot allows it to easily adhere to the constant curvature assumption. The Yoshimura origami structure provides high axial stiffness, while the cable-driven design enables pose control through low-level motor PID control. The modular characteristic of the robot brings attention to the significance of customization. Additionally, the use of a piece-wise constant curvature model allows for the handling of various configurations of the entire robot.

In the latter half of this chapter, we will delve into the kinematic model of continuum robots. We will begin by discussing the constant curvature assumption, which serves as the foundation for all models. All of our models are built upon this assumption. Subsequently, we will explore two models and examine the interrelationships between task space, configuration space, and actuator space.

3.1 OrCA: Origami Continuum Arm

The initial prototype of the origami continuum arm is presented in the work by Santoso et al. [8] The majority of our SLInKi experiments (see Sec. 4) were conducted using this robot. experiments ran on this robot. This robotic module demonstrates exceptional capabilities, including a considerable alteration in length (approximately 1.25 times its original length) and exceptional torsional resistance (73 times stronger compared to a silicone module of comparable size). These desirable traits are owed to the tubular, accordion-like tessellated origami structure, which is manufactured from a 0.178 mm-thin PET sheet. The design of the origami structure adheres to the Yoshimura origami crease pattern. Each origami continuum module consists of a pliant plastic body, an acrylic end plate, and a custom-made PCB that employs an 8-bit ATmega32U4 microprocessor for embedded control. Additionally, four electrical wires are meticulously arranged in a helical shape within the interior cavity of the origami structure, serving the function of providing power and facilitating communication between continuum modules. The module control boards are able to establish communication with a desktop computer by employing the Inter-Integrated Circuit (I2C) protocol.

Each of the origami structures is bordered on both sides by acrylic plates in the shape of equilateral triangles, measuring 78.35 mm in height. These plates serve the dual purpose of providing structural support and serving as mounts for the electronics. To facilitate movement, cable actuator motors are attached at each corner of the plate, and the cables traverse holes within the origami structure. The distance between the center of the plates and the cable spool measures 40 mm. The length of an origami continuum module can vary between 40 mm when compressed and 100 mm when fully extended. However, the flexibility of the origami structure is insufficient to sustain its own weight when in the fully extended configuration. To address this, we have established a maximum cable length of 80 mm for all three cables. This constraint results in a maximum bending angle of 38.20°. The overall robot manufacturing and

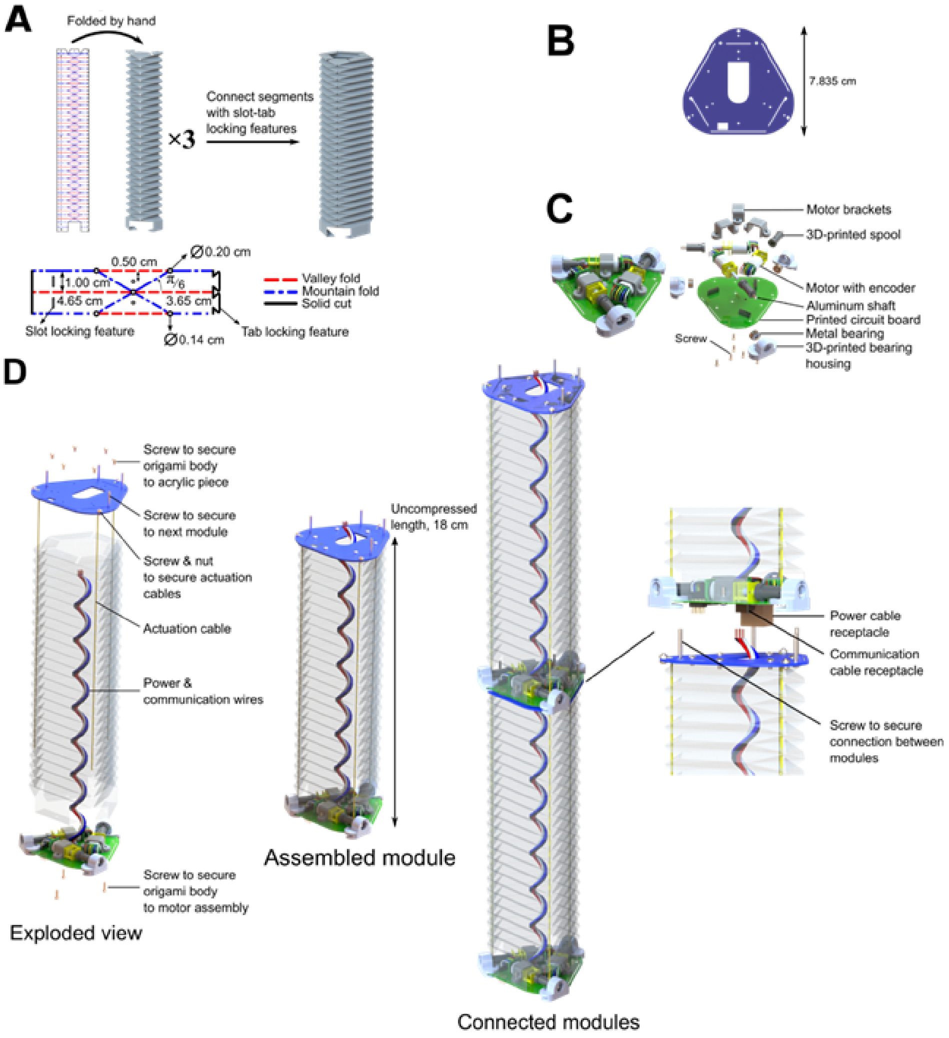


Figure 3.1: (A) Origami body assembly, (B) acrylic top plate, (C) motor assembly, (D) origami module assembly. [8]

assembly process is presented in Fig. 3.1.

Later on, we carried out electrical and mechanical enhancements based on this prototype. In order to improve efficiency and stability, we devised a novel printed circuit board (PCB) which minimized the distance of connections between modules, commonly referred to as deadlength, and reduced their overall weight. To ensure proper functioning, each module was controlled by three cables, each with varying lengths ranging from 80 mm to 200 mm. During our experimentation phase, we assembled three modules to construct a three-section extensible continuum robot, successfully achieving a remarkably low deadlength of merely 10 mm. All the AMoRPH experiments (see Sec. 5) were conducted using this robot.

The PCB is a symmetrical hexagon with a diameter of 55 mm that encompasses its outermost points. It possesses mounting holes situated at each corner and accommodates all components on a single side. Its primary purpose is to operate and control up to three brushed DC motors. It accomplishes this by providing each motor with a pulse width modulation (PWM) drive signal capable of reaching values of up to 18 V and 1 A. Additionally, it reads the output of the motor quadrature encoder and keeps track of the bidirectional motor current. Furthermore, it features an RS485 communication interface that facilitates the interconnection of multiple boards through a single bus. This is achieved via a process known as daisy-chaining. The custom USB/RS485 converter printed circuit board assembly (PCBA) allows communication with the application software. It achieves this by implementing a USB HID class interrupt endpoint that ensures all motor drivers on the bus maintain a synchronized 1 ms update rate.

3.2 Modeling

Based on the constant curvature assumption, we employ the kinematic model of arc-length (s), curvature (κ), and bending direction (ϕ) to represent a single module. Additionally, we utilize the prismatic-joint, universal-joint, prismatic-joint (PUP) model to represent equivalent configurations. It is important to note that both of these models are robot-independent, meaning they are not contingent upon the specific mechanism of the robot. As long as the robot satisfies the constant curvature assumption, these models can be applicable. In other words, the inverse kinematics solver based on these models is of a general-purpose nature. While our design and testing are specifically tailored to our self-manufactured Origami Continuum Arm, the inverse kinematics solvers can be utilized on other continuum robots as well.

3.2.1 Constant Curvature Assumption

In the context of robotics and mechanical systems, curvature refers to a fundamental geometric property that characterizes the degree of bending or deviation from a straight line in a given curve or trajectory. Mathematically, curvature (κ) at a specific point on a curve is defined as the reciprocal of the radius of the osculating circle at that point.

In the study of continuous and flexible structures, particularly in the field of robotics, the curvature of a deformable component plays a crucial role in determining its kinematics and behavior during motion. The concept of curvature is essential for understanding the geometric deformation of flexible elements, such as soft robotic arms or continuum manipulators, as they navigate through space.

The Constant Curvature Assumption posits that, for a certain segment, a flexible component of a robotic system maintains a consistent curvature. This assumption simplifies the modeling and analysis of the system by assuming a relatively uniform curvature, facilitating the application of mathematical techniques and control methodologies.

3.2.2 Task Space, Configuration Space, and Actuator Space

Task space is the work space, in robotic kinematics, we use it to represent the attitude of the robot, usually more precise is the pose of the End-Effector, that is, the position and orientation of the 6 DoF. Usually, it is represented by Cartesian coordinates.

The variables employed to represent the configuration space according to Jones' definition [19] are denoted as s, κ, and ϕ. By employing Jones' methodology, the connection between task space and configuration space for a continuum module is determined through the following Equation 3.1. It is worth noting that in Figure 3.2, the arc length is represented as l instead of s. In this article, we will consistently use s to denote arc length, while l will be used to represent cable length.

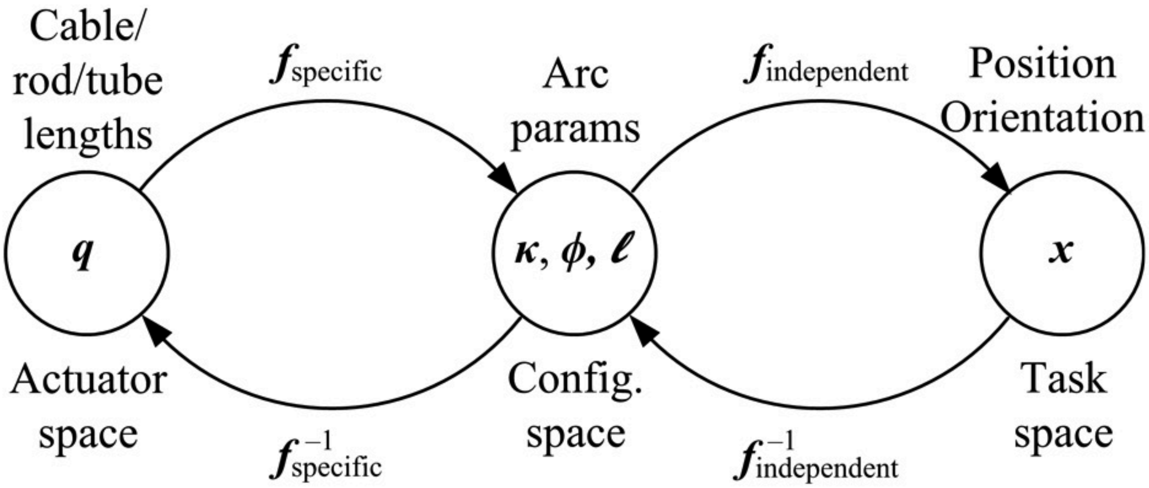


Figure 3.2: The kinematics of constant-curvature robots are determined by three spaces and the mappings between them. A robot-specific mapping converts the variables q in the actuator space to variables κ, ϕ, l in the configuration space. Subsequently, a robot-independent mapping transforms these configuration space variables into the task space. [9]

$$T = \begin{bmatrix} \cos^2\phi(\cos\kappa s - 1) + 1 & \sin\phi\cos\phi(\cos\kappa s - 1) & \cos\phi\sin\kappa s & \frac{\cos\phi(1-\cos\kappa s)}{\kappa} \\ \sin\phi\cos\phi(\cos\kappa s - 1) & \cos^2\phi(1 - \cos\kappa s) + \cos\kappa s & \sin\phi\sin\kappa s & \frac{\sin\phi(1-\cos\kappa s)}{\kappa} \\ -\cos\phi\sin\kappa s & -\sin\phi\sin\kappa s & \cos\kappa s & \frac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

In Jones' study, an actuator space is defined that can be used to manipulate the pose of the continuum module using pressure (in the case of an artificial muscle) or cable-length. However, since this study specifically examines cable-driven continuum robots, the term cable-length is used for convenience throughout the document. In Fig. 3.2, Webster and Jones clearly explain the relation of these spaces.

Based on the geometrical definition, the Jones team set three cables with controllable lengths on a plane with a 120-degree angle between them, and the distance between these three cables will not change during the length change process, that is, the plane will be

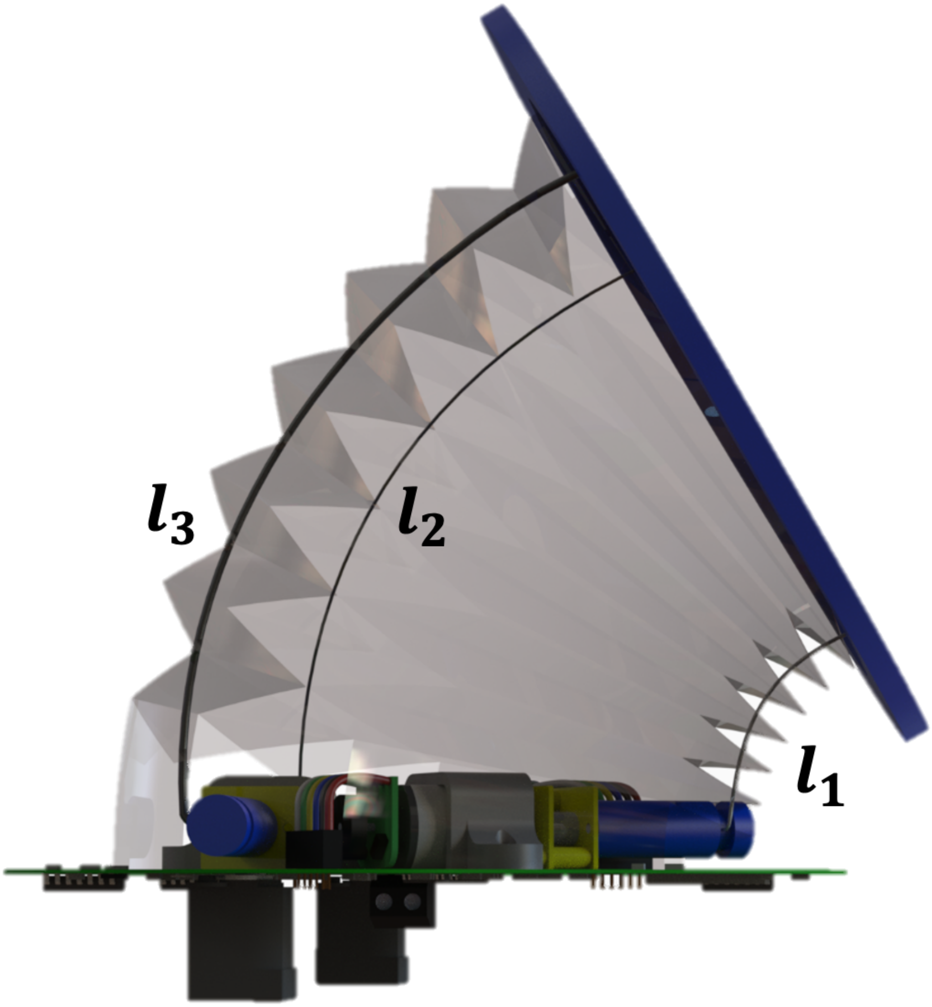


Figure 3.3: A computer-aided design (CAD) visualization depicting a module of a continuous origami robot with a consistent curvature. The illustration includes clearly marked cable lengths denoted as l_1, l_2, l_3.

orthogonal to the three cables and the three points will be maintained in an equilateral triangle as seen from the center of the circle in the bending direction.(Fig. 3.3)

$$\kappa = 2\frac{\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3}}{d(l_1 + l_2 + l_3)} \tag{3.2}$$

$$\phi = \tan^{-1}\left(\frac{\sqrt{3}}{3}\frac{l_3 + l_2 - 2l_1}{l_2 - l_3}\right) \tag{3.3}$$

$$s = \frac{(l_1 + l_2 + l_3}{3} \tag{3.4}$$

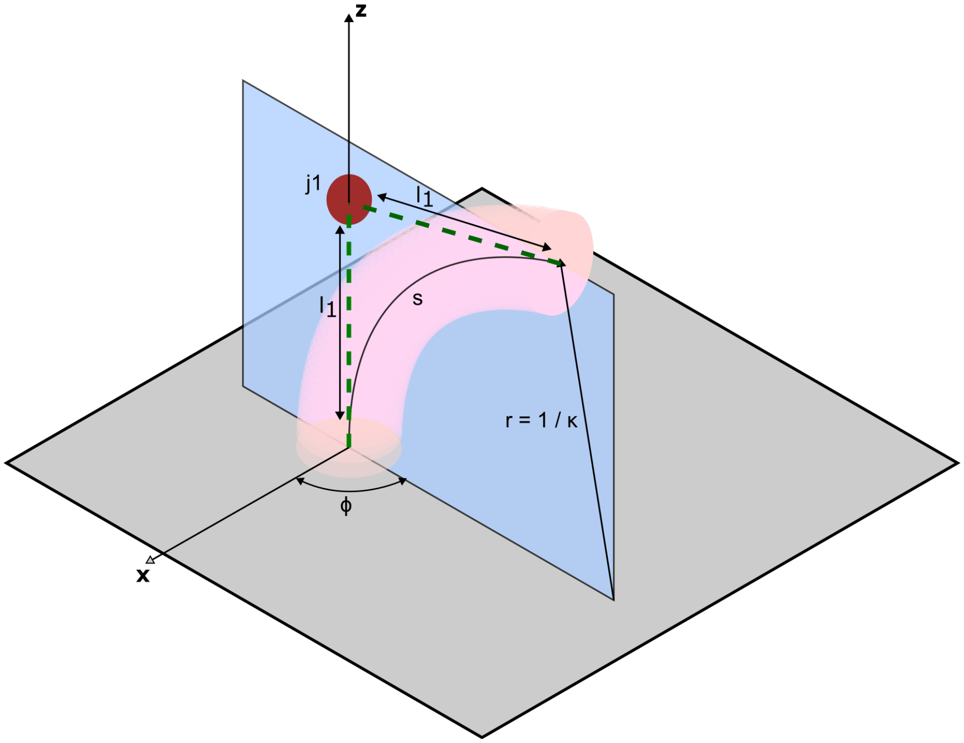


Figure 3.4: A constant curvature continuum robot module can be represented by arc length(s), curvature(κ), and bending direction(ϕ) or equivalently by a set of virtual joints/links. (*j1*) formed at the intersection of the two equal-length virtual links (l_1). Where r is the radius of the arc.

Zhang et al. [10] introduce the virtual PUP model (Fig. 3.4) as an additional layer of configuration space. Under the constant curvature assumption, these configuration space parameters are equivalent to the convention arc length s, curvature κ and bending direction ϕ parameters and can be used to describe the same configuration one-to-one. The calculation method involves treating the base and tip of the virtual prismatic joints as two separate links and setting their prismatic joint extension to be equal in the direction of the base and tip. This ensures that they satisfy the constant curvature assumption. By finding the intersection point, we can identify the virtual universal joint. After determining the location of the base, the virtual universal joint's position, and the bending angle, we can represent s, κ, ϕ accordingly.

Based on the virtual link/joint method introduced by FABRIKc, we have an alternative representation of configuration space. Under the constant curvature assumption, these two configuration space parameters are equivalent and can be used to describe the same configuration one-to-one. The calculation method involves treating the base and tip of the virtual link as two separate links and setting their virtual link lengths to be equal in the direction of the base and tip. This ensures that they satisfy the constant curvature assumption. By finding the intersection point, we can identify the virtual joint. After determining the location of the base, the virtual joint's position, and the bending angle, we can represent s, κ, ϕ accordingly. The exFABRIKc and Analytical Solution paragraphs that follow will utilize this configuration space for inverse kinematics calculations.

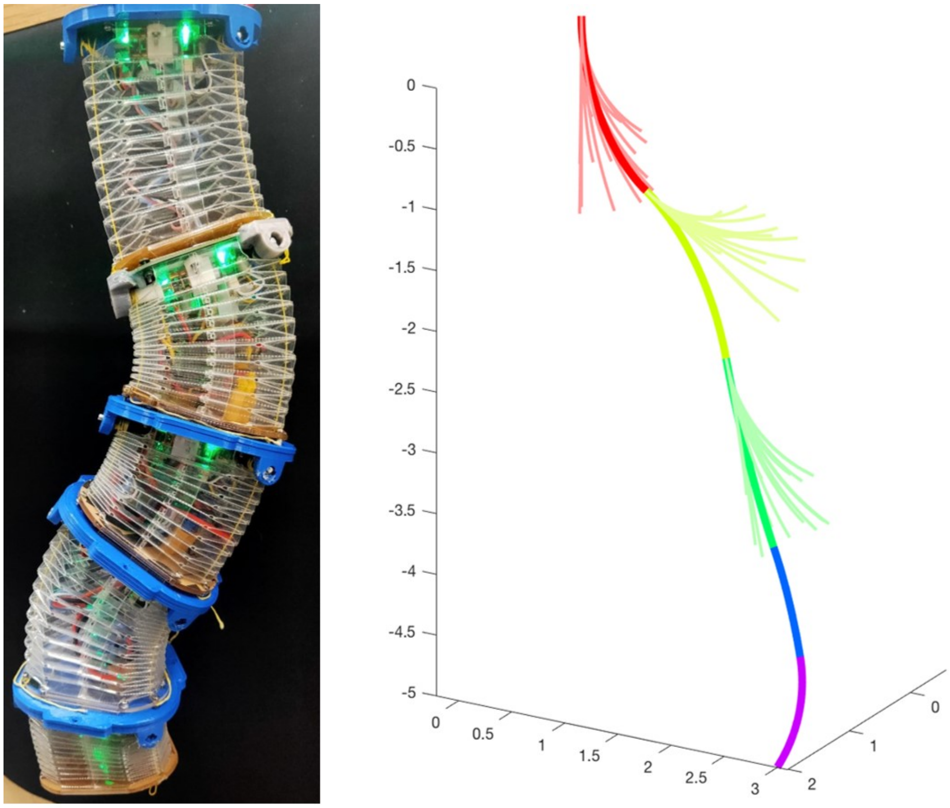
Chapter 4

SLInKi

We propose a heuristic approach, the State Lattice based Inverse Kinematics Solver (SLInKi). This approach treats the configuration profiles of continuum robot as a path. Due to its hyper-redundant nature, the end-effector is able to reach a desired pose with multiple configurations. The inverse kinematic solver is implemented to find proper configurations for each section of the robot. From this aspect, the path-finding algorithms share a similar objective: to explore and connect adjacent nodes until arriving at the destination node.

4.1 Algorithm Description

As previously stated, the IK algorithm employed in this study considers the configuration as a path and addresses the IK problem accordingly. However, it should be noted that pathfinding algorithms incur a substantial computational cost when attempting to attain a desired orientation. To overcome this limitation, we combine a state-lattice approach with a supplementary method that readily accounts for the desired orientation. Nonetheless, this alternative technique is less efficient and tailored for larger quantities of modules. Specifically, our strategy involves employing a state-lattice approach to calculate the IK for reaching the vicinity of the goal without utilizing all modules of



(a)

(b)

Figure 4.1: (a) An origami-inspired continuum robot is modular and scalable. The current prototype consists of five modules, each being able to bend in 3-D and change in length. (b) An instance of robot pose simulated by SLInKi search algorithm, with alternate module configurations searched.

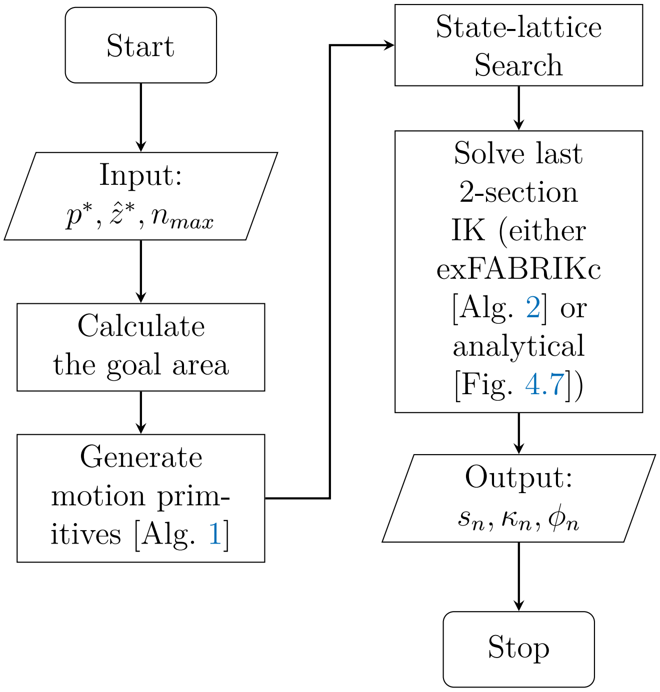


Figure 4.2: Flowchart for the SLInKi algorithm. Note that p^* & $\hat{z}^*$ denote goal position & orientation separately, while n_{max} represents the overall number of modules. s_n, κ_n, ϕ_n denote the configurations for all modules. The details of the goal area, motion primitives (Alg. 1), state-lattice searching and 2-section IK solvers - exFABRIKc (Alg. 2) and analytical (Fig. 4.7) - are presented in Sec. 4.1.5 and Sec. 4.1.6.

the continuum manipulator. Subsequently, we employ a dissimilar algorithm that expeditiously reaches the final goal position and orientation by employing the remaining manipulator modules. In this investigation, we evaluate the performance of SLInKi using 1) the iterative algorithm exFABRIKc, involving the utilization of the final two modules, as well as 2) a purely analytical approach employing solely the final module. Fig. 4.2 illustrates the flowchart describing the workflow of the proposed algorithm.

4.1.1 State-lattice planner

In our application, each module of the continuum robot has 6 DoF in the task space. However, the position and orientation are correlated, which constrains the end-effector orientation when translating to the desired position. In this regard, we consider the IK of our robot to be a kinodynamic issue, which can be solved via a motion planning algo-

Table 4.1: Nomenclature used in kinematics modeling and algorithm

s, s_{max}	Arc length and maximum allowable value
κ, κ_{max}	Curvature and maximum allowable value
ϕ, ϕ_{max}	Bending orientation and maximum allowable value
m	Index of motion primitives
T_m	Transformation matrix of mth (s, κ, ϕ) setting
$\delta_s, \delta_\kappa, \delta_\phi$	snap-width of s, κ, ϕ
n	Number of modules
t	Index of the modules, t=1,2...n.$t = 1$ for the most proximal module and $t = n$ for the most distal module
s_t	Arc length of tth module
κ_t	Curvature of tth module
ϕ_t	Bending orientation of tth module
e	Position error
ϵ	The threshold of the error
l_t	Length of the virtual links on tth module
p^*	The target position in the world coordinate
$\hat{z}^*$	The target orientation in the world coordinate
p_{tj}	Virtual joint position of tth module
p_{te}	End position of tth module
p_{tb}	Base position of tth module
$\hat{z}_{te}$	End orientation of tth module
$\hat{z}_{tb}$	Base orientation of tth module
k, k_{max}	The iteration index and the maximum allowable iterations

rithm, state lattice-based planner [31]. Compared with many other graph-based search algorithms (such as 8-connected grids), the feasibility requirement of lattice connections guarantees that any solutions found with a state lattice-based approach will be feasible. [32] The disadvantage is that completeness cannot be guaranteed.

We generate the next motion primitive with the parameters (s, κ, ϕ), and calculate the corresponding poses with forward kinematics. To avoid discontinuities and singularities, when the curvature (κ) is equal to or very close to zero, we only generate motion primitives with rotation angle $\phi = 0$.

Algorithm 1: Generate Motion Primitives

```
Result: List of transformation matrix and the index of motion primitives
$\kappa \leftarrow 0, \phi \leftarrow 0, s \leftarrow 0, m \leftarrow 0$;
while $s < s_{max}$ do
    $T_m = ForwardKinematics(s, \kappa, \phi)$;
    $index_m = [s, \kappa, \phi]$ ;
    $s = s + \delta_s$;
    $m = m + 1$;
end while
$\kappa \leftarrow \delta_\kappa$ ;
while $\kappa < \kappa_{max}$ do
    $s \leftarrow \delta_s$;
    while $s < s_{max}$ do
        $\phi \leftarrow 0$;
        while $\phi < \phi_{max}$ do
            $T_m = ForwardKinematics(s, \kappa, \phi)$;
            $index_m = [s, \kappa, \phi]$;
            $\phi = \phi + \delta_\phi$;
            $m = m + 1$;
        end while
        $s = s + \delta_s$;
    end while
    $\kappa = \kappa + \delta_\kappa$;
end while
```

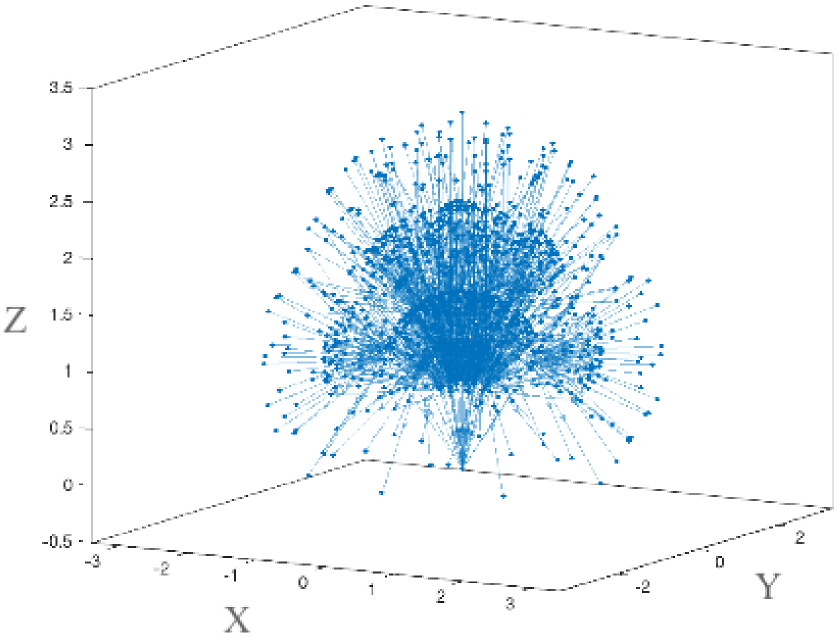


Figure 4.3: A visualization of the motion primitives for a 2 sections continuum robot.

4.1.2 Weighted A* search

With the state-lattice structure of the search scope, we ensure that each node searched is actually reachable. Connecting each node is a feasible configuration.

Any searching algorithm can be used to connect each node, here we use A* algorithm to search. Because of its heuristic nature, the search path can be oriented towards the target in the general direction, and it is easy to add parameters to adjust the priority of the path selection.

Since our goal is to find feasible configurations quickly, global optimization is not a concern at this stage, so we choose Weighted A* for the calculation, with $\epsilon > 1$, so the weighting of the paths will be more towards the end point, and less towards the end point. The Weighted A* is used to calculate the $\epsilon > 1$, so the path selection will be weighted more towards the endpoint and less towards the cost function.

We use a greedy way to adjust the A* algorithm, that is, we will directly select the local minima as the best option when selecting nodes in each section, and then proceed to the selection of the next section. This approach ensures a solution in $\mathcal{O}(n)$, but it is not necessarily global optimization.

4.1.3 Cost function

Although heuristics help us exploit the shortest path from current tip to desired end, for continuum robots we focus on the configuration profile, as introduced above, which is highly correlated to the function $g(n)$. According to our previous work [33], to achieve smooth trajectory control, the IK solver of a continuum robot can be treated as an optimization problem. Our robot manipulator consists of a tendon-driven actuator. Each module has three cable-connected motors which are used to drive them to the desired state. When performing IK, we can use curvature, bending direction and arc length as intermediate variables to calculate the desired cable lengths. These lengths can be sent to the DC motors on each module to drive the module to the desired configuration. These equations are as follows:

$$l_1 = 2\sin\left(\frac{\kappa s}{2}\right)\left(\frac{1}{\kappa} - d\sin\phi\right) \tag{4.1}$$

$$l_2 = 2\sin\left(\frac{\kappa s}{2}\right)\left(\frac{1}{\kappa} + d\sin\left(\frac{\pi}{3} + \phi\right)\right) \tag{4.2}$$

$$l_3 = 2\sin\left(\frac{\kappa s}{2}\right)\left(\frac{1}{\kappa} - d\cos\left(\frac{\pi}{6} + \phi\right)\right), \tag{4.3}$$

where l_1, l_2 and l_3 are the lengths of cables 1, 2 and 3, κ,s,ϕ represent curvature, arc length and bending direction respectively, d is the original distance from the top center of a module to the cable attachment point. Still, the deviation of equations above is stated in our prior work [33]. Notice that in cable length domain, a straight line is the shortest distance from initial to target. Thus, we can derive the minimized cable length change cost function:

$$g_{cable}(n) = \sum_{i=1}^{n}[(l_{i1} - l_{i1p})^2 + (l_{i2} - l_{i2p})^2 + (l_{i3} - l_{i3p})^2] \tag{4.4}$$

where l_{i1}, l_{i2}, l_{i3} are cable lengths for the current state of each module and $l_{i1p}, l_{i2p}, l_{i3p}$ are for previous state.

In addition, the arc length (Minimum Arc-length Cost) and curvature for each module (Minimum Curvature Cost) can also be effective cost functions, as discussed in our previous work [8]. The equations are shown below, and we will evaluate their performance with minimum cable change cost in the experiment section.

$$g_{arc}(n) = \sum_{i=1}^{n} s_i, \quad g_{cur}(n) = \sum_{i=1}^{n} \kappa_i \tag{4.5}$$

Although obstacles are rare in our proposed application, obstacle avoidance is still a useful feature for an inverse kinematics solver. The state lattice-based approach makes planning around obstacles a simple matter of incorporating the proximity to or overlap with an obstacle into the cost function. We also calculate and add a "distance to obstacle" d_{obs} term to the total cost function $f(n) = g(n)+h(n)+d_{obs}(n)$. We introduce a parameter clear radius(r_{clr}) to define a safety distance from the robot to obstacles.

$$d_{obs}(n) = k(r_{clr} - dist(Pos_{obs}, Pos_n)) \tag{4.6}$$

where k is a large positive value as a coefficient, Pos_{obs} is the position of the obstacle, and Pos_n) is the state of node n, which is the tip position of the nth potential configuration. If d_{obs} is zero or negative, the obstacle is out of the clear radius and won't obstruct the robot. In that case, we set $d_{obs}(n) = 0$, and the cost function is identical to the function without obstacle avoidance.

4.1.4 Combine 2 methods

The purpose of state lattice-based search is to quickly approach the desired goal with respect to its precise position and orientation. To combine the two algorithms and to match the characteristics of state lattice-based search, we set the goal area so that if the

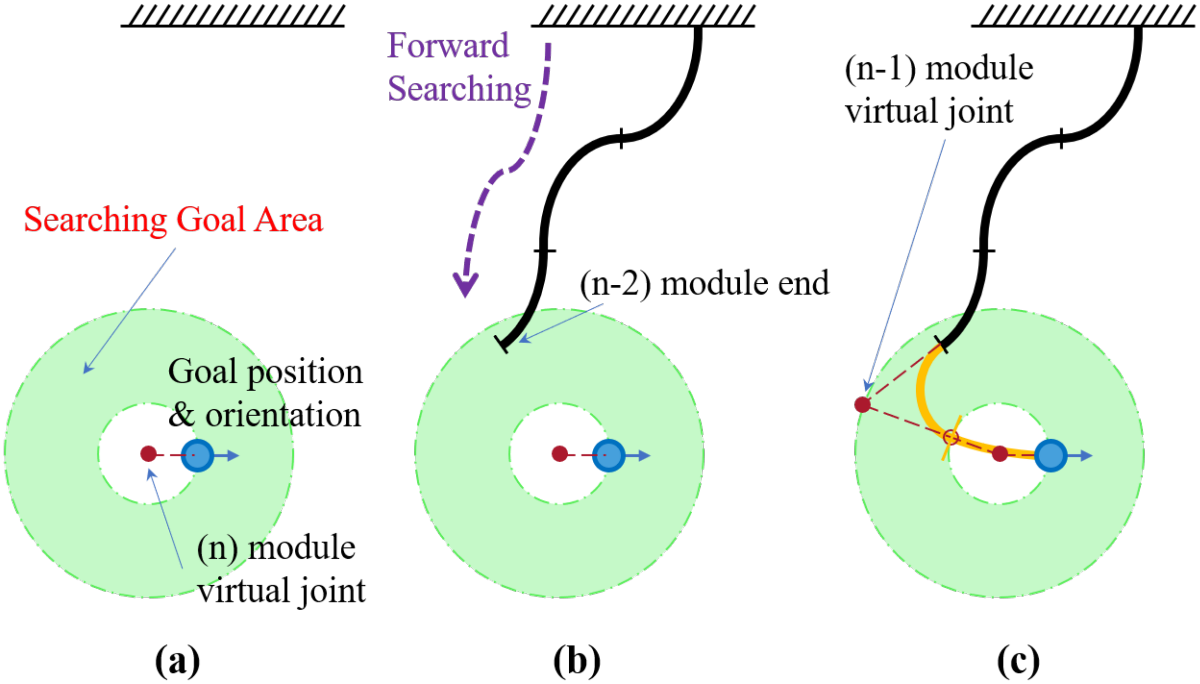


Figure 4.4: (a) The concept of Goal Area. (b) The search phase. (c) The fine-tuning phase.

search part successfully enters the goal area, it can be connected to the goal position in proper orientation regardless of its orientation.

As for the fine-tuning algorithm, we first try to use exFABRIKc, the detailed computation of which has been mentioned in the previous chapters, and its algorithmic properties make the orientation accuracy guaranteed. The computation is very fast when there are only 2 sections. However, its disadvantage is that we cannot control the bending angle. We would like to have more controllable parts, similar to the cost function in the search section. Therefore, we have developed the Geometry Analytical IK Solver to replace the fine-tuning of exFABRIKc. The details of the algorithm will be explained in a later section.

1. Goal Area

   Given the goal position and orientation, we take a length of the virtual link as the virtual link to the goal, and since we already know the orientation, we can get the position of the virtual joint from this side. Using this position as the center of the circle and the virtual link length as the radius, we can get the inner circle, and using twice the virtual link length as the radius, we can get the outer circle. As long as any point in the ring between the inner and outer circles is at a distance of d from the center of the circle, $l \leq d \leq 2l$ is satisfied. The goal area concept is shown in Fig. 4.4(a).

2. Fine-Tuning

   Make sure that the search section reaches the goal area and then use exFABRIKc or Geometry Analytical IK solver to solve the last 2 configurations. The concept is shown in Fig. 4.4(c).

4.1.5 exFABRIKc

exFABRIKc is an extension of the FABRIKc algorithm [10] that we have developed as a geometry-based heuristic iterative inverse kinematics solver. While FABRIKc effectively addresses the inverse kinematics problem in multisection continuum robots, it lacks the capability to handle extensible continuum modules. Therefore, we have extended this inverse kinematics algorithm to be applicable to our OrCA system. In this section, we will first explain the operation of FABRIKc, followed by an elucidation of its extension, namely exFABRIKc.

FABRIKc

Zhang's approach [10] is to add constraints to the configuration space so that the linkage conforms to the constant curvature assumption, and then use the FABRIK method to derive the attitude of the continuous module. Because of the characteristics of FABRIK, the EEs can have accurate position and orientation.

This method loses the control of one degree of freedom, and we cannot precisely control the rotation along the z-axis in the IK calculation. This is not a big problem because the grippers usually can cope with various orientations, or we usually put a revolute joint in the EE with this rotation direction. Moreover, in FABRIKc the settings are based on constant arc length modules, which are not suitable for extensible continuum robots. We introduce exFABRIKc to solve this problem. The details will be mentioned in the next section.

The FABRIKc method is divided into two parts: forward and backward. The steps are shown in Fig. 4.5, where (a)-(f) are in the forward phase, (g)-(i) are in the backward phase.

The forward phase moves the tip to the desired position and orientation, and then points the bottom half of the virtual link to the virtual joint of the (n-1) module. The forward part moves the tip to the desired position and orientation, and then points the lower half of the virtual link to the virtual joint of (n-1) module. Since each module has a fixed arc length, we can find the virtual link length from the bending angle. We can then get the position and orientation of the base of module (n). We can get the position and orientation of the base of module (n). Repeat this for the previous module and find the bending angle of the virtual joint. This is because there is no more virtual joint in the bottom section, and the pose of the first module has to be maintained. This completes the forward part of the process.

The backward phase is simpler, just shift the base to the correct position. After the forward step, the position of the first section will be changed, and in the backward step, all the modules will be shifted to the initial position, and this step will be completed.

After iteration, the EE can be approximated gradually. To summarize, Forward sets the bending angle of each section and ensures that the last section can be turned back to the base direction after the final link, while Backward corrects the offset position after many bends. Repeating this will minimize errors.

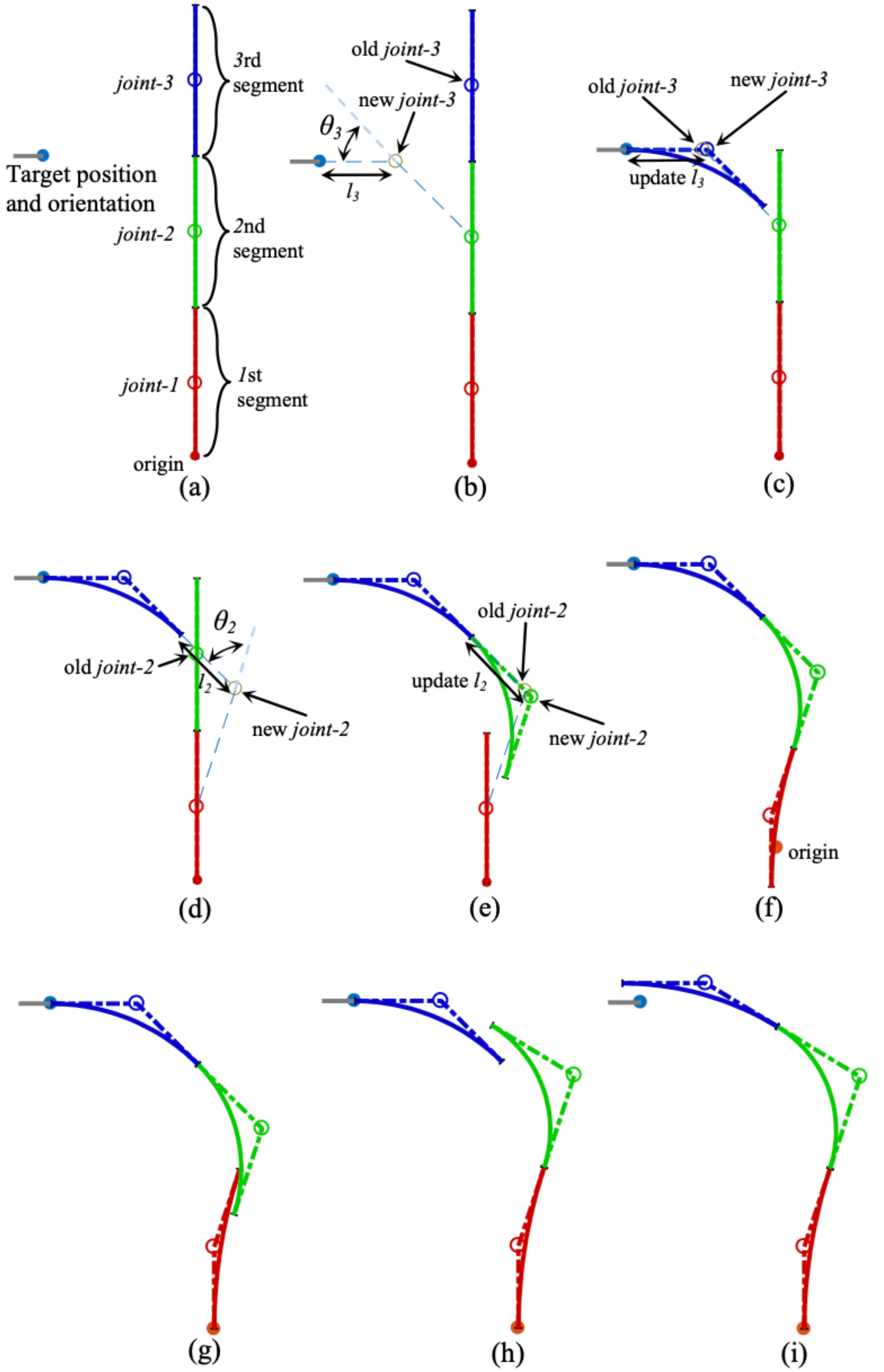


Figure 4.5: An example of the two-phase iteration of the proposed FABRIKc algorithm on a 3-segment continuum robot [10]

exFABRIKc

With our Origami Continuum Arm, it is necessary to handle the extensible arc length. Therefore, we modified the Forward part, when connecting virtual joints, we not only decide the direction, but also adjust the length of the virtual link according to the points of the two virtual joints. Since the virtual links on both sides of the virtual joint in each section have to be of equal length (to comply with the constant curvature assumption), this change will also affect the position of the virtual joint.

In practice, even though it is an extensible arc length module, there is still a range of movement, so here we can calculate both max and min virtual links and keep them within the range. The backward part is the same and does not need to be changed. This will minimize errors in the iteration and take advantage of extensible.

4.1.6 Geometry Analytical IK Solver

Taking a target direction and two virtual joints, a triangle can be formed based on the law of cosine and the constant curvature assumption, given a virtual link length to determine the triangle.

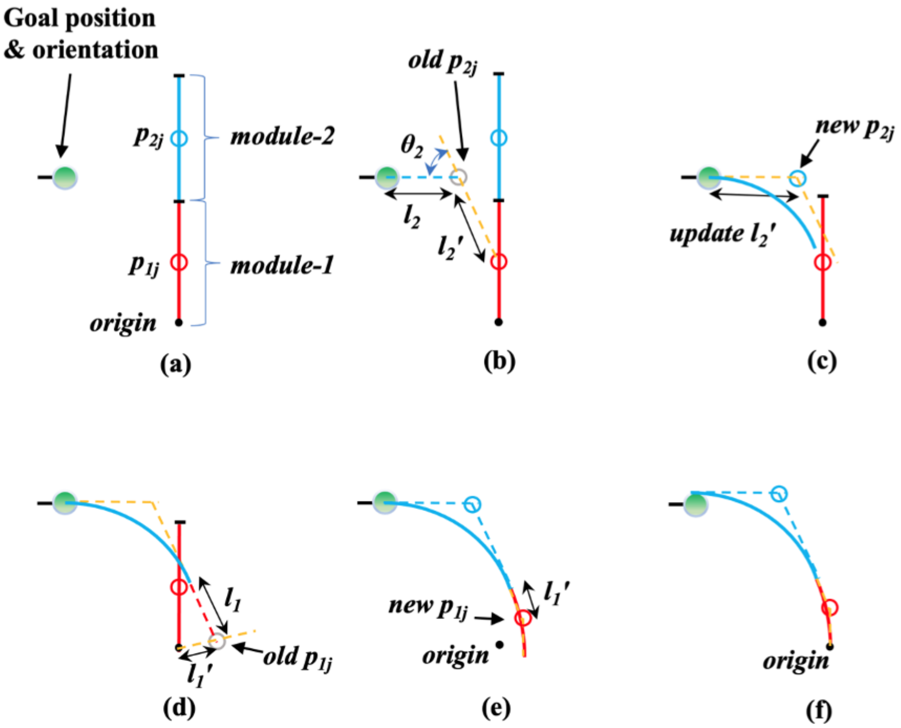


Figure 4.6: A visualization of the exFABRIKc on a 2-segment continuum robot. (a) The initial and the target poses, where p_{1j} and p_{2j} are the virtual joints for module 1 and 2. (b) Forward reaching phase: Update module-2 to satisfy the desired position and orientation. The upper virtual link 2 rotates and translates to reach the goal pose. The p_{2j} follows the link. The lower virtual link 2 rotates toward the virtual joint 1, and adjusts the length to connect virtual joint 2 and 1, where is the extending step that does not happen in FABRIKc. (c) Update the length of upper visual link 2 with l_2. Due to the length changing, translate the p_{2j} and lower virtual link 2. Module-2 is done with forward reaching phase. (d) Update upper virtual link 1 to connect module-2. The two virtual links should be in an identical orientation. Translate p_{1j} along with the virtual link. Define the distance between the p_{1j} and the origin as $l1$. (e) Update the module-1 with l_1 and have new p_{1j}. Because module-1 is the module that connects to the base, the orientation of the lower virtual link is always identical to the base orientation. Module-1 is done with forward reaching phase. (f) Backward reaching phase: Translate the whole robot to the origin. When it is done, the end-effect should be in a proper orientation with little position errors. Then, repeat from (b) to minimize the error.

Algorithm 2: exFABRIKc

```
Result: List of $p_e, p_j, p_b$ for each module
while $e > \varepsilon$ and $k < k_{max}$ do
    /* Forward reaching */
    $p_e \leftarrow p^*$ ;
    $\hat{z}_e \leftarrow \hat{z}^*$ ;
    $t \leftarrow 1$ ;
    while $t \leq n$ do
        /* translate and rotate the end of $t$th module */
        $p_{te} = p_e$ ;
        $\hat{z}_{te} = \hat{z}_e$ ;
        // get old virtual joint position and base orientation
        $p_{tj} = p_{te} - l_t \cdot \hat{z}_{te}$ ;
        $\hat{z}_{tb} = (p_{tj} - p_{(t+1)j})/||p_{tj} - p_{(t+1)j}||$ ;
        // adjust the length of virtual link
        $l_t = ||p_{tj} - p_{(t+1)j}||$ ;
        // update virtual joint and base of $t$th module
        $p_{tj} = p_{te} - l_t \cdot \hat{z}_{te}$ ;
        $p_{tb} = p_{tj} - l_t \cdot \hat{z}_{tb}$ ;
        // set the pose for the next module
        $p_e = p_{tb}$ ;
        $\hat{z}_e = \hat{z}_{tb}$ ;
        $t = t + 1$ ;
    end while
    /* Backward reaching */
    $p_b = p_{base}$;
    while $t \geq 1$ do
        $\nu = p_b - p_{tb}$ ;
        // translate $t$th module
        $p_{tb} = p_{tb} + \nu$ ;
        $p_{tj} = p_{tj} + \nu$ ;
        $p_{te} = p_{te} + \nu$ ;
        // set the base position of the next module
        $p_b = p_{te}$ ;
        $t = t - 1$ ;
    end while
    $e = ||p^* - p_e||$ ;
    $k = k + 1$ ;
end while
```

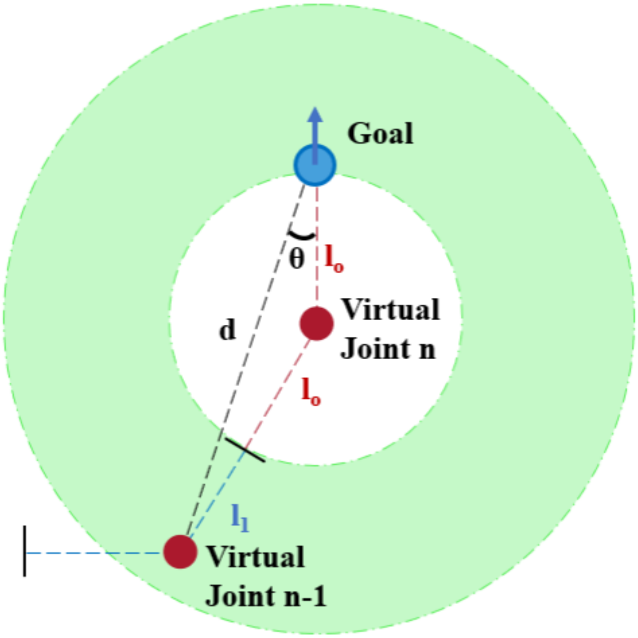


Figure 4.7: A close-up view of the virtual links used in the analytical solution for the SLInKi algorithm. We can use the law of cosines to calculate l_o, allowing us to determine the exact arc parameters (s, κ, and ϕ) of the final two links required to reach the goal position and orientation.

From the law of cosines, we have:

$$(l_1 + l_0)^2 = l_0^2 + d^2 - 2l_0 d\cos(\theta), \tag{4.7}$$

where l_0 is the unknown virtual link length of the module n, d is the distance between the goal and the virtual joint of the module $n-1$, and l_1 is the length of the virtual link of the module $n-1$. l_1 is known and d, allows us to solve for l_0 as follows:

$$l_0 = \frac{d^2 - l_1^2}{2(l_1 + d\cos(\theta))} \tag{4.8}$$

Using the virtual link length of the final module, we can calculate the bending angles and continuum lengths of the final 2 modules required to reach the desired position and orientation.

This method can be used to connect any two poses in space using two constant curvature curves. This method is fast and accurate, given a virtual link length. This method is fast and accurate, given a virtual link length. It can be used in place of exFABRIKc.

4.2 Algorithm Limitations

State lattice-based search algorithms can guarantee the feasibility of the solutions at the expense of strict completeness. If the nodes in the lattice are not dense enough, we may not find the solution even if it exists. To avoid this, we introduce snap-width variable and searching goal area. The snap-width variable controls the density of nodes in the lattice, and the searching goal area should be large enough to cover at least one node.

We employ a combination of the weighted $A^\star$ algorithm and the greedy algorithm simultaneously. It can decrease computational cost but might not find global optimal path. Our approach can find a feasible IK solution for continuum robots with low computational cost.

The exFABRIKc is used to refine the end effector's pose. It is fast and precise; however, it has singularities while the kinematic chain is straight and the target lies on the line. This situation is discussed in [34]. The searching goal area can decrease the likelihood of the singularities happening because the searching target is the virtual joint, which makes the last two sections configuration is usually not a straight line. We also set a maximum iteration value, even exFABRIKc can not converge, SLInKi still generate a solution.

We replace exFABRIKc with analytic solution because not only is it faster but also no singularities issue. The limitation of this geometry based IK solver is it is not optimal. The analytic IK solver needs a virtual link length to find the other virtual link length. Different initial guesses will cause different results. Now we are using the state lattice based searching results as the initial guess. Sometimes the solutions have large curvature or arc length.

4.3 Experiments and Discussions

In this section, we conduct several experiments to verify the capabilities of our IK algorithm - SLInKi, as well as assessing multiple aspects of its properties. First, we adopt minimum cable length change as the cost function in the SLInKi program, as it is determined having the best performance in the previous work [8]. We compare the overall performance among four different IK solvers in simulation. Simulations were performed in Matlab 2020b on an Apple MacBook Pro with a 2.3 GHz Quad-Core i5 with 8ăGB Ram.

We set the values of parameters which are needed in SLInKi algorithm based on our origami continuum robot. The maximum arc length (s_{max}) is 80 mm, and the range of orientation angle (ϕ) is 0 to 2π. The curvature (κ) range is 0 to 0.0125 mm^{-1} due to hardware limitation. We set maximum iteration value (k_{max}) as 50 times. In our experiments, SLInKi-exFABRIKc usually reaches the goal pose with less than 10 iterations. The error threshold (ϵ) is $1mm$.

Finally, a trajectory following test for a physical robot is conducted to showcase SLInKi's real-time human-robot collaboration capability. In this test, an origami-inspired modular continuum robot with remarkable flexibility is employed, as depicted in Fig. 4.1. SLInKi is employed to generate inverse kinematics solutions for a prescribed rectangular trajectory, allowing us to evaluate the performance of the physical robot.

4.3.1 Cost Function Comparison: Single Pose

By investigating the difference among the three cost functions, we shall determine the best performer and incorporate it into SLInKi. We aim to assign random target tip position and orientation within the robot's workspace, apply multiple distinctive cost

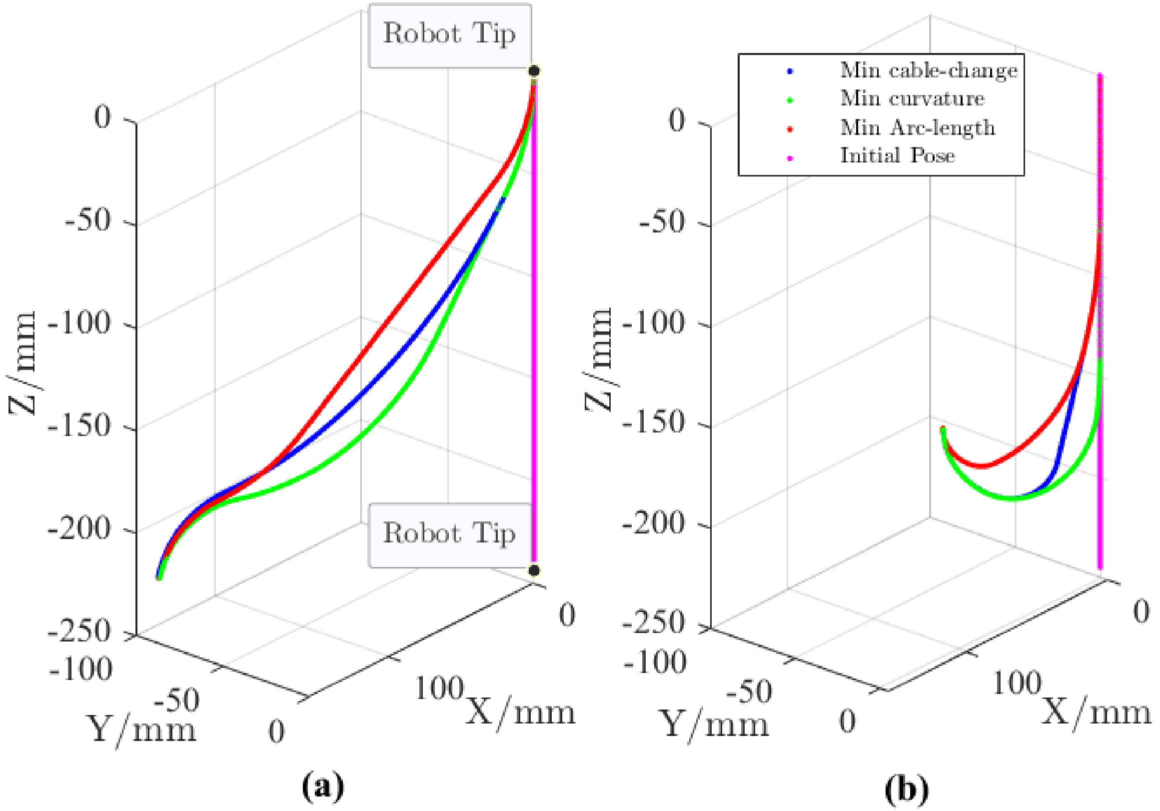


Figure 4.8: Experimental results for comparing single pose performance of multiple cost functions. The magenta line is the continuum robot in the home position. Other curves are the final configurations from SLInKi with different cost-functions. The red, green, and blue curves are respectively optimized for minimum arc length, minimum curvature, and minimum cable length change. All three configurations lead the end-effector to the desired pose. The red curve tends to move toward the goal in the shortest path. The green curve leans to stay straight and bends only when it is essential. The blue curve is similar to the green one because the configuration moves from the home position, where all modules are straight. Detailed arc-length and curvature data for each robot module in designated optimal cost function settings are presented in tables below, where (a) corresponds to Table 4.2 and (b) corresponds to Table 4.3.

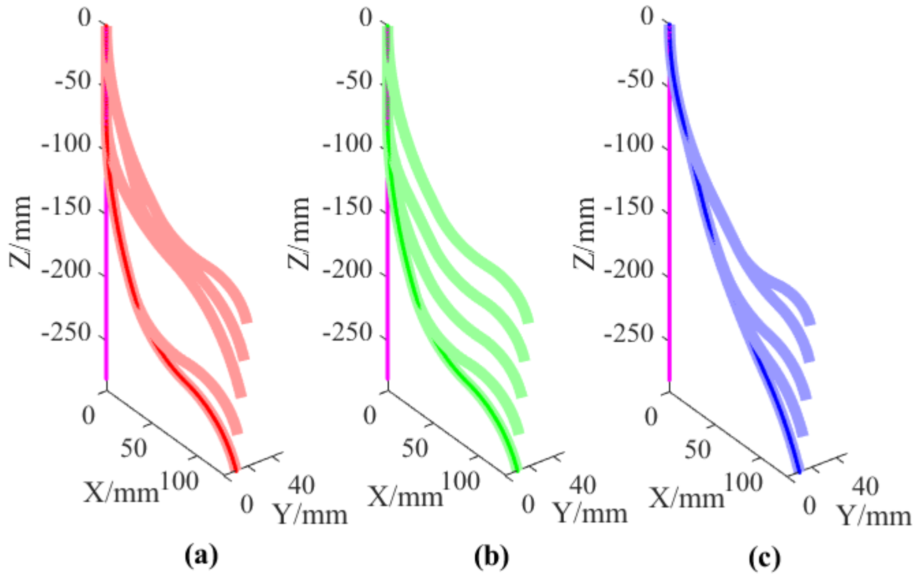


Figure 4.9: A demonstration of the effect of changes in the soft function of the state-lattice search sequence of SLInKi. The task is to sweep through a sequence of poses from (50,20,-230) at orientation (0.5547,0,-0.8321) to (90,-40,-222) at orientation (-0.2250,0,-0.9744) in 4 steps. (a) is optimization for arc length. (b) is optimization for minimum curvature. (c) is optimization for minimum cable length change. The configurations tend to stay in the previous status and only move when it needs to. This shows that (a) and (b) are less ideal trajectories because some modules have to move back and forth frequently during the task.

functions to SLInKi and observe the features. The three cost functions are as follows:

1. Minimize Arc Length

   Shorter arc-length maximizes the stiffness of our origami structure, making it resilient to external forces

2. Minimize Curvature

   Less curvature means less cable tension and less stress on the manipulator, resulting in greater working life

3. Minimize Change in Cable-length

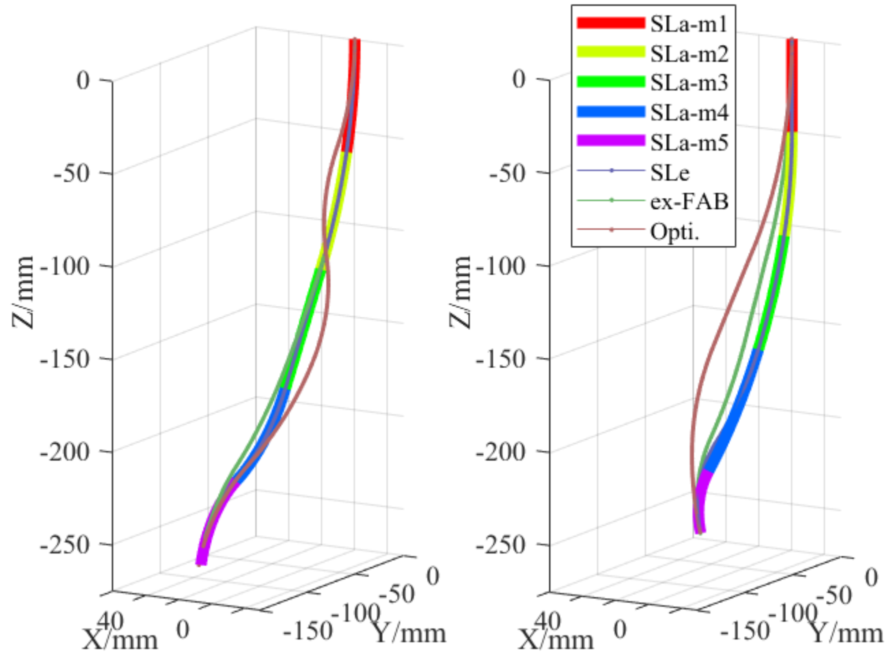
   smaller changes in cable length means smoother robot motion and less power consumption when transitioning between poses

Equations for each of these cost functions can be found in the previous section.

In Fig. 4.8, we show range configurations that achieve the same target, each calculated using a different cost functions. Meanwhile, to certificate that minimum cable change cost being most appropriate for our system, another multi-pose test needs to be conducted to let the robot follow a designated trajectory.

4.3.2 Cost Function Comparison: Multiple Poses

During operation, a manipulator transitions between points. To better mimic this, created a linear trajectory and compared the behavior of the different cost functions in smoothly following it. As shown in Fig. 4.9, the minimum cable change function involves the most fluent shape changing, or to say, the robot uses minimum overall motion to switch its poses. Quantitatively, we found that the average change in cable length was 4.9 mm for minimum cable length, 14.2 mm for minimum curvature, and 9.8 mm for minimum arc length functions (with maximums recorded as 29, 89, and 45 mm respectively.



(a) **(b)**

Figure 4.10: Results of the algorithms' performance experiments. The green curve is the IK solution from exFABRIKc. The brown curve is from the Jacobian method based IK solver optimized for minimum change. The gray curve is SLInKi-exFABRIKc. The multicolor curve is the result by SLInKi-analytic optimized for minimum change, with each color representing a manipulator module. These figures only show the configurations, while the details of time usage and errors are shown in the tables below. (a) corresponds to Table 4.4. (b) corresponds to Table 4.5. Notice that errors measured in these two charts are calculated from vector norms of either positions or orientations.

While minimizing curvature and arc-length may be useful in other circumstances, we will focus on smoothness of motion in this paper and use a cost function that minimizes change in cable length for the rest of this article.

4.3.3 Performance Evaluation of SLInKi

The next step is to compare our algorithm with existing IK solvers and evaluate the results. To achieve this, multiple single pose experiments are conducted. In each experiment, we randomly set a desired end-effector position and orientation, then used four different IK algorithms: exFABRIKc, numerical method based minimum cable optimized IK solver [8], SLInKi-exFABRIKc and SLInKi-analytic to create joint configuration profiles. The overall robot shape generated from these profiles are compared, thus the algorithm performances can be assessed. Experiment results, including robot shapes and intermediate virtual joints produced from separate algorithms are shown in Fig. 4.10, while the overall calculation time, actual end-effector position & orientation values and errors between desired ones are presented in Tables 4.4-4.5.

We found that all of the tested algorithms reached the goal with acceptable errors. The optimization based IK solver generates the smoothest motions, but is significantly slower than all the others and is too slow for real-time usage. Both SLInKi with exFABRIKc and SLInKi with the analytical solution are faster than pure exFABRIKc. The analytical solution runs much faster than using the 2-module exFABRIKc, while the state-lattice search takes a similar amount of time for each case.

4.3.4 Customizable Evaluation of SLInKi

In addition to the low computational cost, the state-lattice approach is also highly customizable. We designed the SLInKi with changeable arc length continuum modules in mind. For creating feasible configurations, the user needs to set the maximum arc length (s_{max}) for motion primitives generation (Algorithm 1). It is also easy to modify the num-

Table 4.2: Performance Comparison for Cost Functions, Single Pose, Goal Position: (150;-90;-220) & Orientation: (0.5661;0.2265;-0.7926)

		min-arc len	min-curv	min-cable change
arc-length (mm)	1	56.00	60.67	60.67
	2	70.00	70.00	65.33
	3	70.00	60.67	65.33
	4	48.71	58.04	54.69
	5	45.57	52.98	49.51
curvature (10^{-3}/mm)	1	12.50	7.692	7.692
	2	0.000	0.000	3.571
	3	0.000	7.692	3.571
	4	9.470	8.750	6.141
	5	21.85	22.34	20.41

Table 4.3: Performance Comparison for Cost Functions, Single Pose, Goal Position: (40;-60;-180) & Orientation: (0;0;1)

		min-arc len	min-curv	min-cable change
arc-length (mm)	1	60.67	65.33	65.33
	2	70.00	70.00	65.33
	3	46.67	51.33	56.00
	4	35.47	47.99	43.40
	5	30.30	52.04	48.92
curvature (10^{-3}/mm)	1	7.692	3.571	3.571
	2	0.000	0.000	3.571
	3	18.03	12.03	7.217
	4	25.69	27.25	34.41
	5	51.97	20.68	24.49

Table 4.4: Algorithm Comparison 1, Single Pose, Goal Position: (-146.15,9.32,-255.65) & Orientation: (-0.391,-0.018,-0.920)

	Opti. based IK	exFABRIKc	SLInKi1	SLInKi2
time(ms)	3628.6	31.8	14.0(9.5+4.5)	11.6(9.1+2.5)
position(mm)	(-146.15,9.32,-255.65)	(-146.15,9.32,-255.65)	(-146.15,9.32,-255.65)	(-146.15,9.32,-255.65)
pos-err(mm)	0	0	0	0
orientation	(-0.391,-0.018,-0.920)	(-0.391,-0.018,-0.920)	(-0.391,-0.018,-0.920)	(-0.391,-0.018,-0.920)
ori-err	0	0	0	0

Table 4.5: Algorithm Comparison 2, Single Pose, Goal Position: (-4.86,52.63,-270.79) & Orientation: (0.191,-0.117,-0.975)

	Opti. based IK	exFABRIKc	SLInKi1	SLInKi2
time(ms)	2490.3	45.8	28.9(19.4+8.5)	16.9(12.7+4.2)
position(mm)	(-4.86,52.63,-270.79)	(-4.86,52.66,-270.58)	(-4.86,52.63,-270.79)	(-4.86,52.63,-270.79)
pos-err(mm)	0	0.220	0	0
orientation	(0.192,-0.117,-0.975)	(0.192,-0.117,-0.975)	(0.192,-0.117,-0.975)	(0.192,-0.117,-0.975)
ori-err	0	0	0	0

ber of sections for different length continuum robots. It only affects the node numbers in the state lattice based search and has minimal effect on computation time. We tested SLInKi with different size robots (5, 7, and 9 segments) to reach the same position and orientation.

The other customizable aspect is the cost function. In previous section, we discussed different cost function for different purposes. Since the cost function is modulized. We can set different cost function on each module independently. For example, we could use min-arc-length on the middle two modules to limit the torque exerted on them, and min-cable-change on the first modules to ensure easier transitions between states.

Table 4.6: SLInKi Segment Comparison, Single Pose, Goal Position: (-100,-10,-350) & Orientation: (-0.377,-0.270,-0.886)

	5 segments	7 segments	9 segments
time(ms)	13.3(10.9+2.4)	28.3(25.8+2.5)	29.7(27.0+2.1)
position(mm)	(-100.0,-10.0,-350.0)	(-100.0,-10.0,-350.0)	(-100.0,-10.0,-350.0)
pos-err(mm)	0	0	0
orientation	(-0.377,-0.270,-0.886)	(-0.377,-0.270,-0.886)	(-0.377,-0.270,-0.886)
ori-err	0	0	0

4.3.5 Tip-first SLInKi

One goal when using SLInKi for trajectory following is ensuring that the cable-lengths calculated result in smooth motion on the physical robot. This was the reason we investigated minimum cable length change as a cost function. SLInKi does not use the same search method for all areas of the workspace, meaning that there may be

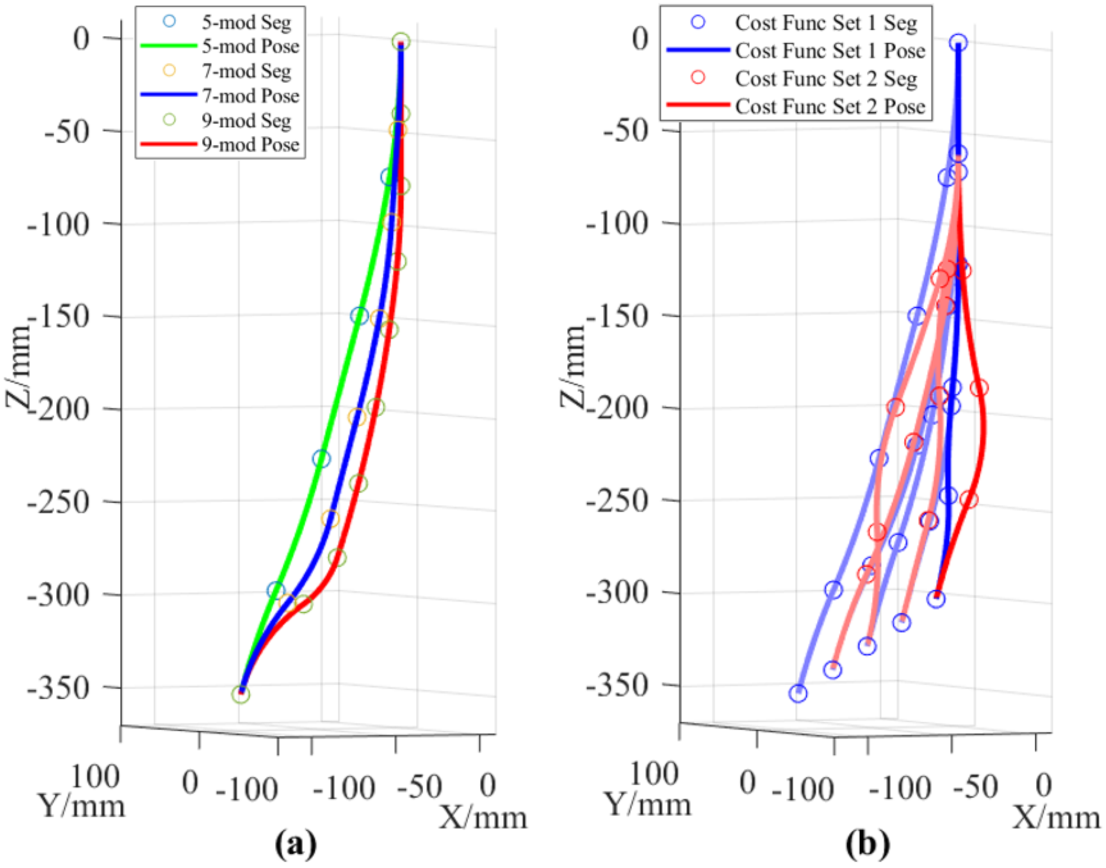


Figure 4.11: Examples that demonstrate the customizability and flexibility of SLInKi. (a) exhibits that our algorithm can deal with IK for different input sizes. In this scenario, given the same goal position and orientation, we prove that SLInKi is eligible to solve the IK for 5, 7, and 9-segment separately, within a reasonable timescale. Time consumption and error performance are listed in Table 4.6. (b) illustrates that the cost function selection for SLInKi can be quite flexible, remarking on the modularity of our algorithm design. Note that in this experiment, the robot tip follows a line trajectory interpolated by 5 waypoints. Two trials are implemented, each of which with different sets of cost functions applied. Cost function set 1, plotted in blue, involves minimum arc length for the first two segments and minimum cable change for the rest three. Meanwhile, set 2, plotted in red, applies all five segments with minimum cable change. As the figure shows, customized cost function sets lead to absolutely distinct configuration profiles, which indicates that SLInKi is highly flexible and customizable according to task requirements.

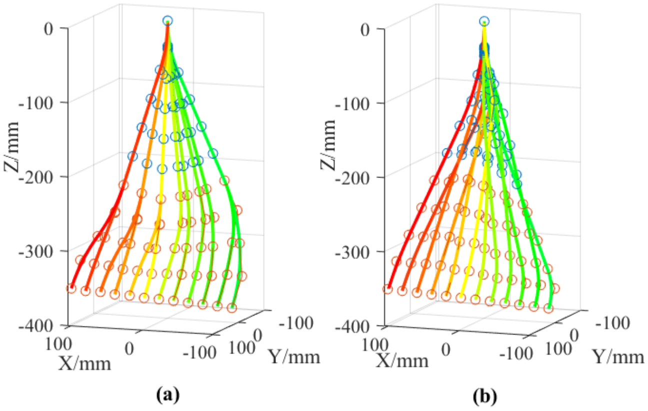
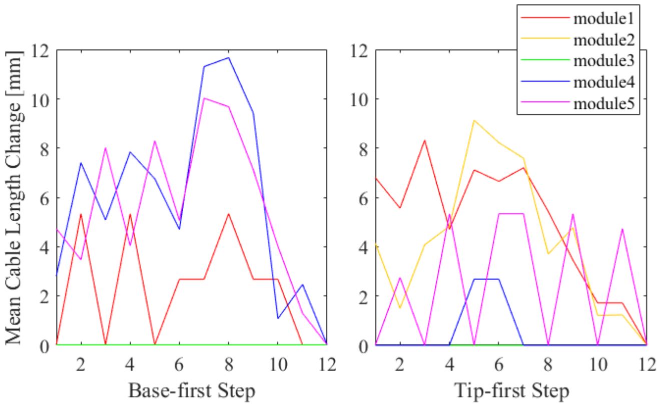


Figure 4.12: A comparison of the IK results from base-first (a) and tip-first (b) SLInKi when following a straight line. SLInKi uses different techniques to search through different parts of the workspace, using, for this graph, exFABRIKc near the goal and using state-lattice for the rest. Thus, switching whether or not we consider the robot tip or the robot base to be the goal can affect the path the manipulator will take to reach the desired state. We can see that the tip-first SLInKi results in a smoother trajectory. More detail can be found in Fig. 4.13.

discontinuities that weighting struggles to solve. To investigate this, we ran SLInKi in the opposite direction: using the goal position and orientation as the starting point and the base of the robot as the goal. We call this Tip-first SLInKi, and compared it to the original direction (so-called Base-first SLInKi) in Fig. 4.12 to see if changing the search direction effected the change in cable length. We found that tip-first SLInKi followed the trajectory by moving the base joint continuously, while the tip of the base-first SLInKi kept having to bend back and forth to reach the goal. This behavior resulted on average a 13% lower change in cable lengths for the tip-first SLInKi compared to the base-First SLInKi (2.39 mm/step vs 2.71 mm/step. A full graph of the change in cable lengths of each module can be seen in Fig. 4.13.

4.3.6 Obstacle Avoidance

To highlight the adaptability and customizability of the state-lattice, we performed experiments of IK in an environment containing an obstacle. A demonstration of this can



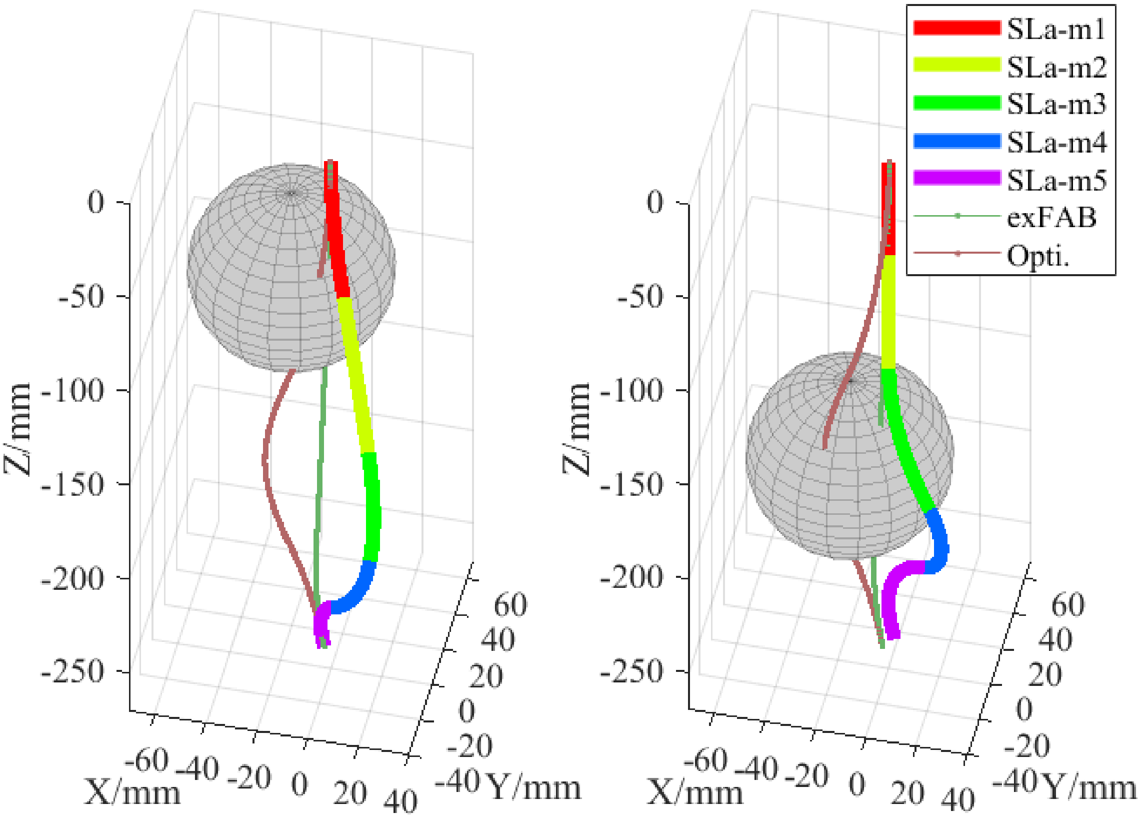
(a) **(b)**

Figure 4.13: A plot of the changes in cable length for the states, shown in Fig. 4.12 (cable lengths are averaged between the three cables of each module). (a) shows the behavior of base-first SLInKi, while (b) shows the behavior of tip-first SLInKi. Tip-first SLInKi results in a smother progression, with an average change in cable length of 2.38 mm/step vs 2.72 mm/step for base-first.

be found in Fig. 4.14, where we show SLInKi avoiding an obstacle to reach a goal. In these simulation experiments, we set $r_{clr} = 150\ mm$ and $k = 100$ in obstacle avoidance cost function Equations 4.6. We also performed a demonstration on the physical robot, where the robot attempted to stay at a goal location while avoiding a movable obstacle in real time, which can be seen in the attached video (example images shown in Fig. 4.15). In contrast, neither pure exFABRIKc nor our optimization-based IK solver can take into account obstacles.

4.3.7 Physical Robot Path Following Demonstration

To validate the multi-poses performance of SLInKi, we conduct an experiment comparing the simulation and real robot performance. This validation was performed using an OptiTrack® Motion Capture System to record the final states of the robot. As shown in Fig. 4.16, the real robot configurations shown at the four corners correspond respectively to the simulation results of the four corners of the rectangular trajectory. The experiment shows that the robot is able to follow the trajectory in real time within an acceptable



(a) **(b)**

Figure 4.14: Two examples of IK with obstacle avoidance using the SLInKi algorithm, a feature incompatible with pure exFABRIKc or traditional optimization approaches.

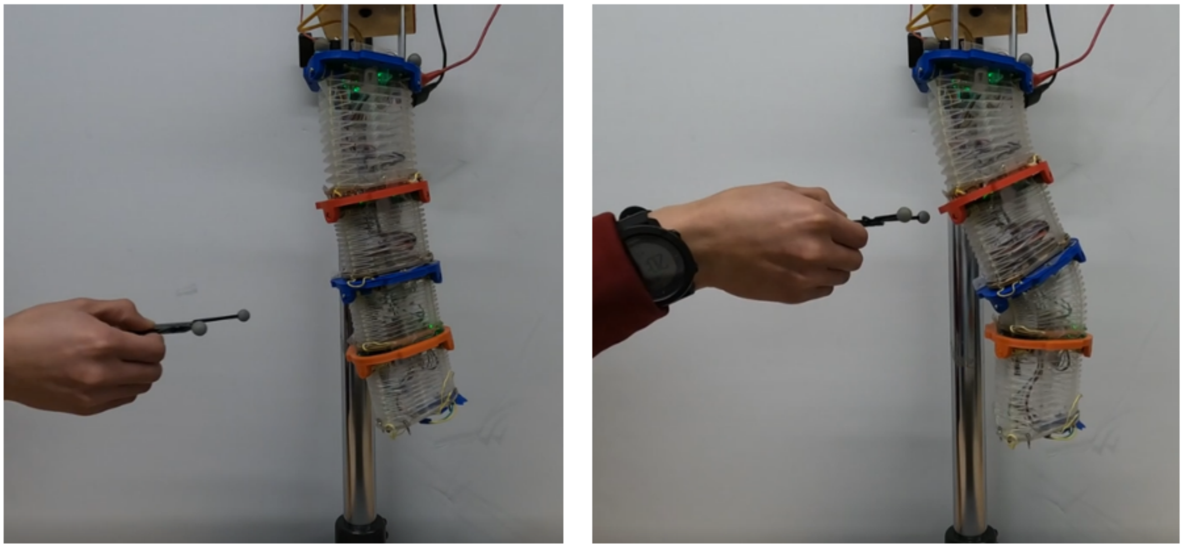


Figure 4.15: Two images from our real-time obstacle avoidance demonstration on the physical manipulator. The robot attempts to maintain the same end-effector position while avoiding the human-held obstacle. A video of this experiment can be seen in our supplementary files.

error performance, which certificates the real-time capability of SLInKi.

4.3.8 Real-time IK Demonstration

For a final validation of the SLInKi, we demonstrated its ability to follow user-supplied poses in real time (Fig. 4.17. The experimenter moved a target capable of being tracked by the motion capture system around the manipulator's workspace. Using the OptiTrack® system, the control computer calculated the position and orientation of this target and used it as the goal state of the manipulator (with a vertical offset of 4 cm to prevent collisions). The control computer then commanded the manipulator to reach the desired goal using the output of the SLInKi algorithm. Once the goal was reached, the process was repeated. A video of this real-time position tracking can be found in the supplemental files, where we demonstrated both Base-first and Tip-first SLInKi.

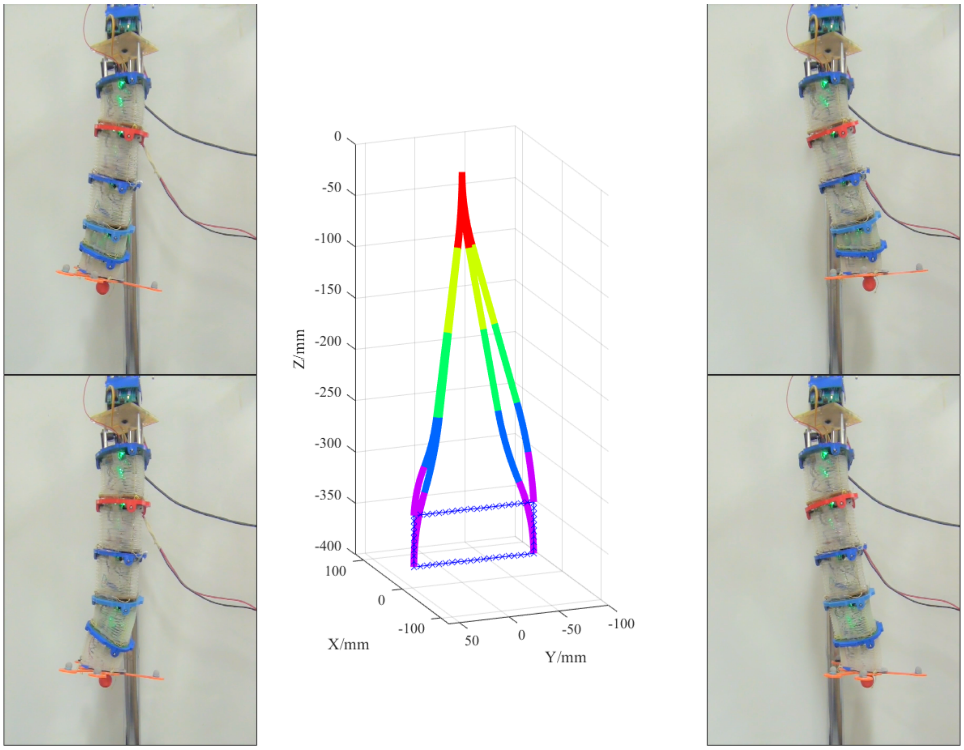
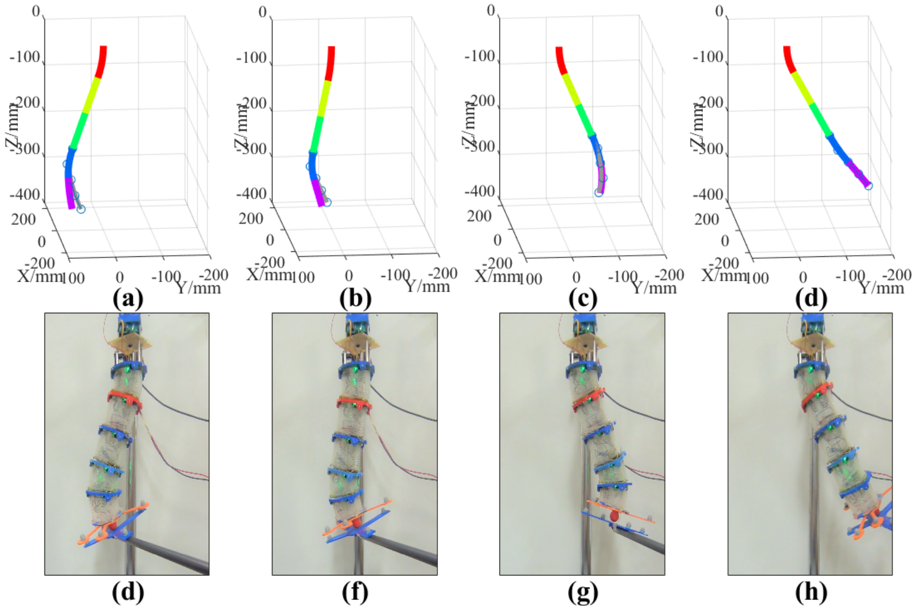


Figure 4.16: Rectangular trajectory following demonstration for both simulation and real robot trials. The simulation trial, as shown in the middle, documents what the shapes of our 5-module robot look like in each corner of a rectangular path, as well as the way points following top-left, top-right, bottom-right, bottom-left, top-left sequence. Given target waypoints, the robot configurations can be calculated by SLInKi in real time. The real robot shapes shown in the figure respectively correspond to the configurations at each corner of the rectangle.



(a) (b) (c) (d)

(d) (f) (g) (h)

Figure 4.17: Here we present four individual pose tracking demonstration trials, to further illustrate the capability of SLInKi. In this experiment, SLInKi solves the IK for random poses and moves the end-effector to reach the target position in the specific orientation. (a)-(d) present the simulation results, where the gray curve lines with skeletons are the solutions by SLInKi without hardware limitations and the colorful curve lines represent the 5-segment manipulator with bending and extending limits. The searching part (upper 3 segments) of SLInKi can embrace the limits and generate reachable poses. The exFABRIKc part (lower 2 segments) doesn't take account of the limits so that the colorful curve lines may not coincide with the gray curve lines. (e)-(h) are the photos of the target and the origami continuum arm in the representative poses corresponding to (a)-(d).

4.4 Summary

In our simulation, we conduct a comparative analysis of various cost functions employed in both single-pose and multi-pose sequential actions. Through this analysis, we are able to observe the impact of these diverse cost functions on the selection of routes within the search section. Furthermore, we undertake a thorough comparison of our proposed algorithm with other existing algorithms. Remarkably, we have discovered that our algorithm demonstrates exceptional performance in terms of both speed and accuracy. Additionally, the Geometry based analytical IK solver exhibits superior efficiency compared to exFABRIKc.

Due to the inherent traits of the search algorithm employed, initial obstacle avoidance is readily achievable. By assigning a higher cost to nodes in close proximity to obstacles, the algorithm effectively prioritizes nodes that are further away from obstacles. Furthermore, simulations and physical experiments involving robotic agents were conducted to validate the effectiveness of this approach.

SLInKi has been proven to effectively and efficiently address the IK problem in multi-section continuum robots. However, certain issues arise due to its discretized path search methodology. Specifically, the intricate actions are delegated predominantly to the last two sections, resulting in the remaining $n-2$ sections transitioning abruptly from one state to another. Moreover, the use of a greedy algorithm for computational expediency compromises the overall smoothness of the pose.

To overcome these limitations, we introduce the AMoRPH (see Sec. 5) approach. With AMoRPH, we aim to rectify these challenges and achieve improved performance.

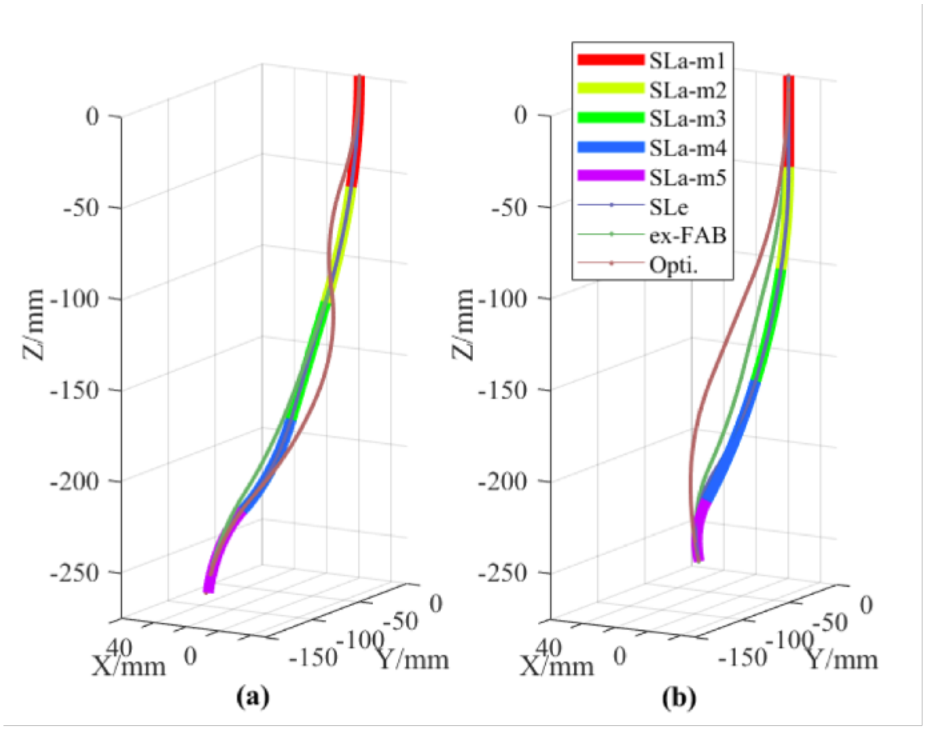


Figure 4.18: Some results of the algorithms' performance experiments. The green curve is the IK solution from exFABRIKc. The brown curve is from the Jacobian method based IK solver optimized for minimum change. The gray curve is SLInKi-exFABRIKc we present in the paper [11]. The multicolor curve is the result by SLInKi-analytic optimized for minimum change, with each color representing a manipulator module. These figures only show the configurations, while the details of time usage and errors are shown in the tables below. (a) corresponds to Table 4.4. (b) corresponds to Table 4.5. Notice that errors measured in these two charts are calculated from vector norms of either positions or orientations.

Chapter 5

AMoRPH

While we wanted to solve the aforementioned SLInKi problem, we found that the Analytical Solution we used in SLInKi could be solved quickly and accurately. We extended this to solve the problem of pose smoothness and motion smoothness.

In light of these extensive predecessors' studies, we have conducted a profound analysis of the inherent nature of the IK problem. As a result, we propose the Analytical Model-based Real-time Posture Heuristic (AMoRPH) IK solver for continuum robots. This innovative approach leverages a virtual joints model based on FABRIKc [10], effectively transforming the IK problem into a purely geometric challenge. By employing a root-finding method, we can identify configurations that exhibit enhanced smoothness.

Building upon these foundations, our research delves into the practical realm through rigorous experimentation. We have implemented and thoroughly evaluated our proposed IK solver on a physical robot, putting it to the test in real-world scenarios. In addition to this, we have created and assessed an efficient whole-body collision detection mechanism based on the prismatic-universal-prismatic joints model. Furthermore, we have introduced and experimentally evaluated an obstacle avoidance algorithm within a simulated environment. These experiments collectively reinforce the effectiveness and viability of our approach in addressing the complex challenges of real-time posture control for extensible multisection continuum robots.

Contributions of this method:

1. Novel, fast and accurate geometric analytical IK solver for extensible multisection continuum robots
2. Guaranteed solutions for achievable end-effector position and orientation
3. Smooth motions and shapes by optimizing configurations
4. Implementation an entire robot body collision detection method

5.1 Algorithm Description

For extensible continuum robots (CRs) with variable section lengths, configurations are akin to sequences of constant curvature curves. Our task is to minimize the number of these curves needed to achieve the desired start and end poses. In 3D space, this amounts to a 5 DoF IK problem, neglecting z-axis rotation, as we often assume an omnidirectional final gripper. Nonetheless, we retain this degree of freedom for optimizing pose smoothness.

In this approach, we use the virtual serial linkage method (Fig. 3.4) to represent constant curvature curves as a series of prismatic-universal-prismatic joints (PUP) serial linkage robotic arms, which is described at Sec. 3.2.2. Based on this representation, a coarse geometry-based analytical IK solution is obtained for two sections. There are multiple solutions for 5 DoF IK problem in 6 DoF space. The curvature balancing method is used for picking a constant curvature pair. Then we segment the PUP series by adding virtual universal joint or dividing constant curvature curve to satisfy the section number. The flowchart shown in Fig. 5.1.

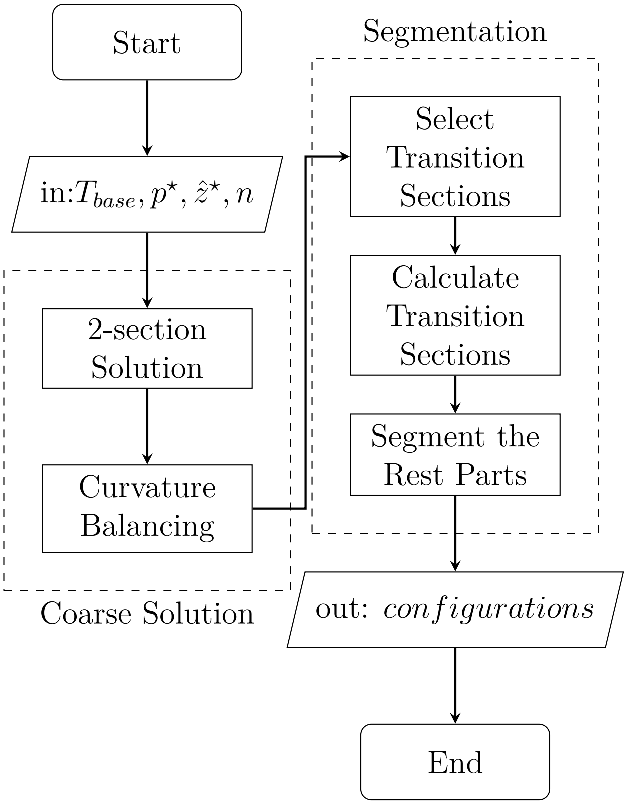


Figure 5.1: The Flowchart of AMoRPH

5.1.1 Coarse 2-section IK

The inverse kinematics problem is solved given only the start (p_{start}, *orientation-1*) and end (p_{goal}, *orientation-2*) pose. We establish a virtual joint ($j2$) on the *orientation-2* axis and two virtual links (l_2) with arbitrary initial lengths (Fig. 5.2). The $j2$ position is set by varying l_2. A temporary link of length a connects $j2$ to p_{start}, forming a triangle. According to the law of cosines, the relationship can be expressed as Eq. 5.1.

$$(l_1 + l_2)^2 = l_1^2 + a^2 - 2al_1 \cos\theta \tag{5.1}$$

Upon organizing the information, we can obtain that l_1 is a function of the variables a, θ, and l_2, as Equation 5.2.

$$l_1 = \frac{a^2 - l_2^2}{2(l_2 + a\cos\theta)} \tag{5.2}$$

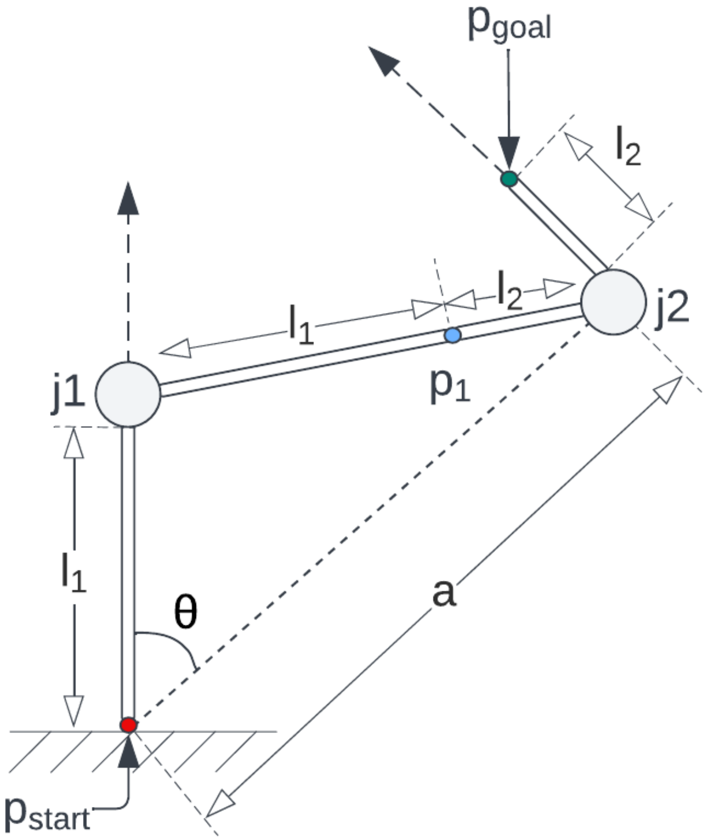


Figure 5.2: The virtual joints *j1* and *j2* move along the *orientation-1* and *orientation-2* respectively, with accompanying virtual links of equal on both sides. A coarse IK is obtained per Eq. 5.2. Point j' illustrates the geometry of the adding-segment algorithm.

The given variables, p_{start}, p_{goal}, can be represented as points in space $\mathbf{X_b}$ and $\mathbf{X_e}$, *orientation-1*, *orientation-2* can be represented as unit vectors indicating direction in space $\mathbf{u_1}$ and $\mathbf{u_2}$. This is shown in Equation 5.3.

$$\begin{aligned}
\mathbf{X_b} &= p_{start} \\
\mathbf{X_e} &= p_{goal} \\
\mathbf{u_1} &= orientation_1 \\
\mathbf{u_2} &= orientation_2
\end{aligned} \tag{5.3}$$

The location of the virtual joints $j1$ and $j2$ can be represented by $\mathbf{X_1}$ and $\mathbf{X_2}$, respectively. These variables are dependent on $\mathbf{X_b}$, $\mathbf{X_e}$, $\mathbf{u_1}$, $\mathbf{u_2}$, l_1, and l_2, as indicated by their functional relationship in Equation 5.4.

$$\begin{aligned} \mathbf{X_1} &= \mathbf{X_b} + l_1\mathbf{u_1} \\ \mathbf{X_2} &= \mathbf{X_e} - l_2\mathbf{u_2} \end{aligned} \tag{5.4}$$

The temporary link, denoted as a, represents the distance between points $\mathbf{X_b}$ and $\mathbf{X_2}$. In Equation 5.5, we introduce the result from Equation 5.4 to express a as a function of l_2. It is important to note that $\mathbf{X_e}$, $\mathbf{u_2}$ and $\mathbf{X_b}$ are all given variables.

$$\begin{aligned} a &= \|\mathbf{X_2} - \mathbf{X_b}\| \\ &= \|\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b}\| \end{aligned} \tag{5.5}$$

The value of $\cos\theta$ can be obtained via the dot product between u_1 and the vector $\mathbf{X_2} - \mathbf{X_b}$. Similarly, in Equation 5.6, after introducing the results of Equation 5.4 and Equation 5.5, $\cos\theta$ becomes a function of l_2.

$$\begin{aligned} \cos\theta &= \frac{\mathbf{u_1} \cdot (\mathbf{X_2} - \mathbf{X_b})}{\|\mathbf{u_1}\|\|\mathbf{X_2} - \mathbf{X_b}\|} \\ &= \frac{\mathbf{u_1} \cdot (\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b})}{\|\mathbf{u_1}\|\|\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b}\|} \end{aligned} \tag{5.6}$$

From Equation 5.5 and Equation 5.6, it can be observed that in Equation 5.2, both a and $\cos\theta$ are functions of l_2, therefore l_1 is also a function of l_2 (Equation 5.7). In other words, given a value for l_2, we can obtain a corresponding value for l_1, thereby obtaining a complete 2-section Coarse IK solution.

$$l_1 = \frac{\|\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b}\|^2 - l_2^2}{2(l_2 + \|\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b}\| \frac{\mathbf{u_1} \cdot (\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b})}{\|\mathbf{u_1}\|\|\mathbf{X_e} - l_2\mathbf{u_2} - \mathbf{X_b}\|})} \tag{5.7}$$

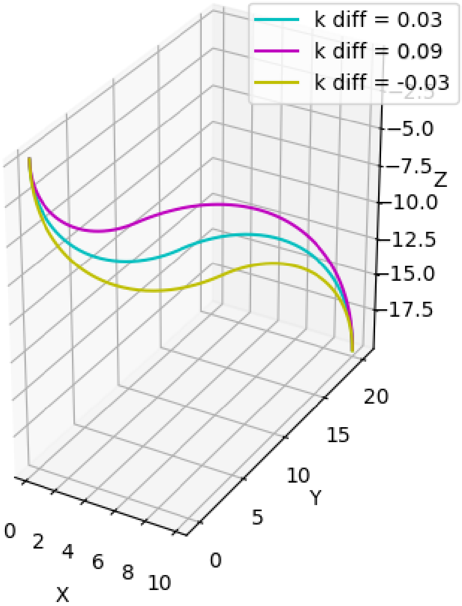


Figure 5.3: Coarse IK solutions share identical start/end poses. The curves have different curvatures.

5.1.2 Curvature Balancing

Since the calculation is done in virtual PUP configuration space, we only calculate 5 DoF during the calculation, where z-axis rotation is not considered, so we have an infinite number of pairs that can satisfy this IK problem. Fig. 5.3 shows multiple IK solutions with different curvature pairs. This provides us with more room to choose more suitable configurations.

Curvature characterizes the extent of curvature within a curve. Maintaining uniform curvature distribution between two curves is desirable, making the quest for a balanced curvature pair an optimal solution. For varying values of l_2, distinct curvature pairs can be identified. To obtain the nearest curvature pair, we employ a root-finding algorithm.

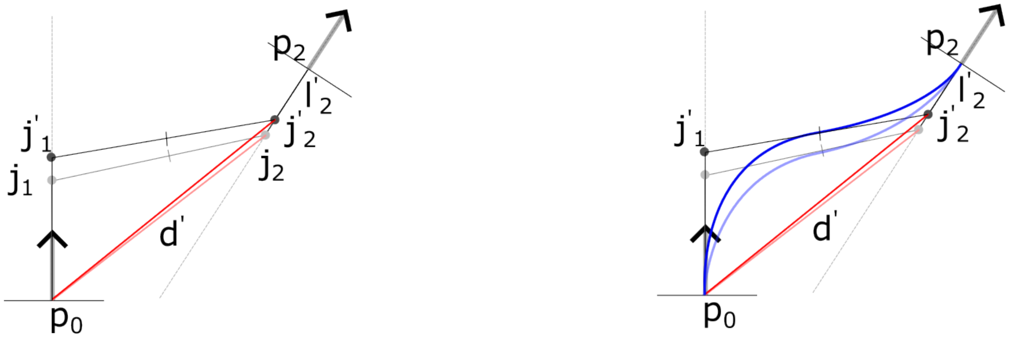


Figure 5.4: With varying l_2 values, under the constraints of equal link lengths on either side of the joint and joints moving solely along the orientation direction, we achieve a unique triangular configuration. This configuration facilitates the determination of l_1 length and the position of j_1, allowing us to identify a curve pair. A root-finding algorithm is subsequently utilized to pinpoint the curve pair with the most similar curvatures.

5.1.3 Addition of virtual joints

Using the same calculation, we can arbitrarily add a new joint (j_{new}) between joint 1 and joint 2. By given l_new, the calculation can be done on both sides to get updated joint 1 and joint 2 with proper l_1 and l_2. Fig. 5.5 shows different configurations by adding different virtual joint. This powerful method allows us to add sections arbitrarily, thus changing the overall attitude, but maintaining the position and orientation of the tip.

5.1.4 Sections Dividing

The constant curvature curve pair discussed earlier maintains uniform curvature along its entire length, making it suitable for division into multiple continuum modules. When two curves of comparable length are each divided into two segments, a four-section configuration can be generated.

However, a 2-section coarse IK solution does not always coincide with the division point. A uniform cut may result in unequal section lengths; for example, dividing into three configurations might yield an arc length distribution of 0.25, 0.25, 5, which is

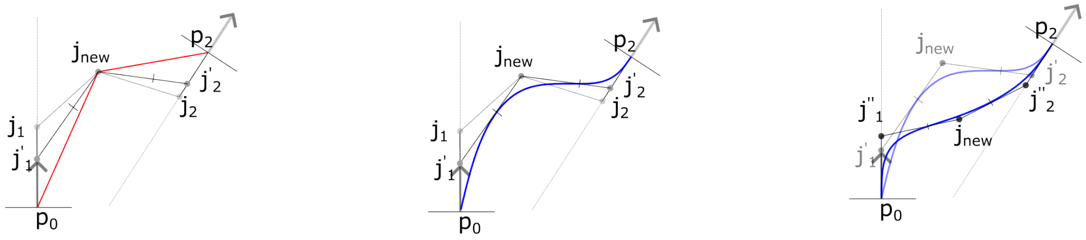


Figure 5.5: Addition of virtual joints. (a) We arbitrarily add a new joint (j_{new}). To connect to p_0 and p_2 from j_{new}, we get the distance and the angle between the connection line and the orientations. (b) With a given link length l_new, we can update l_1 and l_2 to satisfy the PUP model constraints, which can convert to constant curvature curve pairs. Then, we have a new, 3-section configuration. (c) If we assign the joint to another position, it generates another configuration.

highly uneven. We employ the virtual joint addition method described previously to address this issue. This entails assigning a new virtual joint between joint 1 and joint 2, positioned at the projection point from the midpoint of the total arc length for the 3-section case. The virtual link length is determined based on the current bending angle and extrapolated from the expected arc length. This approach ensures that the shapes remain similar and maintains a consistent arc length distribution.

Our proposed approach initially calculates a 2-section coarse IK solution, which is then divided into multiple sections to align with the physical robot's requirements. This eliminates the need for iterative adjustments to approach the desired pose, and the solvability of the solution can be confirmed from the outset. Moreover, in the division process, the complexity of computation remains constant and unrelated to the number of sections, presenting an $\mathcal{O}(1)$ complexity.

5.1.5 Obstacle Avoidance

Obstacle avoidance is predicated on collision detection. We first assess whether the obstacle is in close proximity to the robot, and accordingly employ the appropriate avoidance maneuvers based on its distance. In this regard, we will introduce an efficient and conservative collision detection method based on the PUP model. Furthermore, we will elucidate how the outcomes of this method can be utilized for effective obstacle avoidance.

Collision Detection

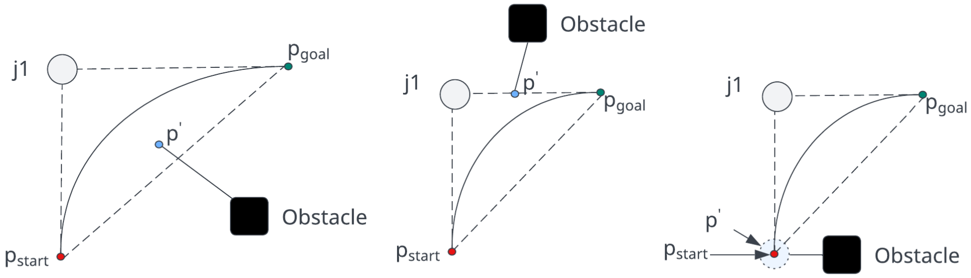


Figure 5.6: In the spatial domain, the distance between an obstacle and a triangle can be categorized into three scenarios: the distance between the obstacle and the triangle's plane, the distance between the obstacle and the sides of the triangle, and the distance between the obstacle and the triangle's vertices.

Following the methodology described in Section 3.2.2, it is feasible to transform all constant curvature curves into P-U-P combinations, typically composed of three segments. In a three-segment configuration, these segments form a planar triangle, while a bending angle of 0 results in a straight line. All constant curvature curves can be characterized by these triangular representations, simplifying the collision detection process.

To ascertain the distance between an arbitrary point in space and a triangle, the following procedure is applied. Fig. 5.6 shows the visualization of the scenarios. Firstly, the projection of the point onto the plane where the triangle is situated is computed. Subsequently, if the projection point falls within the confines of the triangle, the distance between the point and the plane is determined. If the projection point lies outside the triangle, the distances between the point and each of the three sides of the triangle are calculated. This approach parallels the previously discussed distance calculation method, where distances from the obstacle to the sides of the triangle are computed. If the point is projected onto the line representing a particular side and is within the side's boundaries, the distance between the point and the line is computed. If, on the other hand, the projected point is outside the side, the distance from the point to the triangle's vertices is computed, selecting the smallest of these values.

Obstacle Avoidance

After detecting where a collision may occur, we choose the detour location based on the obstacle's position and the robot body. If the proposed pulling point is at the corresponding length of the original virtual joints, we change the original virtual joint. If it is on a smooth curve, we add a pulling point. However, there are two exceptions: firstly, if the obstacle is close to the center axis of the first segment, we consider it unavoidable. Secondly, if the obstacle is close to the center axis of the last segment, the system cannot reach the desired goal. In such cases, the system will reset the endpoint to the nearest safe range point along the center axis to the original goal.

5.2 Algorithm Limitations

The proposed approach is applicable to extensible multisection continuum robots. During the segmentation phase, the arc length of the modules is adjusted. This method results in larger errors in the IK configurations for fixed arc length multisection continuum robots. Additionally, since it is based on a two-section PUP model to calculate the Coarse IK solution, a singularity exists. When the desired end-effector position is located on the negative z-axis of the base and has the same orientation as the base, the bending angle is 2π. In such a posture, the continuum module is unable to meet the requirements.

5.3 Experiments and Discussions

Our experiments comprised two main categories: (i) predefined trajectory tests and (ii) real-time interaction scenarios. In the predefined trajectory experiments, we compared different IK algorithms in identical environments and measured their computational efficiency. We also validated our approach on a physical robot using predefined trajectories. The random interaction experiments assessed our method's capability to handle random end-effector poses in both simulation and physical robot settings. Additionally, obstacle avoidance performance was evaluated in the simulated environment. Our results demonstrated that our approach yields feasible and real-time human-operated responses.

We evaluated our IK method through Matlab simulations on a MacBook Pro 13" (CPU: 2.3 GHz quad-core Intel Core i5). The experiments compared computation times and solution accuracy with existing algorithms. We employed predefined trajectories for pick-and-place and circular motions. Computation time was measured from pose input to joint configuration output, excluding trajectory generation and visualization overhead. The simulated robot mirrors the physical counterpart detailed in Sec. 3.1.

Table 5.1: Algorithm comparison for the pick-and-place trajectory

Algorithm	Pick-and-Place		
	Time (s)	Position Error (μm)	Orientation Error
AMoRPH	0.0408 ± 0.0257	16.056 ± 0.02279	$2.7502 \times 10^{-7} \pm 3.9039 \times 10^{-6}$
exFABRIKc	0.3482 ± 0.0232	151.3 ± 244.6	$4.8003 \times 10^{-13} \pm 3.6572 \times 10^{-12}$
Opti. based	9.9867 ± 0.4237	248.9 ± 2482.9	$1.9953 \times 10^{-4} \pm 0.0019$
SLInKi	0.9236 ± 0.0444	655.0 ± 5087.2	0.0338 ± 0.2548

5.3.1 Predefined Trajectory Experiments

The first pick-and-place trajectory moved from point A down 100 mm, horizontally to point B, and down another 100 mm, with the end-effector fixed downward. A return path to A was also incorporated before moving to B. The second, circular, trajectory

Table 5.2: Algorithm comparison for the circular trajectory

Algorithm	Circular		
	Time (s)	Position Error (μm)	Orientation Error
AMoRPH	0.0107 ± 0.0043	$(6.3230 \pm 12.207) \times 10^{-10}$	$(4.4439 \pm 8.4120) \times 10^{-15}$
exFABRIKc	0.0789 ± 0.0090	1.1 ± 1.7	$(1.6169 \pm 1.1115) \times 10^{-15}$
Opti. based	35.5736 ± 3.1228	0.60198 ± 0.5624	$(2.3791 \pm 3.1665) \times 10^{-5}$
SLInKi	0.2352 ± 0.0123	$(1.4705 \pm 1.4367) \times 10^{-10}$	$(1.2003 \pm 1.1470) \times 10^{-15}$

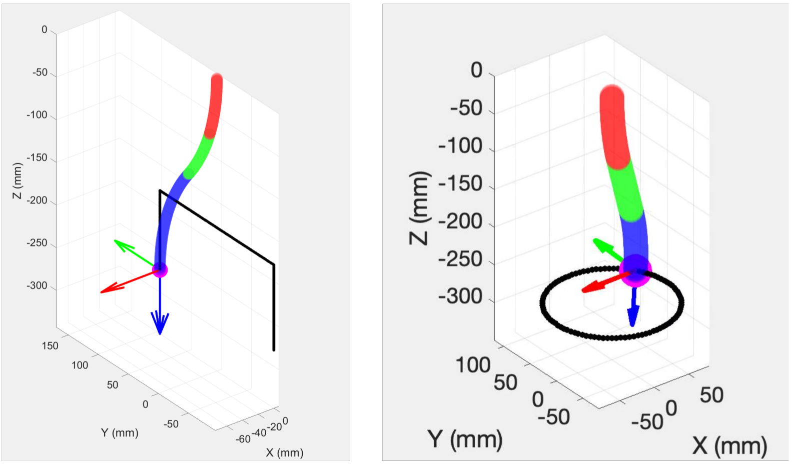


Figure 5.7: Visualization of pick-and-place and circular trajectories.

had an 80 mm radius, with the end-effector orientation rotating 45ř around the x-axis. The pick-and-place trajectory has 404 steps; the circular trajectory has 100 steps. For both trajectories, the primary metric was time to compute inverse kinematics solutions from desired poses.

We tested (1) our proposed approach, (2) cable-length-based optimization IK [8], (3) exFABRIKc [11], modified from FABRIKc [10] (4) search-based IK SLInKi [11] in the same experimental setting.

The trajectories are tested for 100 times with each IK solver. The average and standard deviation of the computational time are listed in Table 5.1,5.2. The proposed approach needs 40.8 milliseconds to solve the pick-and-place trajectory, which means it takes about 0.1 millisecond to solve one step. As a comparison, exFABRIKc is 8 times

Table 5.3: Mean of Variance of Curvatures (km^{-1})

Algorithm	Pick-and-place	Circular
AMoRPH	11.699	1.3005
exFABRIKc	4.8012	10.816
Opti. based	10.624	12.672
SLInKi	153.21	24.571

slower, SLInKi is 12 times slower, and the optimization-based IK is 244 times slower than AMoRPH.

Regarding accuracy, we evaluate the accuracy of the IK solutions by comparing the position and orientation of the end-effector obtained through forward kinematics with the given desired end-effector pose. The positional error and orientation error are calculated at each step. The mean and standard deviation of the errors are computed statistically. The results are presented in Table 5.1,5.2. It can be observed that all four algorithms achieve good accuracy, but the proposed approach leverages analytical solutions and thus produces negligibly small errors.

In hyper-redundant systems, where numerous configurations can achieve a desired end-effector pose, balanced solutions are preferred. To quantify smoothness, we analyzed sectional curvatures using the variance of curvature within each section. Smaller variances indicate smoother profiles. Table 5.3 displays the results, with curvature variances measured in units of km^{-1}. Notably, AMoRPH closely matches Optimization-based IK and outperforms other algorithms in circular trajectories, while exFABRIKc excels in Pick-and-Place trajectories.

In continuous motions, motion smoothness depends on cable length changes between steps. To assess smoothness, we analyze the differences in cable lengths between steps for a 3-section CR. Smaller differences indicate smoother motion. The results, shown in Table 5.4, demonstrate AMoRPH's superior performance in both pick-and-place and circular tasks compared to other algorithms, with Optimization-based IK used as the benchmark due to its optimization cost function relating to cable length changes.

Table 5.4: Cable Length Changing (mm)

Algorithm	Pick-and-Place	Circular
AMoRPH	0.9933 ± 0.1787	2.2528 ± 0.6884
exFABRIKc	1.4363 ± 0.8652	2.0316 ± 0.4710
Opti. based	1.1097 ± 1.2573	4.6115 ± 2.3384
SLInKi	16.2027 ± 36.3850	18.8711 ± 30.0343

5.3.2 Real-time Interaction Demonstration with Joystick

Simulations were based on kinematic models. Physical limitations like deadlength and bending angle constraints were addressed through optimized mechanical designs. The random trajectory helps us evaluate the fidelity not only the real-time response but also the configurations in the physical world.

The Nintendo Switch Pro Controller (NSPC) is a pivotal input device in this research. It possesses an internal Inertial Measurement Unit (IMU), which intuitively manipulates the desired end-effector orientation. Furthermore, the controller incorporates a pair of analog joysticks, affording precise control over the relative desired end-effector position in Cartesian coordinates. Notably, the controller features a designated button purposed for transitioning the point of focus in manipulation from the desired end-effector to the obstacle.

Random Pose Following

We tested the AMoRPH IK solver with NSPC on both simulated and physical robots (Fig. 5.8), and the AMoRPH smoothly handles the 3-section extensible continuum robot on-the-fly, yielding shape and motion in both random position and orientation. The AMoRPH can handle the 3-section extensible continuum robot in random position and orientation smoothly and instantly.

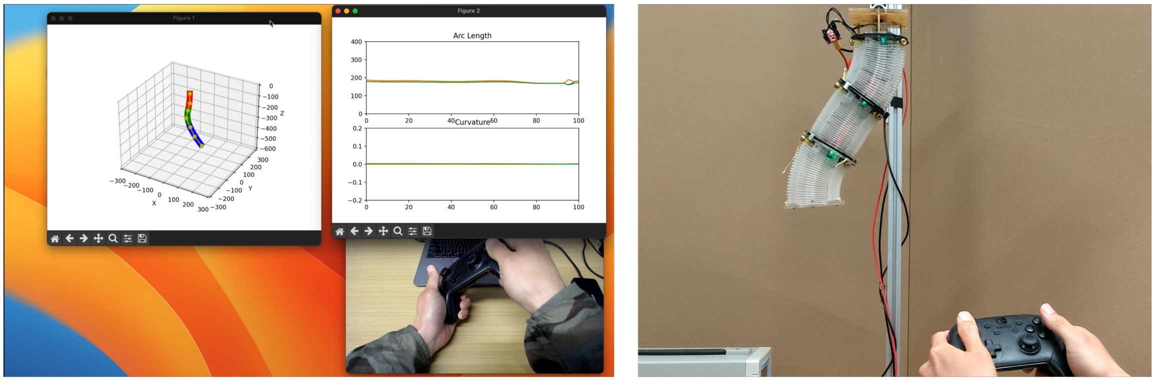


Figure 5.8: The experimental configuration for random interaction experiments involved user-driven manipulation of CRs employing NSPC on both simulated and physical robotic platforms.

Obstacle Avoidance

We have also tested the obstacle avoidance function in the simulation environment (Fig. 5.9). The operator can switch between operating the end-effector and the obstacle by pressing the ZR button. In practice, the AMoRPH can handle the configuration to avoid the obstacle. If the obstacle is close to the desired end-effector, AMoRPH abandons the given goal position and backs off to a position far enough from the obstacle according to the specified orientation.

5.3.3 Real-time Interaction Demonstration with Motion Capture System

In order to enhance the intuitive operation of this system, we have employed a motion capture system as the input interface for the EE pose. This enables a more direct interaction with the robot, allowing us to observe the real-time performance of the AMoRPH algorithm and the smoothness of the calculated pose. In this demonstration, we initially utilize a rigid body's pose as the desired pose, which is freely manipulated and rotated by the user. Subsequently, we introduce another rigid body, using its position as the obstacle's position, and manipulate it to approach the robot. This allows us to observe

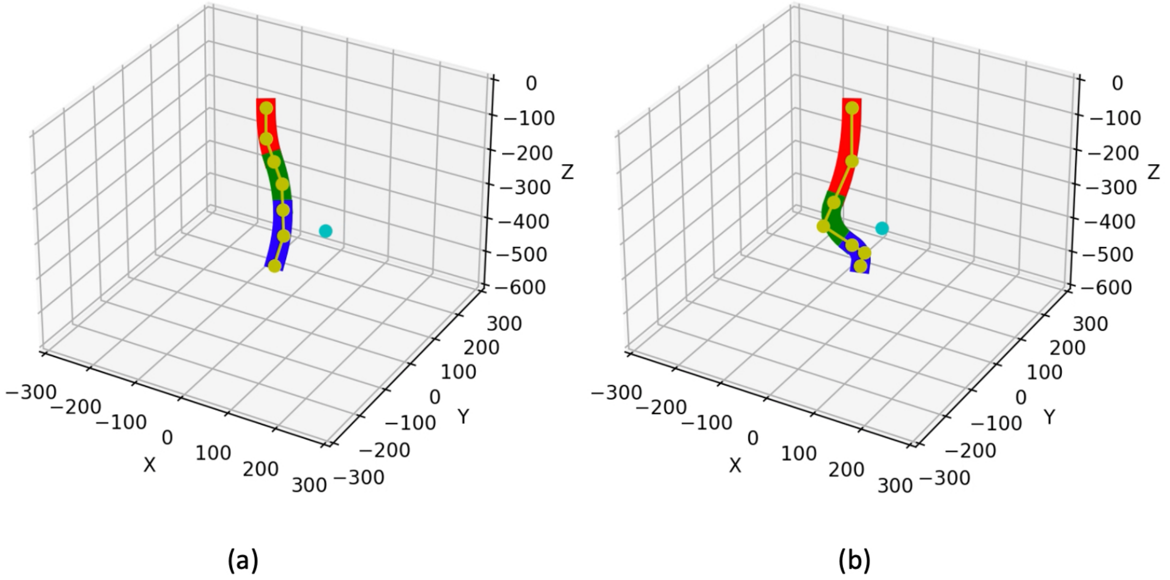


Figure 5.9: Obstacle Avoidance Demonstration: Users control the obstacle's position with NSPC. In Scenario (a), where the obstacle is distant from the CRs, AMoRPH ensures smooth configurations. In Scenario (b), when the obstacle approaches the CRs, AMoRPH steers the robot clear of the obstacle.

the obstacle avoidance posture computed by AMoRPH.

Random Pose Following

We utilize the OptiTrack® system to track a handheld marked rigid body. The pose of this rigid body is then employed as the input for the IK algorithm, serving as the desired end-effector pose, to evaluate its real-time tracking capability. In order to prevent collision between the marker and the tip of the robot, we have implemented a 10 cm offset along the z axis. We utilized a 4-section OrCA configured in the toward down direction. During the operation, the user has the freedom to manipulate and rotate the designated rigid body, and the OrCA can accurately track the changes in both its position and orientation in real-time. It is worth noting that in this particular demonstration, the physical robot solely relies on low-level cable-length control, with no additional feedback mechanism. Therefore, it can be categorized as an open-loop control system. Nevertheless, the robot demonstrates remarkable responsiveness to changes in position and orientation, ensuring smooth motion and maintaining its intended shape throughout the process.

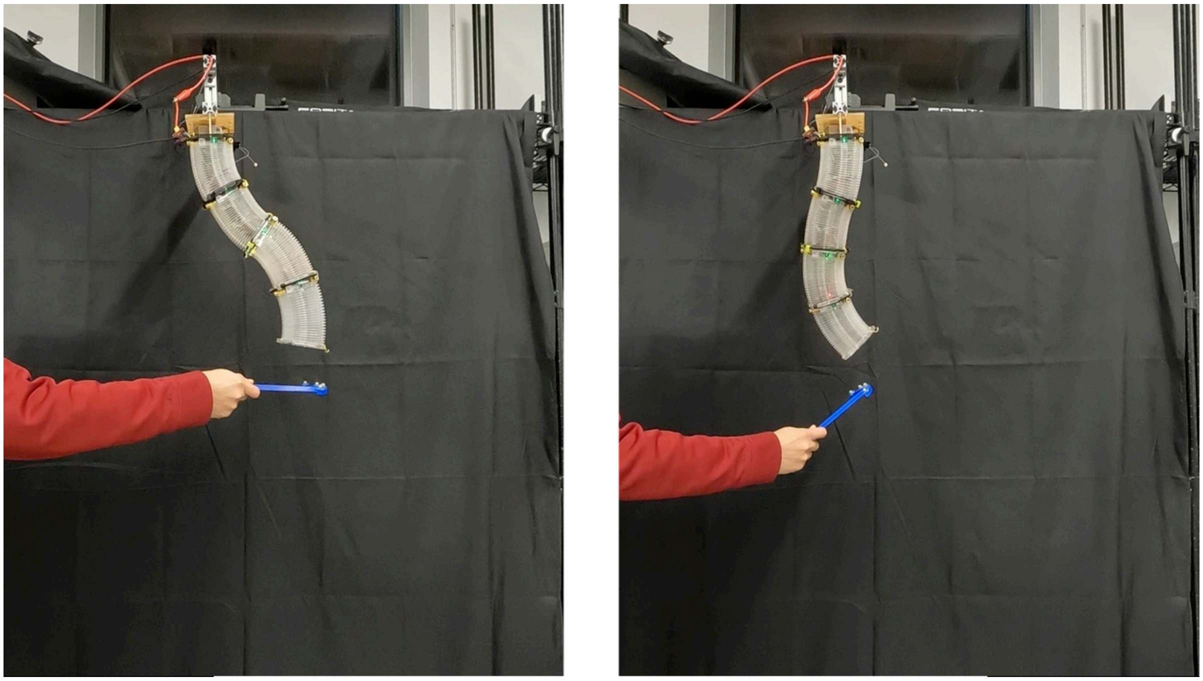


Figure 5.10: The user manipulates the designated rigid body to input the desired poses, while the OrCA, with the AMoRPH IK algorithm, enables real-time tracking of the desired end-effector pose.

Obstacle Avoidance

Continuing from the preceding demonstration, we introduce another rigid body, utilizing its position as the obstacle's position, and manipulate it to approach the robot. In Fig. 5.11, it can be observed that as the obstacle approaches the robot body, several intermediate segments undergo changes in posture, while the end effector (EE) maintains the same pose. In Fig. 5.12, it is evident that as the obstacle comes closer to the EE position, we alter the desired EE position to make it retreat, all the while preserving its original orientation.

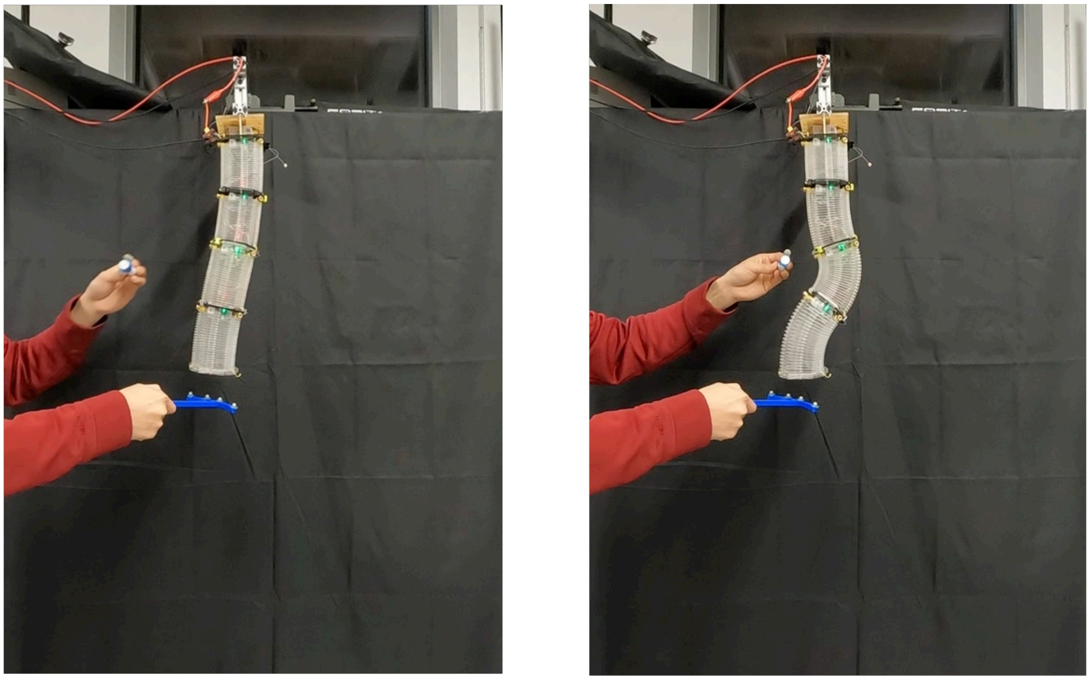


Figure 5.11: The user manipulates the specified rigid bodies to input the desired positions and obstacles, while AMoRPH performs real-time configuration adjustments to avoid obstacles and maintain the target end-effector pose.

5.4 Summary

We introduce AMoRPH, a novel model-based real-time posture heuristic approach designed for solving inverse kinematics problems in extensible CRs capable of managing both position and orientation. In performance and smoothness comparisons with three other inverse kinematics solvers, AMoRPH demonstrates superior speed, accuracy, and maintains high levels of smoothness in both shape and motion. The algorithm evaluated on a physical robot. Additionally, we have implemented an efficient entire robot body collision detection mechanism based on the PUP model. Building upon collision detection, we present the concept of an obstacle avoidance algorithm, which has also been experimentally evaluated in a simulation environment.

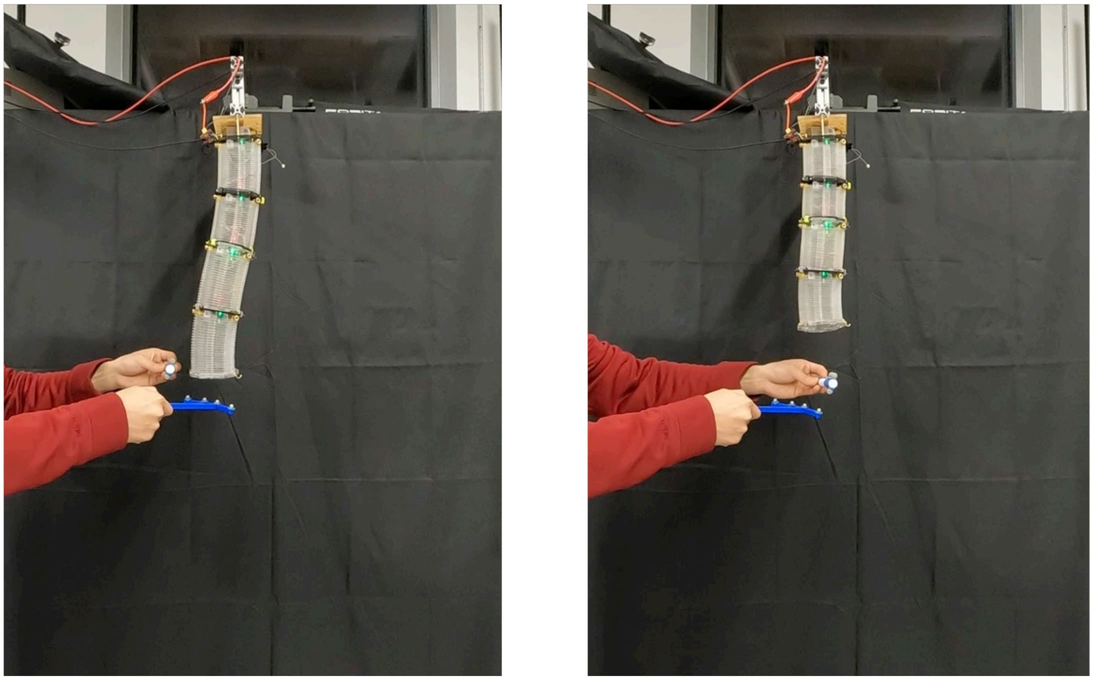


Figure 5.12: The user controls the designated rigid bodies to input desired positions and obstacles. When an obstacle approaches the end effector (EE) position, AMoRPH adjusts the EE position while maintaining its original orientation to avoid the obstacle.

Chapter 6

Conclusions

In order to enhance the practical application of continuum robots in real-world scenarios, several essential requirements have been identified, including fast operation, high accuracy, guaranteed solutions, customization capabilities, and obstacle avoidance. Existing inverse kinematics (IK) algorithms have been found inadequate in meeting these specified requirements.

To address these limitations, we have developed SLInKi, grounded in motion planning principles. Employing a greedy strategy, SLInKi expeditiously identifies viable solutions, demonstrating both promptness and precision when coupled with exFABRIKc or Geometry Analytical methodologies. This system exhibits real-time, interactive performance, with commendable accuracy in determining position and orientation.

Exploring various cost functions, we've found that different configurations can be generated for diverse purposes by adjusting the order of the two components (search and refine). Successful obstacle avoidance has also been achieved, yet the discrete results of state-lattice search may lead to less smooth motion, necessitating further consideration.

Recognizing the importance of smoothness, we extended the geometry analytical solver in SLInKi to develop a second IK algorithm, AMoRPH. Built upon the PUP model, AMoRPH achieves rapid IK solutions, outperforming the state of the art in simulations and real-world setups. Notably, AMoRPH's IK computation is 8 to 2000 times faster, making it highly suitable for real-time tasks.

To ensure smooth shapes, AMoRPH incorporates curvature balancing into the Coarse IK solution, utilizing a spline-inspired segmenting method during the segmentation phase. Our method consistently ranks among the top tier in shape smoothness assessment, employing the mean value of the variance of curvatures. Motion smoothness is similarly validated by measuring the average cable length difference between adjacent steps during continuous movements.

Moreover, we have leveraged the PUP model for efficient collision detection, achieving real-time performance. The modular nature of the PUP model enables simultaneous inverse kinematics for both intermediate and end points, along with successful demonstrations mapping the human arm.

In summary, our contributions include the development of SLInKi and AMoRPH, addressing the shortcomings of existing IK algorithms. Our methods exhibit real-time performance, rapid computation, and superior smoothness in both shape and motion, making them valuable tools for the practical application of continuum robots.

Key Points:

- Developed SLInKi grounded in motion planning principles for real-time, precise solutions.

- Introduced AMoRPH, a second IK algorithm achieving rapid computation (0.1 ms) and substantial speed enhancement.

- Addressed the importance of smoothness, incorporating curvature balancing and achieving top-tier results in shape and motion assessments.

- Leveraged the PUP model for efficient collision detection, enabling real-time performance and simultaneous inverse kinematics for various points.

- Demonstrated the applicability of our methods through simulations and real-world setups, highlighting their suitability for diverse tasks.

Chapter 7

Broader Impacts

In this chapter, we delve into the profound implications of our study across various crucial domains. These impacts extend beyond the realms of academic inquiry, encompassing societal, educational, and health-related dimensions. Our focus is particularly directed towards the fields of assistive robotics, STEM education, and the development of ventilators during the COVID-19 pandemic. Research in these areas holds significant relevance for enhancing human life and addressing specific circumstances.

Firstly, we will explore the application of assistive robotic technology in society, specifically in aiding and supporting individuals requiring special care. Assistive robots not only contribute to improved quality of life but also broaden the social participation of beneficiaries. Research in this area is poised to propel technological advancements while fostering inclusivity and diversity within society.

Secondly, we will discuss the importance of STEM education and how our research contributes to the advancement of this field. STEM education is pivotal in nurturing future scientists, technologists, engineers, and mathematicians, and our research findings aim to enrich educational content by providing students with opportunities to apply scientific knowledge in practical settings, cultivating innovative thinking and problem-solving skills.

Lastly, we will examine the impact of the ventilator developed during the COVID-19 pandemic. This research direction directly responds to the ongoing global health crisis, underscoring the critical role of technological innovation in crisis response. Our outcomes hold promise in offering practical and viable solutions for future crisis management, emphasizing the potential applications of technology in emergency situations.

The discussions in this chapter showcase how our research deeply and diversely influences society, offering a broader perspective and emphasizing the crucial role of technological innovation in addressing real-world challenges.

7.1 Assistive Robots

A variety of studies have investigated the amalgamation of robotic technology with wheelchairs to aid individuals with disabilities. Hillman [35] focused on the attachment of a robotic arm to a powered wheelchair, addressing the limitations and prerequisites of this integration. Sinyukov [36] presented a navigation framework for electric wheelchairs, combining alternative input interfaces with assistive navigation to facilitate secure indoor movement. Kuo [37] proposed a human-oriented design for an autonomous navigation-assisted robotic wheelchair, highlighting user authority and comfort. Collectively, these studies underscore the potential of robot-assisted wheelchairs in augmenting mobility and self-sufficiency for individuals with disabilities.

7.1.1 Motivation and Goal

This research is positioned as a pertinent contribution to the development of a secure, economical, and user-friendly assistive robot, with the overarching goal of mitigating obstacles to independent living experienced by individuals grappling with mobility impairments or muscle dysfunction. Conditions such as spinal cord injuries, traumatic brain injuries, cerebral palsy, neurological disorders like muscular dystrophy, and the natural aging process often render individuals incapable of self-care, necessitating the intervention of a home caregiver. The prevailing shortage of home care workers compounds this challenge, presenting a significant issue within the healthcare system. Several studies, including those by Mohamadzadeh et al. [38] and Haghgoo et al. [39], underscore the impact of activities of daily living (ADL) on mental health, further positing that an economically viable assistive robot holds promise as an efficacious means of enhancing overall quality of life.

The envisaged assistive robot stands out as a meticulously crafted solution tailored to address the multifaceted impediments faced by individuals with mobility impairments, thereby fostering a heightened degree of autonomy in their day-to-day lives. A primary focus of this innovative solution lies in furnishing support for ADL, with the specific aim of augmenting the routine functionality of individuals affected by diverse conditions such as spinal cord injuries, traumatic brain injuries, cerebral palsy, muscular dystrophy, and the inexorable progression of aging. By facilitating assistance with routine activities, the assistive robot strategically addresses a spectrum of physical limitations, ultimately contributing to an enhanced quality of life for its end users.

The JACO® robotic arm [40] has demonstrated substantial effectiveness in enhancing the quality of life and societal integration of individuals with upper extremity disabilities [41]. Its intuitive usability makes it particularly accessible and has the potential to decrease the amount of time caregivers devote to assisting individuals by a remarkable 41% [42]. Moreover, an extended employment of the JACO® arm has been shown to enhance manipulation capabilities and facilitate the accomplishment of daily activities, thereby reducing reliance on caregiver support [43]. Collectively, these findings suggest that the JACO® robotic arm positively affects the lives of individuals with upper extremity impairments by promoting independence and bolstering social participation.

Beyond its role in physical assistance, the assistive robot assumes a strategic posture in response to the prevailing shortage of home care workers, functioning as a technological augmentation designed to bridge gaps in care provision. This forward-thinking solution endeavors to supplement human assistance, particularly in instances where the scarcity of caregivers threatens the seamless delivery of requisite support services. Moreover, the scope of the assistive robot extends beyond mere physical aid, as it conscientiously acknowledges and addresses the intricate interplay between ADL and mental health. This acknowledgment underscores the profound impact that routine activities can exert on an individual's psychological well-being.

In its quest for societal impact, the overarching vision for the assistive robot underscores affordability as a cornerstone principle. The imperative is to engineer a cost-effective robotic solution, ensuring that the advantages of augmented mobility and support with daily activities are accessible to a broad demographic grappling with mobility challenges. By steadfastly striving for affordability, this project endeavors to democratize access to assistive technology, thereby engendering a notable enhancement in the quality of life for those who stand to derive the greatest benefits from its implementation.

7.1.2 Key Insights from Interviews

We believe that the Origami Continuum Arm, when paired with SLInKi and AMoRPH, can serve as an assistive robot for wheelchair users. To clarify the real needs, as a preliminary study, we reached out and informally interviewed an engineer from a wheelchair manufacturing company in Taiwan, as well as two Taiwanese wheelchair users one using a power wheelchair and the other using a manual wheelchair. This was a general inquiry to gain an understanding of the potential requirements, in order to determine the direction for further research and development. No quantitative or qualitative research analysis was conducted at this stage.

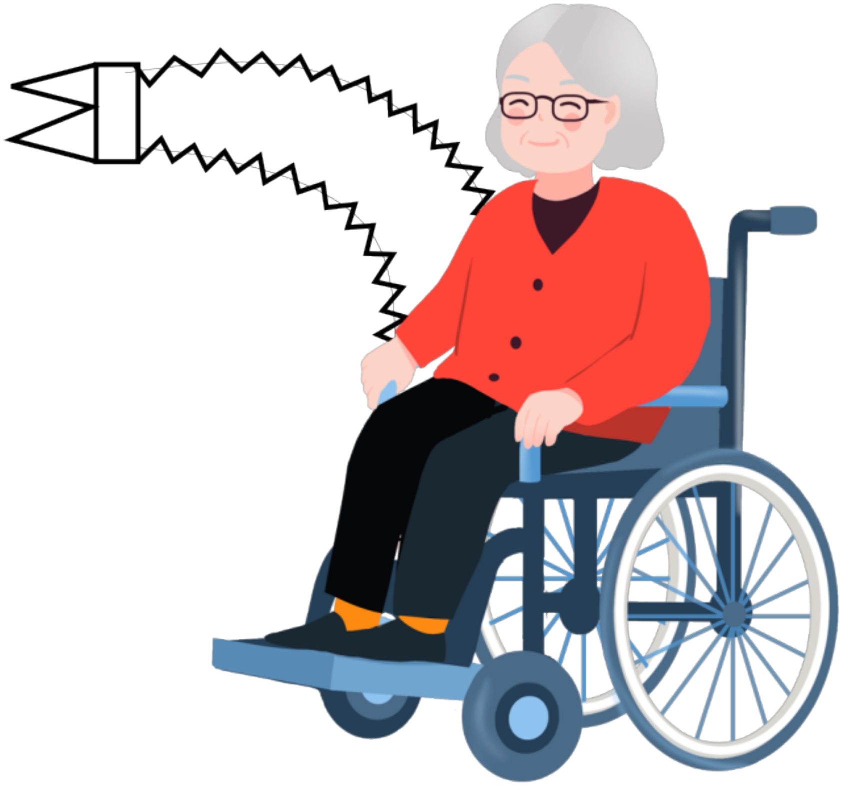


Figure 7.1: Assistive robot makes grandma happy.

The Assistive Robot for Wheelchair Users, as elucidated in the presented frame, is informed by insightful interviews conducted with wheelchair engineer and users. The obtained qualitative data has illuminated several salient points, encapsulated within the subsequent key insights:

Firstly, a recurring challenge articulated by wheelchair users pertains to the impediment in accessing items positioned at elevated heights. This manifests as a noteworthy hindrance in the daily lives of users, necessitating the development of solutions to address this common difficulty.

Secondly, users have identified door operations as a significant source of inconvenience. The act of reaching and operating doors is reported as a notable challenge, emphasizing the importance of incorporating functionalities that facilitate seamless interaction with doors.

Thirdly, the interviews underscore the value of providing posture support for wheelchair users. The nuances of how this posture support is implemented require further elaboration, particularly in delineating control methods that align with the limitations in hand functionality experienced by wheelchair users. This aspect becomes pivotal in ensuring the effectiveness and user-friendliness of the assistive robot.

Fourthly, a substantial proportion of wheelchair users encounter limitations in hand functionality. The imperative, therefore, lies in the design and operation of the assistive robot, necessitating simplicity to accommodate these constraints and enhance user accessibility.

Lastly, concerns have been raised by wheelchair engineers regarding the payload capacity of the Continuum Arm, a crucial component of the Assistive Robot. The frame calls for the provision of pertinent details and potential solutions to address these concerns, thereby ensuring the optimal functionality and safety of the Continuum Arm in various usage scenarios.

In essence, the insights gleaned from these interviews provide valuable guidance for the refinement and enhancement of the Assistive Robot for Wheelchair Users. By addressing these identified challenges and concerns, the development of the assistive robot can be tailored to more effectively meet the specific needs and requirements of wheelchair users, contributing to a more inclusive and user-centric assistive technology.

7.1.3 Potential for Broader Applications

In comparison to articulated assistive robot arms such as the JACO Arm [40], the advantages of the lightweight, compliant, and redundant mechanisms of OrCA system are noteworthy. The lightweight design allows for easy installation onto manual wheelchairs, negating the need for large motors and electrical control systems. Consequently, this reduces overall costs, making it more affordable. While force control can ensure compliance in articulated arms, mechanical compliance ensures safety in various situations, including instances where the electrical control system fails. Being redundant enables this system to handle a wider range of everyday needs and accomplish tasks in narrow or constrained environments. Conversely, a drawback is the limited payload capacity, which may make control challenging when subjected to external forces and prevent the handling of heavier objects. To enhance payload capacity, employing stronger materials and motors is necessary, but this would compromise the benefits of lightweight design and compliance.

The combination of OrCA with the presented IK solvers has the potential to become an ideal assistive robot for wheelchair users. However, despite its affordable raw material costs, we currently lack appropriate large-scale production methods. The origami structure of OrCA is relatively complex for factory automation, requiring manual assembly in the prototype stage and preventing mass production in factories. The simplicity of its design, coupled with its already published designs in the academic community, pose challenges for establishing patent barriers in commercialization. This increases the difficulty of profiting from this project while simultaneously helping those in need. These are important considerations to keep in mind during the commercialization process.

Furthermore, by making certain adjustments, this particular system framework can be applied to various other domains, including manufacturing, service robotics, and teleoperated applications. Such adaptations have the potential to significantly alter the landscape of robotics innovation, fostering a broader range of applications in the context of human-in-the-loop operating environments.

7.2 STEM Education

Educational robots, including intelligent assistants, virtual simulations, and multi-functional suite robots, are characterized by their flexibility, digitization, and natural interactivity

[44]. They are primarily utilized in STEM education, language learning, and special education. However, these robots also face challenges such as gender bias, technical issues, and resource constraints Pei [44]. Within the learning process, these robots can function as tutors, tools, or peers, proving particularly valuable in language, science, and technology education Mubin [45].

Recent advancements in electronics and computer technology have made it possible for educational institutions to obtain a variety of novel and versatile robotics modules within a limited budget. This allows for the creation of interest in STEM learning among students and provides expanded opportunities for those pursuing STEM careers Tuluri [46]. Educational robots offer a low-cost, user-friendly, modular, and scalable platform that enhances collaborative learning [47]. Consequently, the use of soft robotics in STEM education presents a promising avenue.

The utilization of soft robotics in STEM education has gained increasing attention due to its potential for pioneering pedagogical approaches and real-world applications. Agrawal [48] explores the challenges and possibilities of integrating soft robotics into educational settings, emphasizing the need for innovative instructional practices. Souza [49] provides practical examples of STEM education involving robotics, specifically tailored for children and adolescents. These studies collectively highlight the untapped potential of soft robotics in optimizing STEM education.

The presented project aligns with the aforementioned characteristics of a low-cost, user-friendly, modular, and scalable platform, making it highly suitable for application in educational robots. The unique attributes and benefits of the current project in the field of STEM education are elaborated upon as follows:

7.2.1 Contributions to STEM Education

The project makes noteworthy contributions to STEM education through the strategic incorporation of the following elements:

- Utilization of Affordable and Accessible Materials:

  OrCA, a pivotal component of the project, is designed to harness affordable and easily accessible materials. This deliberate choice not only facilitates cost-effectiveness but also ensures that the educational benefits of the project are widely accessible, aligning with the overarching goal of democratizing STEM learning resources.

- AMoRPH and its Applicability to High School Students:

  AMoRPH, grounded in the law of cosines, is specifically tailored for high school students. This deliberate alignment with the high school curriculum ensures that the educational content is contextually relevant and age-appropriate. By integrating mathematical principles into the learning experience, AMoRPH contributes to a nuanced understanding of STEM subjects.

- Educational Toolkit for STEM Learning:

  The project, collectively, functions as an educational toolkit meticulously curated for STEM learning. By encompassing diverse components and aspects, it offers a comprehensive learning experience. This multifaceted toolkit is poised to enrich the educational landscape by providing students with a holistic understanding of STEM concepts and applications.

7.2.2 Mentoring High School Students

The project assumes a mentoring role in cultivating the skills and knowledge of high school students across various domains within STEM education. This mentorship extends to the following key areas:

- Robot Design:

  Through practical engagement and hands-on activities, the project imparts valuable insights into the intricacies of robot design. This experiential learning approach not only fosters creativity but also cultivates a deeper understanding of the engineering principles governing robotic systems.

- Manufacturing Processes:

  The mentoring initiative encompasses the elucidation of manufacturing processes relevant to STEM domains. By familiarizing high school students with these processes, the project instills a practical understanding of how theoretical concepts manifest in real-world applications, fostering a bridge between theory and practice.

- Modeling Techniques:

  The project serves as a platform for mentoring students in advanced modeling techniques. This facet of the mentorship program contributes to the development of analytical and problem-solving skills, essential for success in STEM disciplines.

- Programming Skills:

  Acknowledging the pivotal role of programming in contemporary STEM fields, the project dedicates attention to mentoring high school students in programming skills. This equips students with the computational prowess necessary for navigating the evolving landscape of STEM-related technologies.

In summation, the project emerges as a multifaceted educational endeavor with a dual emphasis on accessible educational resources and targeted mentorship, thus positioning itself as a valuable asset in the realm of STEM education.

7.3 Ori-Vent

The global impact of the COVID-19 pandemic, resulting from the SARS-CoV-2 virus, has been of significant magnitude, as evidenced by the reporting of 414,179 confirmed cases and 18,440 deaths by March 2020 [50]. To mitigate the risk of transmission, the implementation of social distancing measures, particularly in healthcare settings, has been deemed necessary [51]. The expeditious and accurate diagnosis of the disease through laboratory means assumes great importance in the realm of disease control and the assurance of patient well-being [52]. Consequently, the dissemination of knowledge pertaining to the disease and its management holds pivotal importance in our collective efforts to contain the pandemic [53].

During the COVID-19 pandemic, our attention temporarily shifted away from other tasks to modify the origami module, leading to the development of the Ori-Vent [12]. This initiative was driven by the global shortage of ventilators at that time. Our main goal was to create a mechanism that was simple to manufacture, easily portable due to its foldable nature, and made from commonly available materials. The use of 3D printing was limited to only a few components, and even this process was straightforward. We conducted an initial investigation that included examinations of both the manufacturing process and the performance of the device. Subsequently, we made all the production methods openly accessible. Although the pandemic has since abated, our intention is for this concept to continue contributing to worldwide safety.

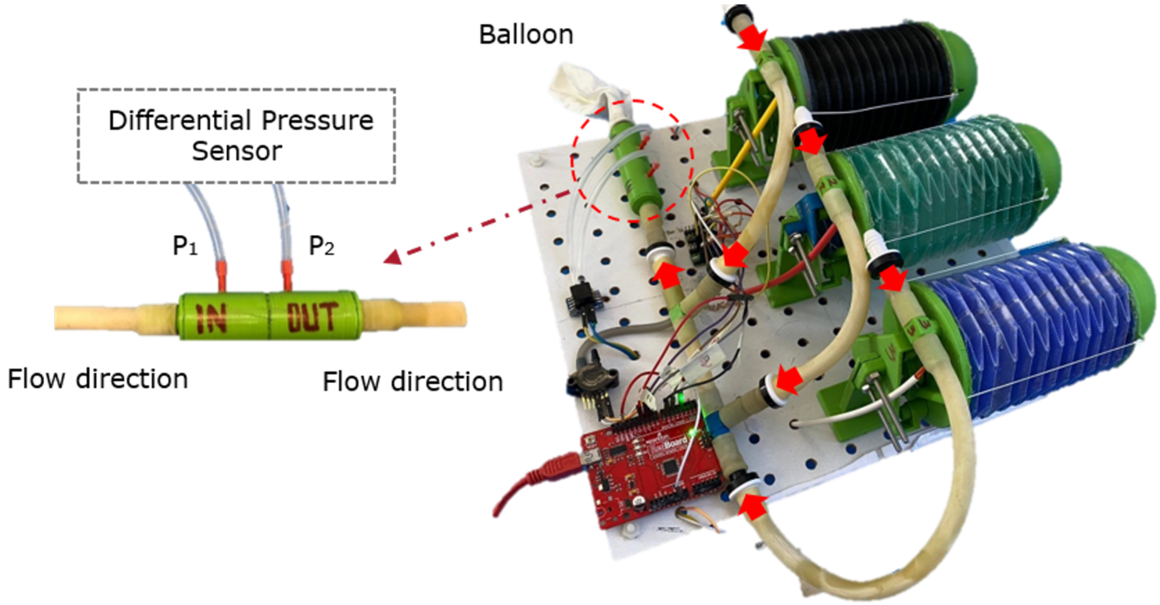


Figure 7.2: Prototypes of a low-cost ventilator. [12]

7.3.1 Requirements

The primary prerequisites for our novel ventilator design encompass affordability, accessibility, and functionality. Our design must employ cost-effective materials and straightforward methods of production. Furthermore, it should not rely on any medical components that may experience scarcity or divert resources from their primary purposes, such as bag valve masks (BVMs). Additionally, the design should be compact and portable, devoid of any dependence on a hospital's infrastructure or bulky, inflexible machinery. To actualize a fully operational ventilator, we aspire to fulfill the subsequent criteria [54]:

- Capable of functioning for at least 48 hours
- Have a connection method to allow an adjustable O_2 source to be connected
- Capable of providing a respiratory rate of 12-40 breaths per minute (BPM)
- Capable of providing 5-24 cm-H_2O of positive end expiratory pressure (PEEP)
- Capable of delivering tidal volumes range 160-1500 cc with every breath
- Capable of limiting the peak inspiratory pressure to 18-40 cm-H_2O
- Powered using 120 V or lower (ideally with a 12 V car battery)

7.3.2 Related Work

Due to the global COVID-19 pandemic, the topic of alternative ventilator design has emerged as a pressing concern, attracting the attention of numerous organizations and laboratories. In response, certain organizations have promptly released open-source ventilator projects, aiming to promptly save lives. For instance, the team from the University of Florida is presently focusing their efforts on developing a bellows-based ventilator [55]. MIT's E-VENT project utilizes a linkage mechanism to compress a bag valve mask (BVM) as the primary pressure source [56]. Another affordable open-source ventilator design employs vacuum pressure to generate a differential [57]. In contrast, our objective is to create a portable design that utilizes readily available materials and can be easily integrated into the existing stockpile of manual and automated ventilation machines. Furthermore, our design facilitates construction from scratch using commonly accessible off-the-shelf resources.

Origami structures, which encompass intricate, folded two-dimensional patterns, have gained popularity as actuator designs due to their advantageous characteristics such as low weight, precise control, and cost-effectiveness in manufacturing. One existing approach pertains to the attributes and functionalities of origami springs and bellows, as emphasized in studies by Morgan et al. [58] and Butler et al. [59] respectively. Additionally, there is a growing body of research exploring the various applications of origami structures, as elaborated upon in the work of Santoso [33].

7.3.3 Ori-Vent

Our proposed ventilator design takes advantage of our previous advancements in origami continuum manipulators. The ventilator is constructed using a module crease pattern and consists of a flexible bellows that is capable of maintaining an airtight seal. This bellows can be created using either folded plastic or NinjaFlex, a pliable material that can be 3D printed.

The functionality of the ventilator is achieved through the incorporation of a single L-Bend geared DC Motor with a gear ratio of 224:1, which is installed within the structure. This motor drives a cable that is connected to the opposite end of the bellows. By winding the cable around a spool, the bellows contracts, allowing the system to fulfill its intended purpose of ventilating air. The cables are pulled by rotating the motor in a clockwise direction, while they are released by rotating it counterclockwise.

The accompanying figure, labeled Fig. 7.3, showcases the prototypes of the proposed design.

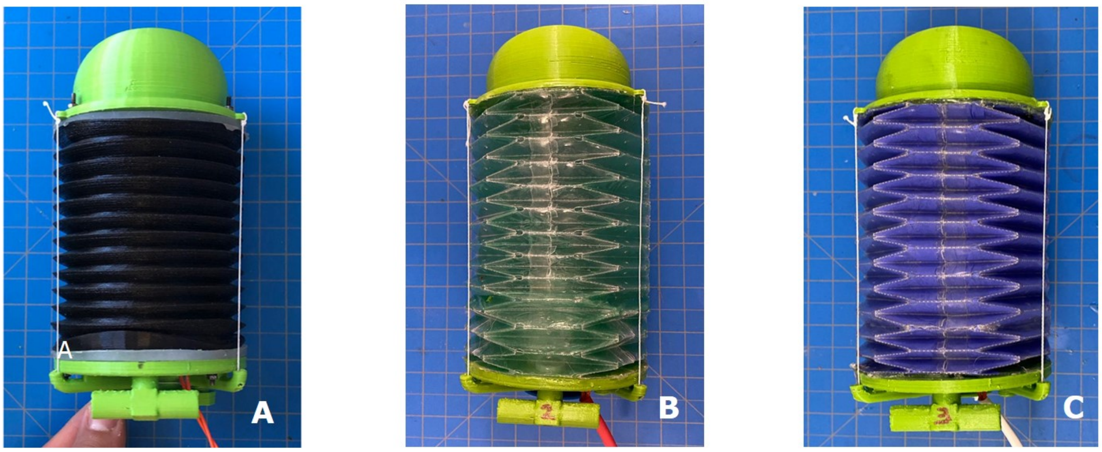


Figure 7.3: Prototypes of a low-cost ventilator. (A) 3D Printed NinjaFlex Bellows. (B) Origami Bellows sealed with Stretchlon200 bagging film.(C) Origami Bellows sealed with Nitrile Latex.

The Ori-Vent project has successfully reached the stage of publication, signifying a significant milestone in our journey. In our commitment to fostering collaboration and innovation, we have taken the courageous step of making all our design and manufacturing processes open source, thereby extending an invitation to a diverse community to engage with and contribute to the advancement of this pioneering initiative. Although the practical implementation of Ori-Vent is yet to be realized despite the gradual easing of the pandemic, our unwavering conviction in the transformative impact of this project remains steadfast. We firmly believe that during times of scarcity, Ori-Vent possesses the inherent potential to serve as a catalyst, igniting and stimulating the development of inventive engineering solutions that can effectively address pressing challenges and contribute to the improvement of global healthcare systems.

Bibliography

[1] W. McMahan, B. Jones, and I. Walker, “Design and implementation of a multisection continuum robot: Air-Octor,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005, pp. 2578–2585.

[2] M. Pritts and C. Rahn, “Design of an artificial muscle continuum robot,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, vol. 5, Apr. 2004, pp. 4742–4746 Vol.5.

[3] “Bionic Handling Assistant | Festo USA,” https://www.festo.com/us/en/e/about-festo/research-and-development/bionic-learning-network/highlights-from-2010-to-2012/bionic-handling-assistant-id_33759/.

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

[5] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, “A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm,” *Bioinspir. Biomim.*, vol. 7, no. 2, p. 025006, Jun. 2012.

[6] P. Qi, C. Qiu, H. Liu, J. S. Dai, L. D. Seneviratne, and K. Althoefer, “A Novel Continuum Manipulator Design Using Serially Connected Double-Layer Planar Springs,” *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 3, pp. 1281–1292, Jun. 2016.

[7] K. Zhang, C. Qiu, and J. S. Dai, “An Extensible Continuum Robot With Integrated Origami Parallel Modules,” *Journal of Mechanisms and Robotics*, vol. 8, no. 031010, Mar. 2016.

[8] J. Santoso and C. D. Onal, “An Origami Continuum Robot Capable of Precise Motion Through Torsionally Stiff Body and Smooth Inverse Kinematics,” *Soft Robotics*, Jul. 2020.

[9] 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, Nov. 2010.

[10] W. Zhang, Z. Yang, T. Dong, and K. Xu, “FABRIKc: An Efficient Iterative Inverse Kinematics Solver for Continuum Robots,” in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2018, pp. 346–352.

[11] S.-S. Chiang, H. Yang, E. Skorina, and C. D. Onal, “SLInKi: State Lattice based Inverse Kinematics - A Fast, Accurate, and Flexible IK Solver for Soft Continuum Robot Manipulators,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, Aug. 2021, pp. 1871–1877.

[12] A. Aihaitijiang, S.-S. Chiang, E. H. Skorina, and C. D. Onal, “Ori-Vent: Design and Prototyping of Accessible and Portable Origami-Inspired Ventilators,” in *Experimental Robotics*, ser. Springer Proceedings in Advanced Robotics, B. Siciliano, C. Laschi, and O. Khatib, Eds. Cham: Springer International Publishing, 2021, pp. 126–136.

[13] G. Robinson and J. B. C. Davies, “Continuum robots - a state of the art,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, 1999, pp. 2849–2854 vol.4.

[14] I. D. Walker, “Continuous Backbone “Continuum” Robot Manipulators,” *International Scholarly Research Notices*, vol. 2013, p. e726506, Jul. 2013.

[15] S. Kolachalama and S. Lakshmanan, “Continuum Robots for Manipulation Applications: A Survey,” *Journal of Robotics*, vol. 2020, pp. 1–19, Jul. 2020.

[16] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. Walker, “Closed-Form Inverse Kinematics for Continuum Manipulators,” *Advanced Robotics*, vol. 23, no. 15, pp. 2077–2091, Jan. 2009.

[17] R. J. Webster, J. P. Swensen, J. M. Romano, and N. J. Cowan, “Closed-Form Differential Kinematics for Concentric-Tube Continuum Robots with Application to Visual Servoing,” in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Berlin, Heidelberg: Springer, 2009, pp. 485–494.

[18] P. Sears and P. E. Dupont, “Inverse Kinematics of Concentric Tube Steerable Needles,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 1887–1892.

[19] B. A. Jones and I. D. Walker, “Kinematics for multisection continuum robots,” *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, Feb. 2006.

[20] A. Aristidou and J. Lasenby, “Inverse Kinematics: A review of existing techniques and introduction of a new fast iterative solver,” *Technical Report F-INFENG/TR. 632, CUED*, p. 74, 2009.

[21] G. S. Chirikjian, “Inverse Kinematics of Binary Manipulators Using a Continuum Model,” *Journal of Intelligent and Robotic Systems*, vol. 19, no. 1, pp. 5–22, May 1997.

[22] J. Lai, K. Huang, and H. K. Chu, “A Learning-based Inverse Kinematics Solver for a Multi-Segment Continuum Robot in Robot-Independent Mapping,” in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2019, pp. 576–582.

[23] S. Neppalli, M. Csencsits, B. Jones, and I. Walker, “A geometrical approach to inverse kinematics for continuum manipulators,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice: IEEE, Sep. 2008, pp. 3565–3570.

[24] A. Aristidou and J. Lasenby, “FABRIK: A fast, iterative solver for the Inverse Kinematics problem,” *Graphical Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011.

[25] X. Wang, Y. Ding, C. Zhou, and K. Xu, “FABRIKc Delta: Extending the FABRIKc Algorithm to Solve the Forward Kinematics of a Continuum Delta Robot,” in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2022, pp. 2009–2014.

[26] D. Kolpashchikov, O. Gerget, and V. Danilov, “FABRIKx: Tackling the Inverse Kinematics Problem of Continuum Robots with Variable Curvature,” *Robotics*, vol. 11, no. 6, p. 128, Dec. 2022.

[27] H. Wu, J. Yu, J. Pan, G. Li, and X. Pei, “CRRIK: A Fast Heuristic Algorithm for the Inverse Kinematics of Continuum Robot,” *J Intell Robot Syst*, vol. 105, no. 3, p. 55, Jun. 2022.

[28] H. J. Zhang, M. Giamou, F. Marić, J. Kelly, and J. Burgner-Kahrs, “CIDGIKc: Distance-Geometric Inverse Kinematics for Continuum Robots,” *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7679–7686, Nov. 2023.

[29] J. Xiao and R. Vatcha, “Real-time adaptive motion planning for a continuum manipulator,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 5919–5926.

[30] J. Li and J. Xiao, “A General Formulation and Approach To Constrained, Continuum Manipulation,” *Advanced Robotics*, vol. 29, no. 13, pp. 889–899, Jul. 2015.

[31] M. Pivtoraiko and A. Kelly, “Generating near minimal spanning control sets for constrained motion planning in discrete state spaces,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005, pp. 3231–3237.

[32] M. Likhachev and D. Ferguson, “Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles,” *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, Aug. 2009.

[33] J. Santoso, E. H. Skorina, M. Luo, R. Yan, and C. D. Onal, “Design and analysis of an origami continuum manipulation module with torsional strength,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2098–2104.

[34] A. Aristidou, Y. Chrysanthou, and J. Lasenby, “Extending FABRIK with Model Constraints,” *Comp. Anim. Virtual Worlds*, vol. 27, no. 1, pp. 35–57, Jan. 2016.

[35] M. Hillman, K. Hagan, S. Hagan, J. Jepson, and R. Orpwood, “The Weston Wheelchair Mounted assistive robot - the design story,” *Robotica*, vol. 20, no. 2, pp. 125–132, Mar. 2002.

[36] D. A. Sinyukov, R. Li, N. W. Otero, R. Gao, and T. Padir, “Augmenting a voice and facial expression control of a robotic wheelchair with assistive navigation,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. San Diego, CA, USA: IEEE, Oct. 2014, pp. 1088–1094.

[37] C.-h. Kuo and H. Chen, “Human-Oriented Design of Autonomous Navigation Assisted Robotic Wheelchair for Indoor Environments,” in *2006 IEEE International Conference on Mechatronics.* Budapest, Hungary: IEEE, Jul. 2006, pp. 230–235.

[38] M. Mohamadzadeh, V. Rashedi, M. Hashemi, and V. Borhaninejad, “Relationship Between Activities of Daily Living and Depression in Older Adults,” *Iranian Journal of Ageing*, vol. 15, no. 2, pp. 200–211, Jul. 2020.

[39] H. A. Haghgoo, E. S. Pazuki, A. S. Hosseini, and M. Rassafiani, “Depression, activities of daily living and quality of life in patients with stroke,” *Journal of the Neurological Sciences*, vol. 328, no. 1-2, pp. 87–91, May 2013.

[40] “Robotic arm,” https://assistive.kinovarobotics.com/product/jaco-robotic-arm.

[41] F. Routhier, P. Archambault, M. L. C. Cyr, V. Maheu, M. Lemay, and I. Gélinas, “Benefits of JACO Robotic Arm on Independent Living and Social Participation : An Exploratory Study,” in *RESNA Annual Conference*, 2014.

[42] V. Maheu, J. Frappier, P. S. Archambault, and F. Routhier, “Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities,” in *2011 IEEE International Conference on Rehabilitation Robotics.* Zurich: IEEE, Jun. 2011, pp. 1–5.

[43] M. Beaudoin, J. Lettre, F. Routhier, P. S. Archambault, M. Lemay, and I. Gélinas, “Long-term use of the JACO robotic arm: A case series,” *Disability and Rehabilitation: Assistive Technology*, vol. 14, no. 3, pp. 267–275, Apr. 2019.

[44] Z. Pei and Y. Nie, “Educational Robots: Classification, Characteristics, Application Areas and Problems,” in *2018 Seventh International Conference of Educational Innovation through Technology (EITT)*, Dec. 2018, pp. 57–62.

[45] O. Mubin, C. J. Stevens, S. Shahid, A. A. Mahmud, and J.-J. Dong, "A Review of The Applicability of Robots in Education," *Technology for Education and Learning*, vol. 1, no. 1, 2013.

[46] F. Tuluri, "STEM Education by Exploring Robotics," in *Robotics in STEM Education: Redesigning the Learning Experience*, M. S. Khine, Ed. Cham: Springer International Publishing, 2017, pp. 195–209.

[47] P. Plaza, E. Sancristobal, G. Fernandez, M. Castro, and C. Pérez, "Collaborative robotic educational tool based on programmable logic and Arduino," in *2016 Technologies Applied to Electronics Teaching (TAEE)*, Jun. 2016, pp. 1–8.

[48] A. Agrawal and N. Gupta, "Soft Robotics in Education," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).* Noida, India: IEEE, Oct. 2022, pp. 1–5.

[49] T. Lobato de Souza and L. Sato Elisiario, "Educational Robotics Teaching with Arduino and 3D Print Based on Stem Projects," in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, Oct. 2019, pp. 407–410.

[50] S. Baloch, M. A. Baloch, T. Zheng, and X. Pei, "The Coronavirus Disease 2019 (COVID-19) Pandemic," *Tohoku J. Exp. Med.*, vol. 250, no. 4, pp. 271–278, 2020.

[51] S. E. Pambuccian, "The COVID-19 pandemic: Implications for the cytology laboratory," *Journal of the American Society of Cytopathology*, vol. 9, no. 3, pp. 202–211, May 2020.

[52] J. L. D. Pizzol, V. P. da Hora, A. J. Reis, J. Vianna, I. Ramis, A. von Groll, and P. A. da Silva, "Laboratory diagnosis for Covid-19: A mini-review," *Rev Soc Bras Med Trop*, vol. 53, p. e20200451, 2020.

[53] F. Begum, “Covid19 A Pandemic of 2020 – A Review,” *International Journal of Trend in Scientific Research and Development*, May 2020.

[54] “DIY Hack-a-Vent Innovation Challenge : North Carolina Military Business Center,” http://www.ncmbc.us/announcements/diy-hack-a-vent-innovation-challenge/.

[55] “Open-Source Ventilator Project,” https://simulation.health.ufl.edu/technology-development/open-source-ventilator-project/.

[56] M. E. V. P. , “MIT E-VENT | Emergency ventilator design toolbox,” https://e-vent.mit.edu/.

[57] J. Lee, “Low-Cost Open-Source Ventilator-ish Device or PAPR,” https://github.com/jcl5m1/ventilator, Jun. 2020.

[58] J. Morgan, S. P. Magleby, and L. L. Howell, “An Approach to Designing Origami-Adapted Aerospace Mechanisms,” *J. Mech. Des*, vol. 138, no. 5, May 2016.

[59] J. Butler, J. Morgan, N. Pehrson, K. Tolman, T. Bateman, S. P. Magleby, and L. L. Howell, “Highly Compressible Origami Bellows for Harsh Environments,” in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* American Society of Mechanical Engineers Digital Collection, Dec. 2016.