UNIVERSITY OF CALIFORNIA

Los Angeles

Exploration of the Synergy between Computational Mechanics and Robotics for Slender Structures

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Dezhong Tong

2023

ABSTRACT OF THE DISSERTATION

Exploration of the Synergy between Computational Mechanics and Robotics for Slender Structures

by

Dezhong Tong
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2023
Professor Mohammed Khalid Jawed, Chair

Slender structures, widely found from natural environments (e.g., tendrils) to engineering applications (e.g., flexible electronics), frequently experience geometrically nonlinear deformations and substantial topological changes when exposed to simple boundary conditions or modest external stimuli. On one hand, the nonlinear dynamics of slender structures present considerable challenges for the automated manipulation of these structures by robots. On the other hand, the automated interactions between robots and such structures also open up opportunities to enhance our understanding of the mechanics governing slender structures. This dissertation focuses on the synergy between computational mechanics and robotics for the manipulation and study of slender structures. Specifically, it delves into discrete differential geometry (DDG)-based simulations, an emerging field in computational mechanics, to develop a comprehensive sim2Real manipulation framework for generating task-oriented deformable manipulation strategies. Moreover, we conduct automated experiments to gain valuable insights into the behavior of slender structures. Our contributions can be categorized into three main areas:

First, we develop a penalty-energy-based method and combine it with Kirchoff rod's theory to simulate rod assemblies with frictional contact responses. Our simulation method is validated, demonstrating its robustness, accuracy, and efficiency across diverse scenarios. These scenarios include modeling flagella bundling, a significant biological phenomenon for bacterial navigation, as well as tying knots. These numerical validations underscore the potential of our approach as a significant step toward the ultimate goal of a computational framework for sim2real manipulation tasks. We then combine our numerical framework with desktop experiments to investigate the mechanics of various types of knots.

Second, we combine DDG-based simulations, scaling analysis, and machine learning to develop a sim2Real framework for various deformable manipulation tasks, including paper folding and the deployment of deformable linear objects onto rigid substrates. Our sim2Real framework harnesses the precision of physical simulations, the rapid inference capabilities of neural networks, and the enhanced adaptability conferred by scaling analysis. This synergy yields robust, accurate, and efficient solutions for these manipulation tasks. In the paper folding task, a physics-informed model is learned using scaled simulation data, enabling the creation of a model predictive control system for precise paper folding. We validate the effectiveness of this physics-based approach through extensive robotic experiments. In addition, we construct a physics-informed manipulation policy within the same framework for the deployment task. This policy proves to be robust, accurate, and efficient in controlling the shape of various deformable linear objects during deployments. Furthermore, we demonstrate the potential of this deployment scheme in various engineering applications including cable management and knot tying.

Finally, we delve into the application of automation science to explore the nonlinear mechanics of slender structures. Traditional experimental platforms (e.g., optical platforms) struggle to systematically capture the numerous boundary conditions and corresponding equilibriums of slender structures. To address this challenge, we've designed a robotic system for automated experiments. This system allows us to investigate one of the fundamental problems in solid mechanics: the buckling of an elastic rod with a helical centerline. We answer this problem with a combination of theoretical analysis, numerical simulation, and automated robotic experiments. Then, significant advances are made in understanding this phenomenon, uncovering different buckling types within this system, including continuous buckling and snap buckling. Given the distinct behaviors of these two types of buckling, our exploration is particularly meaningful in demonstrating how various buckling can be triggered within a single system. Our automated robotic experiments highlight the potential of robotic technology in advancing our understanding of mechanics through intelligent interactions with the physical world.

The dissertation of Dezhong Tong is approved.

Artur R. Davoyan

Veronica J. Santos

Pei-Yu Chiou

Mohammed Khalid Jawed, Committee Chair

University of California, Los Angeles

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my advisor, Prof. Mohammed Khalid Jawed, whose support and guidance have been the cornerstone of my Ph.D. journey. His patience, motivation, and dedication have not only shaped my academic growth but also inspired me to aspire for excellence in my research.

I would also like to thank my committee members, Prof. Veronica Santos, Prof. Artur Davoyan, and Prof. Peiyu "Eric" Chiou, for their invaluable insights and advice throughout my academic journey. Their expertise and feedback have enriched my research and contributed to its depth and quality.

Then, I would like to extend my thanks to Dr. Andrew Choi, Prof. Jungseock Joo, Prof. Andy Borum, Prof. Bashir Khoda, Prof. Demetri Terzopoulos, Prof. Weicheng Huang, Prof. Longhui Qin, and all members of the SCI lab, for all of their continuous support.

Finally, my deepest gratitude goes to my family and friends, whose unwavering love, deep care, steadfast support, and profound understanding have sustained me throughout this challenging journey. Your encouragement and belief in me have been my most significant source of strength, and I am immensely thankful for your presence in my life.

We also gratefully acknowledge financial support from the National Science Foundation (grant numbers 1925360, 2101751, and 2047663).

VITA

2014–2018	B.S. in Mechanical Engineering, Shanghai Jiao Tong University, China.
2018–2019	M.S. in Mechanical Engineering, University of California, Los Angeles, Los Angeles, California
2019–2023	Ph.D. candidate in Mechanical Engineering, University of California, Los Angeles, Los Angeles, California

PUBLICATIONS

[1] **Tong, D**, Borum, A., Jawed, M.K., 2021. “Automated stability testing of elastic rods with helical centerlines using a robotic system.” *IEEE Robotics and Automation Letter*, 7(2), 1126-1133

[2] **Tong, D***, Choi*, A., Joo, J., and Jawed, M.K., 2022. “A Fully Implicit Method for Robust Frictional Contact Handling in Elastic Rods.” *Extreme Mechanics Letter*, 58, p.101924.(∗ denotes equal contribution)

[3] **Tong, D**, Choi, A., Joo, J., Andy, B., and Jawed, M.K., 2022. “Snap Buckling in Overhand Knots.” *Journal of Applied Mechanics*, pp.1-23.

[4] Choi, A.*, **Tong, D***, Terzopoulos, D., Joo, J. and Jawed, M.K., 2023. “Deep Learning of Force Manifolds from the Simulated Physics of Robotic Paper Folding.” Submitted to *IEEE Transactions on Automation Science and Engineering* (∗ denotes equal contribution)

[5] **Tong, D**, Choi, A., Longhui, Q., Weicheng, H, and Jawed, M.K., 2023. "Sim2Real Physically Informed Neural Controllers for Robotic Deployment of Deformable Linear Objects." Resubmitted to *International Journal of Robotics and Research* (Under Review)

[6] **Tong, D**, Ibrahim Khalil, Silva. M., Q., Khoda, B., and Jawed, M.K., 2023. "Mechanical Response of Fisherman's Knots during Tightening." Submitted to *Journal of Applied Mechanics*

[7] **Tong, D***, Borum, A.*, Jawed, M.K., 2023. "How the helix buckles." Submitted to *The Proceedings of the National Academy of Sciences* ($*$ denotes equal contribution)

[8] A. Choi, **Tong, D**, B. Park, D. Terzopoulos, J. Joo and M. K. Jawed, 2023. "mBEST: Realtime Deformable Linear Object Detection Through Minimal Bending Energy Skeleton Pixel Traversals," *IEEE Robotics and Automation Letters*, 8(8), 4863-4870

[9] Choi, A., **Tong, D**, Jawed, M.K., and Joo, J., 2021. "Implicit contact model for discrete elastic rods in knot tying" *Journal of Applied Mechanics*, 88(5).

[10] Ibrahim Khalil, **Tong, D**, Wang, G., Jawed, M.K., and Khoda, B., 2022. "Systematic Variation of Friction of Rods" *Journal of Applied Mechanics*, 89(11), p.111007.

[11] Lim, S., Du, Y., Lee, Y., Panda, S.K., **Tong, D**, and Jawed, M.K., "Fabrication, control, and modeling of robots inspired by flagella and cilia." *Bioinspiration & Biomimetics*,, 18(1), p.011003.

CHAPTER 1

Introduction

Elastic slender structures are three-dimensional objects characterized by at least one dimension vastly exceeding the others. This can include objects like rods (length $\gg$ width $\sim$ thickness) and plates (length $\sim$ width $\gg$ thickness) [1]. Due to their distinctive geometry, these slender structures find widespread application in many practical scenarios. Some examples include surgical suturing [2], sailing knots [3], packaging [4], and soft robotics [5]. However, due to their unique geometry, these structures usually exhibit large geometrically nonlinear deformation even under moderate external forces. Manipulation of those elastic structures becomes a challenging and attractive problem in robotics.

Many human-engineered systems, e.g., soft robots, knots, and knitted structures, are constructed from thin deformable objects. These entities possess mechanical versatility that is impossible with rigid bodies, endowing them with huge potential in robotics. Nonetheless, acquiring novel manipulation skills to actuate or maneuver these objects is still significant even though the robotic manipulation of rigid objects has made great strides, and mature methods have been created [6]. Therefore, a robust and efficient strategy is required to augment our comprehension of these slender deformable structures. Prior efforts to simulate deformable structures mainly focused on Finite Element Method and voxel-based discretization. Discrete differential geometry (DDG), an emerging mathematical discipline prevalent in the computer graphics community, is well-suited for constructing robust, efficient, and accurate numerical frameworks for modeling elasticity [7]. Building upon existing DDG-based numerical frameworks, we expand their applicability within computational mechanics. This numerical framework is later used to deepen our insight into the nonlinear deformations of various elastic structures with complicated topologies. Our relevant research validates the precision and robustness of the designed DDG-based simulator, simultaneously offering profound insights into the behavior of elastic structures with intricate topologies. This effort paves the way for exploring the practical utility of such structures within the realm of robotics.

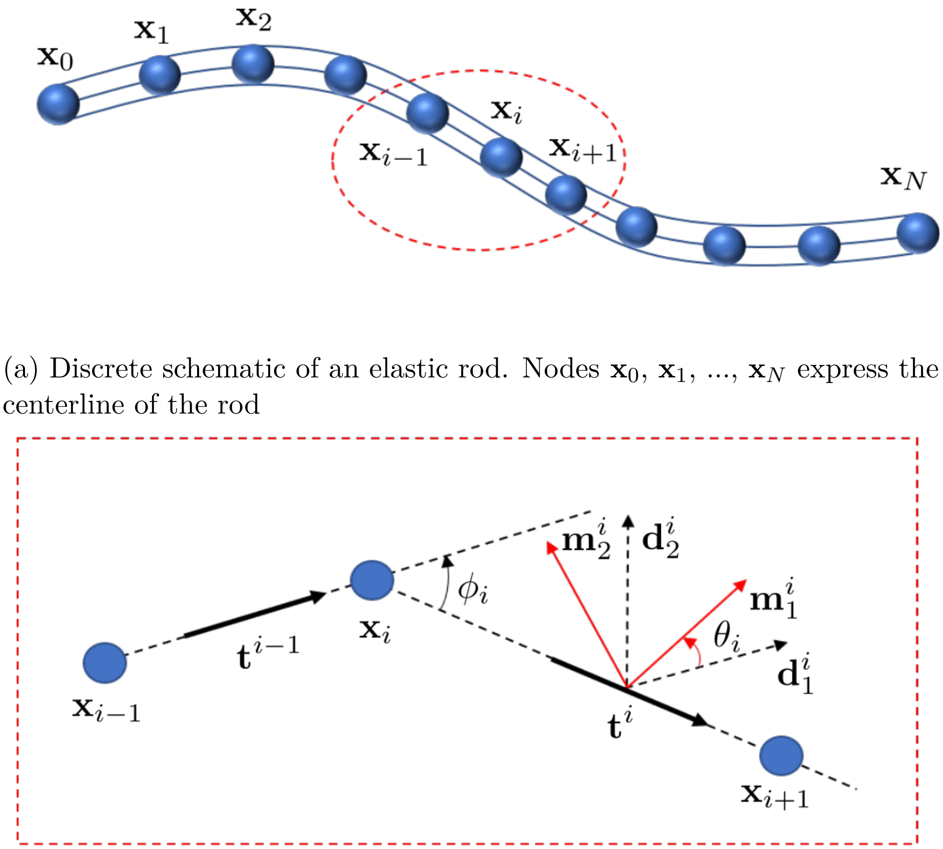
In the robotics community, one of the keys to the robotic manipulation of deformable objects is to construct a mapping from observations of manipulated objects to the robot's action space [8]. This necessitates the development of a reliable model that serves as the robotics knowledge base. Prior efforts usually focus on reproducing this correlation through exhaustive human-made demonstrations, which is usually inefficient and inadequate when faced with myriad potential deformations [9, 10, 11]. Given the robustness and physical precision of our designed DDG-based simulators, this study introduces a novel framework. This framework combines physically accurate simulations, scaling analysis, and machine learning techniques to forge predictive models aimed at advancing deformable manipulation skills. The synergy of those components benefits the combined framework from the physical predictive accuracy of the DDG simulator, compact mapping through scaling analysis, and the high inference speed of the neural network. We later demonstrate the efficacy of such combined frameworks in multiple challenging deformable manipulation tasks including paper folding and shape controlling of deformable linear objects.

In addition to the development of manipulation skills, there exists a parallel fascination among researchers for slender structures with controllable compliant properties. These structures adhere to physical principles dictating their elastic deformations, thereby allowing for interpretable mappings between deformations and applied external forces or boundary conditions. Such mappings have proved valuable for innovative engineering applications. For instance, soft actuators are designed based on the elastic instability inherent in such structures [12], a phenomenon wherein stored elastic energies rapidly convert into kinematic energy. However, deriving such mappings solely through simulations is often challenging. Traditional mechanical experiments usually require extensive manual labor and often fail to encompass the full spectrum of deformation scenarios. Herein, the introduction of robotic systems capitalizes on their automation and dexterity to collect experimental data, enabling an exhaustive exploration of elastic structure mechanics. During my Ph.D. study, I showcased the capabilities of robotics to solve a challenging fundamental solid mechanics problem experimentally. This research trajectory illuminates the potential of integrating robotic technology into AI4sicence research.

1.1 Discrete Elastic Rods (DER)

The origins of rod theory trace back to the 18th century when Euler and Bernoulli proposed simple one-dimensional beam models. Building upon their foundation, Kirchhoff and Cosserat extend and generalize these concepts in the 19th century. Recently, researchers in computer graphics communities developed a fast DDG-based simulator known as Discrete Elastic Rods (DER) [13, 14]. This simulator is adept at capturing the intricate nonlinear deformations inherent in elastic rods. Here, we provide a brief overview of the fundamental formulations of this physically precise DDG-based simulation approach.

As shown in Fig. 1.1a, an elastic rod can be discretized into $N+1$ nodes: $\mathbf{x}_0$, $\mathbf{x}_1$, ..., $\mathbf{x}_N$, and N edges: $\mathbf{e}^0$, $\mathbf{e}^1$, ..., $\mathbf{e}^{N-1}$. These edges are related to nodes as $\mathbf{e}^i = \mathbf{x}_{i+1} - \mathbf{x}_i$ for $0 \leq i \leq N-1$. Hereafter, we denote node-related quantities with subscripts and edge-related quantities with superscripts. Each edge, $\mathbf{e}^i$, is associated with two orthogonal frames: a material frame $[\mathbf{m}_1^i, \mathbf{m}_2^i, \mathbf{t}^i]$ and a reference frame $[\mathbf{d}_1^i, \mathbf{d}_2^i, \mathbf{t}^i]$, as illustrated in Fig. 1.1b. Both frames share the same tangent $\mathbf{t}^i = \mathbf{e}^i / \|\mathbf{e}^i\|$. The reference frames are arbitrarily initialized at initial time $t = 0$s, and their updates at each time step are performed using time parallel transport [14]. The material frame, utilized for evaluating the rod's centerline rotation, is derived by rotating the reference frame along the shared tangent direction $\mathbf{t}^i$ by a rotation



(a) Discrete schematic of an elastic rod. Nodes $\mathbf{x}_0$, $\mathbf{x}_1$, ..., $\mathbf{x}_N$ express the centerline of the rod

(b) A zoomed-in snapshot of two edges showcasing their reference frame, material frame, turning angles, and twist angles.

Figure 1.1: Discrete schematic of a rod and its relevant notations.

angle θ^i, as shown in Fig. 1.1b. Detailed information on the DER algorithm can be found in Ref. [13, 14]. Here, the collective discrete nodal positions $\mathbf{x}_i$ and rotation angles θ^i are employed to construct a degree of freedom (DOF) vector of size $4N + 3$ to represent the rod's configuration:

$$\mathbf{q} = \left[\mathbf{x}_0, \theta^0, \mathbf{x}_1, ..., \mathbf{x}_{N-1}, \theta^{N-1}, \mathbf{x}_N\right]^T. \tag{1.1}$$

where T is the transpose operator.

Based on DER [13, 14], the strains of a deformed rod can be divided into three categories, each corresponding to a distinct type of elastic energy: stretching, bending, and twisting.

Within the DOF vector $\mathbf{q}$, we can outline the elastic strains and energies of the rod to construct equations of motion (EOM). First, the stretching strain of an edge is:

$$\epsilon^i = \frac{\|\mathbf{e}^i\|}{\|\tilde{\mathbf{e}}^i\|} - 1. \tag{1.2}$$

Hereafter, quantities with a $\tilde{(\cdot)}$ indicate the undeformed status, e.g. $\|\tilde{\mathbf{e}}^i\|$ is the undeformed length of the i-th edge. To capture the bending strain, a curvature binormal vector is introduced, representing the misalignment between consecutive edges:

$$(\kappa\mathbf{b})_i = \frac{2\mathbf{e}^{i-1} \times \mathbf{e}^i}{\|\mathbf{e}^{i-1}\|\|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i}. \tag{1.3}$$

The norm of this vector, $\|(\kappa\mathbf{b})_i\| = 2\tan(\phi_i/2)$, represents the curvature magnitude, where turning angle ϕ_i is the angle of rotation between consecutive edges which can be seen in Fig. 1.1b. The material curvatures are then determined by the inner products between the curvature binormal vector and the material frame directors $[\mathbf{m}_1^i, \mathbf{m}_2^i]$:

$$\begin{aligned} \kappa_i^1 &= \frac{1}{2}(\mathbf{m}_2^{i-1} + \mathbf{m}_2^i) \cdot (\kappa\mathbf{b})_i, \\ \kappa_i^2 &= -\frac{1}{2}(\mathbf{m}_1^{i-1} + \mathbf{m}_1^i) \cdot (\kappa\mathbf{b})_i. \end{aligned} \tag{1.4}$$

Finally, the twisting strain of an edge is

$$\tau_i = \theta^i - \theta^{i-1} + \Delta\tau_i^{\text{ref}}, \tag{1.5}$$

where $\Delta\tau_i^{\text{ref}}$ is the discretely integrated reference twist, which can be obtained through the parallel transport between two neighboring reference frames.

Given the elastic strains, the expressions for the elastic energies of a rod can be formulated as

$$E_s = \frac{1}{2} \sum_{i=0}^{N-1} EA(\epsilon^i)^2 \|\hat{\mathbf{e}}^i\|, \tag{1.6a}$$

$$E_b = \frac{1}{2} \sum_{i=1}^{N-1} \frac{1}{\|e^i\|} \left[EI_1(\kappa_i^1 - \hat{\kappa}_i^2)^2 + EI_2(\kappa_i^2 - \hat{\kappa}_i^2)^2 \right], \tag{1.6b}$$

$$E_t = \frac{1}{2} \sum_{i=1}^{N-1} \frac{1}{\|e^i\|} GJ(\tau_i - \hat{\tau}_i)^2, \tag{1.6c}$$

where A is the rod's cross-sectional area; E is Young's modulus; G is the shear modulus; J is the geometrical factor of the torsional rigidity which is equal to the second polar moment of the area along the tangent $\mathbf{t}^i$ when the cross-section is circular, and I_1 and I_2 are the second moments of the area along the material frames $\mathbf{m}_1^i$ and $\mathbf{m}_2^i$, respectively.

At each degree of freedom q_i, the elastic forces (associated with the node-relevant quantities) and elastic moments (associated with edge-relevant quantities) are:

$$F_i^{\text{int}} = -\frac{\partial (E_s + E_b + E_t)}{\partial q_i}, \tag{1.7}$$

where j is an integer from 0 to $4N + 3$.

To numerically solve the equations of motion and update the DOF vector $\mathbf{q}$ and its velocity $\mathbf{v} = \dot{\mathbf{q}}$ from time step t_k to $t_{k+1} = t_k + \Delta t$, implicit Euler integration is employed:

$$\mathcal{R}(\mathbf{q}) \equiv \frac{\mathbb{M}}{\Delta t} \left(\frac{\mathbf{q}(t_{k+1}) - \mathbf{q}(t_k)}{\Delta t} - \dot{\mathbf{q}}(t_k) \right) - \mathbf{F}^{\text{int}} - \mathbf{F}^{\text{ext}} = 0, \tag{1.8a}$$

$$\dot{\mathbf{q}}(t_{k+1}) = \frac{\mathbf{q}(t_{k+1}) - \mathbf{q}(t_k)}{\Delta t}, \tag{1.8b}$$

where $\mathbb{M}$ is a square lumped mass matrix of size $4N + 3$; $\mathbf{F}^{\text{int}}$ is a $(4N + 3) \times 1$ elastic force vector (from Eq. 1.7), and $\mathbf{F}^{\text{ext}}$ is a $(4N + 3) \times 1$ external force vector. The $\dot{()}$ operator represents the derivative of a quantity with respect to time, i.e., $\dot{\mathbf{q}}(t_k)$ is the velocity vector at time t_k. Note that the subscript in Eq. 1.8 is the time stamp. We can solve Eq. 1.8 with

Newton's method for the nonlinear dynamics of a rod over time accurately.

In summary, DER is formulated based on the classical Kirchhoff theory and discrete differential geometry, e.g., the curvatures of a rod centerline give the elastic bending energies, and the internal elastic forces required by equations of motion are derived from the energies in a discrete format. Given the accurate treatments of elasticity, DER's physical accuracy is validated in many prior works. Some instances include coiling of elastic rods on a moving belt [7], buckling of rotating flagella in the fluid [15], instability of rods [16] and plates [17], analyzing the motions of soft robots [12], and etc. Our research commences within the well-established DER simulation and advances: in Ch. 2, we develop a novel implicit frictional contact framework, augmenting DER algorithm for simulating rod assemblies in contact accurately; in Ch. 3, we leverage our augmented DER algorithm to delve into the mechanics in knots; in Ch. 4, we harness insights gathered from DER to develop a learned model for completing a challenging task in robotics – folding paper; in Ch. 5, we construct a robust robotic deployment scheme for controlling the shape of deformable linear objects with the help of DER; in Ch. 6, we use DER simulation to guide a 7 DOF robotic arm to explore the buckling of an elastic rod with helical centerline.

1.2 Outline of the Thesis

This chapter introduces the well-established numerical tool for simulating the nonlinear dynamics of rods that serves as the main research objective of this Thesis. We explore the nonlinear dynamics of rods and propose different avenues to manipulate & study them with robotics. Subsequent chapters present the extensions and applications of the DER method for investigating the synergy between computational mechanics and robotics for slender elastic structures. The main contributions of this thesis are as follows:

Ch. 2 delineates the Implicit Contact Model (IMC), a fully implicit frictional contact model we designed. IMC is designed based on Coulomb's friction law, capable of enforcing non-penetration constraints and approximating Coulomb friction ideally for contacting bodies. Given the significance of frictional contact in real-world scenarios, this work is crucial to predict and understand the nonlinear behaviors of slender elastic structures in which frictional contact occurs. We simulate several challenging frictional contact scenarios pertaining to rods such as tying an overhand knot and flagella bundling. The physical accuracy of IMC is validated in knot-tying scenarios, and the computational efficiency of the IMC is proved by comparing IMC to a state-of-the-art frictional contact framework. IMC emerges not only as robust, efficient, and precise in addressing frictional contact, particularly in rod assemblies but also as an indispensable tool in the mechanics study of slender elastic structures. It lays the foundation for exploring robust Sim2Real task-oriented robotic manipulation strategies, recognizing the importance of frictional contact in practical scenarios.

Ch. 3 focuses on the mechanics of knots. Knots are complex geometric configurations formed by slender elastic rods in self-contact. These structures serve a versatile array of applications in our daily lives, ranging from practical (securing shoelaces) to decorative applications (Chinese knotting). Due to the self-contact and complex topology, the knot can have extreme nonlinear behaviors under manipulation. Through a combined study that includes the simulations expounded in Ch. 2 and desktop experiments, we explore various unique tightening behaviors of some fundamental but popular knots in daily lives, including the snap buckling in an overhand knot induced by the simple pulling boundary conditions and the various motion patterns of a tightening fisherman's knots. The finding of knots strengthens our understanding of the knot's nonlinear behaviors and sheds light on handling knotted structures with automation science.

Ch. 4 addresses a challenging deformable manipulation task in robotics – folding paper. Paper proves resistant to manipulation due to its high bending stiffness and slippery surface. In this context, we combine physically accurate simulation, scaling analysis, and machine learning to forge a reduced-order model primed for predicting the paper's various states. The learned model derives its robust physical accuracy from the underpinning physically precise simulations, boasts accelerated inference speeds courtesy of the computational efficiency of the machine learning component, and boasts a generalizable framework rooted in scaling analysis. We proceed to craft a closed-loop control system by harmonizing the acquired model with a perception algorithm. Subsequently, this closed-loop control scheme finds application in a series of robotic experiments, thereby vividly demonstrating the efficacy of our control strategy. Furthermore, we extend our analysis by contrasting our meticulously devised optimal control scheme with a state-of-the-art folding approach [18], effectively showcasing the unequivocal superiority of our proposed scheme.

Chapter 5 states a novel neural controller designed for deploying deformable linear objects (DLOs) onto rigid substrates, following diverse and viable patterns. Deploying DLOs is instrumental for controlling the shape of a DLO in the practical world, e.g., drawing or writing on cakes with icing [19], deploying marine cables [20], depositing carbon nanotubes [21], and melting electrospinning for advanced manufacturing [22]. A meticulous examination of the deployment task is conducted, entailing a comprehensive physical analysis. This analysis enables us to identify the optimal robotic grasp, which is pivotal in mitigating the substantial nonlinear deformations and buckling tendencies inherent in manipulating DLOs. Subsequently, a neural controller is formulated, drawing upon numerical simulation, scaling analysis, and machine learning. The efficacy of our developed controller is validated across a spectrum of deployment tasks, effectively demonstrating its versatility and adaptability. Furthermore, we highlight its potential by realizing two practical applications: cable management and knot tying. These real-world scenarios underscore our proposed neural controller's tangible utility and broad applicability in tangible contexts.

Ch. 6 explores the usage of robotics technology in solid mechanics study. Given the slenderness of rods, it can have a lot of unexplored behaviors, for example, buckling with some specific boundary conditions. Here, we look into a fundamental mechanics problem: when an elastic rod with helical centerline buckles and how it buckles. This problem is especially challenging when we try to do the experimental study since the required boundary conditions should have multiple (6) controllable degrees of freedom and the potential boundary conditions are numerous. Here, we propose to leverage the automation and dexterity of robots to study this problem. We use a combined theoretical analysis and simulations to find that the buckling points of an elastic rod with a helical centerline, and then we design an automated robotic experimental system to study the buckling types of the system. The robotic experiments show high robustness and accuracy in this project, which paved the path for using automation science for studying unknown physics in various systems.

CHAPTER 2

A Fully Implicit Frictional Contact Framework

Accurate modeling of frictional contact is crucial for simulating the assembly of rod-like structures in real-world applications, including knots, hairs, flagella, and more. Due to their high geometric nonlinearity and elasticity, achieving accurate rod-on-rod contact remains challenging for researchers in both computational mechanics and computer graphics. Typically, frictional contact is treated as a constraint in a system's equations of motion. These constraints are usually computed independently at each time step during dynamic simulations, potentially slowing down the simulation and leading to numerical convergence issues. In this work, we propose a fully implicit penalty-energy-based frictional contact method known as the Implicit Contact Model (IMC). This approach efficiently and robustly captures accurate frictional contact responses. We demonstrate the performance of our algorithm in two distinct scenarios: flagella bundling and knot tying. Flagella bundling is a novel and complex contact phenomenon in biology, inspiring potential engineering applications in soft robotics [23]. We provide a side-by-side comparison with the Incremental Potential Contact (IPC), a state-of-the-art contact handling algorithm [24], to illustrate that IMC achieves similar physical accuracy while converging faster. Regarding knot tying, we simulate the tightening process of an overhand knot and validate our approach's physical accuracy against knot theory as outlined in Audoly et al. [25] and Jawed et al. [26]. The simulator we develop is employed in Ch. 3 to explore the mechanics of knots.

The underlying motivation is in § 2.1. The numerical framework is detailed in § 2.2. Then, the results and discussions are stated in § 2.4. Next, we give the conclusions in § 2.6.

The content of this chapter is from Ref. [27].

2.1 Motivation

Throughout the course of human history, the significance of flexible filamentary structures has been deeply ingrained in human society, serving multiple purposes ranging from fastening and sailing to climbing, weaving, and hunting. As our comprehension of material properties advanced, so too did our capacity to engineer rods with heightened material characteristics, encompassing qualities like flexibility, strength, and resilience. This evolution has intensified the need to investigate and gain a more profound understanding of filaments' intricate mechanics.

However, conducting real-world experiments for such investigations often proves tedious and costly. This implies the increasing demand for accurate physics-based numerical simulations that can accurately replicate real-world behavior.

Discrete Elastic Rods (DER), which has emerged from the computer graphics community [13, 14], excels in capturing the nonlinear dynamics of rod-like structures. Several prior studies have sought to unravel the mechanics of rod-like structures through DER, encompassing areas such as rod deployment [7, 28, 15], elastic gridshells [29, 30, 31], and the propulsion of bacterial flagella [32, 33, 34]. However, a comprehensive understanding of frictional contact handling remains an area that requires further exploration.

Frictional contact formulations are usually diverse and based on the scaling of the system and/or the physical scenario. This chapter focuses on Coulomb friction, a suitable approximation of dry friction. Coulomb friction emerges as a standard in non-cohesive contact due to its simplicity and empirical accuracy. It finds extensive application across various engineering domains, including contact in elastic structures, simulations involving granular media, and more.

The mainstream of creating a frictional contact handling framework can be divided into three categories: impulse methods, constraint-based optimization methods, and penalty energy methods. Impulse methods determine contact forces based on the required impulse to prevent rod segments from penetrating, with an example being the impulse force model by Ref. [35]. However, these methods can result in unrealistic visual jittering during simulations with large time steps [36]. Therefore, impulse methods often suffer from a trade-off between physical accuracy and time efficiency.

Constraint-based methods regard frictional contact as a constrained optimization problem. Jean and Moreau [37, 38] implemented convex analysis to propose using unilateral constraints to solve dry friction in granular media initially. Then, Alart and Curnier [39] implemented a Lagrange multiplier for solving constrained-based contact dynamics as a root-finding problem for nonsmooth functions. In computer graphics, Daviet et al. [40] combined an analytical solver with the complementary condition from [39] to capture Coulomb friction in elastic fibers. In Ref. [41], the algorithm from [40] was incorporated with a nonlinear elasticity solver to simulate frictional contacts in assemblies of Discrete Elastic Rods [13, 14]. Based on previous work, Daviet [42] proposed a general constraint-based framework for simulating contact in thin nodal objects. Constraint-based methods treat frictional contact as a constrained optimization problem, which can yield realistic results but is more complex to implement than impulse or penalty methods. An additional drawback of constraint-based methods is that the frictional contact responses usually are treated as independent unknown variables of the system so the computation costs increase because of the expanded system.

The penalty energy method utilizes a formulated artificial contact energy whose gradient can be regarded as the frictional contact responses. Since he contact energy is usually obtained from the deformed configurations of the rod directly, the penalty energy method has a simpler mathematical formulation compared to the constrained-based algorithm. Given the differentiability of the contact energy, a second-order optimization is usually implemented to solve the rod assemblies with frictional contact [24, 36, 3]. These methods have become popular recently since they have shown the capabilities of generating accurate frictional contact responses with the appropriate formulated contact energies. Given the simplicity of the formulation and efficient computation, we decided to develop our frictional contact handling framework with penalty-based energy methods.

In this Chapter, we present the Implicit Contact Model (IMC) as a fully implicit penalty-based contact model integrated into the DER framework. IMC enhances the simulation of rod assemblies in frictional contact, demonstrating robustness, efficiency, and accuracy. To assess the efficacy of IMC, we juxtapose it with the state-of-the-art penalty-energy-based algorithm for managing frictional contacts, known as IPC [24]. Our findings reveal that IMC achieves a comparable level of physical accuracy to IPC while also demonstrating superior computational efficiency. Augmented with IMC, the DER algorithm facilitates advanced mechanical studies, such as intricate structures like knots (as explored in Ch. 3). Moreover, it broadens the horizon for sim-to-real manipulation of deformable objects, a concept elaborated in Chs. 4 and 5.

2.2 Numerical Framework

In this section, we present the details of constructing a penalty-based energy formulation for computing Coulomb friction. Our formulation directly links frictional contact responses with the rod's configuration, enabling us to compute these responses based on the current deformation of the rod.

To describe an edge-to-edge contact pair, we use vector concatenation denoted as $\mathbf{x}_{ij} := (\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_j, \mathbf{x}_{j+1}) \in \mathbb{R}^{12}$, where $|j-i| > 1$ excludes consecutive edges when enforcing contact, as illustrated in Fig. 2.1. We use a cylinder as the contact mesh for a discrete edge in DER (Discrete Elastic Rod). Lumelskey's algorithm [43] calculates the minimum distance Δ between two discrete edges. We denote the set of all valid edge combinations as $\mathcal{X}$. In subsequent equations, we represent the subscriptless $\mathbf{x}$ as an arbitrary edge combination for clarity.

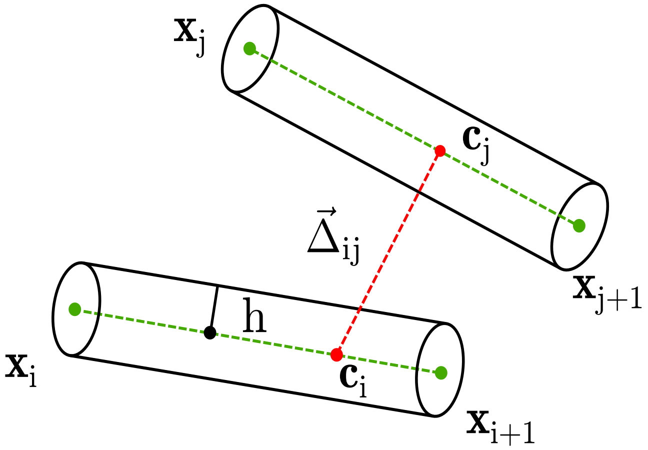


Figure 2.1: Discrete schematic of one contact pair. The green dots showcase the nodes of the edges while the green dashed lines denote the centerlines of the edges. The red dashed line denotes the vector $\vec{\Delta}$ whose norm is the minimum distance Δ between the edges. $\vec{\Delta}$ is connected to edges i and j by $\mathbf{c}_i = \mathbf{x}_i + \beta_i(\mathbf{x}_{i+1} - \mathbf{x}_i)$ and $\mathbf{c}_j = \mathbf{x}_j + \beta_j(\mathbf{x}_{j+1} - \mathbf{x}_j)$ where $\beta_i, \beta_j \in [0, 1]$. As Δ approaches the contact threshold $2h$, repulsive forces increase at an exponential rate, thus enforcing non-penetration.

We design a contact energy function $E(\Delta(\mathbf{x}))$ that increases as the minimum distance Δ between two bodies approaches a contact threshold ($2h$ for our application, with h being the rod radius). The gradient of the contact energy $-k\nabla_{\mathbf{x}} E(\mathbf{x}) \in \mathbb{R}^{12}$ serves as the contact forces, while the Hessian of the contact energy $-k\nabla^2_{\mathbf{x}} E(\mathbf{x}) \in \mathbb{R}^{12\times 12}$ functions as the contact force Jacobian. Here, k is the contact stiffness, which appropriately scales the contact forces to enforce non-penetration. In the following sections, we will detail the formulation of contact energy $E(\Delta(\mathbf{x}))$, the minimum distance between two edges $\Delta(\mathbf{x})$, and address the topic of friction.

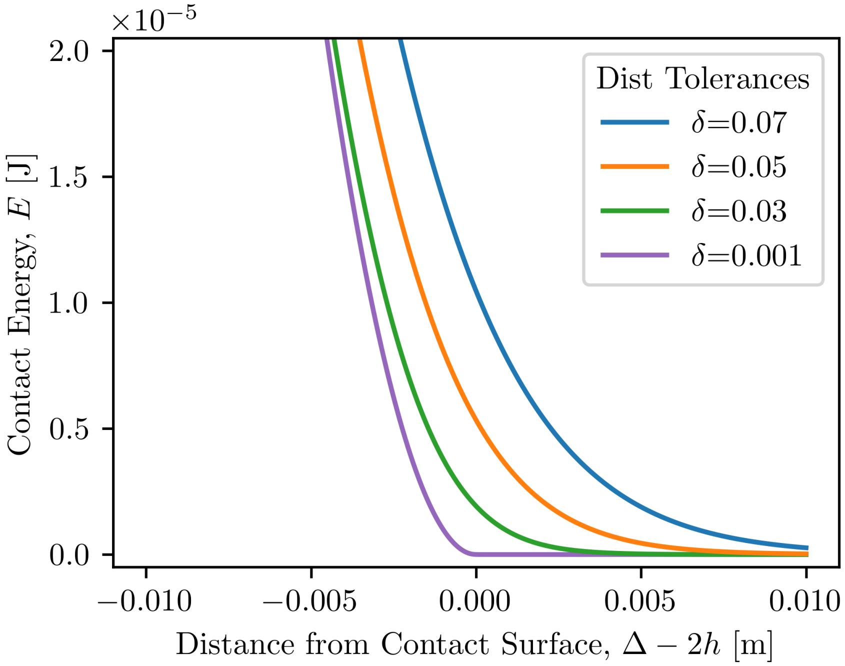


Figure 2.2: Plots for Eq. 2.2 with varying distance tolerance δ values.

2.2.1 Contact Energy Formulation

In an ideal setting, contact energy must satisfy two properties: (1) it equals zero for any distance $\Delta > 2h$ and (2) it becomes non-zero precisely at distance $\Delta = 2h$. These properties can be effectively described using a Heaviside step function. However, this function is non-smooth and exhibits a sudden, discontinuous change in value, making it unsuitable for root-finding algorithms like Newton's method. Now, let's revisit the formulation of IPC (Incremental Potential Contact), which represents the state-of-the-art framework for handling frictional contact [24]. To approximate Coulomb friction smoothly, IPC employs the following energy formulation:

$$E^{\text{IPC}}(\Delta, \delta) = \begin{cases} -(\Delta - (2h + \delta))^2 \ln\left(\frac{\Delta}{2h+\delta}\right), & \Delta \in (2h, 2h + \delta) \\ 0 & \Delta \geq 2h + \delta, \end{cases} \tag{2.1}$$

where δ is the distance tolerance that defines the region $(2h, 2h+\delta)$ for which non-zero forces are experienced. As Δ decreases towards $2h$, the contact energy approaches infinity, rendering it undefined for $\Delta \leq 2h$. While this barrier formulation in IPC effectively enforces non-penetration, it's critical for the solver to avoid any contact pairs entering the penetration zone or reaching the undefined region during the optimization process. This is achieved through a custom line search method that conservatively sets an upper limit for the Newton update coefficient α.

In contrast to this, our energy formulation is designed to permit optimization into the penetrated region, expanding the range of contact forces from $\Delta \in (2h, 2h+\delta)$ to $\Delta \in (0, 2h+\delta)$. This extension allows us to use more aggressive line search methods, resulting in faster convergence, particularly for the some complex rod assemblies simulation problem, e.g., flagella bundling. While this theoretically makes our model susceptible to penetration, we address this concern by ensuring a sufficient contact stiffness parameter k. We describe a method for adaptively setting an appropriate stiffness value in § 2.3.3. Additionally, to enhance non-penetration enforcement, we take the energy formulation from Ref. [36] and square it, causing our gradient to grow exponentially rather than linearly. Therein, our smooth approximation is:

$$E(\Delta, \delta) = \begin{cases} (2h-\Delta)^2 & \Delta \in (0, 2h-\delta] \\ \left(\frac{1}{K_1}\log(1+\exp(K_1(2h-\Delta)))\right)^2 & \Delta \in (2h-\delta, 2h+\delta) \\ 0 & \Delta \geq 2h+\delta, \end{cases} \tag{2.2}$$

where $K_1 = 15/\delta$ is the stiffness of the energy curve.

We incorporate the piecewise term $(2h-\Delta)^2$ into our energy formulation for two key reasons. First, this term serves as an equivalent representation for our energy formulation in the region $\Delta \leq 2h-\delta$. It offers a simpler gradient and Hessian, resulting in computational efficiency. Secondly, and of greater significance, the piecewise term plays a crucial role in ensuring numerical stability by preventing the exponential term in Eq. 2.2 from diverging. In Fig. 2.2, we illustrate the plotted energy term for various δ values. As depicted, the energy starts to increase exponentially as Δ approaches the contact limit, represented as 0 here. As δ decreases, more realistic contact behavior is achieved, enhancing accuracy, albeit with a stiffer equation that may be more challenging to converge.

2.2.2 Computing Piecewise Distance

As mentioned in Li et al. [24], the minimum distance between two edges $(\mathbf{x}_i, \mathbf{x}_{i+1})$ and $(\mathbf{x}_j, \mathbf{x}_{j+1})$ can be formulated as the constrained optimization problem

$$\Delta = \min_{\beta_i, \beta_j} ||\mathbf{x}_i + \beta_i(\mathbf{x}_{i+1} - \mathbf{x}_i) - (\mathbf{x}_j + \beta_j(\mathbf{x}_{j+1} - \mathbf{x}_j))|| \ni 0 \leq \beta_i, \beta_j \leq 1, \tag{2.3}$$

where β_i and β_j represent the contact point ratios along the respective edges. As shown in Fig. 2.1, contact point ratios are $\beta_i = \frac{\|\mathbf{c}_i - \mathbf{x}_i\|}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}$ and $\beta_j = \frac{\|\mathbf{c}_j - \mathbf{x}_j\|}{\|\mathbf{x}_{j+1} - \mathbf{x}_j\|}$ for contact points $\mathbf{c}_i$ and $\mathbf{c}_j$, respectively. Minimum distance between two edges can be classified into three distinct categories: point-to-point, point-to-edge, and edge-to-edge. As the names suggest, these classifications depend on which points of the edges the minimum distance vector $\vec{\Delta}$ lies as described by β_i and β_j shown in Fig. 2.1.

In our prior research [36], we modified Lumelsky's edge-to-edge minimum distance algorithm [43], which implicitly computes the β values, to make it fully differentiable by introducing smooth approximations. However, in this current work, we have transitioned to using piecewise analytical functions, as depicted in Eqs.2.4,2.5, and 2.6, similar to the methodology presented in [24]. This change was motivated by our observation of more stable performance, even though it resulted in a non-smooth Hessian when transitioning between contact categories.

We now describe the conditions for each contact type classification. First, if $\vec{\Delta}$ lies at the ends of both edges, meaning that both β constraints are active, the distance formulation simplifies to the point-to-point case. This case can be readily solved using the Euclidean distance formula,

$$\Delta^{PP} = ||\mathbf{x}_a - \mathbf{x}_b||, \tag{2.4}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the nodes for first and second edges in contact, respectively.

If $\vec{\Delta}$ only lies on one end of one rod (i.e. only one β constraint is active), then the contact type degenerates to point-to-edge. This can be solved as

$$\Delta^{PE} = \frac{||(\mathbf{x}_a - \mathbf{x}_b) \times (\mathbf{x}_b - \mathbf{x}_c)||}{||\mathbf{x}_a - \mathbf{x}_b||}, \tag{2.5}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the nodes of the edge for which the minimum distance vector does not lie on an end and $\mathbf{x}_c$ is the node of the edge which the minimum distance vector does lie on. Finally, edge-to-edge distance (i.e. no active constraints) for the i-th and j-th edges can be solved as

$$\begin{aligned} \mathbf{u} &= (\mathbf{x}_{i+1} - \mathbf{x}_i) \times (\mathbf{x}_{j+1} - \mathbf{x}_j), \\ \Delta^{EE} &= |(\mathbf{x}_i - \mathbf{x}_j) \cdot \hat{\mathbf{u}}|, \end{aligned} \tag{2.6}$$

where $\hat{}$ indicates a unit vector. With Δ fully defined, this concludes our contact energy formulation. To correctly classify contact pairs, we use Lumelsky's algorithm to compute β values.

2.2.3 Adding Implicit Friction

We model friction according to Coulomb's friction law, which describes the conditions necessary for two solids to transition between sticking and sliding. This law states that the frictional force F^{fr} is (1) equal to μF^{n} during sliding, (2) is in the region of $[0, \mu F^{\text{n}})$ when sticking, and (3) is independent of the magnitude of velocity. Here, μ is the friction coefficient and F^{n} is the normal force experienced by the body.

Let's introduce the following equivalences for clarity: $\mathbf{F}^{\text{c}} \equiv k\nabla_{\mathbf{x}} E$ and $\mathbf{J}^{\text{c}} \equiv k\nabla^2_{\mathbf{x}} E$. With

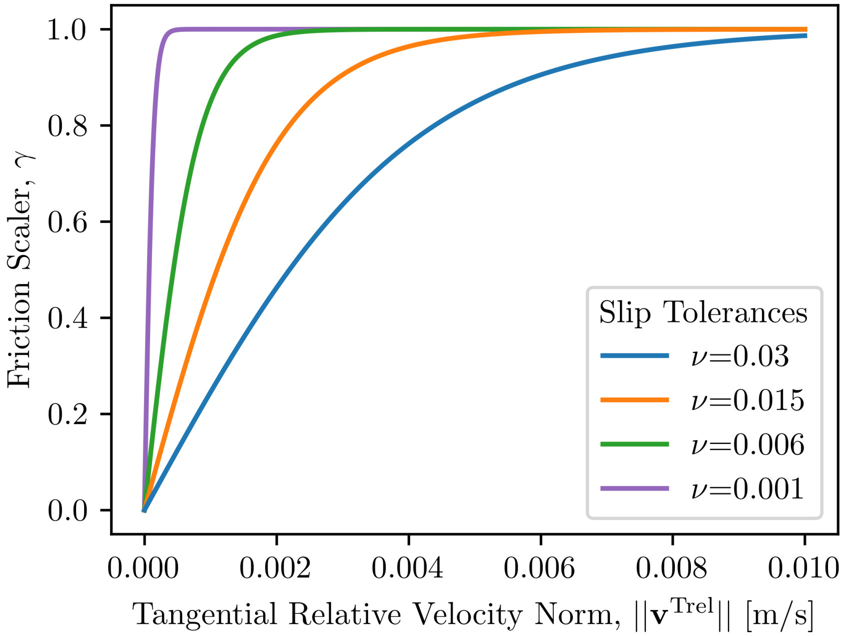


Figure 2.3: Plots for Eq. 2.10 with varying distance tolerance δ values, and γ controls the transition from sticking friction to sliding friction.

these notations, for a contact pair $\mathbf{x}_{ij} := (\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_j, \mathbf{x}_{j+1})$, we can calculate the normal force on the i-th and $i+1$-th nodes as $F_i^{\mathrm{n}} = \|\mathbf{F}_i^{\mathrm{c}}\|$ and $F_{i+1}^{\mathrm{n}} = \|\mathbf{F}_{i+1}^{\mathrm{c}}\|$, respectively. This, in turn, allows us to determine the contact norm vector

$$\mathbf{n}_i = \frac{\mathbf{F}_i^{\mathrm{c}} + \mathbf{F}_{i+1}^{\mathrm{c}}}{\|\mathbf{F}_i^{\mathrm{c}} + \mathbf{F}_{i+1}^{\mathrm{c}}\|}. \tag{2.7}$$

The direction of friction is then the tangential relative velocity between edges i and j. To compute this, we must first compute the relative velocities of the edges at the point of contact, which can be done using $\beta_i, \beta_j \in [0,1]$ as shown below: The direction of friction is determined by the tangential relative velocity between edges i and j. To calculate this velocity, we first need to compute the relative velocities of the edges at the point of contact.

This can be done using $\beta_i, \beta_j \in [0, 1]$ as shown below:

$$\begin{aligned}
\mathbf{v}_i^{\mathrm{e}} &= (1-\beta_i)\mathbf{v}_i + \beta_i \mathbf{v}_{i+1}, \\
\mathbf{v}_j^{\mathrm{e}} &= (1-\beta_j)\mathbf{v}_j + \beta_j \mathbf{v}_{j+1}, \\
\mathbf{v}^{\mathrm{rel}} &= \mathbf{v}_i^{\mathrm{e}} - \mathbf{v}_j^{\mathrm{e}},
\end{aligned} \tag{2.8}$$

where $\mathbf{v}_i, \mathbf{v}_{i+1}, \mathbf{v}_j$, and $\mathbf{v}_{j+1}$ are the velocities of the i-th, $i+1$-th, j-th, and $j+1$-th nodes, respectively. The tangential relative velocity of edge i with respect to edge j can then be computed as

$$\mathbf{v}^{\mathrm{Trel}} = \mathbf{v}^{\mathrm{rel}} - (\mathbf{v}^{\mathrm{rel}} \cdot \mathbf{n}_i)\mathbf{n}_i, \tag{2.9}$$

where $\hat{\mathbf{v}}^{\mathrm{Trel}} = \mathbf{v}^{\mathrm{Trel}}/||\mathbf{v}^{\mathrm{Trel}}||$ is our friction direction.

Now, our contact model needs to handle the transition between sticking and sliding. According to Coulomb's law, $\|\mathbf{v}^{\mathrm{Trel}}\| = 0$ during static friction, while $\|\mathbf{v}^{\mathrm{Trel}}\| > 0$ is indicative of sliding friction. Sticking continues until the tangential force exceeds the threshold of μF^{n}, at which point sliding commences. This relationship, much like the ideal contact energy, can also be represented using a modified Heaviside step function. However, for the same reasons as mentioned earlier, we opt for another smooth approximation, as described by:

$$\gamma\left(\|\mathbf{v}^{\mathrm{Trel}}\|, \nu\right) = \frac{2}{1 + \exp\left(-K_2 \|\mathbf{v}^{\mathrm{Trel}}\|\right)} - 1, \tag{2.10}$$

where ν (m/s) is our desired slipping tolerance and $K_2(\nu) = 15/\nu$ is the stiffness parameter. As shown in Fig. 2.3, $\gamma \in [0, 1]$ smoothly scales the friction force magnitude from zero to one as $\|\mathbf{v}^{\mathrm{Trel}}\|$ increases from zero. The slipping tolerance describes the range of velocities $(0, \nu)$ for which a friction force $< \mu F^{\mathrm{n}}$ is experienced. In other words, we consider velocities within this range to be "sticking".

Finally, the friction experienced by a node i for a contact pair $\mathbf{x}_{ij}$ can be described as

$$\mathbf{F}_i^{\text{fr}} = -\mu\gamma\hat{\mathbf{v}}^{\text{Trel}} F_i^{\text{n}}. \tag{2.11}$$

With friction fully defined, we can proceed to formulate the friction Jacobian $\nabla_{\mathbf{x}}\mathbf{F}^{\text{fr}}$. It's worth noting that our formulation depends on $\beta(\mathbf{x})$, which implies the need to compute the gradient $\nabla_{\mathbf{x}}\beta$. However, we can simplify this process by recognizing that the magnitudes of the contact forces $\mathbf{F}^{\text{c}}i$ and $\mathbf{F}^{\text{c}}i+1$ exhibit a linear relationship with β, where

$$\begin{aligned} \mathbf{F}_i^{\text{c}} &= (1-\beta)(\mathbf{F}_i^{\text{c}} + \mathbf{F}_{i+1}^{\text{c}}), \\ \mathbf{F}_{i+1}^{\text{c}} &= \beta(\mathbf{F}_i^{\text{c}} + \mathbf{F}_{i+1}^{\text{c}}). \end{aligned} \tag{2.12}$$

Therefore, we can obtain β by simply solving

$$\beta = \frac{\|\mathbf{F}_{i+1}^{\text{c}}\|}{\|\mathbf{F}_i^{\text{c}} + \mathbf{F}_{i+1}^{\text{c}}\|}. \tag{2.13}$$

We can now treat β as a function of $\mathbf{F}^{\text{c}}$, resulting in a simplified chain ruling procedure. Let us denote Eq. 2.11 as the functional $f(\mathbf{x}, \mathbf{F}^{\text{c}}(\mathbf{x}))$. The friction Jacobian can then be computed through chain rule as

$$\nabla_{\mathbf{x}}\mathbf{F}^{\text{fr}} = \nabla_{\mathbf{x}} f + \nabla_{\mathbf{F}^{\text{c}}} f \nabla_{\mathbf{x}}\mathbf{F}^{\text{c}}. \tag{2.14}$$

This concludes our fully implicit friction scheme. Full psuedocode for the IMC algorithm can be found in Algo. 1.

2.3 Augmenting Numerics

Although Coulomb friction is perfectly defined in IMC, numerical issues can still arise when simulating complicated rod assemblies. To address these challenges and ensure the robustness and efficiency of IMC, we have implemented a series of numerical treatments, which are

```
Algorithm 1: Implicit Contact Model
  Input: $\mathbf{x}, \mathbf{x}_0, k, \delta, \nu$
  Output: $\mathbf{F}^{\text{c}}, \mathbf{J}^{\text{c}}, \mathbf{F}^{\text{fr}}, \mathbf{J}^{\text{fr}}$
  Function IMC($\mathbf{x}, \mathbf{x}_0, k, \delta, \nu$):
      $\mathbf{v} \leftarrow \mathbf{x} - \mathbf{x}_0$                                  // compute velocity
      $\mathbf{F}^{\text{c}}, \mathbf{J}^{\text{c}} \leftarrow$ genContact($\mathbf{x}, \delta$)                    // Eq.2.2
      $\mathbf{F}^{\text{c}} \leftarrow k\mathbf{F}^{\text{c}}$                          // scale by contact stiffness
      $\mathbf{J}^{\text{c}} \leftarrow k\mathbf{J}^{\text{c}}$                                      // $\mathbf{J}^{\text{c}} \equiv \nabla_{\mathbf{x}}\mathbf{F}^{\text{c}}$
      $\mathbf{F}^{\text{fr}} \leftarrow$ genFriction($\mathbf{x}, \mathbf{v}, \mathbf{F}^{\text{c}}, \nu$)                 // Eq.2.11
      $\nabla_{\mathbf{x}} f, \nabla_{\mathbf{F}^{\text{c}}} f \leftarrow$ genFrictionPartials($\mathbf{x}, \mathbf{v}, \mathbf{F}^{\text{c}}, \nu$)   // $\mathbf{F}^{\text{fr}} \equiv f(\mathbf{x}, \mathbf{F}^{\text{c}})$
      $\mathbf{J}^{\text{fr}} \leftarrow \nabla_{\mathbf{x}} f + \nabla_{\mathbf{F}^{\text{c}}} f \nabla_{\mathbf{x}} \mathbf{F}^{\text{c}}$                      // Eq.2.14
      return $\mathbf{F}^{\text{c}}, \mathbf{J}^{\text{c}}, \mathbf{F}^{\text{fr}}, \mathbf{J}^{\text{fr}}$
```

outlined in the following paragraphs.

2.3.1 Scaling of Contact Potential

As the radius of the rod can be very thin, often within the range of a few millimeters, collision detection, and gradient and Hessian computations may be susceptible to floating-point inaccuracies. To address this, we employ a scaling approach by multiplying all nodes by a factor of $1/h$ to enhance numerical stability. This scaling introduces normalized variables $\bar{\Delta} = \Delta/h$ and $\bar{\delta} = \delta/h$, resulting in the normalization of Eq. 2.2 to:

$$E\left(\bar{\Delta}, \bar{\delta}\right) = \begin{cases} (2 - \bar{\Delta})^2 & \bar{\Delta} \in (0, 2 - \bar{\delta}] \\ \left(\frac{1}{K_1} \log\Big(1 + \exp\Big(K_1(2 - \bar{\Delta})\Big)\Big)\right)^2, & \bar{\Delta} \in (2 - \bar{\delta}, 2 + \bar{\delta}) \\ 0 & \bar{\Delta} \geq 2 + \bar{\delta}. \end{cases} \tag{2.15}$$

Accounting for this scaling in the gradient and Hessian computation can easily be taken care of through the product and chain rule which ends in us multiplying the gradient by $1/h$ and the Hessian by $(1/h)^2$. With this normalization in mind, we choose a stiffness K_1 for a desired δ through the relation

$$K_1(\delta, h) = \frac{15h}{\delta}. \tag{2.16}$$

2.3.2 Collision Detection

To perform collision detection, we identify the set of all edge combinations whose minimum distance is less than $2h + \delta$, resulting in the contact set

$$\mathcal{C} = \{\mathbf{x}_{ij} \in \mathcal{X} \mid \Delta_{ij} < 2h + \delta\}. \tag{2.17}$$

Given that this operation can be computationally intensive, we adopt an optimization strategy. At the beginning of each time step, we compute a candidate set $\hat{\mathcal{C}} = \{\mathbf{x}_{ij} \in \mathcal{X} \mid \Delta_{ij} < 2h + \hat{\delta}\}$, where $\hat{\delta}$ is chosen to be significantly larger than δ. Then, at the start of each iteration, we derive the actual contact set $\mathcal{C}$ from the candidate set $\hat{\mathcal{C}}$. This approach substantially reduces computational costs

Note that if $\hat{\delta}$ is not set large enough, certain edge combinations not belonging to the initial set $\hat{\mathcal{C}}$ may enter the contact zone or even penetrate by the end of the optimization. Our energy formulation in Eq. 2.2 is capable of dealing with this as minor penetrations do not lead to simulation failure and will be remedied in the next time step. This is in contrast to IPC, which may require a significantly larger $\hat{\delta}$ and/or more robust collision checking during each iteration of the optimization process.

2.3.3 Adaptive Contact Stiffness

As a penalty method, we utilize a contact stiffness parameter k to scale both the contact force and Jacobian. It's significant to select an appropriate value for k to prevent issues such as penetration (due to a value too low) or excessive hovering (due to a value too high) To achieve this, we start by defining the set $\mathcal{C}_i$ as the collection of all node indices i and j associated with the contact set edge combinations $\mathbf{x}_{ij} \in \mathcal{C}$. Next, in our pursuit of an appropriate scaling for the contact stiffness, we calculate the norm of the total forces (excluding contact and friction) $\left|\left|\mathbf{F}_i^{\text{total}}\right|\right|$ experienced by each node within the contact set $\mathcal{C}$. These forces can be represented as:

$$\mathcal{F} = \left\{ \left\|\mathbf{F}_i^{\text{total}}\right\| \mid i \in \mathcal{C}_i \right\}. \tag{2.18}$$

The maximum force magnitude of these forces can then be used to determine the contact stiffness

$$k = \max(\mathcal{F})s, \tag{2.19}$$

where s is a constant scaling factor. In all our experiments, we set s to be 1×10^5. The intuition behind this contact stiffness formulation is to achieve non-penetration through force equilibrium. Furthermore, by using the maximum of $\mathcal{F}$, if non-penetration can be achieved for the edge with the largest value in $\mathcal{F}$, then this k value should be large enough to prevent penetration for all other contact pairs as well.

2.3.4 Line Search Algorithm

Once the internal forces (e.g., bending force, twisting force, and stretching force), external forces (e.g., viscous dragging force and contact forces), and their respective Jacobians are computed, we can simply use Newton method to find the solution of the equations of motion. However, due to the high nonlinearity of the governing equations, convergence for Newton's method may suffer without a line search method. To rectify this, we perform Goldstein-Price line search in the Newton direction to ensure that the square of the Euclidean norm of total force $\|\mathbf{F}\|^2$ in Eq. 7 decreases.

We design an inner loop for each Newton iteration where we deploy the line search algorithm. This inner loop returns an optimal search step size α until $\|\mathbf{F}\|^2$ is smaller than a certain tolerance or until a maximum number of iterations is reached. As mentioned in § 2.2.1, our energy formulation allows us to use a more aggressive line search strategy compared to Ref. [24], resulting in larger search step sizes and faster convergence. Pseudocode for the line search method can be found in detail in Algo.2.

Algorithm 2: LineSearch

```
Parameters: $\alpha_l, \alpha_u, m_1 = 0.1, m_2 = 0.9$          // Initial interval for $\alpha$
Input: $\mathbf{q}, \Delta\mathbf{q}$                                  // DOFs from DER
Output: $\alpha$                                           // Newton search magnitude
Function LineSearch($\mathbf{q}, \Delta\mathbf{q}$):
    iter ← 0
    $\alpha \leftarrow 1$
    success ← False
    $\mathbf{d_0} \leftarrow \mathbf{F}^T(\partial\mathbf{F}/\partial\mathbf{q})\Delta\mathbf{q}$
    while success is False do
        if $\alpha m_2 \mathbf{d}_0 <= \frac{1}{2}\|\mathbf{F}(\mathbf{q} - \alpha\Delta\mathbf{q})\|^2 - \frac{1}{2}\|\mathbf{F}(\mathbf{q})\|^2 <= \alpha m_1 \mathbf{d}_0$ then
            success ← True
        else if $\frac{1}{2}\|\mathbf{F}(\mathbf{q} - \alpha\Delta\mathbf{q})\|^2 - \frac{1}{2}\|\mathbf{F}(\mathbf{q})\|^2 < \alpha m_2 \mathbf{d}_0$ then
            $\alpha_l \leftarrow \alpha$
        else
            $\alpha_u \leftarrow \alpha$
        if $|\alpha_l - \alpha_r| <$ small value or iter $>$ iterMax then
            success ← True
        $\alpha \leftarrow 0.5(\alpha_l + \alpha_u)$
        iter ← iter + 1
    return $\alpha$
```

2.4 Numerical Study for Flagella Bundling

In this section, we showcase simulation results for a challenging scenarios of rods in frictional contact – flagella bundling. Flagella bundling a significant phenomena in the nature [44, 45, 46, 47, 48, 49, 50, 51, 52]. We capture this phenomenon with a specific designed simulator combining DER, IMC, and solid-fluid interactions [53]. We then evaluate IMC''s performance with the state-of-the-art frictional contact framework – IPC [24].

When micro-organisms with multiple flagella swim in fluid, their multiple flagella will become a bundle for navigation (e.g., *Escherichia coli* and *Salmonella typhimurium* [54]). Each flagellum consists of a rotary "head", a short flexible hook, and a helical filament. By rotating their filaments, these micro-organisms can navigate their environments through

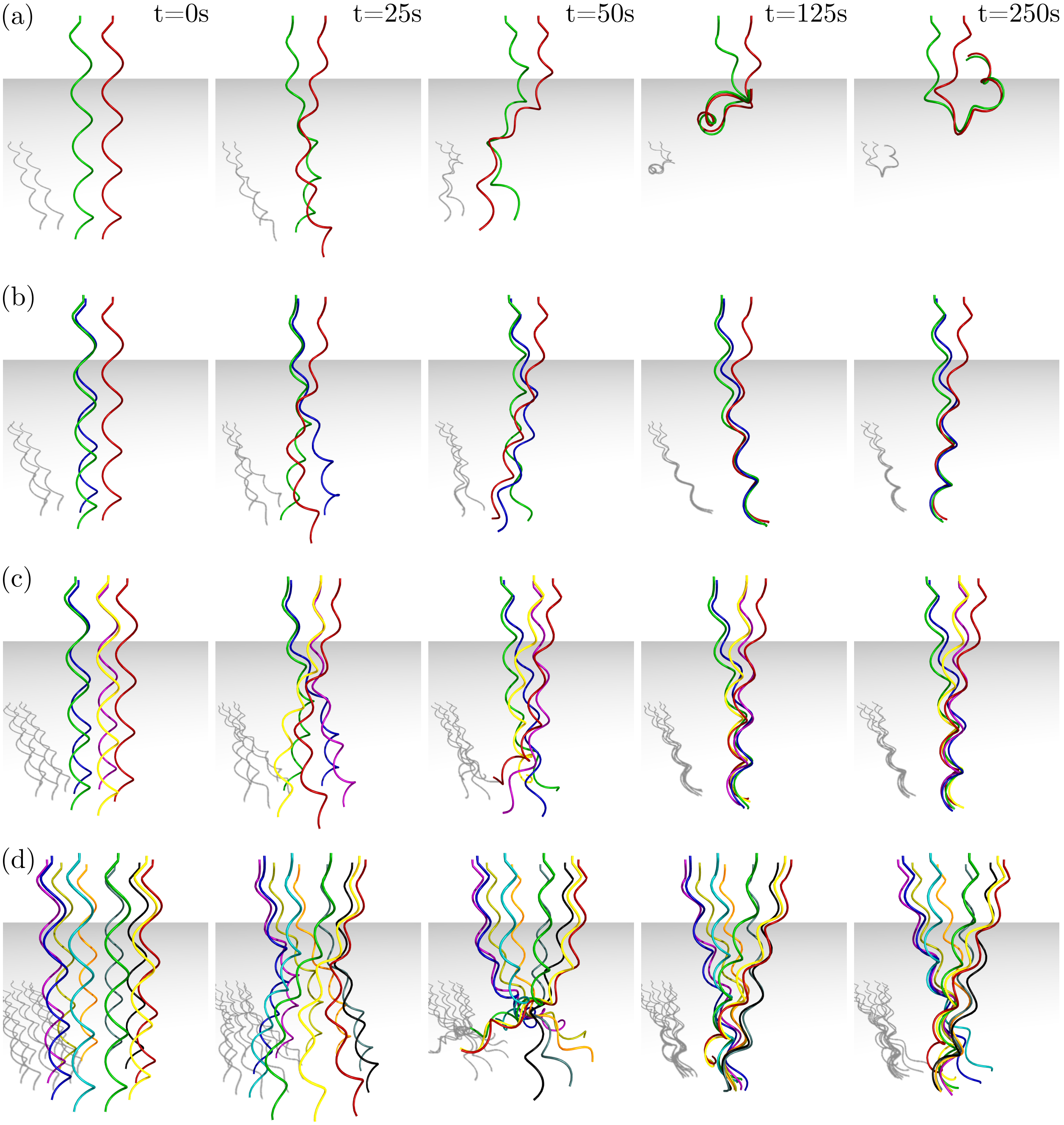


Figure 2.4: Rendered snapshots of flagella bundling with varying amounts of flagella. Rows contain (a) $M = 2$, (b) $M = 3$, (c) $M = 5$, and (d) $M = 10$ flagella. Each column indicates the flagella configuration at the moment of time indicated in the top row.

sophisticated manipulation of the solid-fluid interaction between their flexible structures and the surrounding flows. This has led to biomimicry, where flagella have inspired the design of several soft robot locomotion strategies in viscous fluids [23, 55, 56, 57, 58]. Here, we developed a specific numerical framework combined DER, regularized stokeslet segment (RSS) (which is for simulating fluiding-solid interaction), IMC for simulating flagella bundling, as shown in Fig. 2.4. To validate the IMC's performance, we also switch IMC to IPC in the numerical framework to observe the performance with different contact handling model. The details are given in the following paragraph.

2.4.1 Solid-Fluid Interaction

As stated before, the simulator for flagella bundling is a combination of three components: DER for the flagella, IMC for the contact handling, and RSS for solid-fluid interaction. Since DER and IMC are detailed formulate before. Here, we show how to use RSS to simulate solid-fluid interaction.

When simulating the flulid-solid interactions, we formulate the relationship between velocity and hydrodynamic forces for each node in a discretized rod. We utilize Regularized Stokeslet Segments (RSS) [53] method to capture the viscous drag forces exerted on a slender rod moving in a viscous fluid. The Stokeslet, the primary Green's function of Stokes flow, represents solutions for the Stokes equations as a superposition of fundamental solutions. The methods of regularized Stokeslets have been implemented in numerous studies such as self-propelled microorganisms [59] [60], hyperactivated sperm motility [61] [62], and flagella bundling [44] [51].

For a particular choice of regularization [53], the relationship of the velocity at an evaluation point $\hat{\mathbf{x}}$ and the corresponding regularized force $\mathbf{f}$ exerted on a point $\mathbf{x}$ is the regularized Stokeslet:

$$8\pi\eta\mathbf{u}(\hat{\mathbf{x}}) = \left(\frac{1}{R} + \frac{\epsilon^2}{R^3}\right)\mathbf{f}(\mathbf{x}) + \frac{(\mathbf{f}(\mathbf{x})\cdot\mathbf{r})\mathbf{r}}{R^3}, \tag{2.20}$$

where η is the fluid viscosity; $\mathbf{r} = \hat{\mathbf{x}} - \mathbf{x}$; $R^2 = \|\mathbf{r}\|^2 + \epsilon^2$, and ϵ is the regularization parameter.

For an edge $\mathbf{e}^i = \mathbf{x}_{i+1} - \mathbf{x}_i$ in a viscous fluid, a point along this edge can be defined by

$\mathbf{x}_\alpha = \mathbf{x}_i + \alpha(\mathbf{x}_{i+1} - \mathbf{x}_i)$, where $\alpha \in [0, 1]$. We assume a linear density of the force $\mathbf{f}_\alpha$ applied on the edge: $\mathbf{f}_\alpha = \mathbf{f}_i + \alpha(\mathbf{f}_{i+1} - \mathbf{f}_i)$. With this, we obtain the relationship between the velocity at $\hat{\mathbf{x}}$ and the linear force density as

$$8\pi\eta\mathbf{u}(\hat{\mathbf{x}}) = \|\mathbf{e}\| \int_0^1 \left[\left(\frac{1}{R_\alpha} + \frac{\epsilon^2}{R_\alpha^3} \right) \mathbf{f}_\alpha + \frac{(\mathbf{f}_\alpha \cdot \mathbf{r}_\alpha)\mathbf{r}_\alpha}{R_\alpha^3} \right] d\alpha, \tag{2.21}$$

where $\|\mathbf{e}\|$ is the length of the edge $\mathbf{e}$. Through integration, we can then rewrite Eq. 2.21 as

$$8\pi\eta\mathbf{u}(\hat{\mathbf{x}}) = \mathbf{f}_0(T_{0,-1} + \epsilon^2 T_{0,-3}) + \mathbf{f}_1(T_{1,-1} + \epsilon^2 T_{1,-3}) + \sum_{n=0}^{3} c_n T_{n,-3}, \tag{2.22}$$

where the coefficients $\mathbf{c}_n$ are

$$\begin{aligned}
\mathbf{c}_0 &= (\mathbf{f}_0 \cdot \mathbf{r}_0)\mathbf{r}_0, \\
\mathbf{c}_1 &= (\mathbf{f}_0 \cdot \mathbf{e})\mathbf{r}_0 + (\mathbf{f}_0 \cdot \mathbf{r}_0)\mathbf{e} + (\mathbf{f}_1 \cdot \mathbf{e})\mathbf{r}_0, \\
\mathbf{c}_2 &= (\mathbf{f}_0 \cdot \mathbf{v})\mathbf{v} + (\mathbf{f}_1 \cdot \mathbf{r}_0)\mathbf{e} + (\mathbf{f}_1 \cdot \mathbf{e})\mathbf{r}_0, \\
\mathbf{c}_3 &= (\mathbf{f}_1 \cdot \mathbf{e})\mathbf{e},
\end{aligned} \tag{2.23}$$

and the sequences of $T_{k,l}$ are

$$\begin{aligned}
\mathbf{T}_{0,-1} &= \frac{1}{\|\mathbf{e}\|} \log[\|\mathbf{e}\| R_\alpha + (\mathbf{r}_\alpha \cdot \mathbf{e})] \Big|_0^1, \\
\mathbf{T}_{0,-3} &= - \frac{1}{R_\alpha[\|\mathbf{e}\| R_\alpha + (\mathbf{r}_\alpha \cdot \mathbf{e})]} \Bigg|_0^1, \\
\mathbf{T}_{1,-1} &= \frac{R_\alpha}{\|\mathbf{e}\|^2} \Bigg|_0^1 - \frac{\mathbf{r}_0 \cdot \mathbf{e}}{\|\mathbf{e}\|^2} T_{0,-1}, \\
\mathbf{T}_{1,-3} &= - \frac{1}{R_\alpha \|\mathbf{e}\|^2} \Bigg|_0^1 - \frac{\mathbf{r}_0 \cdot \mathbf{e}}{\|\mathbf{e}\|^2} T_{0,-3}, \\
\mathbf{T}_{2,-3} &= - \frac{\alpha}{R_\alpha \|\mathbf{e}\|^2} \Bigg|_0^1 + \frac{1}{\|\mathbf{e}\|^2} T_{0,-1} - \frac{\mathbf{r}_0 \cdot \mathbf{e}}{\|\mathbf{e}\|^2} T_{1,-3}, \\
\mathbf{T}_{3,-3} &= - \frac{\alpha^2}{R_\alpha \|\mathbf{e}\|^2} \Bigg|_0^1 + \frac{2}{\|\mathbf{e}\|^2} T_{1,-1} - \frac{\mathbf{r}_0 \cdot \mathbf{e}}{\|\mathbf{e}\|^2} T_{2,-3}.
\end{aligned} \tag{2.24}$$

Recall that in DER a flexible rod constitutes $N+1$ nodes and N edges each with length $\|\mathbf{e}^i\|$. Therefore, we can formulate hydrodynamic forces for the whole rod through the summation of Eq. 2.22 for each segment, resulting in

$$8\pi\eta\mathbf{u}(\hat{\mathbf{x}}) = \sum_{i=0}^{N-2} \left(\mathbb{A}_1^i \mathbf{f}_i + \mathbb{A}_2^i \mathbf{f}_{i+1} \right), \tag{2.25}$$

where coefficient matrices $\mathbb{A}_1^i$ and $\mathbb{A}_2^i$ are

$$\begin{aligned}
\mathbb{A}_1^i &= \|\mathbf{e}^i\| \left[\left(T_{0,-1}^{i,i+1} + \epsilon^2 T_{0,-3}^{i,i+1} \right) + T_{0,-3}^{i,i+1} \left(\mathbf{r}_i \mathbf{r}_i^T \right) + T_{1,-3}^{i,i+1} \left(\mathbf{r}_i (\mathbf{e}^i)^T + \mathbf{e}^i \mathbf{r}_i^T \right) + T_{2,-3}^{i,i+1} \left(\mathbf{e}^i (\mathbf{e}^i)^T \right) \right] - \mathbb{A}_2^i, \\
\mathbb{A}_2^i &= \|\mathbf{e}^i\| \left[\left(T_{1,-1}^{i,i+1} + \epsilon^2 T_{1,-3}^{i,i+1} \right) + T_{1,-3}^{i,i+1} \left(\mathbf{r}_i \mathbf{r}_i^T \right) + T_{2,-3}^{i,i+1} \left(\mathbf{r}_i (\mathbf{e}^i)^T + \mathbf{e}^i \mathbf{r}_i^T \right) + T_{3,-3}^{i,i+1} \left(\mathbf{e}^i (\mathbf{e}^i)^T \right) \right].
\end{aligned} \tag{2.26}$$

Finally, by using Eqs. 2.25 and 2.26, we can describe the balance between the velocity and the force density along a discretized rod as the linear system

$$\begin{aligned}
\mathbf{U} &= \mathbb{A}\mathbf{f}, \\
\mathbf{F}^{\text{hydro}} &= [\mathbf{f}_0 \Delta l_0, \mathbf{f}_0 \Delta l_1, ..., \mathbf{f}_{N-1} \Delta l_{N-1}]^T,
\end{aligned} \tag{2.27}$$

where $\mathbf{U} = [\dot{\mathbf{x}}_0, \dot{\mathbf{x}}_1, ..., \dot{\mathbf{x}}_{N-1}]^T$ (the velocity of the liquid at nodal position $\mathbf{x}_i$ has the same speed of the rod due to non-slip boundary); $\mathbf{f} = [\mathbf{f}_0, \mathbf{f}_1, ..., \mathbf{f}_{N-1}]^T$ is the force density; $\mathbf{F}^{\text{hydro}}$ is the hydrodynamic force vector, and Δl_i is the Voronoi length at node $\mathbf{x}_i$. We assume the density of the fluid to be equal to the density of the rod so that buoyant forces are negligible. As stated in Ref. [53], we choose the regularization parameter $\epsilon = 1.02h$ based on the value of $\Delta l_i/h$ where Δl_i is the Voronoi length.

With the solid-fluid interaction model formulated, we can design a complete simulator for flagella bundling. The full pseudocode for flagella bundling is given in Algo. 3.

Algorithm 3: Flagella Simulation

```
Parameters: $\delta, \hat{\delta}, \nu$, tolerance
Input: $\mathbf{q}(t_i), \dot{\mathbf{q}}(t_i)$
Output: $\mathbf{q}(t_{i+1}), \dot{\mathbf{q}}(t_{i+1})$
Require      : boundary conditions $\leftarrow$ free
Function FlagellaSim($\mathbf{q}(t_i), \dot{\mathbf{q}}(t_i)$):
    Guess $\mathbf{q}^{(0)} \leftarrow \mathbf{q}(t_i)$
    $n \leftarrow 0,\ \epsilon \leftarrow \infty$
    $\mathbf{F}^{\text{hydro}} \leftarrow \mathbb{A}^{-1}\dot{\mathbf{q}}(t_i)\Delta l$                // Eq. 2.27
    $\mathbf{F}^{\text{total}} \leftarrow \mathbf{0}^{\text{DOF}}$
    $\mathbf{J}^{\text{total}} \leftarrow \mathbf{0}^{\text{DOF}\times\text{DOF}}$
    while $\epsilon >$ tolerance do
        $\mathbf{F}^{\text{int}} \leftarrow$ genForces($\cdot$)                            // Eq. 1.7
        $\mathbf{J}^{\text{int}} \leftarrow$ genJacobian($\cdot$)                          // $\frac{\partial^2(E_s+E_t+E_b)}{\partial q_i \partial q_j}$
        if $n == 0$ then                                      // run only on first iter
            $\hat{\mathcal{C}} \leftarrow$ constructCandidateSet($\mathbf{x}, \hat{\delta}$)          // § 2.3.2
            $k \leftarrow$ updateConStiffness($\hat{\mathcal{C}}, \mathbf{F}^{\text{int}}$)            // § 2.3.3
        $\mathcal{C} \leftarrow$ collisionDetection($\hat{\mathcal{C}}, \delta$)
        for $\mathbf{x}, \mathbf{x}_0 \in \mathcal{C}$ do
            $\mathbf{F}^{\text{c}}, \mathbf{J}^{\text{c}}, \mathbf{F}^{\text{fr}}, \mathbf{J}^{\text{fr}} \leftarrow$ IMC($\mathbf{x}^{(n)}, \mathbf{x}_0, k, \delta, \nu$)        // Algo. 1
            $\mathbf{F}^{\text{total}} \leftarrow \mathbf{F}^{\text{total}} + \mathbf{F}^{\text{c}} + \mathbf{F}^{\text{fr}}$
            $\mathbf{J}^{\text{total}} \leftarrow \mathbf{J}^{\text{total}} + \mathbf{J}^{\text{c}} + \mathbf{J}^{\text{fr}}$
        $\mathbf{F}^{\text{total}} \leftarrow \mathbf{F}^{\text{total}} + \mathbf{F}^{\text{int}} + \mathbf{F}^{\text{hydro}}$
        $\mathbf{J}^{\text{total}} \leftarrow \mathbf{J}^{\text{total}} + \mathbf{J}^{\text{int}}$
        $\mathbf{F}^{\text{free}} \leftarrow \mathbf{F}^{\text{der}}$(free)          // Downsize to only include free DOFs
        $\mathbf{J}^{\text{free}} \leftarrow \mathbf{J}^{\text{der}}$(free, free)
        $\Delta\mathbf{q}^{\text{free}} \leftarrow \mathbf{F}^{\text{free}}/\mathbf{J}^{\text{free}}$            // Solve $\mathbf{J}^{\text{free}}\Delta\mathbf{q}^{\text{free}} = \mathbf{F}^{\text{free}}$
        $\alpha \leftarrow$ LineSearch($\mathbf{q}^{\text{free}}, \Delta\mathbf{q}^{\text{free}}$)                   // Algo. 8
        $\mathbf{q}^{(n+1)}$(free) $\leftarrow \mathbf{q}^{(n)}$(free) $- \alpha\Delta\mathbf{q}_{\text{free}}$
        $\epsilon \leftarrow \left\|\mathbf{F}^{\text{free}}\right\|$                                 // update error
        $n \leftarrow n + 1$
    $\mathbf{q}(t_{i+1}) \leftarrow \mathbf{x}^{(n)}$
    $\dot{\mathbf{q}}(t_{i+1}) \leftarrow (\mathbf{q}(t_{i+1}) - \mathbf{q}(t_i))/dt$
    return $\mathbf{q}(t_{i+1}), \dot{\mathbf{q}}(t_{i+1})$
```

2.4.2 Numerical Setup for Simulations

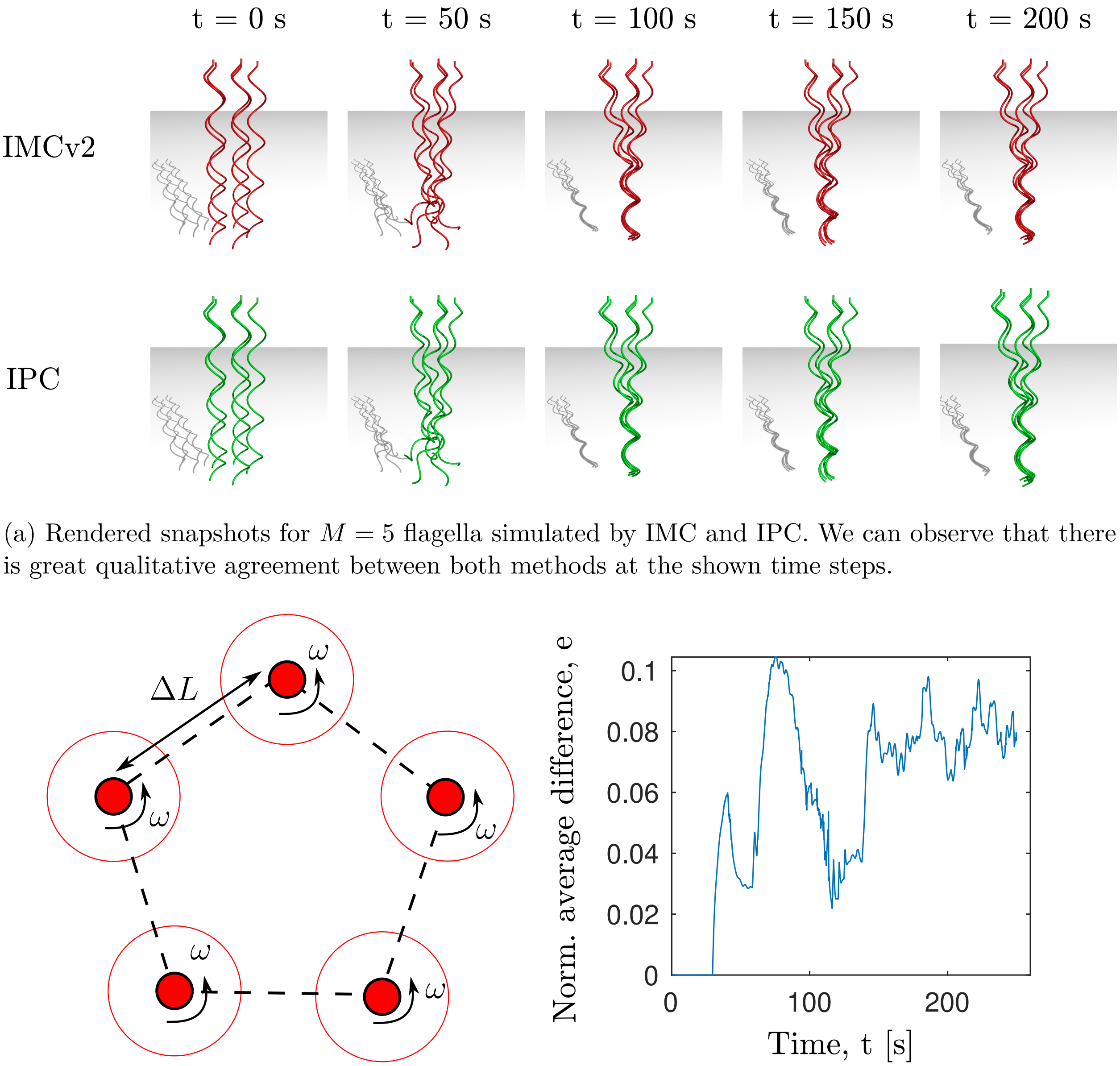
In the simulation, we design the flagella as right-handed helical rods manufactured with linear elastic material. We set the material properties as follows: Young's modulus was set to $E = 3.00$ MPa; Poisson's ratio was set to 0.5; density of the rod was set to $\rho = 1000$ kg/m^3; the cross-sectional radius was set to $h = 1$ mm, and the fluid viscosity was set to 0.1 Pa·s. Here, a Poisson's ratio of 0.5 was chosen to enforce the flagella to be an incompressible material. The topologies of the flagella are helices with helical radius $a = 0.01$ m, helical pitch $\lambda = 0.05$ m, and axial length $z_0 = 0.2$ m. These parameters were chosen as they best mimic the geometries of biological flagella found in nature [32, 34, 63, 51].

We explore the bundling phenomena with M flagella ($M = [2, 3, 5, 10]$) where the rotating ends of each flagella is fixed along the z-axis as shown in Fig. 2.5b. These rotating ends are treated as boundary conditions and are spaced out equidistantly so as to form a regular polygon with M angles with side length $\Delta L = 0.03$ m as shown in Fig. 2.5c. We set the rotation speed of the flagella ends to $\omega = 15$ rad/s which keeps the Reynolds number in our numerical simulation to be always smaller than 4×10^{-2}, thus satisfying the Stokes flow.

Finally, we discretize each flagella into 68 nodes for a total of 67 edges. We found this discretization to have the best trade-off between computational efficiency and accuracy. Furthermore, we set the time step size to $\Delta t = 1$ ms. As the forces generated from our fluid model are handled explicitly, we found 1 ms to be the largest stable time step size before convergence performance became hampered. A distance tolerance of $\delta = 1 \times 10^{-5}$ was used for all simulations.

2.4.3 Comparison between IMC and IPC

Both IMC and IPC were implemented to simulate 250 seconds of rotation for scenarios with 2, 3, 5, and 10 flagella, as illustrated in Fig. 2.4. In the context of viscous fluids, where friction between structures is typically minimal, we focused on purely contact interactions,



(a) Rendered snapshots for $M = 5$ flagella simulated by IMC and IPC. We can observe that there is great qualitative agreement between both methods at the shown time steps.

(b) A top-down visualization of boundary conditions applied to the highest nodes (filled in red circles) of each flagellum as well as the angular rotation ω applied to them. The larger hollow red circles represent the rest of the helical flagella.

(c) The norm of the average difference in the nodal positions for the flagella simulated by IMC and IPC with respect to time.

Figure 2.5: Comparison between IMC and IPC in flagella bundling

neglecting friction ($\mu = 0$). To assess the performance of IMC and IPC, we conducted a visual comparison for the case with 5 flagella ($M = 5$), as demonstrated in Fig. 2.4. At various

Table 2.1: IMC vs. IPC [24] run time data. Simulations are run for a total of 250 seconds with a time step size of $\Delta t = 1$ ms and a rotation speed of $\omega = 15$ rad/s. The contact model used can be seen in the far left column. Next to this, M indicates the number of flagella. AIPTS stands for average iterations per time step. ATPTS stands for average time per time step. Total Iters indicates the total number of Newton's iterations that were necessary to complete the simulation. The Total Run Time is the total computational time to completion. Finally, RTI stands for run time improvement and is the ratio of improvement between IMC's and IPC's total run time.

Model	M	AIPTS	ATPTS [ms]	Total Iters	Total Run Time [hr]	RTI
IMC	2	3.00	10.2	6.01×10^5	0.57	1.82
	3	3.01	21.3	6.04×10^5	1.19	1.82
	5	3.02	67.5	5.39×10^5	3.34	1.40
	10	3.12	389.4	6.56×10^5	22.77	1.22
IPC	2	4.00	18.75	7.98×10^5	1.04	N/A
	3	4.00	39.5	7.93×10^5	2.17	N/A
	5	4.01	95.3	7.09×10^5	4.68	N/A
	10	4.02	477.47	8.45×10^5	27.88	N/A

time steps, it can be observed that the flagella configurations closely resemble each other under both IMC and IPC simulations. This visual similarity suggests that both methods exhibit comparable performance. To quantitatively evaluate this similarity, we introduced a normalized average difference metric, denoted as $\bar{e}$, to measure the dissimilarity in nodal configurations of the flagella between IMC and IPC simulations. This metric is defined as:

$$\bar{e} = \frac{1}{MNh} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left\| \mathbf{x}_j^{i,\text{IMC}} - \mathbf{x}_j^{i,\text{IPC}} \right\|, \tag{2.28}$$

The relationship between the normalized average difference $\bar{e}$ and time t is illustrated in Fig. 2.5c. This plot reveals that the differences between the configurations generated by IMC and IPC remain quite minimal over time, reinforcing the idea that IMC exhibits comparable performance to IPC despite the absence of a non-penetration guarantee.

Where IMC begins to outperform IPC is in terms of computational efficiency. We have provided detailed metrics for all runs in Tab. 2.1, including the average iterations per time step (AIPTS), average time per time step (ATPTS), total iterations, and total run time. All metrics were recorded using time steps with at least one contact. Notably, IMC achieved convergence with fewer average iterations than IPC across all flagella cases, resulting in significant reductions in total run time. These run time improvements are most pronounced for scenarios with $M = 2$ and $M = 3$, and while they start to decrease as M increases further due to the RSS force computation becoming a bottleneck, a clear trend of decreasing run time can still be observed.

2.4.4 Friction Examples

Although friction is typically negligible in a viscous fluid medium, the impact of friction on flagella bundling is still intriguing, especially considering scenarios where the effect of friction can become significant, such as in granular environments. To explore this, we assume an imaginary viscous environment where the friction coefficient between structures is non-negligible. In our simulations, we present data for two flagella ($M = 2$) with friction coefficients ranging from $\mu = 0.1$ to $\mu = 1.0$. A slipping tolerance of $\nu = 1 \times 10^{-4}$ was used for all cases. All other parameters remain the same as before.

We first showcase the sticking and slipping phenomena with snapshots for three different friction coefficients ($\mu = 0$, $\mu = 0.3$, and $\mu = 0.7$) in Fig. 2.6. As intuitively expected, as the friction coefficient μ increases, the amount of sticking also increases. Convergence results for all friction examples can be found in Tab. 2.2, which includes information on average iterations per time step and the simulation length. Two notable trends can be observed from the data. First, for $\mu \geq 0.7$, the simulation ends earlier than the specified 250 seconds. This is because $\mu = 0.7$ marks the point at which the flagella become completely tangled, as shown in the bottom-right frame of Fig. 2.6. As μ increases past 0.7, the tangling occurs even earlier in the simulation. Additionally, we notice that the number of average iterations starts to increase as μ increases. This is in line with our expectations, as larger μ values result

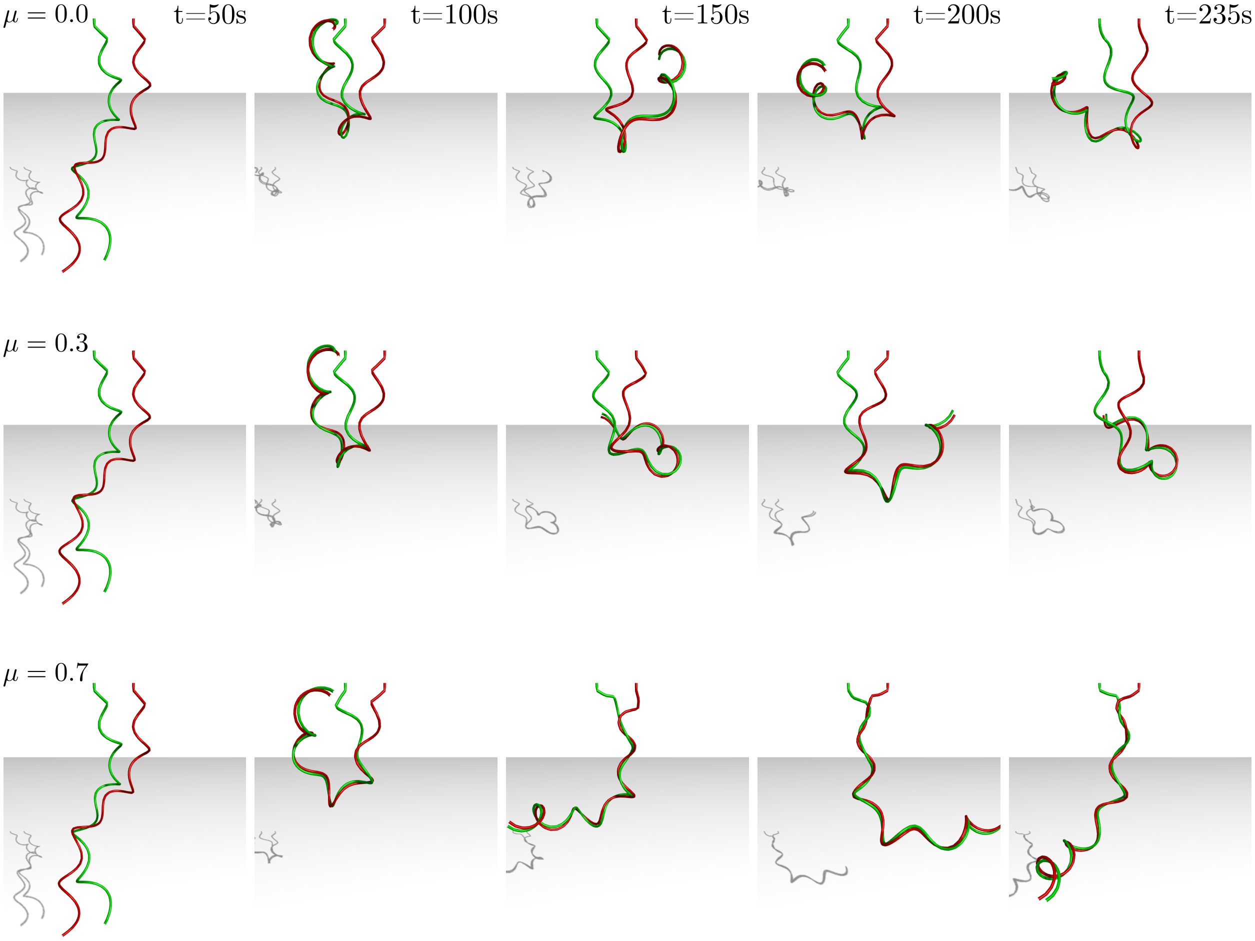


Figure 2.6: Rendered snapshots for $M = 2$ with varying friction coefficients. Each column indicates a moment in time as indicated by the time stamp in the top row. The first row shows the frictionless case $\mu = 0$ as a baseline. The second row has $\mu = 0.3$ where minor sticking can be observed as the point at where the flagella no longer contact is higher than the frictionless case. Still, $\mu = 0.3$ still has plenty of slipping allowing the flagella to not become coiled. As we increase μ to 0.7 in the third row, we can see the amount of sticking increase, ultimately resulting in the flagella becoming completing coiled.

in greater sticking between the flagella, which makes convergence more challenging and, in some cases, results in early termination of the simulation due to flagella entanglement.

Table 2.2: Friction results for varying friction coefficients. AIPTS stands for average iterations per time step. Total Iterations indicate the total number of Newton's method iterations that were necessary to complete the simulation. Sim End indicates the total *simulated* time. All simulations were set to run for 250 seconds. As can be seen, simulations with $\mu \geq 0.7$ end earlier due to excessive tangling of the flagella.

μ	AIPTS	Total Iterations	Sim End [sim s]
0.1	3.01	6.02×10^5	250
0.2	3.01	6.04×10^5	250
0.3	3.61	7.25×10^5	250
0.4	4.89	9.83×10^5	250
0.5	6.67	1.34×10^6	250
0.6	8.71	1.76×10^6	250
0.7	14.47	2.72×10^6	235.89
0.8	14.16	1.89×10^6	180.98
0.9	11.1	1.05×10^6	142.72
1.0	11.65	1.02×10^6	135.32

2.5 Numerical Study for Overhand Knot Tying

In this section, we validate our generated frictional contact forces against established theoretical relations, specifically focusing on the contact scenario of knot tying. We begin by validating our model against the theoretical relation for tightening trefoil knots as formulated in Ref. [25]. This validation is similar to our prior work in Ref. [36]. Additionally, we validate our contact model against the theoretical relation for tightening knots with varying unknotting numbers as formulated in Ref.[26].

2.5.1 Tightening a Trefoil Knot

Audoly et al. [25] developed a theoretical framework that describes the interplay between traction force and the geometric configuration of a trefoil knot as it undergoes tightening.

This theoretical relationship can be succinctly expressed as follows:

$$\frac{Fh^2}{EI} = \frac{\epsilon^4}{2} + \mu\sigma\epsilon^3, \tag{2.29}$$

where F is the traction force induced by friction; h is the rod radius; EI is the bending stiffness; μ is the coefficient of friction; σ is a constant (0.492 for trefoil knots), and $\epsilon = \sqrt{h/R}$ where R is the knot loop radius.

In our simulation, we implement the IMC approach to replicate the process of tightening a trefoil knot. During this simulation, we record the traction forces (F) while maintaining the following key parameters: cross-sectional radius $h = 0.0016$ m, Young's modulus $E = 1.8 \times 10^5$ Pa (resulting in $EI = \frac{E\pi h^4}{4}$), and friction coefficient $\mu = 0.1$. With non-dimensionalization, we define the normalized traction force as $\bar{F} = \frac{Fh^2}{EI}$. The excellent agreement we observed between our simulation results and the theoretical predictions, as demonstrated in Fig. 2.7, suggests the remarkable validation of our model against Audoly et al.'s theory.

2.5.2 Tightening Overhand Knots with Various Unknotting Numbers

In addition to the validation of [25], we also conduct a more rigorous validation against theory for overhand knots of various unknotting numbers stated in Ref. [26]. In this paper, Jawed et al. proposed the relation

$$n^2 \frac{h}{e} = \frac{1}{8\sqrt{3}\pi^2} g\left(\left[\frac{96\sqrt{3}\pi^2}{\mu} \cdot \frac{n^2 F h^2}{EI} \right]^{\frac{1}{3}} \right), \tag{2.30}$$

where n is the unknotting number; e is the end-to-end shortening, and $g(\cdot)$ is a known nonlinear equation detailed in the paper.

We use IMC to run simulations of tightening overhand knots for $n \in [1, 4]$ and $\mu = 0.07$ (all other rod parameters are the same as in 2.5.1. We plot the recorded non-dimensionalized

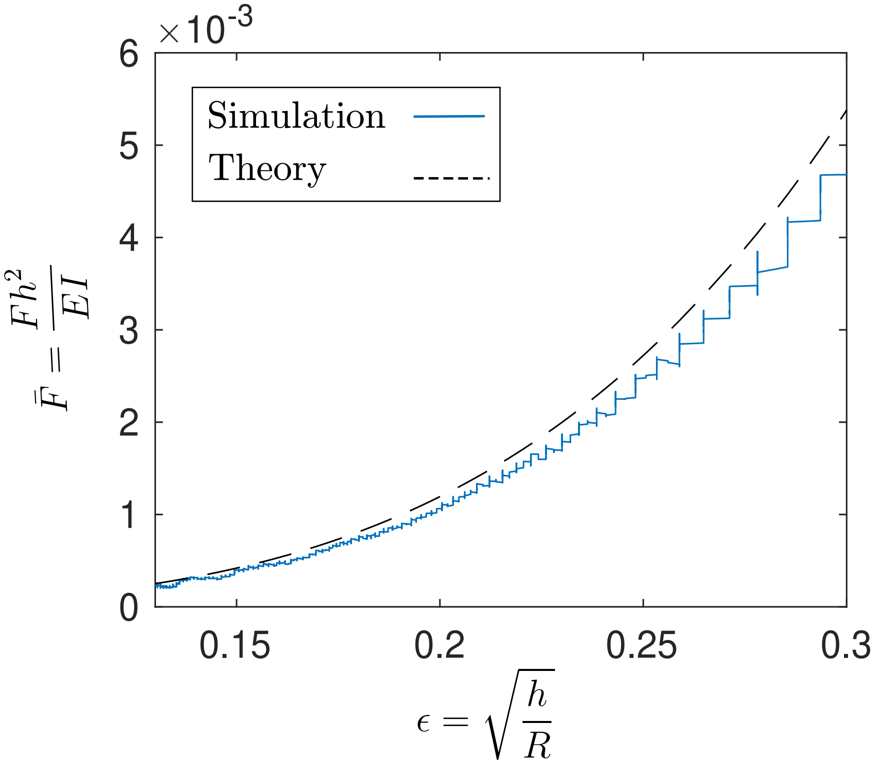


Figure 2.7: Comparison of non-dimensionalized traction force $\bar{F}$ between IMC simulation results and theory for tightening trefoil knots [25].

traction force $n^2\bar{F}$ against the non-dimensionalized end-to-end shortening $\bar{e} = n^2h/e$. Once again, we observe excellent agreement between simulation results and the plotted theory as shown in Fig. 2.8, thus further cementing the physical accuracy of IMC's frictional contact forces.

2.6 Summary and Outlook

In this chapter, we introduced the Implicit Contact Model (IMC), a fully implicit and penalty-based approach for tackling frictional contact handling. IMC can be embedded into a discrete-differential-geometry simulator (e.g., Discrete Elastic Rods) to capture the nonlinear dynamics of slender structures with frictional contact.

To test the performance of our contact model, we formulated an end-to-end simulation

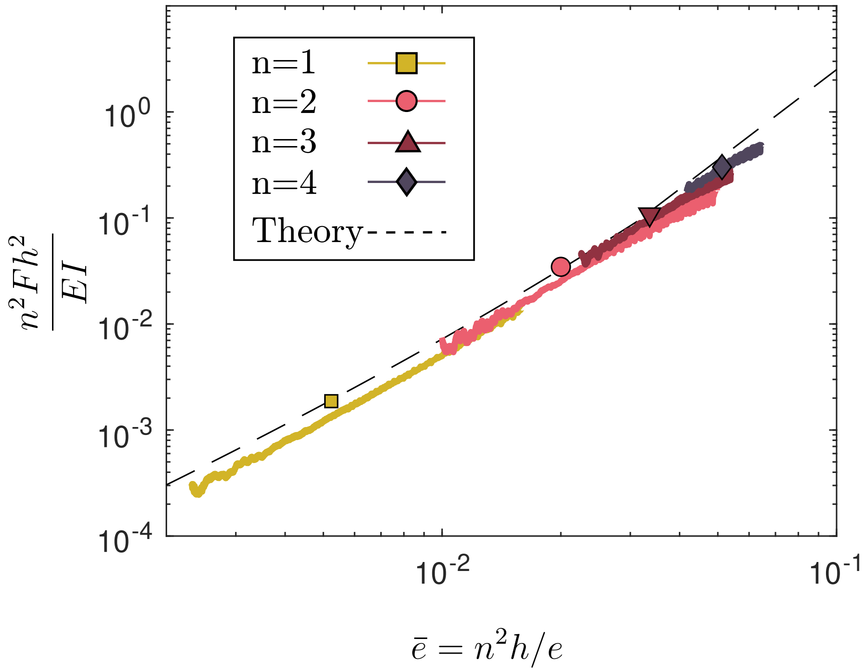


Figure 2.8: Comparison of non-dimensionalized traction force $\bar{F}$ between IMC simulation results and theory for tightening overhand knots with multiple unknotting numbers [26].

framework for the novel and difficult contact scenario of flagella bundling in viscous fluids. For this contact problem, we showed that IMC has comparable performance to the state-of-the-art while maintaining faster convergence. Furthermore, we showcased visually convincing frictional results in an imaginary viscous environment where friction is non-negligible. In addition to flagella bundling, we also demonstrated the efficacy of IMC by simulating the process of tightening an overhand knot. The physical accuracy of our framework is validated by comparing with rigorous theory in prior literature. Given IMC's efficiency and accuracy, this work represents a significant stride toward the ultimate goal of a simulation framework that can faithfully mimic real-world scenarios. It also sets the stage for the study of mechanics in rod assemblies with frictional contact, as explored in Ch. 3.

For future work, we wish to improve upon the stability and robustness of our friction model. Despite the implicit formulation, the number of iterations necessary to converge starts to increase as μ increases. Additionally, the adoption of deep learning to imbue our simulations with physics-based dynamics holds great potential. While our current framework relies on Coulomb friction, an approximation suitable for macroscopic dynamics, there are situations, such as microscale interactions, where it may fall short. Neural networks, with their ability to approximate real-world phenomena, beckon as a promising direction for future exploration.

CHAPTER 3

Snap Buckling in Overhand Knots

When an overhand knot tied in an elastic rod is tightened, it can abruptly transform its shape via a phenomenon called snap buckling. In this chapter, we employ a blend of simulations based on discrete differential geometry (DDG) and hands-on tabletop experiments to delve into the initiation of buckling concerning knot configuration, rod dimensions, and friction. Our experimental setup involves gradually pulling apart the open ends of an overhand knot, inducing snap buckling within the knot's loop. This intriguing occurrence is termed "inversion", as the loop dramatically shifts from one side of the knot to the opposite. This inversion phenomenon emerges due to the coupling of elastic energy between the braid (the section of the knot in self-contact) and the loop (the segment with both ends linked to the braid). We implement a numerical framework that merges Discrete Elastic Rods with a frictional contact-constraint approach to systematically investigate inversion within overhand knots. Remarkably, the numerical simulations accurately replicate knot inversion and exhibit substantial agreement with the results from our physical experiments. To acquire a deeper physical comprehension of the inversion process, we further develop a simplified knot model that obviates the necessity for simulating self-contact. This simplified model facilitates visualizing the bifurcation responsible for snap buckling.

We describe the underlying motivation and relevant literature in § 3.1. The statement of the problem is given in § 3.2. The numerical framework for simulating an overhand knot is given in § 3.3. The experimental setup for studying inversion is stated in § 3.4. Then, a quantitative study for all contributing factors for the system is shown in § 3.5. In § 3.6, we also construct a simplified topology model to explain why snap buckling happens in overhand knots. Finally, we conclude a summary and point out the future avenues in § 3.7. The content of this chapter has appeared in Ref. [64].

3.1 Motivation

Spread across the globe, knots manifest as intricate geometric patterns of slender elastic rods that intertwine with themselves. These formations serve diverse functions in our daily lives, spanning from the utilitarian, as seen in shoelaces, to the ornamental, exemplified by Chinese knotting. Given their pervasive presence and intricate blend of topology and mechanics, knots stand as a compelling yet demanding research subject.

To harness the potential of knots within various structures, it becomes imperative to delve into the knots' mathematical and mechanical properties. In this chapter, we focus on the mechanics of overhand knots – a fundamental knot type – subjected to forceful tightening at their extremities. In certain instances, this tightening triggers a snap buckling instability, abruptly altering the knot's configuration. Self-contact is a pivotal element contributing to this mechanical phenomenon, which occurs within the entangled rod itself. Mathematically, a knot is a theoretically infinitesimal closed loop; however, in practicality, knots possess finite thickness and often sport open ends. Consequently, when addressing knot-related mechanical challenges, the intricacies of self-contact must be navigated. In Ch. 2, we introduced an array of techniques – including the impulse method, constraint-based methods, and penalty-energy-based methods – to simulate interacting rod assemblies. Opting for the constraint-based approach is rooted in its well-defined mathematical representation of contact and its accuracy, demonstrated in prior research like sound generation due to object interactions [65].

The iconic overhand knot takes center stage among the diverse tapestry of knots, emerging ubiquitously in nature and daily activities. Prior studies probing overhand knots have scrutinized the impact of tightening forces and their effects on equilibrium shapes. For instance,

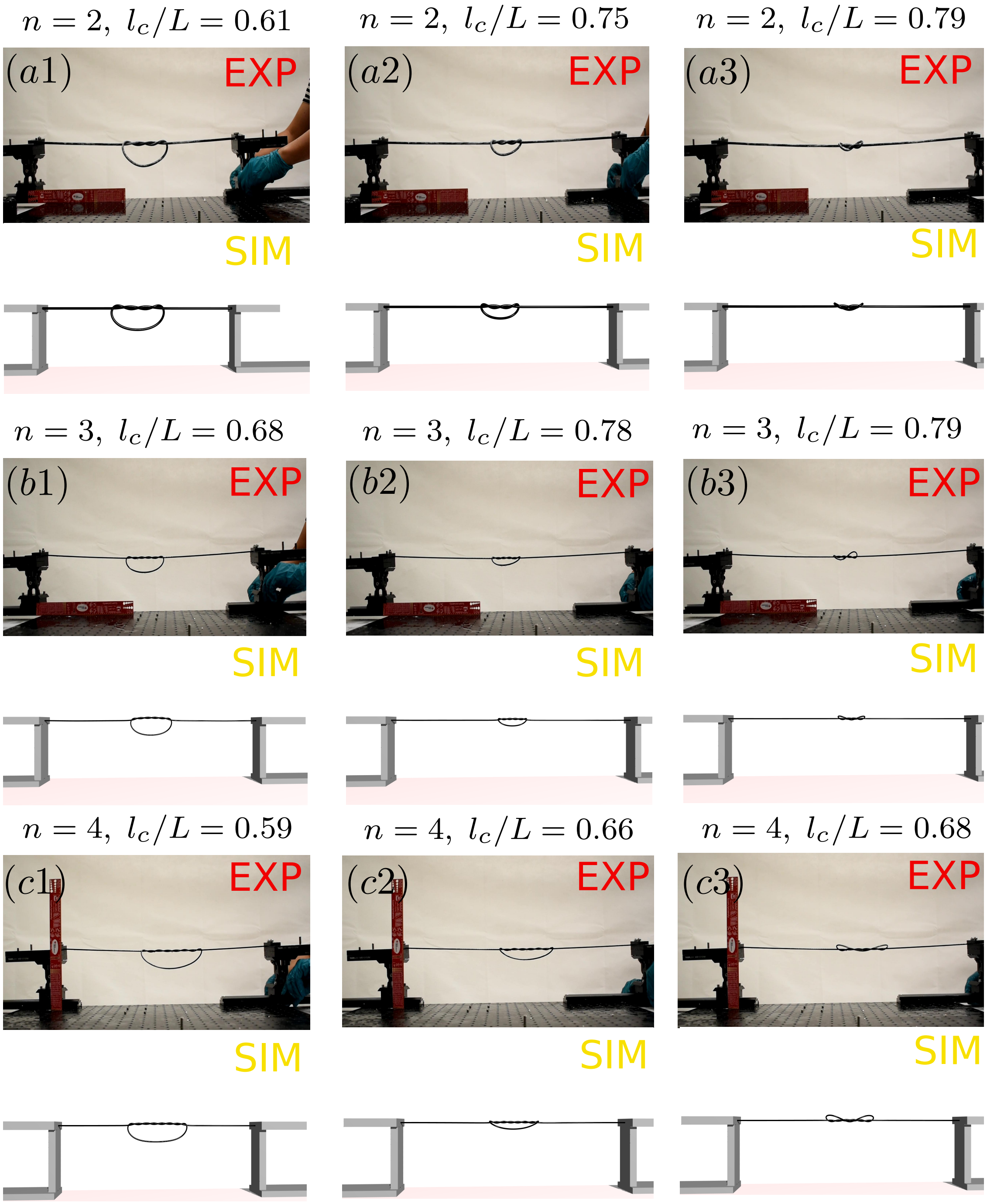


Figure 3.1: Snapshots of overhand knots in the tightening process from both experiments and simulations.

Audoly et al. [25] established the link between traction forces and the shortening of trefoil (3_1) and cinquefoil (5_1) knots. Przybyl et al. [66] delved into the curvature and torsion of a contracting trefoil knot. Jawed et al. [26] formulated an analytical expression intertwining tightening forces, friction, and contraction in overhand knots with varying crossing numbers. Moulton et al. [67] unearthed stable configurations for open trefoil knots sans self-contact. Our focus, however, centers on the snap buckling within self-contacting overhand knots.

The concept of snapping and bifurcation – the rapid transition between stable configurations due to external triggers – finds broad applications across nature and engineering. Instances range from the snapping of Venus flytraps [68] to the bistable states of slap bracelets [69], the snap-through of toy poppers [70], the buckling of helical rods with robotics [16], and the exploration of bistability in soft robots [71].

Self-contact, often expressed as an inequality constraint, significantly influences buckling instabilities in slender structures, though its inclusion complicates buckling analyses considerably. Coleman et al. [72] investigated the stability of self-contacting supercoiled plasmids mimicking DNA structures. Thompson et al. [73] utilized buckling analysis to study the bounds of topoisomerase relaxation in twisted DNA. Closest to our pursuit is the phenomenon depicted in Figure 1 of Bergou et al. [13], followed by Clauvelin et al. [74]. Bergou et al. [13]'s Figure 1 unveiled continuous and discontinuous shape transformations upon rotating the ends of a trefoil knot. Building on this, Clauvelin et al. [74] derived asymptotic solutions for trefoil and cinquefoil knots fastened onto infinitely long rods featuring linear tails and a circular loop. They established that a fusion of tension and twist could lead to asymptotic knot solution buckling via either helical deformations in the tails or out-of-plane buckling in the circular loop.

Inspired by this revelation, we employ simulations and experiments to demonstrate that overhand knots can undergo a rapid shape shift solely through tightening without the imposition of external twisting, as depicted in Fig. 3.1. This transformation, termed "inversion" hereafter, stems from the simplicity of boundary conditions during tightening, suggesting its topology-driven nature. We explore the mechanics and reasons underlying the occurrence of inversion and give a thorough quantification of mechanical responses during overhand knot tightening via simulations and experiments, unraveling the intricacies of inversion.

3.2 Problem Statement

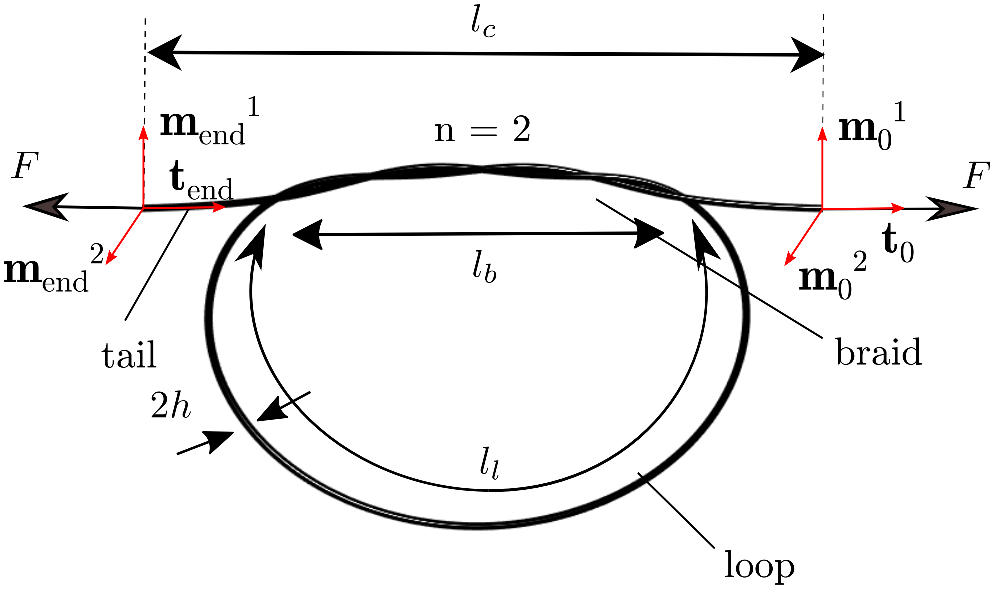


Figure 3.2: Configurations of overhand knots with different unknotting numbers.

In Fig. 3.2, the topology of an overhand knot is shown. These knots consist of a braid, characterized by an arc length l_b, harmonizing with a loop delineated by an arc length l_l, all anchored by two tails, applied by external boundary conditions. The braid's topological intricacies are characterized by the unknotting number n, representing the count of loop traversals required for complete unraveling. The rod constructing the knot has a cross-sectional radius of h, while the overhand knot's ends are clamped, spanning a distance denoted as l_c. While Fig. 3.2 omits its depiction, an essential intrinsic parameter remains concealed—namely, the friction coefficient μ.

As shown in Fig. 3.1, when traction is exerted on both ends of an overhand knot with $n > 1$, the loop abruptly inverts and establishes contact with a specific point along the braid region. This phenomenon, driven by the overhand knot's geometry, is identified as a snap-buckling process. The crux of this problem involves an overhand knot formed from an inherently straight elastic rod, with boundary conditions applied on its clamped ends, inducing a consistent tightening speed. Importantly, during this imposition, the manipulated rod ends maintain uniform material frames, remaining devoid of relative rotation. The introduction of non-zero relative rotation between manipulated ends can usher in more intricate knot behaviors. For instance, Bergou et al. [13] demonstrated that introducing relative rotations to these material frames induces topological changes in a trefoil knot. This study, however, focuses on cases where relative rotation remains nil, shedding light on how the knot's topology governs nonlinear structural behavior rather than delving into boundary rotation influence. Notably, inversion, a geometry-dependent phenomenon, is paramount, overshadowing the influence of Young's Modulus, an assertion validated in the supplementary information of Ref. [64]. The experimental rods, fabricated from rubber, yield a measured Young's Modulus via the assessment of the gravito-bending length $L_{gb} = (\frac{Eh^2}{8\rho g})^{1/3}$, where ρ is volumetric density and g is gravity [7]. Whenever a rod engages a rigid substrate, its form adheres to the gravito-bending length L_{gb}. Based on this, we can compute Young's modulus E from the experiments. For the experimental and simulation scenarios, Young's modulus is set at 1.8 MPa, while Poisson's ratio ν assumes a value of 0.5 (indicative of incompressibility), resulting in a shear modulus $G = E/(2(1 + \nu)) = 0.6$ MPa. The inversion point's experimental measurement hinges on the clamped length l_c, with the comprehensive analysis of the influence of contributing factors, encompassing n, h, and μ, expounded upon in § 3.5.

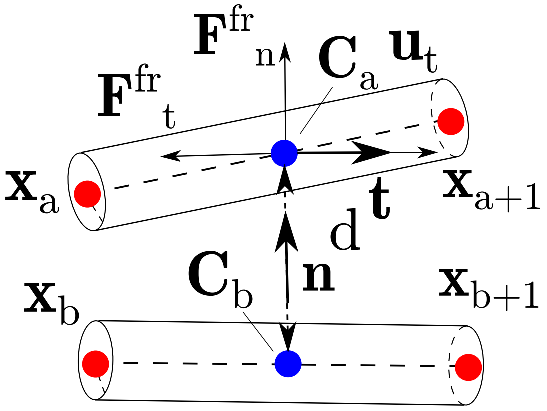


Figure 3.3: Discrete schematic of two contacted rod segments.

3.3 Numerical Framework

In this section, we look into the integration of a constraint-based method, drawing inspiration from the works of Jean and Moreau [37, 38], Bertails et al. [75], and Alart and Curnier [39], with Discrete Elastic Rods (DER), as outlined in Bergou et al. [13, 14], for simulating frictional contact within a knot. The incorporation of constraint-based methods to simulate frictional contact responses typically entails intricate numerical handling. Here, we present a straightforward two-step procedure that seamlessly merges DER with the constraint-based approach. In the initial step, frictional contact forces are explicitly computed, followed by their integration into the DER framework in the subsequent step, enabling the dynamic update of the simulated rod's state. This bifurcated approach not only unveils the equations of motion (EOM) grounded in DER but also elucidates the process of friction force calculation within these EOM. This pragmatic strategy ensures the faithful representation of nonlinear knot behaviors.

Given that the formulation of DER is extensively detailed in § 1.1, we focus our attention predominantly on expounding the formulation pertaining to frictional contact.

Within our numerical framework, friction is modeled using Coulomb's friction model.

A remarkable revelation has been that framing frictional contact responses as linear complementary constraints accurately depicts Coulomb's friction. In this study, we implement the formulation proposed by Alart and Curnier [39] and Bertails et al. [75] to calculate the frictional contact occurring between two contacting edges, illustrated in Fig. 3.3

Fig. 3.3 elucidates that when two edges enter contact, linear complementary constraints materialize between the contact forces $\mathbf{F}^{fr}$ and the contact points. Each segment adopts the form of a cylinder, enabling the computation of contact points $\mathbf{C}_a$ and $\mathbf{C}_b$ via the algorithm articulated by Ref. [43]. The relative tangential velocity ($\mathbf{C}_a$ relative to $\mathbf{C}_b$) is denoted as $\mathbf{u}_t$, and the contact force is demarcated as $\mathbf{F}^{fr} = \mathbf{F}^{fr}_t + \mathbf{F}^{fr}_n$.

Note that $\mathbf{n} = (\mathbf{C}_a - \mathbf{C}_b)/d$ designates the normal direction, while $\mathbf{t} = \mathbf{u}_t/\|u_t\|$ signifies the tangential direction. Grounded in the linear complementary constraints established by Jean and Moreau [37] and Alart and Curnier [39], the correlation between the contact distance d and the normal component of the frictional contact force adheres to Signorini's condition:

$$\mathbf{F}^{fr}_n \cdot \mathbf{n} \geq 0 \perp 0 \leq d - 2h. \tag{3.1}$$

When two segments contact ($d - 2h = 0$), the linear complementary condition between the friction force $\mathbf{F}^{fr}_t$ and tangential relative velocity $\mathbf{u}_t$ is:

$$\mu \|\mathbf{F}^{fr}_n\| + \mathbf{F}^{fr}_t \cdot \mathbf{t} \geq 0 \perp 0 \leq \|\mathbf{u}_t\|. \tag{3.2}$$

The relative velocity $\mathbf{u}$ at the contact point, the minimum distance d, and the contact's normal direction $\mathbf{n}$ and tangential direction $\mathbf{t}$ can all be obtained from the DER formulation. Consequently, we can deduce the frictional contact responses $\mathbf{F}^{fr}$ by satisfying the linear complementary condition. In the study by Alart and Curnier [39], they introduce an augmented Lagrangian formulation, enabling the transformation of the linear complementary constraints in Eq. 3.1 and Eq. 3.2 into the following expressions:

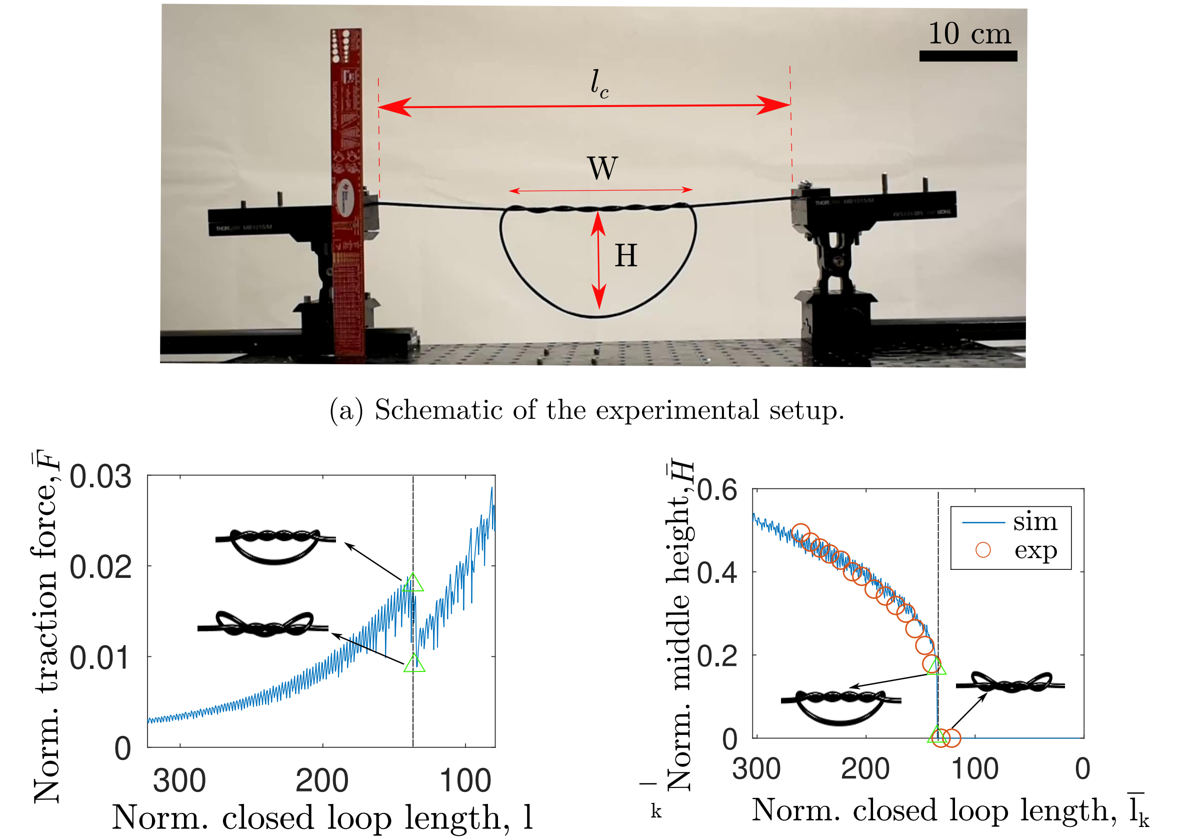
$$\mathbf{F}_{ac}(\mathbf{F}^{fr}) = 0 = \begin{bmatrix} \mathbf{F}^{fr} \cdot \mathbf{n} - P_{\mathbb{R}_+}(\mathbf{F}_n^{fr} \cdot \mathbf{n} - \rho d) \\ -(\mathbf{F}^{fr} \cdot \mathbf{t})\mathbf{t} - P_{D(\mu(\mathbf{F}^{fr} \cdot \mathbf{n} - \rho d)_+)}(-(\mathbf{F}^{fr} \cdot \mathbf{t})\mathbf{t} - \rho \mathbf{u}_t) \end{bmatrix}, \tag{3.3}$$

Eq. 3.3 linearized the formulation of Eq. 3.1 and 3.2 proposed by Alart and Cunier [39]. In Eq. 3.3, $P(\cdot)$ is the projection operator, and ρ is the penalty coefficient users define. Eq. 3.3 can be solved with Newton's method. In this study, an explicit method is used to solve the linear complementary conditions in Eq. 3.1 and 3.2 since we feel interested in the behavior of the knot under quasi-static conditions. The configuration of the knot at the previous time step (which is known) is used to approximate the required quantities, e.g., contact distance d, contact normal $\mathbf{n}$, and tangent direction $\mathbf{t}$ in the linear complementary constraints so that we can solve the frictional contact responses from Eq. 3.3 robustly. Then, the fractional contact responses will be added as external forces in DER for simulating the nonlinear dynamics of knots.

3.4 Experimental Setup and Illustration of Inversion Point

In this section, we show the details of our experimental setup designed to investigate the inversion process quantitatively. As shown in Fig. 3.4a, our methodology employs two fixtures to affix and simultaneously pull the two ends of a suspended overhand knot. These fixtures execute coordinated movements, moving collinearly but in opposite directions.

Recalling that the end-to-end length also termed the clamped length, is denoted by l_c, we introduce two novel parameters, namely H and W, to gauge the geometry of the closed loop within the overhand knot. Fig. 3.4a illustrates that H is the knot's height, delineated by the vertical symmetry axis of the knot, while W represents the knot's width, characterized as the separation between the two ends of the braid region. As the knot progressively tightens,



(a) Schematic of the experimental setup.

(b) Plot of the normalized knot closed loop size plots $\bar{l}_k$ v.s. normalized traction forces $\bar{F}$.

(c) Plot of the normalized knot closed loop size $\bar{l}_k$ v.s. normalized middle height of the knot $\bar{H}$.

Figure 3.4: Schematic of the experimental setup and illustration of the inversion point.

l_c increases while H and W gradually reduce. Notably, for knots with unknotting numbers $n > 1$, a specific point during the tightening process triggers a sharp descent of H to zero, which we define as the inversion point. The formal elucidation of the inversion point is presented in Figs. 3.4b and 3.4c.

Mathematically, the knot's geometry assumes particular significance concerning the closed loop. The size of this closed loop, as depicted in Fig. 3.4a, can be approximated as $L - l_c$, where l_c represents the end-to-end length and L signifies the total rod length utilized for knot formation. We employ $l_k = L - l_c$ to quantify the inversion point in our experimental setup. In our simulations, the inversion point is evaluated using two metrics: traction force and geometric alterations. To enable comparability and dimensionless representation, we introduce the notation $\bar{(\cdot)}$ to signify dimensionless quantities. We normalize l_k, F, and H as follows:

$$\begin{aligned}
\bar{l}_k &= \frac{l_k}{h} = \frac{L - l_c}{h}, \\
\bar{F} &= \frac{Fh^2}{EI}, \\
\bar{H} &= \frac{H}{W},
\end{aligned} \tag{3.4}$$

where $EI = \frac{\pi}{4}Eh^4 (EI = EI_1 = EI_2)$ is the bending stiffness, h is the rod's cross-sectional radius, E is Young's modulus, and F is the traction force. In our simulation setup, the rod's Young's modulus is assigned as $E = 1.8$ MPa, coinciding with the measured value from experimental rods. Furthermore, a Poisson's ratio of 0.5 is applied, assuming incompressibility of the material.

As shown in Fig. 3.4b, it can be seen that the traction force $\bar{F}$ increases initially as the knot is tightened and then proceeds to rapidly decrease and then rebound once $\bar{l}_k$ reaches the inversion point. Fig. 3.4b demonstrates the trajectory of traction force $\bar{F}$, which initially ascends as the knot tightens, only to swiftly decline and subsequently rebound as $\bar{l}_k$ attains the inversion point. A comparable trend can be observed in Fig. 3.4c, where $\bar{H}$ converges to zero at the point of inversion. Remarkably, the inversion points extracted from simulations and experiments coincide. Consequently, in this study, we opt to employ $\bar{H}$ as the benchmark for comparing inversion points derived from simulations and experiments.

Given the geometric foundation of the problem, the contributing factors are those exerting a direct influence on the closed-loop's geometry. In light of this, three variables emerge as pivotal elements shaping inversion: the rod radius h, the unknotting number n, and the friction coefficient μ. In subsequent sections, we investigate how these parameters impact the system's inversion point. For both simulations and experiments, as depicted in Fig. 3.4, we maintain the parameter values of $h = 1.6$ mm, $L = 1$ m, $n = 3$, and $\mu = 0.1$.

3.5 Contributing Factors

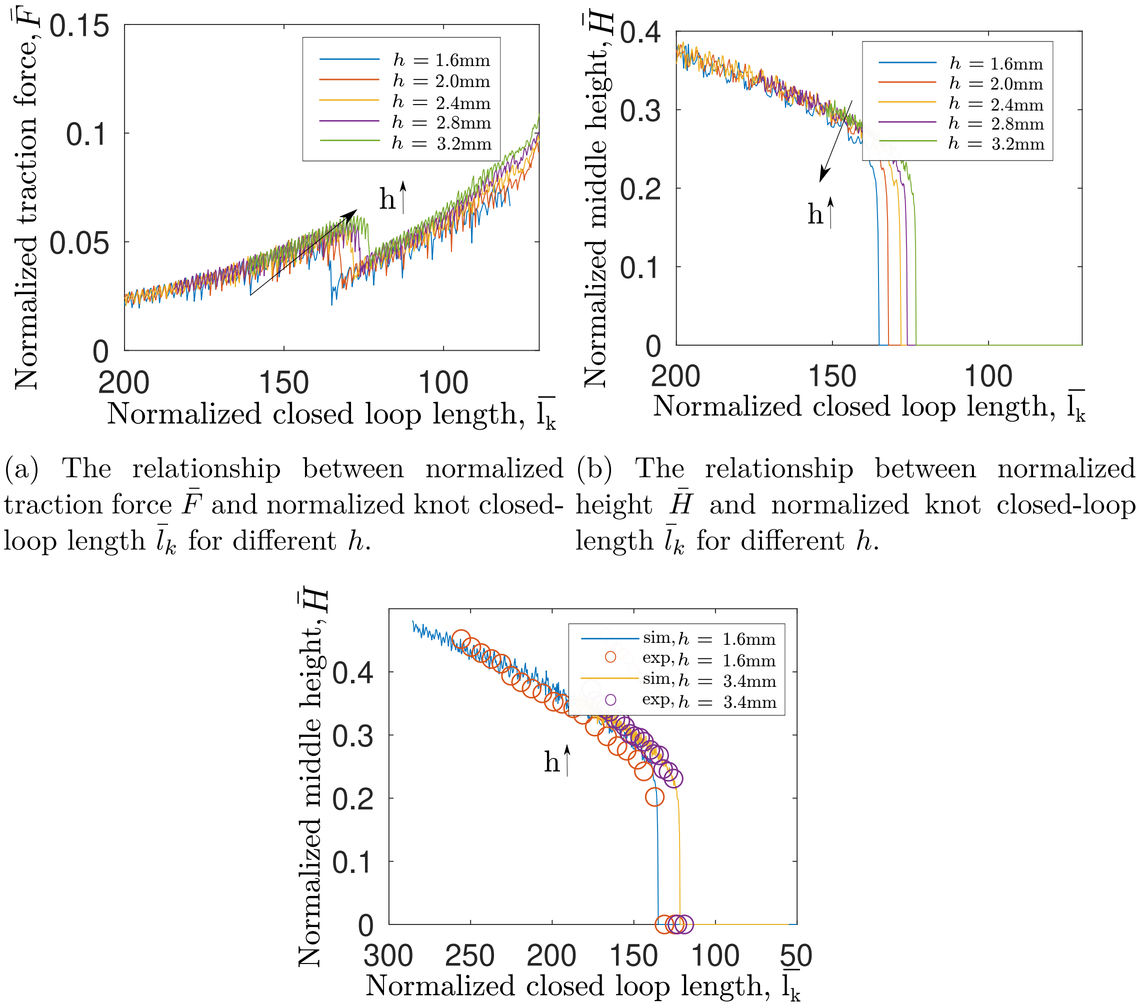
Through both numerical and experimental studies, we comprehensively explore the systemic impact of the three primary contributing factors: h, n, and μ. In our simulation setup, we systematically traverse a range of values for each parameter (n, h, and μ), while keeping the other two parameters constant. However, due to the material constraints inherent in the physical rods, our experimental setup only covers a subset of the entire parameter range encompassed in the simulations. It is worth noting that despite this limitation, our experimental results effectively demonstrate the discernible influence exerted by the contributing factors (n, h, and μ) on the inversion point.

3.5.1 Effect of Rod Radius

We commence by quantitatively exploring the impact of h on the inversion phenomenon. While analyzing each contributing factor, we maintain the other factors at constant levels to accurately discern the isolated influence of the parameter under investigation. Three distinct plots are generated for each contributing factor: simulation results of the normalized traction forces $\bar{F}$ with respect to the normalized knot close-loop length $\bar{l}_k$, simulation results of the normalized height $\bar{H}$ with respect to $\bar{l}_k$, and comparison of simulation and experimental results.

To understand the effect of h, we set the unknotting number n to 3 and fix μ at 0.1. Furthermore, we set the pulling speed at both ends to $\Delta u = 1$ mm/s to ensure quasi-static responses from the system.

Fig. 3.5a shows the evolution of $\bar{F}$ as a function of $\bar{l}_k$ for various rod radius values, specifically $h \in [1.6, 2.0, 2.4, 2.8, 3.2]$ mm. In this plot, a distinct surge in force magnitude manifests at analogous $\bar{l}_k$ values as h increases. Put differently, the inversion point consistently emerges at comparable non-dimensionalized closed-loop lengths, regardless of variations in rod radius. Indeed, the closed knot loop length is normalized by h, rendering inversion more prone



(a) The relationship between normalized traction force $\bar{F}$ and normalized knot closed-loop length $\bar{l}_k$ for different h.

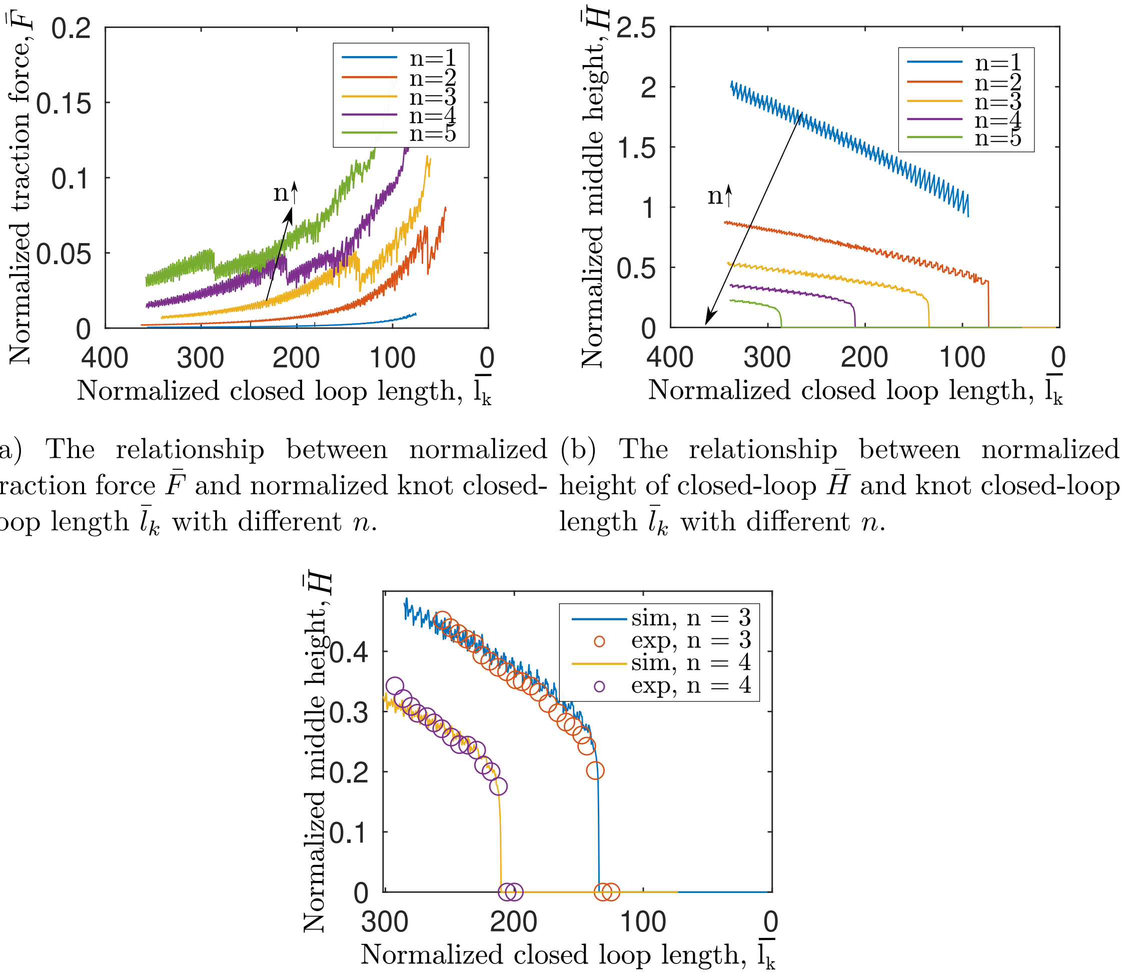
(b) The relationship between normalized height $\bar{H}$ and normalized knot closed-loop length $\bar{l}_k$ for different h.

(c) Comparison between simulated and experimental results of normalized height $\bar{H}$ v.s. normalized knot closed-loop length $\bar{l}_k$.

Figure 3.5: Effect of normalized rod radius h on inversion point.

to occur with larger rod radii. Similarly, Fig. 3.5b shows $\bar{H}$ with respect to $\bar{l}_k$, pinpointing the inversion points through abrupt drops in $\bar{H}$.

For the purpose of experimental validation, two rods with distinct radii were employed: one with $h = 1.6$ mm and another with $h = 3.4$ mm. Notably, as illustrated in Fig. 3.5c, a striking congruence emerges between simulation outcomes and experimental observations,



(a) The relationship between normalized traction force $\bar{F}$ and normalized knot closed-loop length $\bar{l}_k$ with different n.

(b) The relationship between normalized height of closed-loop $\bar{H}$ and knot closed-loop length $\bar{l}_k$ with different n.

(c) Comparison between simulated and experimental results of normalized Height $\bar{H}$ v.s. normalized knot closed-loop length $\bar{l}_k$.

Figure 3.6: Effect of unknotting number n on inversion point.

corroborating the consistency between the two for both rod radii.

3.5.2 Effect of Unknotting Number

Moving forward, we shift our focus to the next contributing factor: the unknotting number n. Just as before, we maintain the other contributing factors at fixed values to isolate the influence of n on the system. Specifically, we set $h = 1.6$ mm and $\mu = 0.1$.

Fig. 3.6a and Fig. 3.6b describe the respective relationships between $\bar{F}$ and $\bar{H}$ with respect to $\bar{l}_k$ for a range of n values. Notably, in both plots, the junctures of abrupt change coincide with the inversion points. For the experimental phase, a rod with $h = 1.6$ mm and $\mu = 0.1$ was utilized to craft overhand knots with n values of 3 and 4. As depicted in Fig. 3.6c, we once again observe a high degree of agreement between the simulated and experimental outcomes.

Overall, our investigation demonstrates that the critical value of $\bar{l}_k$ generally escalates as the unknotting number n increases, concomitant with a decline in clamped length $\bar{l}_c$. Moreover, it is worth noting that an inversion phenomenon requires n to be greater than 1; for cases where $n = 1$, the system does not exhibit inversion.

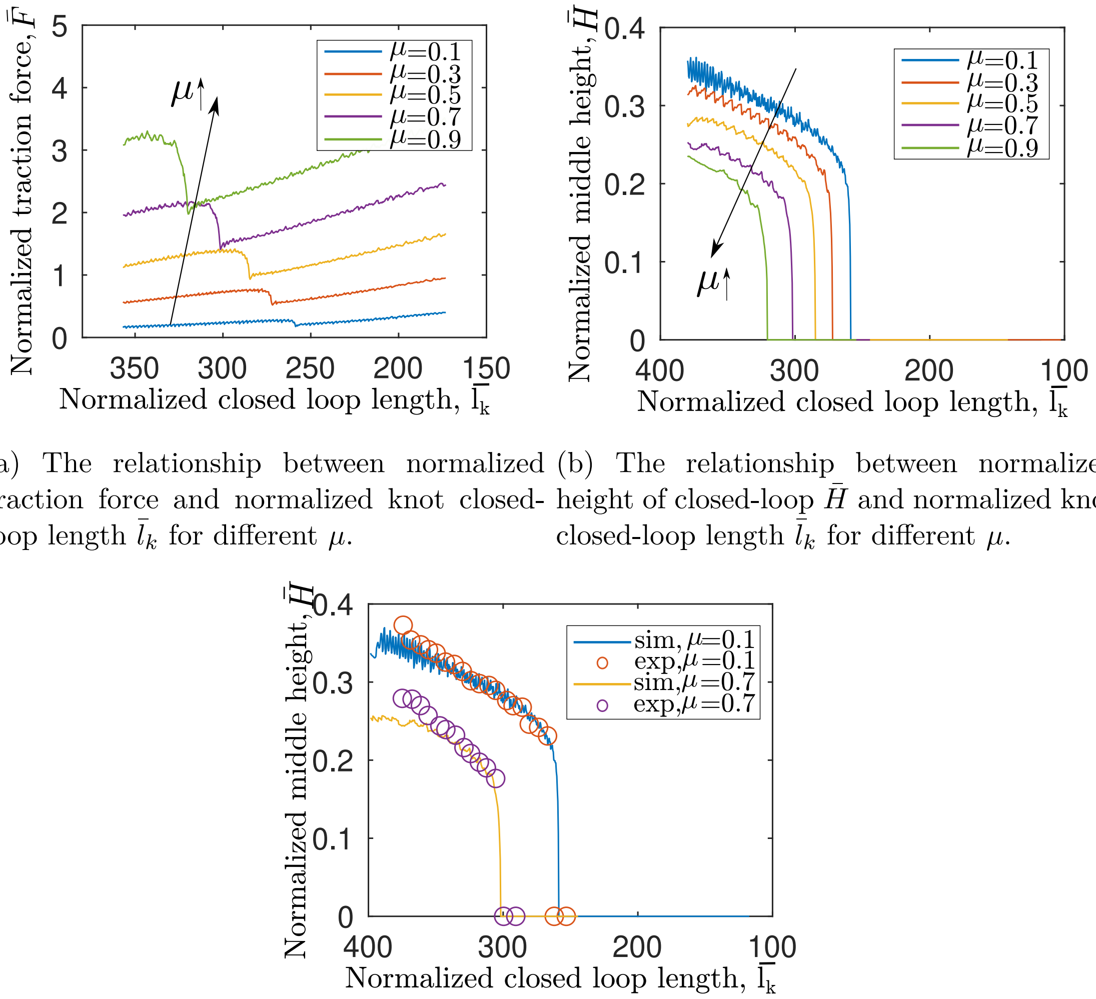
3.5.3 Effect of Friction Coefficient

Finally, we investigate the impact of the friction coefficient μ on the system while keeping the other contributing factors constant: setting n to 3 and h to 3.4 mm.

Figs. 3.7a and 3.7b illustrate the relationships between $\bar{F}$ and $\bar{H}$ respectively concerning to $\bar{l}_k$ for various μ values. Notably, the points of abrupt change correspond to the inversion points. It is evident that as μ increases, the normalized clamped length at which inversion occurs decreases. In the experimental phase, a rod with normalized radius $h = 3.4$ mm and the unknotting number $n = 3$ was employed. By altering the surface of the rod using chalk and glycerin, the friction coefficient μ was manually adjusted. Chalk yielded $\mu = 0.7$, while glycerin resulted in $\mu = 0.1$.

The outcomes are once again in solid agreement between simulation and experiment, as demonstrated in Fig. 3.7c.

Through numerical simulations and desktop experiments, we have ascertained that the parameters h, μ, and n significantly impact overhand knots' inversion process. While we have now obtained the effects of h, μ, and n on the inversion points of overhand knots, the



(a) The relationship between normalized traction force and normalized knot closed-loop length $\bar{l}_k$ for different μ.

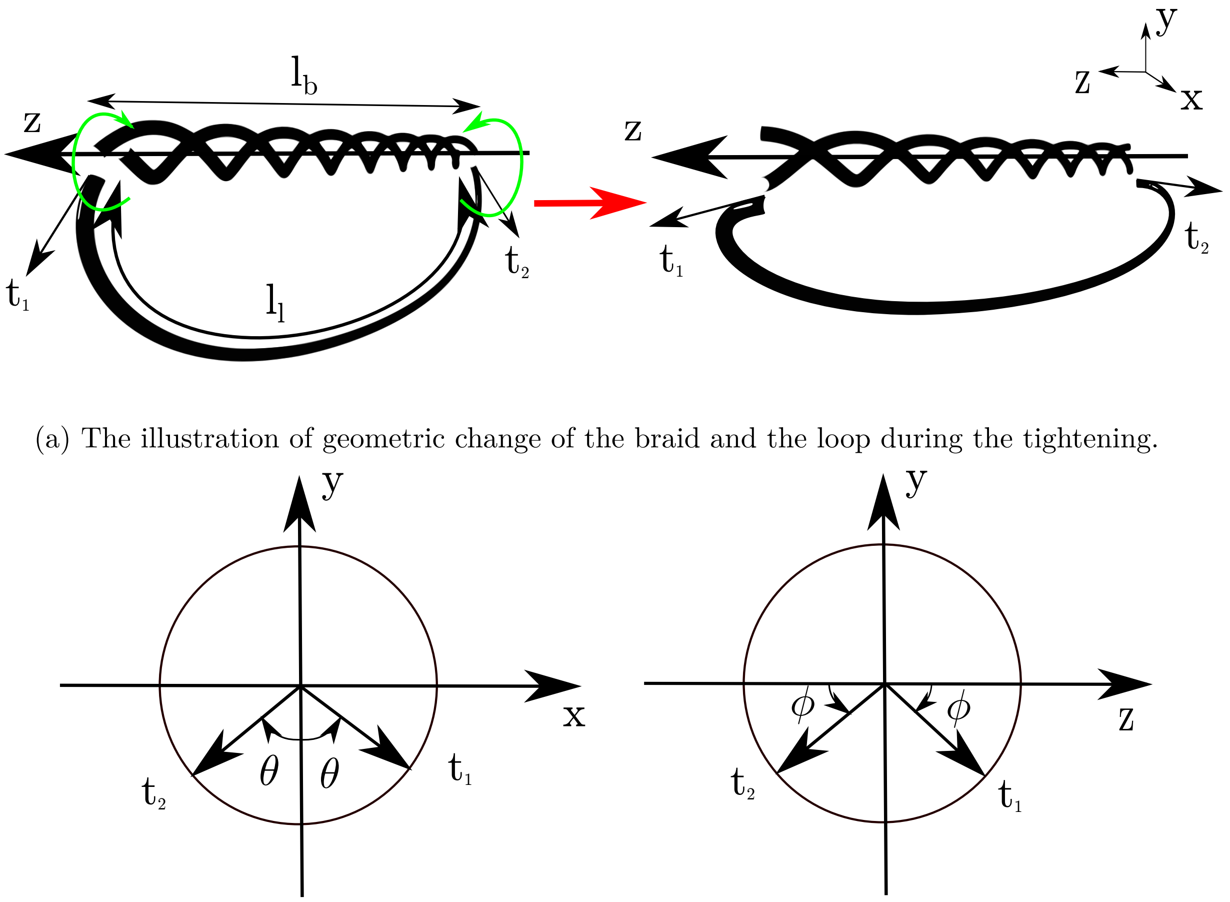
(b) The relationship between normalized height of closed-loop $\bar{H}$ and normalized knot closed-loop length $\bar{l}_k$ for different μ.

(c) Comparison between simulated and experimental results of normalized height $\bar{H}$ v.s. normalized knot closed-loop length $\bar{l}_k$.

Figure 3.7: Effect of friction coefficient μ on inversion point.

underlying mechanisms of these contributing factors and the reasons behind the occurrence of inversion require further investigation.

In § 3.6, we delve into the development of a simplified knot model, one that doesn't necessitate self-contact simulation, to expound upon the mechanisms driving inversion and to delineate the roles played by the contributing factors in the system.



(a) The illustration of geometric change of the braid and the loop during the tightening.

(b) Illustrations of polar angle θ.

(c) Illustrations of rotation angle ϕ.

Figure 3.8: Simplified model of an overhand knot. The overhand knot can be divided into the braid and the loop. During the tightening process, tangents at two ends of the braid will rotate along the x and z-axes. Polar angle θ along z-axis and rotation angle ϕ along x-axis are shown in (b) and (c), respectively

3.6 Topology Analysis

In this section, we present a simplified model for understanding the occurrence of inversion during the tightening process of overhand knots. The primary objective of this simplified model is to bypass the need for simulating frictional self-contact in the numerical simulations of the knot. However, while aiming to capture the geometrically nonlinear behavior of the rod, the simplified model still relies on numerical simulations of the contact-free knot loop.

As shown in Fig. 3.2, an overhand knot can be divided into three segments: the loop, the braid, and the tails. The analysis in the preceding section, drawn from simulations and experiments, revealed that the braided region slackens during inversion, with parts of the braid entering the loop region. Remarkably, the length of the tail regions remains constant throughout inversion. This suggests that the inversion phenomenon emerges from the interplay of elastic energy between the braid and loop regions. Consequently, the simplified model disregards the tails and concentrates solely on the closed loop, encompassing the braided and loop segments.

The schematic of the simplified knot model is illustrated in Fig. 3.8. This model's rods with helical centerlines approximate the braided region, while friction within the braided area and self-contact within the loop region are neglected. In congruence with the formulation in § 3.2, the loop's arc length is denoted as l_l, and the arc length of the braided region is denoted as l_b. The tangent directions at the loop's ends are denoted as $\mathbf{t}_1$ and $\mathbf{t}_2$. As the knot depicted in Fig. 3.8a is tightened, it transforms into the configuration illustrated in Fig. 3.8a. Throughout this process, we assume the knot's shape remains antisymmetric relative to the YZ plane. This permits us to parameterize the rod's boundary conditions using two angles, θ and ϕ, as depicted in Figs. 3.8b and 3.8c.

The angle θ represents the rotation of the tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ around the $\mathbf{z}$-axis. This rotation is mirrored at the opposite end of the braid due to the continuity of the closed loop. Hence, both ends of the braid rotate about the $\mathbf{z}$-axis by the same angle θ. Fig. 3.8c illustrates that the angle ϕ corresponds to the rotation of the tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ around the $\mathbf{x}$-axis.

Given that the simplified model excludes the tails of the rod, six geometric parameters become relevant: l_b, l_l, h, n, θ, and ϕ. It is worth noting that for a sufficiently small radius h, the axial length of the rod within the braided segment approximately equals $2l_b$, leading to an approximation of $l_k \approx l_b + l_l$. Two dimensionless parameters can be derived among the three length-dependent parameters, l_b, l_l, and h. In this case, we opt for $\bar{l}_k = l_k/h$ and $\epsilon = \frac{l_b}{l_l}$

as our chosen dimensionless parameters. It is important to observe that the nondimensional lengths $\bar{l}_b = l_b/h$ and $\bar{l}_l = l_l/h$ can be deduced from $\bar{l}_k$ and ϵ using the relationships:

$$\bar{l}_b = \frac{\epsilon \bar{l}_k}{1+\epsilon} \qquad \bar{l}_l = \frac{\bar{l}_k}{1+\epsilon}. \tag{3.5}$$

As discussed before, our observations indicate that inversion arises from the interplay of elastic energies within the braid and loop regions. Consequently, we denote the elastic energy of the braid as E_b and the elastic energy of the loop as E_l. To quantify these energies in a dimensionless manner, we introduce the following nondimensional energies:

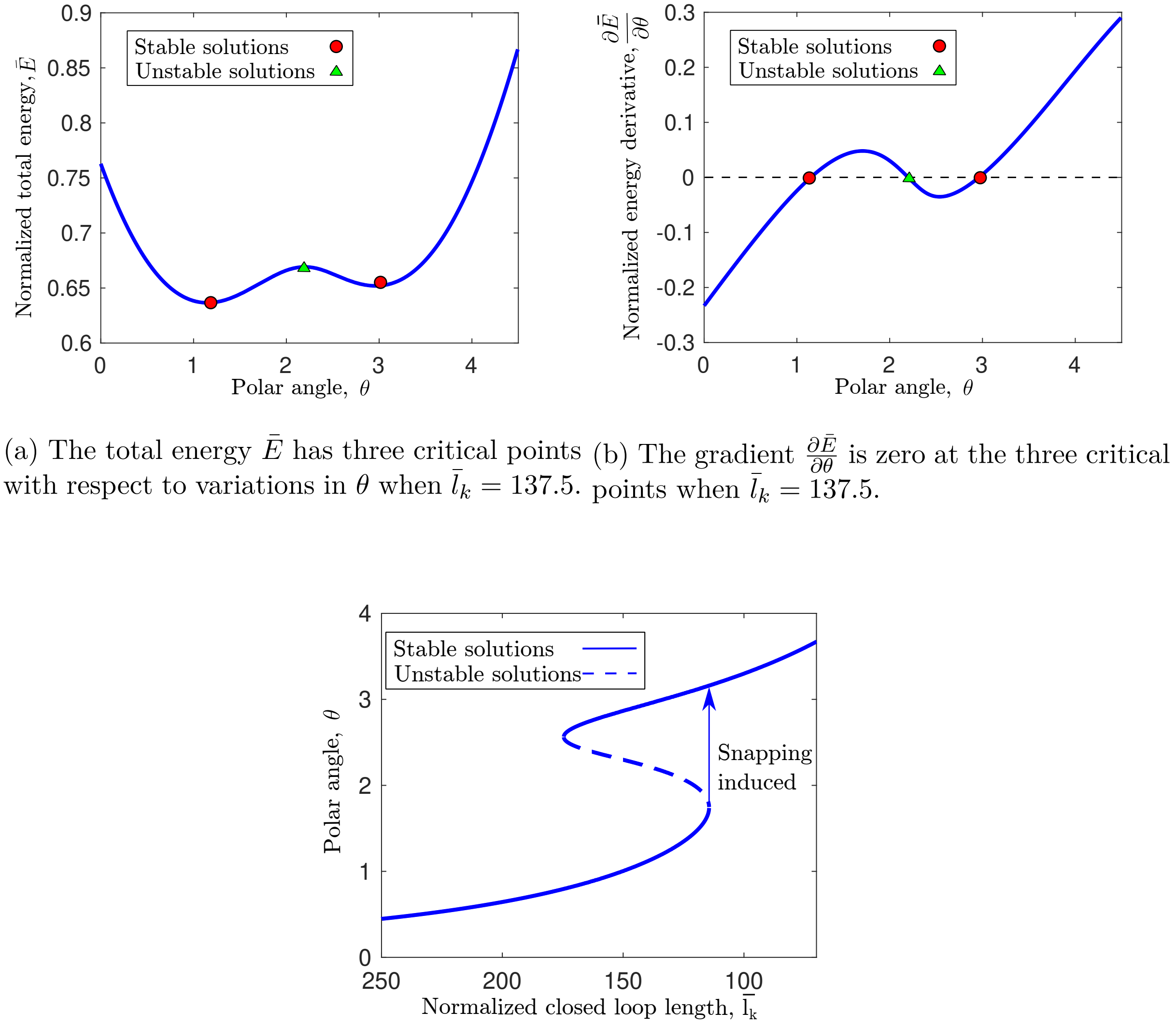
$$\bar{E}_b = \frac{E_b h}{EI} \qquad \bar{E}_l = \frac{E_l h}{EI}. \tag{3.6}$$

To calculate the nondimensional braid energy $\bar{E}_b$, we initiate by parameterizing the helical rods constituting the braid region using the following parameterization:

$$\begin{aligned}
\bar{x} &= \cos(t), \\
\bar{y} &= \sin(t), \\
\bar{z} &= \bar{b}t. \\
\bar{b} &= \frac{\bar{l}_b}{2n\pi - 2\theta}, \\
t &\in [-n\pi + \theta, n\pi - \theta].
\end{aligned} \tag{3.7}$$

By disregarding the twisting energy of the helical rods within the braided region, equation 3.7 can be employed to derive the nondimensional bending energy of the braided region $\bar{E}_b$ in relation to the nondimensional curvature $\bar{\kappa}$:

$$\begin{aligned}
\bar{\kappa} &= \frac{1}{1+\bar{b}^2}, \\
\bar{E}_b &= 2\bar{\kappa}^2(n\pi - \theta)\sqrt{1+\bar{b}^2}
\end{aligned} \tag{3.8}$$



(a) The total energy $\bar{E}$ has three critical points with respect to variations in θ when $\bar{l}_k = 137.5$.

(b) The gradient $\frac{\partial \bar{E}}{\partial \theta}$ is zero at the three critical points when $\bar{l}_k = 137.5$.

(c) The values of θ at the critical points undergo two-fold bifurcations as the normalized closed-loop length $\bar{l}_k$ decreases, with the second fold corresponding to the snap-through instability observed in previous sections.

Figure 3.9: Effects of rotation angle θ and normalized knot closed-loop length $\bar{l}_k$ on the energy $\bar{E}$ in the simplified knot model with $\phi = \pi/3$, $\epsilon = 0.7$, and $n = 3$.

Next, we describe the energy of the loop $\bar{E}_l$. Given that the loop represents a suspended elastic rod within three-dimensional space, its configuration and elastic energy are

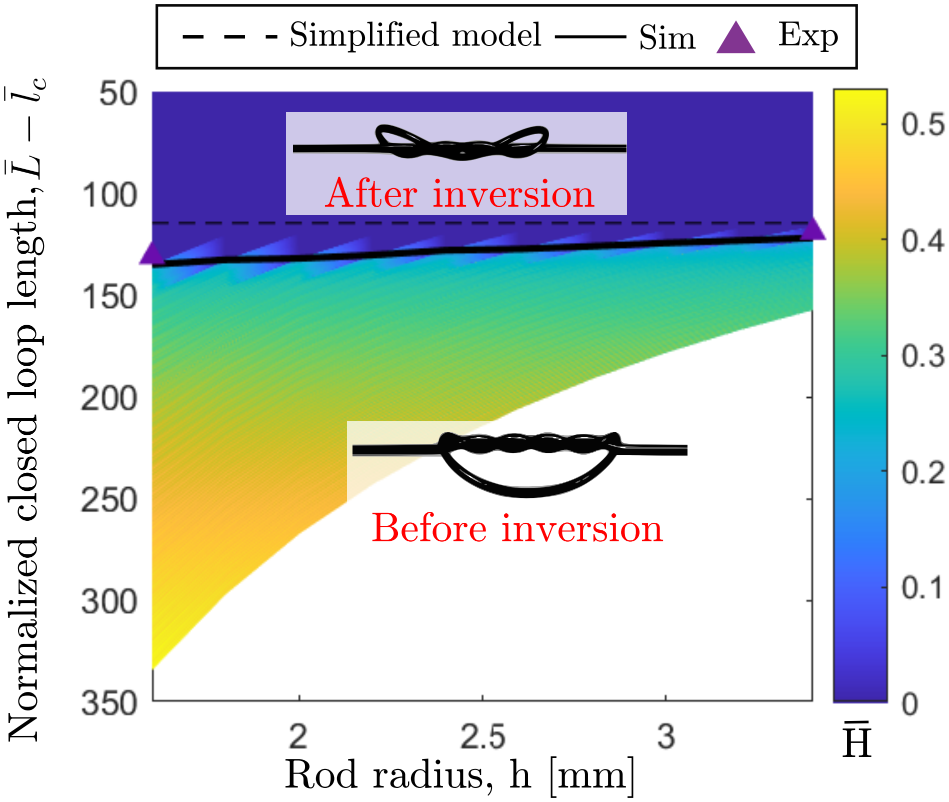


Figure 3.10: Comparison between simplified model, the full DER simulation, and the experimental measurements for different values of h with $n = 3$ and $\mu = 0.1$.

not straightforward to determine analytically. However, we can leverage the DER-based numerical simulations discussed in earlier sections to ascertain the shape of the loop. Consequently, we can formulate $\bar{E}_l$ as a function:

$$\bar{E}_l = f(\phi, \theta, \bar{l}_k, \epsilon), \tag{3.9}$$

where the value of $f(\cdot)$ is found with DER.

We can now express the total knot energy $\bar{E}(\phi, \theta, \bar{l}_k, \epsilon)$ of the closed-loop as $\bar{E}_b + \bar{E}_l$. Equilibrium configurations of the simplified knot must satisfy $\nabla \bar{E} = 0$. However, rather than coupling the analytical expression for the braid energy $\bar{E}_b$ with the numerically computed loop energy $\bar{E}_l$, we will instead focus on the influence of a single variable on the equilibrium

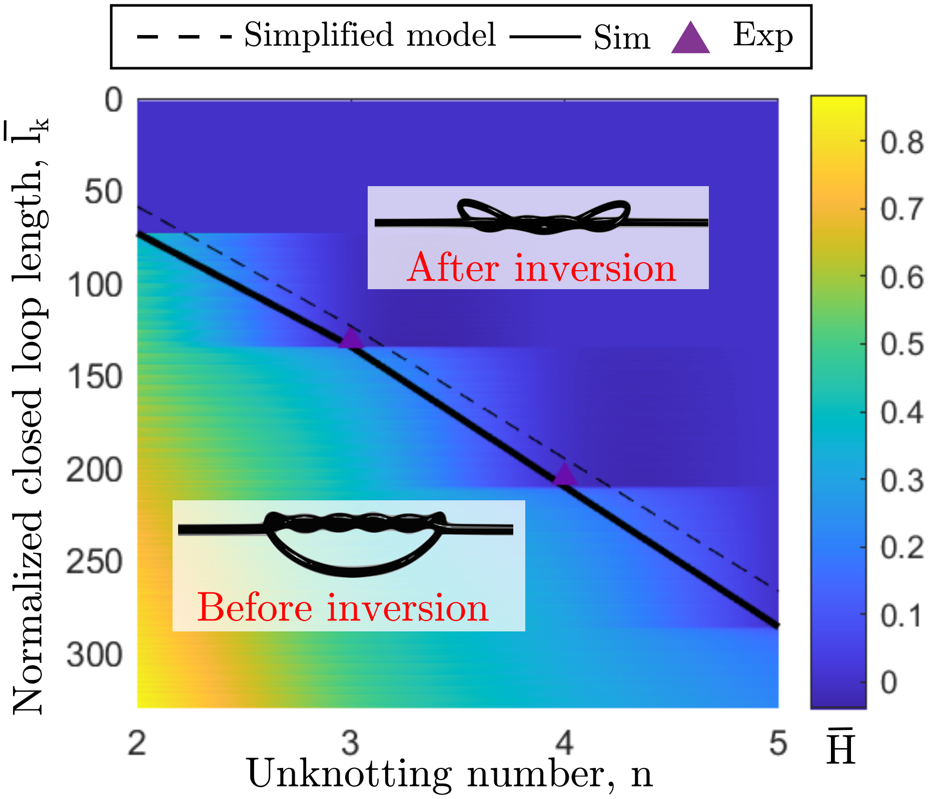


Figure 3.11: Comparison between simplified model, the full DER simulation, and the experimental measurements for different values of n with $h = 1.6$mm and $\mu = 0.1$.

configurations and their stability. Specifically, we will investigate how the rotation angle θ affects knot configurations. The numerical simulations described in previous sections showed that inversion typically occurs at the approximate parameter values $\phi = \pi/3$ and $\epsilon = 0.7$. Therefore, these nondimensional parameter values will be used in the remainder of this section.

To observe the effects of θ on equilibrium configurations for the parameters $\phi = \pi/3$ and $\epsilon = 0.7$, we conducted DER-based numerical simulations to solve for the total energy $\bar{E}$ with a rod radius of $h = 1.6$ mm, an unknotting number of $n = 3$, a clamped length corresponding to $\bar{l}_k = 137.5$, and with fixed rotation angles in the range $\theta \in [0, 5]$ radians. Equilibrium configurations of the knot correspond to rotation angles at which $\partial\bar{E}/\partial\theta = 0$. Fig. 3.9a displays the value of $\bar{E}$ for $\theta \in [0, 5]$. In this plot, we observe three critical points, with the outer two critical points being stable and the intermediate critical point being unstable. It is important to emphasize that these critical points correspond to equilibria with respect to variations in θ only, while the other parameters (namely, ϕ and ϵ) are held constant. Fig. 3.9b illustrates the value of $\partial \bar{E}/\partial \theta$ for the range of θ, confirming the presence of three critical points.

We can investigate how the number of critical points changes as the knot closed-loop's clamped length $\bar{l}_k$ varies. Fig. 3.9c depicts the values of θ at the critical points as the clamped length varies. We observe that there is a single stable critical point for sufficiently large $\bar{l}_k$. As $\bar{l}_k$ decreases, two additional critical points, one stable and one unstable, emerge in a fold bifurcation. Further reduction of $\bar{l}_k$ leads to the unstable critical point colliding with the original stable critical point in a second-fold bifurcation, ultimately resulting in a single stable critical point for sufficiently small $\bar{l}_k$. The second fold bifurcation corresponds to the snap-through instability observed in previous sections. It is worth noting that while the simplified model exhibits hysteresis between the two-fold bifurcations, this hysteretic behavior is not observed in experiments due to the lack of consideration for gravity-induced sag in our model.

The simplified knot model can now be employed to predict how changes in the rod's radius and the unknotting number affect the critical clamped length at which inversion occurs. This allows us to compare the simplified knot model's predictions with the simulated and experimental results from previous sections. For rod radius values in the $[1.5, 3.5]$mm range and an unknotting number of $n = 3$, we calculated the critical closed-loop length $\bar{l}_k$ at which the snap-through instability occurs. These results are shown by the dashed line in Fig. 3.10, alongside the experimental and simulation outcomes based on the full DER model, including contact and friction with a $\mu = 0.1$ friction coefficient. A similar comparison is presented in Fig. 3.11, where the radius is held constant at $h = 1.6$mm while the unknotting number is varied. Notably, the simplified model's predictions are in good qualitative and quantitative agreement with the experimental and simulated results. Specifically, we observe that $\bar{l}_k$

remains unchanged with variations in the rod radius $\bar{h}$ and increases with the unknotting number n, corresponding to a decreasing clamped length $\bar{l}_c$.

3.7 Summary and Outlook

In this chapter, we delved into the phenomenon of snap-through buckling in overhand knots when subjected to tension at their clamped ends. We introduced a discrete differential geometry-based model known as the one-dimensional Discrete Elastic Rods, coupled with a frictional contact constraint method, to investigate how the topology of overhand knots influences their physical behavior during tightening.

Through a combination of experimental observations and numerical simulations, we revealed that intricate knot topologies can give rise to fascinating phenomena such as snap buckling. Moreover, we demonstrated that parameters like μ, h, and n play pivotal roles in shaping the geometry of overhand knots and determining the snap buckling point. Specifically, as μ, h, and n increase, the process of inversion, which corresponds to snap buckling, occurs earlier, implying that looser overhand knots are more prone to inversion.

To offer deeper insights into the mechanics behind inversion, we conducted a topological analysis of overhand knots and constructed a simplified model that predicts the inversion point. This analysis clarified that inversion results from the intricate energy coupling between contact zones and non-contact regions within the knot. Consequently, the sudden loosening of the braid, known as the self-contact zone, leads to inversion. Beyond this point, the loop remains in contact with the braided region.

While some previous research has touched on the effects of self-contact on snap buckling in elastic structures, our work contributes numerically and analytically to this challenging field. The phenomenon of inversion in overhand knots highlights how self-contact-induced topologies can induce captivating behaviors in elastic structures. The energy coupling between contact and non-contact regions triggers inversion, yielding overhand knots that entangle themselves in intricate patterns. This understanding of inversion holds promise in explaining tangles in complex polymers like DNA and proteins, and it could even inspire novel strategies for designing and manipulating knotted structures.

CHAPTER 4

Deep Learning Force Manifolds from the Physical Simulation of Robotic Paper Folding

Robotic manipulation of slender objects presents a significant challenge, particularly when dealing with large and nonlinear deformations. Traditional approaches, such as imitation learning, have been employed to address deformable material manipulation. However, these methods often lack generality and are prone to critical failures when material, geometry, or environmental properties change.

This chapter focuses on a complex deformable manipulation task: folding a piece of paper using only a single robot manipulator. To address this challenge, we propose a data-driven framework that combines physically-accurate simulation and machine learning. They train a deep neural network to predict the external forces acting on the manipulated paper based on the grasp position.

One key innovation is the use of scaling analysis, which results in a control framework that remains robust against variations in material and geometry. We use path planning over a “neural force manifold” generated by their trained model to produce robot manipulation trajectories that prevent sliding. Notably, the offline trajectory generation in this framework is 15 times faster than previous physics-based folding methods.

The trained model’s fast inference speed allows for real-time visual feedback, enabling closed-loop sensorimotor control. The chapter demonstrates the effectiveness of this framework through real-world experiments, showing significant improvements in robotic manipulation performance compared to state-of-the-art folding strategies. This improvement is observed even when manipulating paper objects of varying materials and shapes.

The underlying motivation is in § 4.1. Then, we state the targeting problem in § 4.2. With the problem outlined, we do the physical analysis and outline a learned physics-informed model in § 4.3. Next, we formulate a model predictive control for planning the optimal strategy for the folding task in § 4.4. Next, we introduce our designed robotic system in § 4.5. Finally, we give the conclusions and outlooks in § 4.7. The content of this chapter is from Ref. [76].

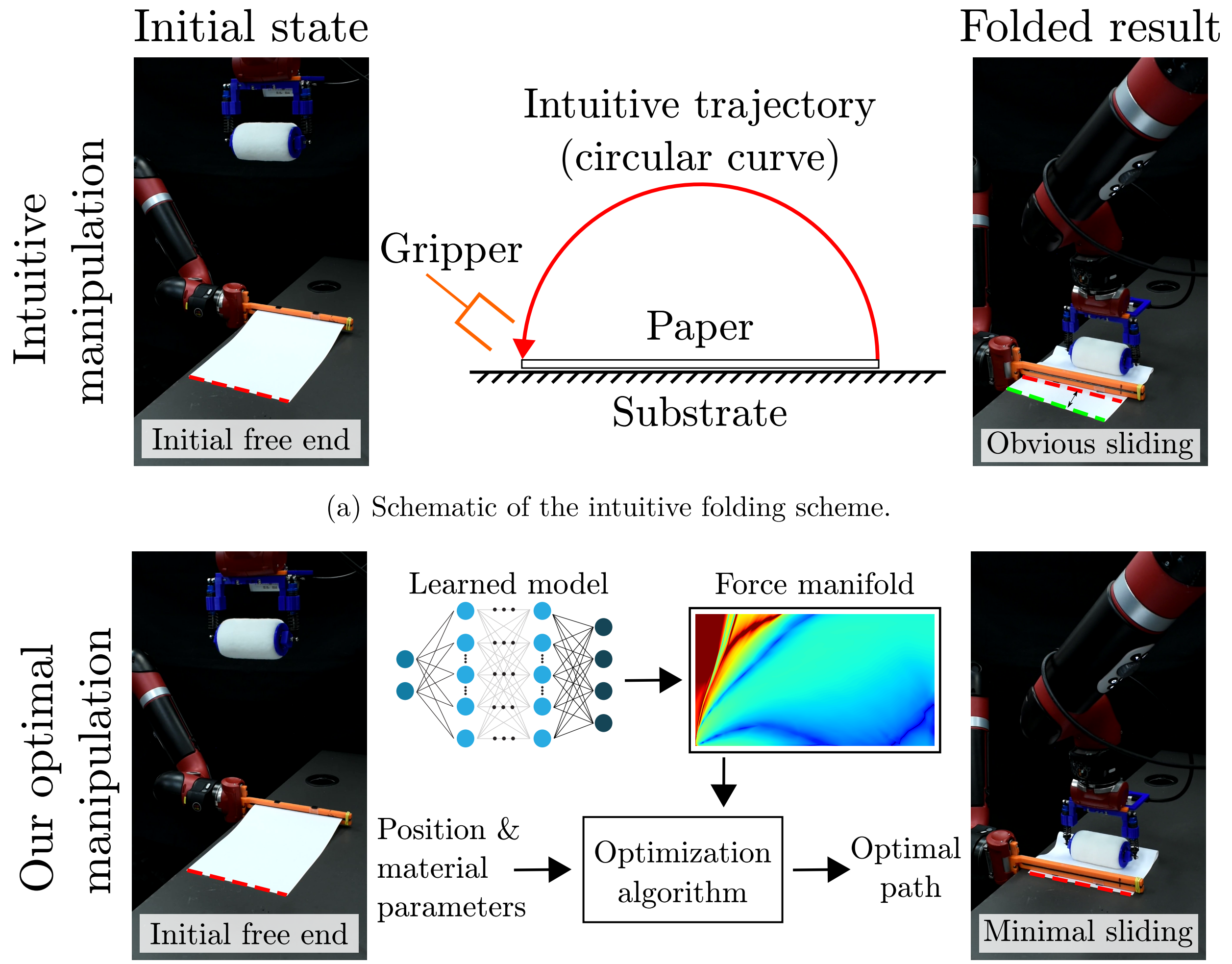
4.1 Motivation

From shoelaces to garments, we encounter slender and pliable structures in our daily lives. These structures are notable for their capacity to undergo significant deformations even under moderate forces, such as the influence of gravity. Consequently, the manipulation of deformable objects by robots poses a formidable challenge, as it necessitates the anticipation of the object's future deformations to successfully complete manipulation tasks.

Previous research has primarily concentrated on the manipulation of two specific categories: cloth [77, 78, 79, 80, 81, 82] and ropes [83, 84, 85, 86, 87, 9, 88, 89]. Consequently, the challenge of effectively manipulating a diverse range of deformable objects remains largely unaddressed. In this chapter, we tackle a particularly intricate deformable manipulation task - the folding of paper.

Historically, research addressing the folding problem can be categorized into four primary approaches: mechanical design-based solutions, vision-based solutions, learning-based solutions, and model-based solutions.

Mechanical design-based approaches typically involve addressing the folding problem using highly specialized manipulators or end effectors. Early methods included the use of specialized punches and dies for sheet metal bending [90]. More recently, there have been



(a) Schematic of the intuitive folding scheme.

(b) Schematic of the optimal folding scheme.

Figure 4.1: Half valley folding for A4 paper with (a) intuitive manipulation and (b) our designed optimal manipulation. An intuitive manipulation scheme such as tracing a semicircle experiences significant sliding due to the bending stiffness of the paper, resulting in a poor fold. By contrast, our optimal manipulation approach achieves an excellent fold by taking into consideration the paper's deformation to minimize sliding.

developments in highly specialized manipulators for robotic origami folding [91]. While these methods reliably achieve repeatable folding, they are often confined to specific fold types, geometries, or materials.

Vision-based approaches entail folding deformable materials based solely on visual input to generate folding motions. These approaches are frequently applied to folding clothes [80, 92, 78] since clothes are highly pliable, and their deformation behavior can be predicted with ease based on specific actions. However, these techniques may not readily extend to paper folding due to paper's higher stiffness, causing it to resist deformation if not handled correctly.

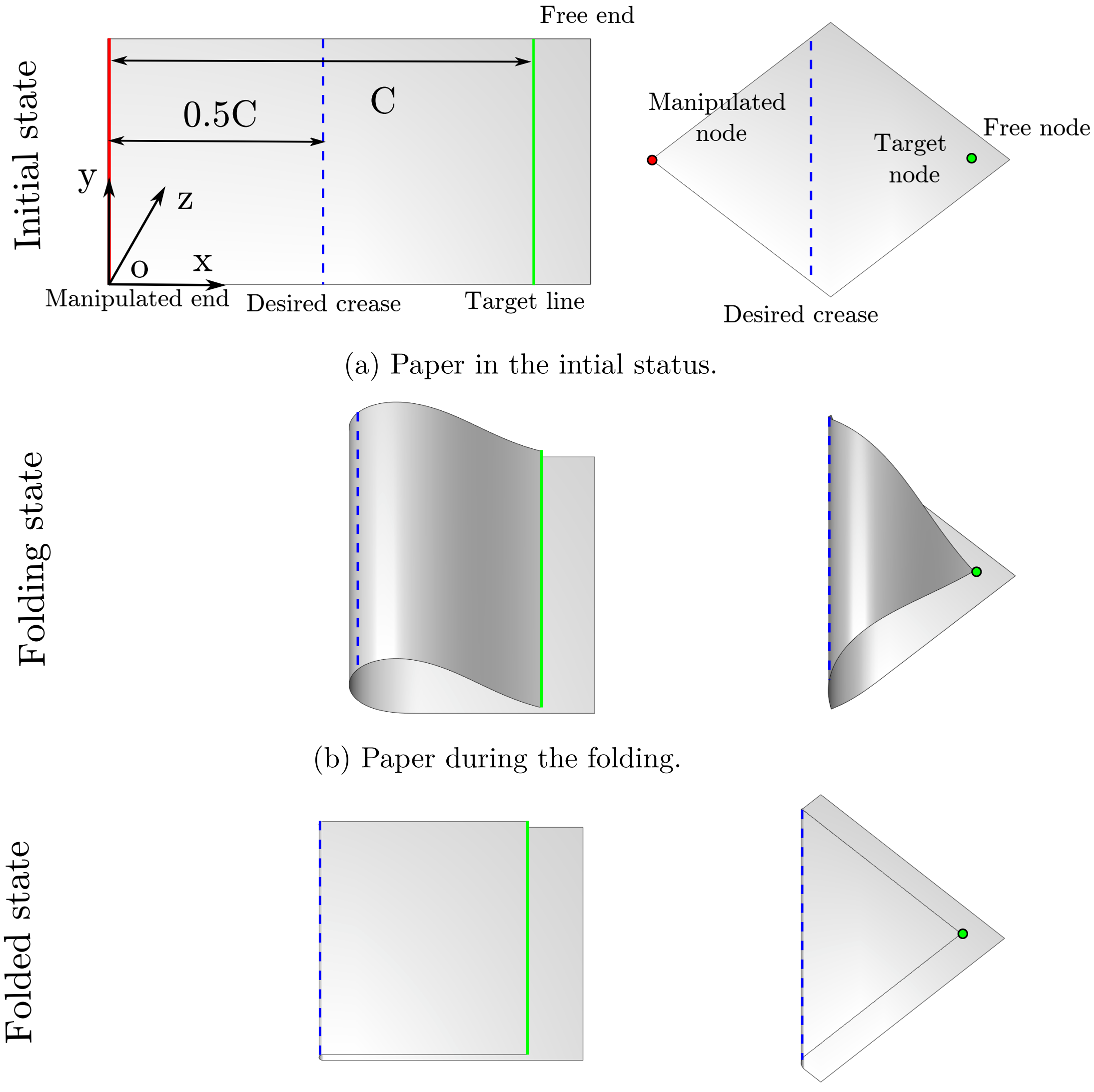
Learning-based approaches involve robots learning how to fold through training data. Learning from demonstrations (LfD), which involves deriving control policies from human demonstrations, has been particularly popular. For examples, prior research has demonstrated flattening and folding towels [93, 94]. Teleoperation demonstrations have also been used for training policies, including the manipulation of deformable linear objects (DLOs) [95] and folding fabric [96]. Researchers have additionally tackled the sim2real problem, using reinforcement learning to train robots to fold fabrics and clothes entirely through simulation [97, 98, 99]. More recently, Zheng et al. [100] employed reinforcement learning to teach a robot to flip pages in a binder using tactile feedback. While pure learning-based methods have shown promise, they tend to struggle with generalization when the material or geometric properties of the objects change significantly.

Model-based approaches, whether based on known or learned models, often employ model predictive control to manipulate deformable objects. Learned models capture the natural dynamics of deformable objects through random perturbations [101]. Although these models are generally fast, they may suffer from inaccuracies when dealing with new states. Theoretical models aim to be as physically accurate as possible, enabling the direct application of their predictive capabilities in the real world. Examples include theoretical models for strip folding [18, 102] and garment folding [103]. Physical models often rely on energy-based formulations [6, 8, 104], where elastic energies are computed based on the topological properties of simulated objects to predict their deformed shape under manipulation. These models can describe complex deformations but often come with computational costs and a trade-off between accuracy and efficiency.

Despite extensive research on 2D deformable object manipulation, the majority of efforts have been confined to soft materials such as towels and cloth. These materials are highly compliant and typically exhibit straightforward, linear deformations, allowing for solutions without the need for profound physical insight. In contrast, our work focuses on the challenge of folding paper with varying stiffness properties using a single manipulator.

While other researchers have attempted single-manipulator folding with a focus on minimizing sliding [18, 97, 102], their methods primarily targeted fabrics secured by taping their ends to a substrate. However, our experiments have shown that their generated trajectories perform inadequately when applied to paper folding. We attribute this to their local optimization strategy, which only considers the current grasp pose. In contrast, we generate folding trajectories through global optimization, underscoring the importance of considering both present and future deformation states during the paper manipulation process.

To address these challenges, we propose a framework that amalgamates physically accurate simulation, scaling analysis, and machine learning to generate folding trajectories optimized to prevent sliding. Scaling analysis renders the problem non-dimensional, reducing dimensionality and enhancing generality. This enables the training of a single nondimensionalized neural network, referred to as a neural force manifold (NFM), to continuously approximate a scaled force manifold derived purely from simulation. NFMs predict external forces on the paper based solely on the grasp position, facilitating the generation of trajectories optimized to minimize forces (and, consequently, sliding) through path planning algorithms. Furthermore, the nondimensional nature of the NFM enables us to generate trajectories for paper with diverse materials and geometric properties, even if these parameters were not part of the training dataset. Our approach demonstrates the capability to fold paper on exceptionally slick surfaces with minimal sliding (Fig. 4.1b).



(a) Paper in the intial status.

(b) Paper during the folding.

(c) Paper is folded.

Figure 4.2: Folding sheets of paper. The manipulation process involves (a) the initial state, where the paper lies flat on the substrate, followed by (b) the folding state, where the manipulated end is moved to the "crease target" line C, and finally (c) the folded state, which involves forming the desired crease on the paper.

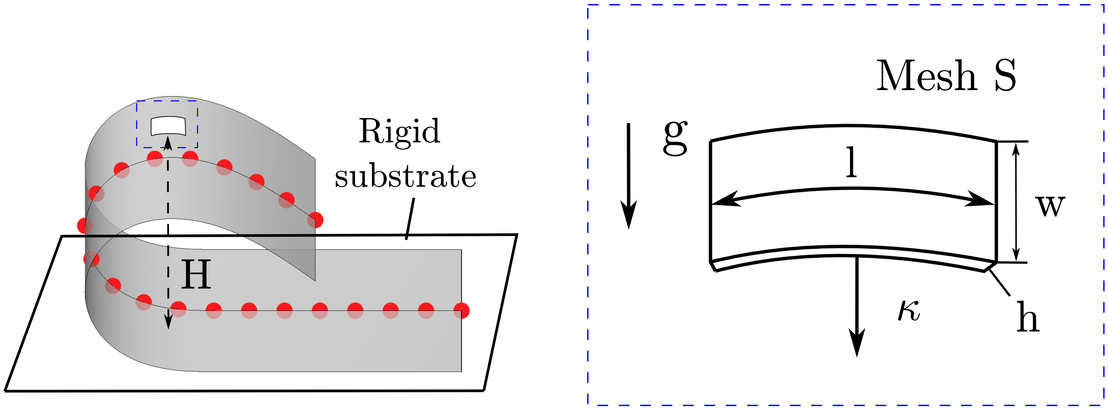
4.2 Problem Statement

This work undertakes an investigation into a seemingly straightforward yet highly challenging task within the realm of robotic folding. Specifically, the objective is to create a predefined crease on a sheet of paper possessing common symmetrical geometries, such as rectangles or diamonds, as depicted in Fig. 4.2. This task involves the manipulation of only one end of the paper, leaving the other end free from any fixtures. This characteristic simplifies the workspace by necessitating the use of a single manipulator. However, it simultaneously introduces the formidable challenge of preventing the paper from slipping against the substrate during the manipulation process.

The task at hand can be dissected into two discrete steps. The first step entails manipulating one end of the paper, transitioning it from its initial flat state, as illustrated in Fig. 4.2a, to the folding state depicted in Fig. 4.2b. The primary objective of this step is to ensure that the manipulated edge or point aligns precisely with the predetermined crease line or point C, as demonstrated in the figure. In the subsequent second step, the paper undergoes permanent deformation, creating the desired crease at the midpoint $C/2$, ultimately achieving the final folded state, as visualized in Fig. 4.2c.

It is essential to recognize that, while the act of creasing the paper may appear straightforward, the principal challenge lies in minimizing the displacement of the free end of the paper during the initial step. The paper's inherent susceptibility to nonlinear deformations, coupled with its slippery surface, accentuates the importance of accurate predictions regarding the paper's folding status to effectively minimize displacement. Given that the initial step does not involve permanent deformations, we opt for a 2D planar rod model under a linear elastic assumption to represent the paper's nonlinear deformations. Further details regarding this modeling approach are expounded upon in the subsequent section.

Subsequently, this physical model is seamlessly integrated with scaling analysis and machine learning techniques, facilitating the generation of folding trajectories that not only incorporate physical principles but are also meticulously optimized to mitigate sliding. Having successfully executed the first step, we employ straightforward motion primitives to finalize the creasing of the paper.



(a) Schematic of paper during the folding state.

(b) Bending deformations of a small piece in the paper.

Figure 4.3: Topology analysis of the energy balance between bending and gravity.

4.3 Physical-based Modelling and Machine Learning

In this section, we present the physical analysis for modelling the paper folding process. First, we analyze the main deformations of the manipulated paper and prove that a 2D model is sufficient to learn the behaviors of the manipulated paper so long as the sheet is symmetrical. The numerical model is based on DER stated in § 1.1 Second, we formulate a generalized strategy for paper folding using scaling analysis. Third, we present the machine learning scheme we used for expressing the generalized strategy in a data-driven manner.

4.3.1 Reduced-Order Model Representation

Paper represents a distinct deformable object with unique properties. Unlike cloth, its surface exhibits developability [105], meaning it can bend but not stretch. Moreover, shear deformations are of minimal significance as paper possesses a negligible thickness-to-length ratio. Therefore, when folding paper in our context, the primary source of nonlinear deformation is bending. We hypothesize that the nonlinear behaviors of paper primarily result from a balance between bending and gravitational energies, denoted as $\epsilon_b \sim \epsilon_g$.

To gain a deeper insight into the energy dynamics of manipulated paper, we scrutinize a finite element of the paper, as illustrated in Fig. 4.3b. The bending energy of this element can be expressed as:

$$\epsilon_b = \frac{1}{2} k_b \kappa^2 l, \tag{4.1}$$

where l represents the undeformed length of the element, κ signifies its curvature, and the bending stiffness is given by:

$$k_b = \frac{1}{12} E w h^3, \tag{4.2}$$

In this equation, w denotes its undeformed width, h is its thickness, and E represents its Young's modulus. The gravitational potential energy of the element can be expressed as:

$$\epsilon_g = \rho w h l g H, \tag{4.3}$$

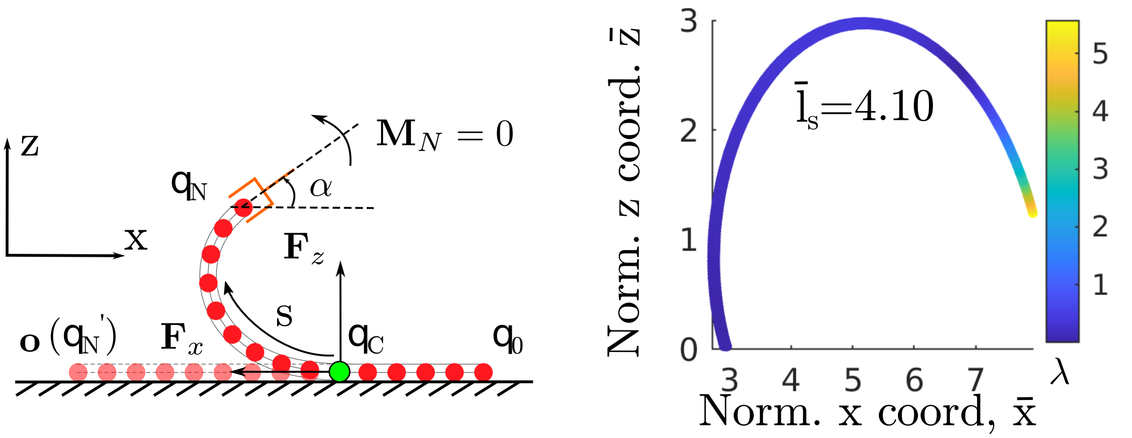
In Eq. 4.3, ρ denotes the volume density, and H signifies its vertical height above the rigid substrate.

From the equations above, we derive a characteristic length referred to as the gravito-bending length, which encapsulates the interplay between bending and gravity:

$$L_{gb} = \left(\frac{E h^2}{24 \rho g} \right)^{\frac{1}{3}} \sim \left(\frac{H}{\kappa^2} \right)^{\frac{1}{3}}. \tag{4.4}$$

This length is measured in meters and scales proportionally with the ratio of vertical height to curvature squared, representing key parameters characterizing the deformed configuration of the manipulated paper. Notably, the formulation of L_{gb} depends solely on one geometric parameter, the paper thickness h, rendering other geometric quantities such as length l and width w irrelevant to the deformed configuration.

Furthermore, due to the paper's symmetrical geometry and material homogeneity, the curvature κ is expected to be uniform for all regions at the same height H. Consequently, we can effectively employ the centerline of the paper, as depicted in Fig. 4.3a, to represent



(a) Side view of a symmetrical paper during folding with coordinate frame and relevant notations.

(b) Sampled λ forces for a particular $\bar{l}_s$ of 4.10. This showcases one of the sampled "partial" force manifolds that we use to train our neural network.

Figure 4.4: Schematic of the folding process and visualization of the sampling training data

the paper's configuration. We model this centerline as a 2D planar rod since deformations are confined to the x-z plane. To simulate the behavior of this 2D planar rod, we implement a discrete differential geometry (DDG) numerical simulation, which will be detailed in the subsequent section.

4.3.2 Generalized Solution and Scaling Analysis

As stated in § 4.2, the crux of the folding task entails maneuvering the end point, denoted as $\mathbf{q}_N$, to the target position C, starting from an initially flat configuration illustrated in Fig. 4.4a. To accomplish this, an analysis of the physical system is conducted to devise a solution that minimizes sliding during manipulation.

Several key quantities are introduced to characterize the deformed configuration of the paper. A point, denoted as $\mathbf{q}_C$, serves as the node linking the suspended region ($z > 0$) and the contact region ($z = 0$) of the paper. Given that deformations primarily occur in the suspended region, the focus is exclusively on this area. An origin, denoted as $\mathbf{o}$, is established for the 2D plane and is situated at the initial manipulated end, $\mathbf{q}_N$, as depicted in Fig. 4.4a. The robot end-effector exerts control over the pose of the manipulated end, specifying both a position, $\mathbf{q}_N = (x, z)$, and an orientation angle, α, as shown in Fig. 4.4a. A crucial constraint is imposed, stipulating that the curvature at the manipulated end remains consistently zero to avert sharp bending deformations, which are vital in preventing permanent deformations during the folding process. At the connective node, $\mathbf{q}_C$, the tangent always aligns with the x-axis.

With these defined parameters, the equations of motion (EOM) for the Discrete Elastic Rod (DER), as outlined in § 1.1, are modified to accommodate the following constraints:

$$\begin{aligned} \mathbb{M}\ddot{\mathbf{q}} + \frac{\partial E_{el}}{\partial \mathbf{q}} - \mathbf{F}^{\text{ext}} &= 0, \\ \text{such that} \quad \mathbf{q}_N &= (x, z), \\ \frac{\mathrm{d}\mathbf{q}_C}{\mathrm{d}s} &= (-1, 0), \\ M_N &= 0, \\ l_s \equiv \int_{\mathbf{q}_C}^{\mathbf{q}_N} \mathrm{d}s &= \mathbf{q}_C \cdot \hat{\mathbf{x}}, \end{aligned} \tag{4.5}$$

where M_N is the external moment applied on the manipulated end, s is the arc length of the paper's centerline, and l_s is the arc length of the suspended region (from $\mathbf{q}_C$ to $\mathbf{q}_N$).

Upon solving Eq. 4.5 through the numerical framework detailed in § 1.1, a unique degree-of-freedom vector $\mathbf{q}$ is obtained. It is noteworthy that once $\mathbf{q}$ is determined, it enables the derivation of external forces from the substrate acting upon the paper, denoted as $\mathbf{F}_{\text{substrate}} = \mathbf{F}_x + \mathbf{F}_z$. The orientation angle α of the manipulated end, as well as the suspended length l_s, can also be deduced. As a reminder, Young's modulus (E), thickness (h), and density (ρ) were determined to be the primary material and geometric properties of the paper through

Eq. 4.4. Subsequently, a physical relationship governing these quantities is outlined:

$$\begin{aligned} \lambda &= \frac{\|\mathbf{F}_x\|}{\|\mathbf{F}_z\|}, \\ (\lambda, \alpha, l_s) &= f(E, h, \rho, x, z), \end{aligned} \tag{4.6}$$

where f is an unknown relationship. It is imperative to observe that, to prevent sliding during manipulation, the condition

$$\lambda \leq \mu_s \tag{4.7}$$

must be satisfied, with μ_s denoting the static friction coefficient between the paper and the substrate. Consequently, a trajectory that minimizes sliding is one that concurrently minimizes λ throughout its path.

A significant challenge arises, however, as the relation f remains unknown, necessitating its determination for trajectory generation. Without an analytical solution, an exhaustive approach involving the numerical framework outlined in § 1.1 could be employed to systematically uncover the mappings between the input and output of f. Nevertheless, this method, reliant on generating tuples, necessitates the solution of the high-dimensional problem presented in Eq. 4.5, rendering it highly inefficient and unsuitable for real-time operation. To circumvent these challenges, we propose the utilization of a regression approximation of f via the fitting of a neural network on simulation data. Nonetheless, this approach is not without its shortcomings. Notably, directly learning f is a time-consuming endeavor due to the high-dimensional nature of Eq. 4.6, dependent on five parameters as input. Furthermore, the formulation's direct dependency on intrinsic paper properties (E, ρ, and h) necessitates a vast range of exhaustive simulations to acquire sufficient data for accurate learning.

To address these challenges and streamline the learning process, we employ scaling analysis. Leveraging the Buckingham π's theorem, we establish five dimensionless groups: $\bar{x} = x/L_{gb}$; $\bar{z} = z/L_{gb}$; $\bar{l}_s = l_s/L_{gb}$; α; and $\lambda = F_t/F_n$, where L_{gb} denotes the gravito-bending length as defined in Eq. 4.4. This formulation results in a dimensionless representation of

Eq. 4.6:

$$(\lambda, \alpha, \bar{l}_s) = \mathcal{F}\left(\bar{x}, \bar{z}\right). \tag{4.8}$$

Crucially, the mapping $\mathcal{F}$ becomes independent of units, effectively disentangling it from material and geometric paper properties. Consequently, with the dimensionality of the problem significantly reduced, training a neural network to model $\mathcal{F}$ becomes straightforward, utilizing non-dimensionalized simulation data from a single paper type. Moreover, the reduced dimensionality of $\mathcal{F}$ facilitates the visualization of the landscape of λ along a non-dimensional 2D-plane.

4.3.3 Data Generation

To learn $\mathcal{F}$, we undertake the solution of Eq. 4.5 for numerous sampled points (x, z). An illustrative representation of the partial force manifold, generated through this sampling procedure for a specific suspended length, is provided in Fig. 4.4b.

For any given (x, z) location, we incrementally introduce rotations along the y-axis and identify the optimal rotation angle, denoted as α, that ensures $M_N = 0$ at the manipulated end. Subsequently, for a specific configuration (x, z, α), we record pertinent data, including the suspended length, l_s, along with the tangential and normal forces acting on the clamped end. This process culminates in the formation of a training dataset, denoted as $\mathcal{D}$, comprising tuples consisting of six elements: $(F_t, F_n, \alpha, l_s, x, z)$.

Following this data collection, we perform a non-dimensionalization of the dataset, transforming it into the following normalized form: $(\lambda, \alpha, \bar{l}_s, \bar{x}, \bar{z})$.

Our simulation framework facilitates the generation of an extensive dataset, denoted as $\mathcal{D}$, encompassing a total of 95,796 training samples. This dataset covers a normalized suspended length range of $\bar{l}_s \leq 6.84$, effectively spanning the workspace of most papers. The generation of this dataset consumed approximately 3.54 hours of computational time, executed on an AMD Ryzen 7 3700X 8-core processor.

4.3.4 Learning Force and Optimal Grasp Orientation

For the training of a neural network model $\mathcal{F}$

$$(\lambda, \alpha, \bar{l}_s) = \mathcal{F}_{\mathrm{NN}}(\bar{x}, \bar{z}), \tag{4.9}$$

we employed a relatively simple fully-connected feed-forward nonlinear regression network architecture. This neural network consisted of four hidden layers, each comprising 392 units. Notably, each hidden layer, with the exception of the final output layer, incorporated rectified linear unit (ReLU) activation functions.

Furthermore, prior to inputting the data into the neural network, we applied a standardization process, as defined in

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}_{\mathcal{D}}}{\boldsymbol{\sigma}_{\mathcal{D}}}. \tag{4.10}$$

This standardization involved adjusting the input values $\mathbf{x}$ by subtracting the mean of the dataset $\mathcal{D}$, denoted as $\bar{\mathbf{x}}_{\mathcal{D}}$, and subsequently dividing them by the standard deviation of $\mathcal{D}$, represented as $\boldsymbol{\sigma}_{\mathcal{D}}$.

Our training procedure commenced with an initial 80-20 split of the dataset $\mathcal{D}$ into training and validation sets, utilizing a batch size of 128. The mean absolute error (MAE) was employed as the training error metric. In the event that training reached a plateau, we alternated between stochastic gradient descent (SGD) and the Adam optimizer. Additionally, we progressively increased the batch size up to 4,096 and conducted training on the complete dataset once the MAE dropped below the threshold of 0.001. Through this training scheme, we successfully achieved an MAE of less than 0.0005.

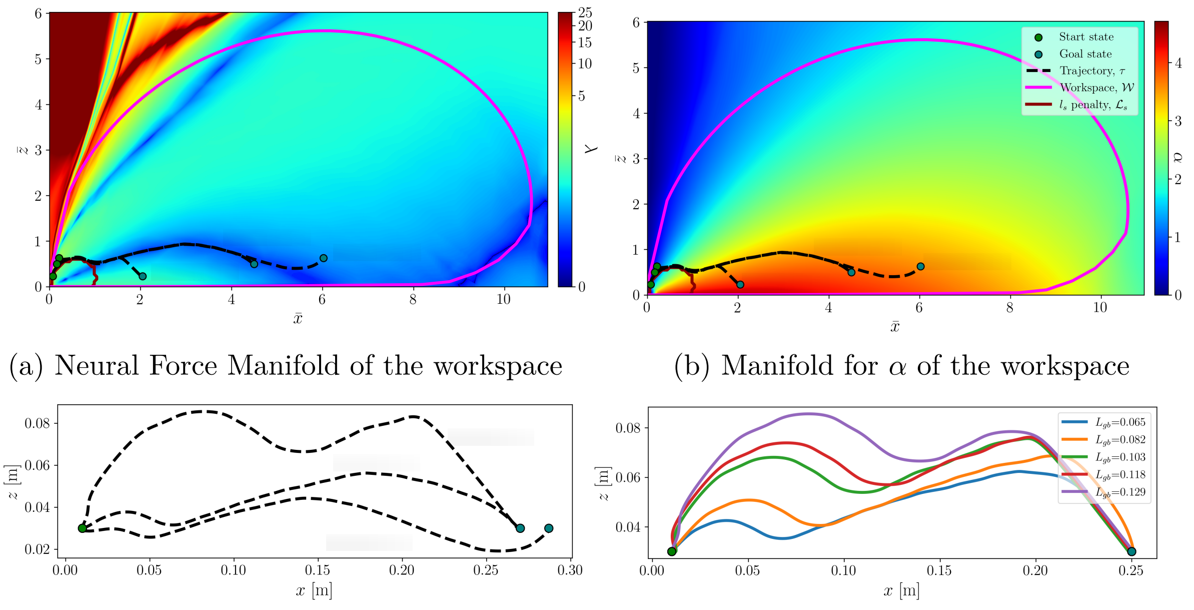
4.4 Optimal Motion Planning

In this section, we begin by constructing a Neural Force Manifold (NFM) using the learned model. The NFM encapsulates essential information regarding the force ratios exerted by the substrates within the manipulation workspace. This information is crucial for devising an optimal motion plan that minimizes the impact of substrate forces on the manipulation process.

4.4.1 Constructing the Neural Force Manifold

The Neural Force Manifold (NFM), which represents the λ – output of $\mathcal{F}_{\text{NN}}$ for the given workspace set, is discretized into a rectangular grid. This grid is composed of $\bar{\delta} \times \bar{\delta}$ blocks, where $\bar{\delta} = \delta / L_{gb}$. Within each block, we calculate and store a single λ value based on the midpoint of that block. This discretization process results in a structured neural force manifold denoted as $\mathcal{M}$, which can be conceptualized as an $m \times n$ matrix.

For path planning, we introduce two additional components to our manifold. Firstly, we restrict exploration to regions covered by our training dataset ($\bar{l}_s \leq 6.84$) and define a workspace, denoted as $\mathcal{W}$. This workspace encompasses all $(\bar{x}, \bar{z})$ pairs within the convex hull of the input portion of the dataset $\mathcal{D}$. Secondly, we establish an exclusion zone based on a certain threshold for $\bar{l}_s$. This is essential to prevent positions characterized by small suspended lengths and large α angles, which could result in high curvatures, potential collisions with our gripper, or undesirable plastic deformation. This exclusion zone is referred to as the penalty region, denoted as $\mathcal{L}_s$. Fig. 4.5a provides a visualization of $\mathcal{M}$ alongside the workspace $\mathcal{W}$ and the penalty boundary $\mathcal{L}_s$ regions. Additionally, Fig. 4.5b displays the corresponding α values associated with the manifold.



(a) Neural Force Manifold of the workspace

(b) Manifold for α of the workspace

(c) Trajectories in (a) and (b)

(d) Arbitrary trajectories for various L_{gb}

Figure 4.5: Visualization of the trained neural network's non-dimensionalized λ force manifold $\mathcal{M}$ (a) and α manifold (b). An extremely low $\bar{\delta}$ discretization is used to showcase smoothness. For the force manifold, we observe two distinctive local minima canyons. Note that regions outside the workspace $\mathcal{W}$ are physically inaccurate but are of no consequence as they are ignored. For the α manifold, we observe continuous smooth interpolation throughout, which is crucial for producing feasible trajectories. Both manifolds showcase the trajectories used in the experiments for folding paper in half for $L_{gb} \in [0.048, 0.060, 0.132]$. (c) The three trajectories in (a) and (b) scaled back to real space. These are the actual trajectories used by the robot. (d) Arbitrary trajectories for various L_{gb} with identical start and goal states, highlighting the effect of the material property on our control policy.

4.4.2 Path Planning over the Neural Force Manifold

In light of the discretized manifold denoted as $\mathcal{M}$, we are now equipped to employ conventional path planning algorithms for the generation of optimal trajectories. Evidently, within the neural force manifold $\mathcal{M}$, we discern the presence of two distinct local minima regions, as visually indicated by the dark blue shading in Fig. 4.5a. However, it is imperative to note that these two minimum regions remain unconnected. This circumstance underscores the potential consequence of suboptimal local optimization, which may inadvertently lead to traversals through regions characterized by high force magnitudes later in the process.

Algorithm 4: Uniform Cost Search

```
Input: $\bar{x}_s, \bar{z}_s, \bar{x}_g, \bar{z}_g, \mathcal{M}$
Output: $\tau^*$
Func UCS($\bar{x}_s, \bar{z}_s, \bar{x}_g, \bar{z}_g, \mathcal{M}$):
    $\mathcal{W} \leftarrow$ valid workspace of $\mathcal{M}$
    $\mathcal{L}_s \leftarrow l_s$ penalty region
    $\mathbf{h} \leftarrow$ initialize min heap priority queue
    $\mathbf{c} \leftarrow$ initialize empty list
    $n_s \leftarrow$ node with location $(\bar{x}_s, \bar{z}_s)$ and cost 0
    $n_g \leftarrow$ node with location $(\bar{x}_g, \bar{z}_g)$ and cost 0
    $\mathbf{h}$.push($n_s$)
    while len($\mathbf{h}$) $> 0$ do
        $n_i \leftarrow \mathbf{h}$.pop()
        if $n_i == n_g$ then
            $\tau^* \leftarrow$ path from start to goal
            break
        $\mathbf{c}$.append($n_i$)
        for $(\bar{x}_j, \bar{z}_j) \in$ neighbors of $n_i$ do
            if $(\bar{x}_j, \bar{z}_j) \notin \mathcal{W} \setminus \mathcal{L}_s$ then
                continue
            $n_j \leftarrow$ node with location $(\bar{x}_j, \bar{z}_j)$ and cost $\lambda_j$ from $\mathcal{M}$
            if $n_j \in \mathbf{c}$ then
                continue
            if $n_j \in \mathbf{h}$ and cost of $n_j$ is higher then
                continue
            $\mathbf{h}$.push($n_j$)
    $\tau^* \leftarrow$ perform trajectory smoothing on $\tau^*$
    return $\tau^*$
```

4.4.3 Dual Manipulator and Perception System

As previously alluded to, prior investigations in the realm of mechanics-based strategies for folding shell-like structures, such as cloth, have relied upon either physical simulations or energy-based optimization techniques to compute the optimal subsequent grasping actions. These methodologies, however, operate solely based on the present configuration of the

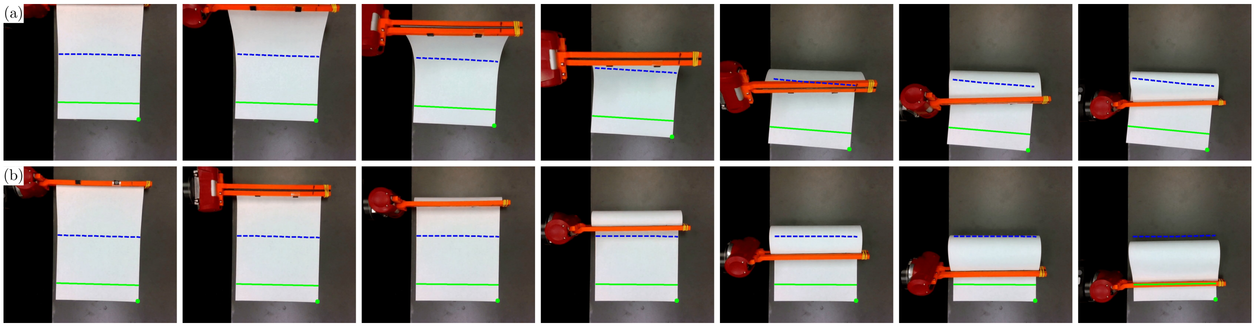


Figure 4.6: Example of our perception system with a top-down view of the folding procedure. (a) Shows the intuitive baseline results while (b) shows our open-loop algorithm for $L_{gb} = 0.048$ and $C = 0.25\,\mathrm{m}$. As in Fig. 4.2, the solid green line indicates the desired end effector position while the dashed blue line indicates the crease location. For this case, we observe that the intuitive baseline suffers from considerable sliding while our open-loop algorithm has near-perfect performance.

manipulated object [18, 102]. Our empirical findings in § 4.6 unequivocally demonstrate the inadequacy of such localized optimization approaches when applied to the context of paper folding. In contrast to the aforementioned local optimization paradigms, our approach centers on the generation of globally optimized trajectories, which account for both the current and future states of the paper sheet. To this end, we formulate the concept of an optimal trajectory denoted as τ^{+}, characterized by its ability to reach the desired goal state while simultaneously minimizing the cumulative sum of a set of cost factors λ_i, defined as follows:

$$\tau^* = \arg\min_{\tau \in \mathcal{T}} \sum_{i=0}^{i=L-1} \lambda_i, \tag{4.11}$$

where L is the length of the trajectory and $\mathcal{T}$ is the set of all valid trajectories from the desired start to goal state. We define a valid trajectory as one that is contained within the acceptable region

$$(x_i, z_i) \in \mathcal{W} \setminus \mathcal{L}_s \;\forall\; (x_i, z_i) \in \tau, \tag{4.12}$$

and whose consecutive states are adjacent grid locations. Given the discretization of the NFM, we can treat $\mathcal{M}$ as a graph whose edge weights consist of λ. Therefore, we use uniform cost search to obtain τ^*. Algo.4 provides the pseudocode of the path planning

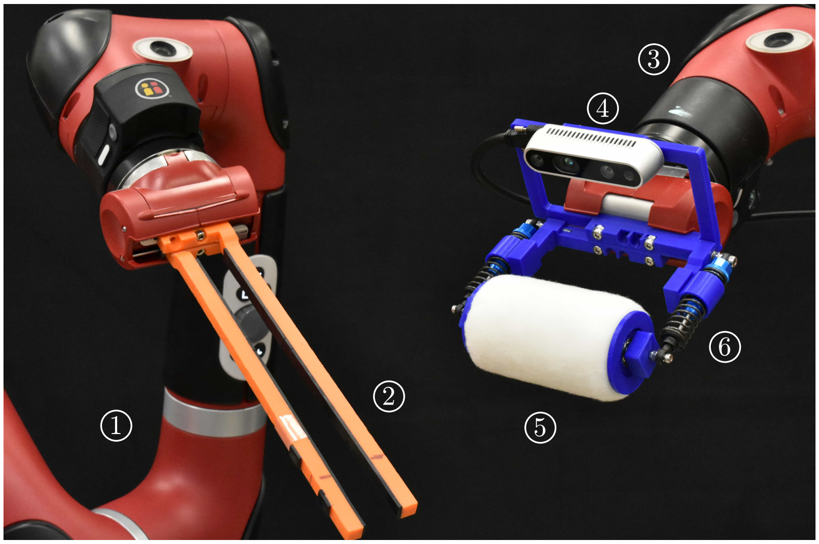


Figure 4.7: Experimental apparatus: Two robot manipulators, one for folding (1) and the other for creasing (3). An elongated gripper (2) is used to grab the manipulated end of the paper. A roller (5) with compliant springs (6) is used to form the crease. An Intel RealSense D435 camera (4) attached to the creasing arm offers visual feedback during the folding procedure. All gripper attachments were 3D printed.

algorithm.

4.5 Robotic Systems

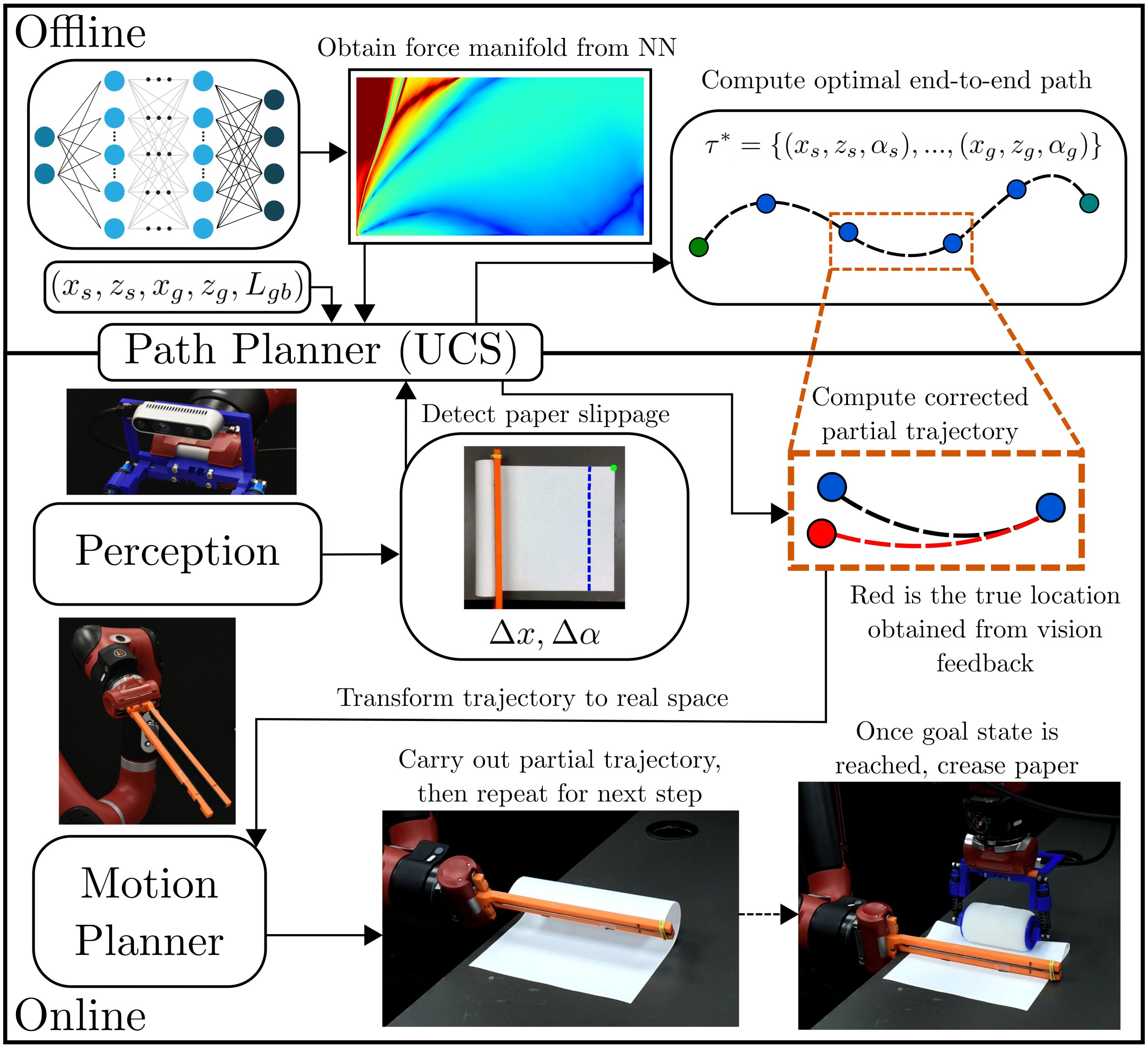
We employ two Rethink Robotics' Sawyer manipulators in our experimental setup, as illustrated in Fig. 4.7. One of these robotic arms is equipped with an elongated gripper designed specifically for folding tasks, while the other arm features a spring-compliant roller for creasing operations. Additionally, the roller arm is fitted with an Intel Realsense D435 camera, providing visual feedback. Notably, the elongated gripper is augmented with rubber lining on the inner surfaces of its fingers to facilitate secure and precise grasping of objects.

For the purpose of perception tasks, we adopt an eye-in-hand approach, involving the attachment of an Intel Realsense D435 camera to the roller arm. It is worth noting that we do not utilize the range output of the camera, as it is oriented downward along the world's z-axis, and the distance from the camera to the table surface is a known parameter in our setup. To ascertain the pose of the paper sheet, we employ a straightforward color detection technique to isolate the paper from its surroundings. Subsequently, we employ Shi-Tomasi corner detection [106] to determine the position of the paper's bottom edge. Fig. 4.6 offers a visual representation of the top-down perspective, along with the detected poses as acquired through our vision system.

4.5.1 Visual-Feedback Control

Despite our efforts to minimize the cost factors represented by λ within our proposed framework, it is crucial to acknowledge the potential occurrence of sliding, which may arise due to a substrate's low-friction surface characteristics and/or slight jittering of the robot's end-effector. It is worth noting that the optimal trajectory τ^*, generated as elaborated in § 4.4.2, is predicated on the assumption of a fixed origin $\mathbf{o}$ within our coordinate system, depicted in Fig. 4.4a. This origin $\mathbf{o}$ can be explicitly defined as $\mathbf{o} = \mathbf{q}_0 - l\hat{\mathbf{x}}$, where l signifies the total length of the paper. The occurrence of any degree of sliding implies that the position of $\mathbf{q}_0$ along the x-axis is changing, thereby necessitating a corresponding adjustment in the position of the origin $\mathbf{o}$. Consequently, deviations from the optimal trajectory ensue during traversal. Moreover, in the absence of adaptive replanning, the amount of sliding denoted as Δx will directly translate into an error of Δx in the creasing operation.

To mitigate the effects of sliding, we introduce a visual-feedback approach into our methodology. We collect visual feedback at N evenly spaced intervals along the trajectory τ^*, as depicted in Fig. 4.8. To achieve this, we divide τ^* into N partial trajectories. With the exception of the initial partial trajectory τ_0^*, we extract the start and goal states of the remaining $1 \leq i \leq N$ partial trajectories, resulting in a sequence of N evenly spaced states $\mathcal{S} = \{(x_1, z_1, \alpha_1), \ldots, (x_N, z_N, \alpha_N)\}$, accounting for potential overlaps. After executing τ_0^*, we assess the extent of sliding, denoted as Δx, and incorporate this error by updating



Offline
Obtain force manifold from NN
Compute optimal end-to-end path
$\tau^* = \{(x_s, z_s, \alpha_s), ..., (x_g, z_g, \alpha_g)\}$
$(x_s, z_s, x_g, z_g, L_{gb})$
Path Planner (UCS)
Detect paper slippage
Compute corrected partial trajectory
Perception
$\Delta x, \Delta \alpha$
Red is the true location obtained from vision feedback
Transform trajectory to real space
Once goal state is reached, crease paper
Carry out partial trajectory, then repeat for next step
Motion Planner
Online

Figure 4.8: Overview of our robotic paper-folding pipeline. The top row shows offline components while the bottom row shows online ones. On the offline side, we use our trained neural network to generate the necessary force manifold for planning. Then, given an input tuple $(x_s, z_s, x_g, z_g, L_{gb})$, we generate an end-to-end trajectory using uniform cost search. This end-to-end trajectory is then split up into partial trajectories that are carried out by the robot. At the conclusion of each partial trajectory, we measure paper sliding and replan the next partial trajectory to rectify the error.

Algorithm 5: Closed-loop Control Pseudocode

```
Input: $(x_s, z_s), (x_g, z_g), L_{gb}, \delta, N, \mathcal{F}_{\text{NN}}$
$\mathcal{M} \leftarrow$ DiscretizeManifold $(\mathcal{F}_{\text{NN}}, \delta)$
$\bar{x}_s, \bar{z}_s, \bar{x}_g, \bar{z}_g \leftarrow$ non-dimensionalize with $L_{gb}$
$\bar{\tau}^* \leftarrow$ UCS $(\bar{x}_s, \bar{z}_s, \bar{x}_g, \bar{z}_g, \mathcal{M})$
update $\bar{\tau}^*$ with $\alpha_s$ using $\mathcal{F}_{\text{NN}}$
$\tau^* \leftarrow$ convert $\bar{\tau}^*$ to real space with $L_{gb}$
$\tau_0^*, ..., \tau_{N-1}^* \leftarrow$ SplitTrajectory $(\tau^*, N)$
$\mathcal{S} \leftarrow$ extract start and goal states
carry out $\tau_0^*$ on robot
for $(x_i, z_i, \alpha_i)$ and $(x_{i+1}, z_{i+1}, \alpha_{i+1}) \in \mathcal{S}$ do
    $\Delta x \leftarrow$ detect sliding of paper
    $x_i^c \leftarrow x_i - \Delta x$
    $\bar{x}_i^c, \bar{z}_i, \bar{x}_{i+1}, \bar{z}_{i+1} \leftarrow$ non-dimensionalize with $L_{gb}$
    $\alpha_i^c \leftarrow \mathcal{F}_{\text{NN}}(\bar{x}_i^c, \bar{z}_i)$
    $\Delta\alpha \leftarrow \alpha_i - \alpha_i^c$
    $\bar{\tau}_i^* \leftarrow$ UCS $(\bar{x}_i^c, \bar{z}_i, \bar{x}_{i+1}, \bar{z}_{i+1}, \mathcal{M})$
    $L \leftarrow \text{len}(\bar{\tau}_i^*)$
    $\boldsymbol{\alpha}_i \leftarrow$ obtain $\alpha$s of $\bar{\tau}_i^*$ using $\mathcal{F}_{\text{NN}}$
    $\boldsymbol{\alpha}_i^c \leftarrow \boldsymbol{\alpha}_i + \Delta\alpha[1, (L-1)/L, ..., 1/L, 0]^T$
    append $\bar{\tau}_i^*$ with $\boldsymbol{\alpha}_i^c$
    $\tau_i^* \leftarrow$ convert $\bar{\tau}^*$ to real space with $L_{gb}$
    carry out $\tau_i^*$ on robot
crease paper with roller
```

the start state and normalizing it as follows:

$$\bar{x}_i^c = \frac{x_i - \Delta x}{L_{gb}}. \tag{4.13}$$

Subsequently, we replan a partial trajectory τ_i^* from the updated start state (x_i^c, z_i) to the next state $(xi + 1, z_{i+1})$ in the sequence and execute this updated trajectory. This process is reiterated until the goal state is reached. This is repeated until reaching the goal state. By properly accounting for sliding, we ensure that the traversal through the NFM is as accurate as possible. Note that this scheme allows us to obtain corrected partial trajectories in near real time once N becomes sufficiently large, as each partial trajectory's goal state approaches its start state, allowing for uniform cost search to conclude rapidly. We direct the reader to our supplementary videos (Footnote 1), which showcase the speed of the feedback loop.

It should be noted that sliding Δx is not the sole error we must rectify. We must also consider that we assume an optimal grasp orientation α for each position within the neural force manifold. When the origin of our neural force manifold shifts, the actual position no longer aligns with the intended position, leading to an angular error $\Delta\alpha$:

$$\begin{aligned} \alpha_i^c &= \mathcal{F}_{\text{NN}}(\bar{x}_i^c, \bar{z}_i), \\ \Delta\alpha &= \alpha_i - \alpha_i^c. \end{aligned} \tag{4.14}$$

Simply applying a correction of $-\Delta\alpha$ to the first point in a partial trajectory would result in a significant abrupt rotation, exacerbating the sliding issue. Furthermore, as long as the sliding remains within reasonable bounds, the incorrect α at the current position within the manifold remains reasonably optimal. Therefore, we opt for a gradual incorporation of the

$\Delta\alpha$ error into the trajectory:

$$\begin{aligned} \tau_i^* &= \mathrm{UCS}(\bar{x}_i^c, \bar{z}_i, \bar{x}_{i+1}, \bar{z}_{i+1}, \mathcal{M}), \\ \boldsymbol{\alpha}_i &= \mathcal{F}_{\mathrm{NN}}(\tau_i^*), \\ \boldsymbol{\alpha}_i^c &= \boldsymbol{\alpha}_i + \Delta\alpha[1, (L-1)/L, ..., 1/L, 0]^T, \end{aligned} \tag{4.15}$$

where UCS denotes uniform cost search and L represents the length of trajectory τ_i^*. This gradual correction approach ensures that we minimize sliding while maintaining the smoothness of the trajectory. Algo.5 provides the pseudocode outlining the entire closed-loop method we have described.

4.6 Experiments and Results

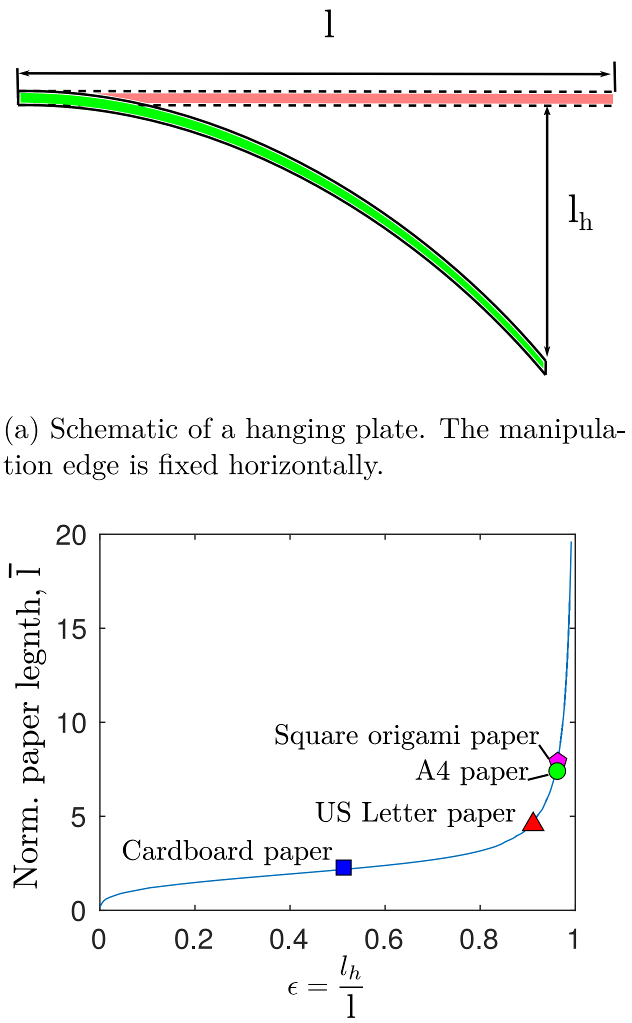
4.6.1 Measuring the Material Property of Paper

To utilize our framework effectively, it's essential to accurately determine the parameter L_{gb}, which comprises both the bending stiffness $k_b = Eh^3/12$ and the density ρ of the paper material. Therefore, by measuring this single quantity, we can sufficiently characterize the material properties of the paper. We next present a straightforward method for measuring this parameter.

As shown in Fig. 4.9a, when one end of the paper is fixed, it undergoes deformation due to the combined effects of bending and gravitational energy. Since L_{gb} represents the combined influence of both bending and gravity on the paper, we can establish the following relationship:

$$\mathcal{L}(\epsilon) = \bar{l} = \frac{l}{L_{gb}}, \quad \epsilon = \frac{l_h}{l}, \tag{4.16}$$

where l_h is the vertical distance from the free end to the fixed end and l is the paper's total length. The mapping $\mathcal{L}(\epsilon)$ can be determined through numerical simulations, as depicted in Fig. 4.9b. Having established this mapping, we can employ straightforward algebra to



(a) Schematic of a hanging plate. The manipulation edge is fixed horizontally.

(b) Relationship between the ratio $\epsilon = l_h/l$ and normalized total length of the paper $\bar{l} = l/L_{gb}$.

Figure 4.9: The measurement scheme for gravity-bending length L_{gb}.

deduce the L_{gb} value. First, we measure the ratio $\epsilon = l_h/l$ for a particular paper to obtain its corresponding normalized total length $\bar{l}$. Then, the value of L_{gb} can be calculated simply by $L_{gb} = l/\bar{l}$. Once we obtain L_{gb}, we can now use the non-dimensionalized mapping Eq. 4.8 to find the optimal path for manipulating the paper.

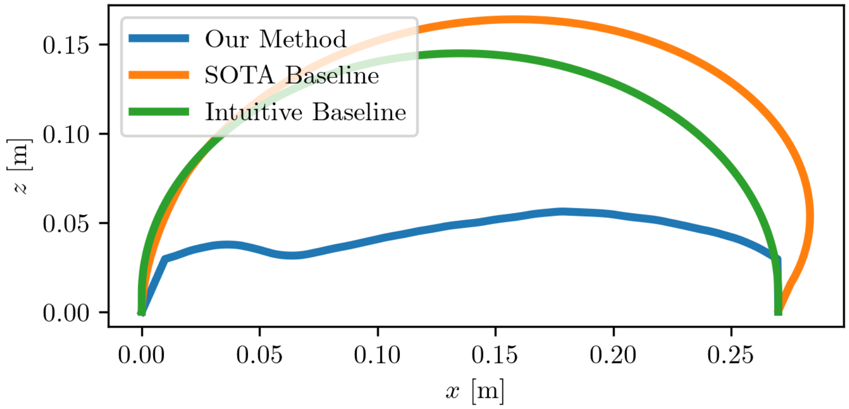


Figure 4.10: Comparison of trajectories computed by the folding algorithms for US letter paper with $\mathcal{C} = 0.27\,\text{m}$.

4.6.2 Baseline Algorithms

To demonstrate the benefits of our folding algorithm, we conducted comparisons with both an intuitive and a state-of-the-art baseline. The intuitive baseline algorithm can be conceptualized as one that would succeed if the opposite end of the paper were fixed to the substrate. In such a scenario, the trajectory would involve gripping the paper's edge and tracing half of a circle's perimeter with a radius of $R = C/2$:

$$\begin{aligned} \mathrm{d}\theta &= \pi/M, \\ \tau_B &= \{(R\cos(i\mathrm{d}\theta), R\sin(i\mathrm{d}\theta), i\mathrm{d}\theta)\ \forall\ i \in [0, M]\}, \end{aligned} \tag{4.17}$$

where M is an arbitrary number of points used to sample the trajectory. We chose $M = 250$ for all our experiments.

Additionally, we conducted comparisons against the state-of-the-art mechanics-based folding algorithm presented by Petrik et al. [18, 102], which we refer to as the "SOTA baseline", that uses a beam model to compute folding trajectories for fabric minimizing sliding. However, this baseline considers only the current status of the deformed material when computing subsequent optimal grasp and, consequently, cannot handle the challenging task of paper folding. Examples of the computed trajectories are shown in Fig. 4.10.

4.6.3 Experimental Setup

We tested folding on four different types of paper:

1. A4 paper, $L_{gb} = 0.048\,\mathrm{m}$,
2. US Letter paper, $L_{gb} = 0.060\,\mathrm{m}$,
3. cardboard paper (US Letter dimensions), $L_{gb} = 0.132\,\mathrm{m}$,
4. square origami paper, $L_{gb} = 0.043\,\mathrm{m}$.

We performed two sets of experiments for the rectangular papers (1–3). The first involved folding the papers to an arbitrary crease location ($C = 0.25\,\mathrm{m}$ for A4 and $C = 0.20\,\mathrm{m}$ for US Letter and cardboard), while the second involves folding the papers in half. We chose an arbitrary crease location of $C = 0.30\,\mathrm{m}$ for the square origami paper. This resulted in a total of 7 folding scenarios. For each scenario, we conducted experiments using four different algorithms—the intuitive baseline, the SOTA baseline, our open-loop approach, and our closed-loop approach. We completed 10 trials for each of these algorithms, resulting in a total of 280 experiments.

4.6.4 Metrics

In the experiments, we employed two critical metrics for evaluation: the average fold length and the spin error. The average fold length was computed by taking the average of the left and right side lengths up to the crease. The spin error, on the other hand, was determined by calculating the angle θ_{err}, representing the difference between the left and right side lengths. In the case of square papers, the fold length was defined as the perpendicular length from the tip to the crease, while the spin error denoted the angular deviation from this line to the true diagonal.

Table 4.1: Offline trajectory computation times for papers and crease types [s]

Algorithm	A4 $\mathcal{C} = 0.25\,\mathrm{m}$	A4 $\mathcal{C} =$ Half	US $\mathcal{C} = 0.20\,\mathrm{m}$	US $\mathcal{C} =$ Half	CA $\mathcal{C} = 0.20\,\mathrm{m}$	CA $\mathcal{C} =$ Half	OR $\mathcal{C} = 0.30\,\mathrm{m}$
Petrik et al.[18, 102]	59.46	51.15	68.14	47.30	80.07	77.28	43.20
Our Method	3.28	4.13	1.80	2.28	1.27	4.19	11.73

4.6.5 Parameters

In our experiments, we discretized the neural force manifold $\mathcal{M}$ using a step size of $\bar{\delta} = 0.0548$. This discretization value was chosen as it offered a good balance between accuracy and computational efficiency. For rectangular papers, we defined a penalty region $\mathcal{L}_s$ with the constraint $\bar{l}_s < 0.958$, whereas square papers used a penalty region defined by $\bar{l}_s < 1.137$. This difference arises from the varying yield strengths of the papers; the square paper has a smaller yield strength compared to the rectangular paper, necessitating a larger suspended length $\bar{l}_s$ range to avoid highly high curvatures.

In our closed-loop control approach, we divided all trajectories into $N = 5$ intervals, regardless of the trajectory length. Additionally, we conducted experiments on a low-friction table to demonstrate the robustness of our method. Notably, lower friction coefficients make the manipulation task more challenging due to the increased likelihood of sliding. We empirically measured the static coefficient of friction between the papers and the substrate, yielding an approximate value of $\mu_s = 0.12$. For comparison, the static coefficient of friction for lubricated steel on steel is typically around $\mu_s = 0.15$.

4.6.6 Results and Analysis

Tab. 4.1 presents the offline trajectory computation times for all experiments conducted using a single Intel i9-9900KF CPU. These results demonstrate an average speed improvement of 15× over the state-of-the-art (SOTA) baseline. In Fig. 4.11, we display the experimental outcomes as box plots, showcasing both achieved fold lengths and spin errors. The fold lengths reveal significant improvements across all folding scenarios over the two baseline

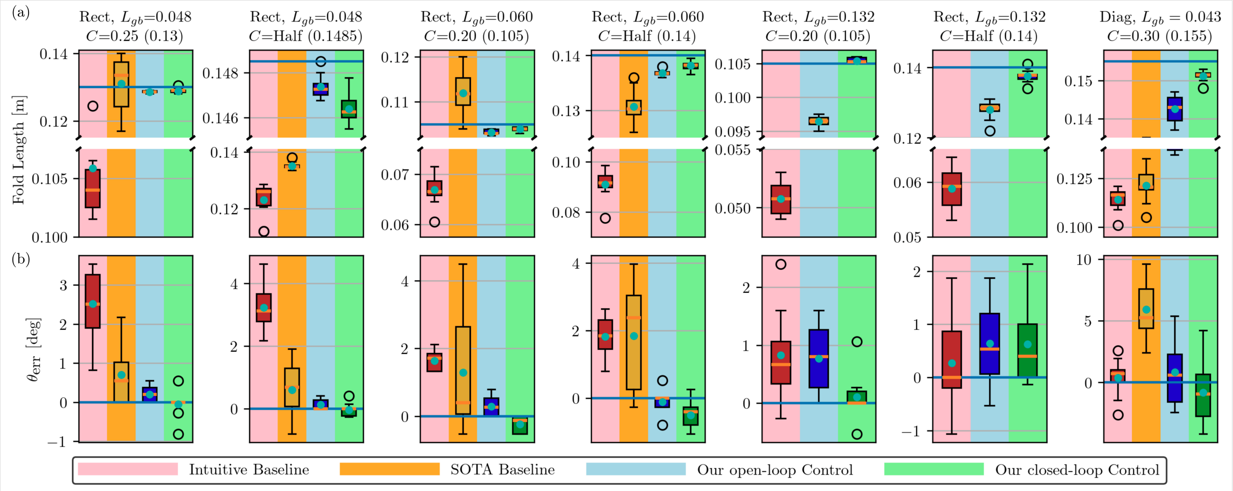


Figure 4.11: Experimental results for all folding scenarios. Each column indicates a folding scenario while the top row (a) shows the fold length and the bottom row (b) shows the spin error. Boxplot results are shown color coded for the intuitive baseline, the SOTA baseline [18], open-loop control, and closed-loop control algorithms. Medians are shown as orange lines, means as turquoise circles, and the desired target value as a light blue horizontal line. Both our open-loop and closed-loop algorithms yield significant improvements over the intuitive baseline and the SOTA baseline, as shown by the broken axis in (a). Our algorithms also exhibit significantly less variance.

methods.

As expected, the SOTA baseline generally outperforms the intuitive method, except for cardboard paper, where the SOTA approach fails to fold it all together. To appropriately represent the data variance, we use broken axes. It's worth noting that our algorithms achieve significantly better average performance and exhibit lower variance, as evidenced by the reduced y-axis resolution after the axis break. The high variances observed in the baseline algorithms can be attributed to the increased influence of friction, which often leads to chaotic and unpredictable outcomes. In essence, deterministic folding can only be attained without sliding.

For the vast majority of cases, incorporating visual feedback yields a clear improvement over the open-loop algorithm. Intuitively, we observe a trend where the performance gap between our open-loop and closed-loop algorithms grows as the material stiffness increases

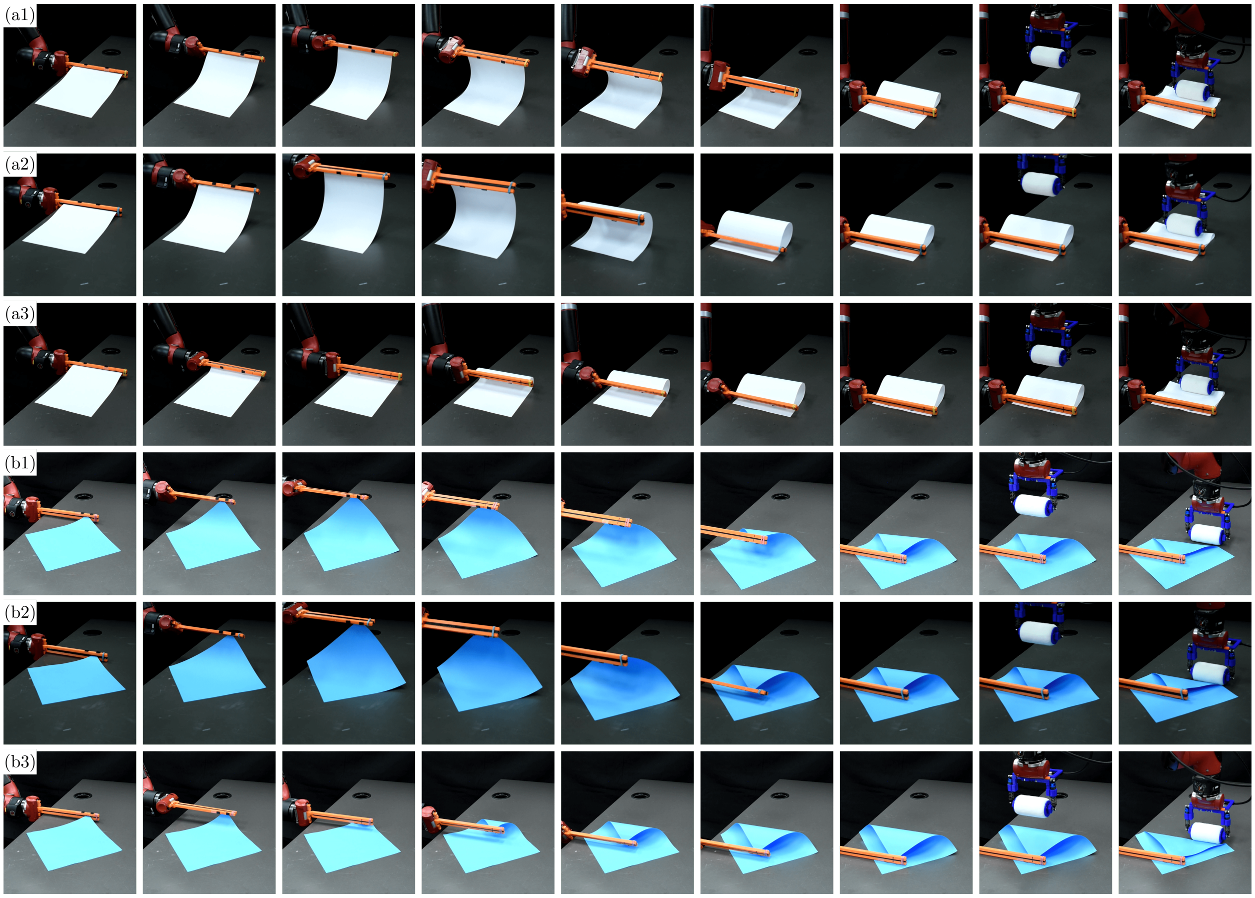


Figure 4.12: Isometric views of different folding scenarios. (a) C = Half folding for $L_{gb} = 0.048$ paper with the intuitive baseline (a1), the SOTA baseline (a2), and our open-loop algorithm (a3). (b) $C = 0.30\,\mathrm{m}$ diagonal folding for $L_{gb} = 0.043$ with the intuitive baseline (b1), the SOTA baseline (b2), and our closed-loop algorithm (b3).

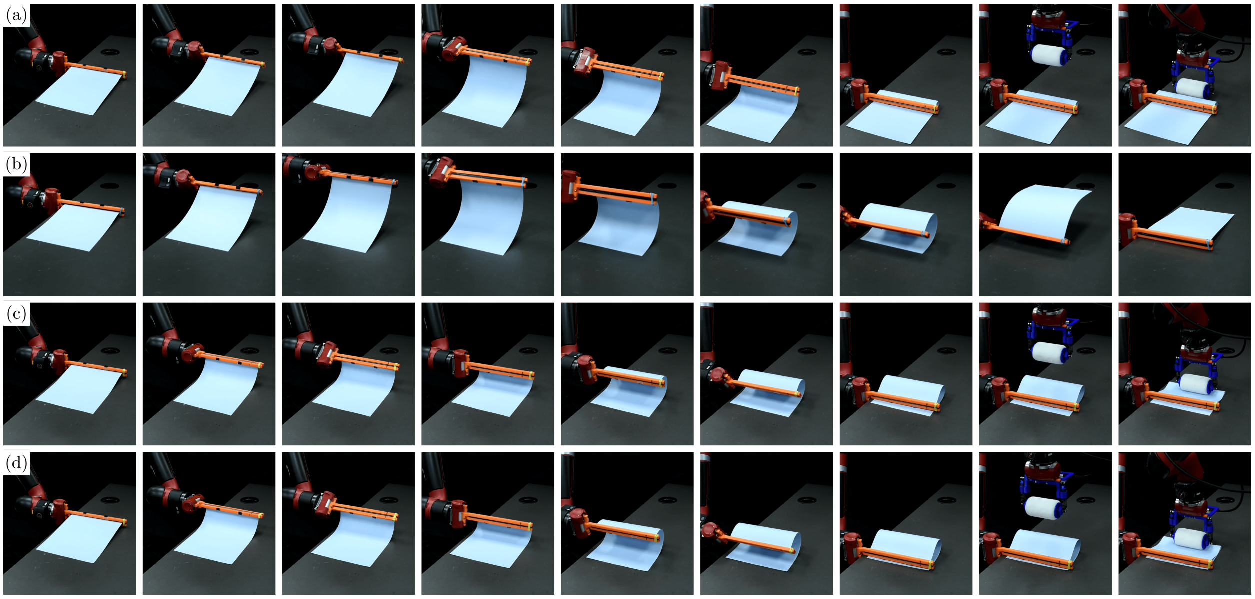


Figure 4.13: Isometric views for folding C = Half with the stiffest paper ($L_{gb} = 0.132$): (a) shows the intuitive baseline, which fails drastically as the stiffness of the paper causes excessive sliding during the folding process, (b) shows the SOTA baseline, which is unable to fold cardboard at all and experiences a high energy snap caused by the large induced deformations, (c) shows our open-loop algorithm, demonstrating significant improvements over both baselines with minimal sliding, and (d) shows our closed-loop algorithm, which improves upon our open-loop results and achieves near perfect folding.

for rectangular folding. For softer materials ($L_{gb} = 0.048$), the open-loop algorithm has near-perfect performance as shown when folding a paper in half in Fig. 4.12(a3). By comparison, Fig. 4.12(a1)–(a2) shows the intuitive and SOTA baselines failing with significant sliding.

The sliding problem is only exacerbated by increasing the material's stiffness ($L_{gb} = 0.132$). Fig. 4.13(a) shows the intuitive baseline algorithm failing to fold the cardboard paper in half by a margin almost as long as the paper itself, while Fig. 4.13(b) shows how the SOTA baseline method experiences complete failure due to a high energy snapping caused by excessive deformation. By comparison, our open-loop algorithm can fold the cardboard with significantly better results albeit with some sliding, as shown in Fig. 4.13(c). As the material stiffness increases, the benefits of visual feedback are more clearly seen as we can achieve near-perfect folding for cardboard, as shown in Fig. 4.13(d). All of our findings for rectangular folding also match the results of our diagonal folding experiment shown in Fig. 4.12(b1)–(b3), where the closed-loop approach once again achieves minimal sliding when compared to the baselines. Overall, the matching findings across all our experiments demonstrate the robustness of our formulation against material and geometric factors.

We observed one oddity for the folding scenario of $L_{gb} = 0.048$ and $C =$ Half, in which the open-loop algorithm outperformed our closed-loop variant, but the decrease in performance is on average only 1mm, which is attributable to repetitive discretization error caused by $N = 5$ replanning. In fact, as we use a discretization of $\delta = 2$mm for the manifold, compounding rounding errors can easily cause 1–2mm errors. With this in mind, our closed-loop method achieves an average fold length performance within a 1-2mm tolerance across all experiments.

We found that softer materials had the greatest error in spin error. As the table's surface is not perfectly flat, any amount of sliding will directly result in uneven spin as shown in Fig. 4.12(a). As the material stiffness increases, the spin errors become more uniform across the methods as the influence of friction is not enough to deform the paper. Nevertheless, we can see that our open and closed-loop algorithms resulted in less sliding than the baseline on average.

4.7 Summary and Outlook

In this chapter, we have introduced an innovative robotic control strategy capable of robustly folding sheets of paper with varying materials and geometries using a single manipulator. Our framework incorporates a combination of techniques from various disciplines, including physical simulation, machine learning, scaling analysis, and path planning. We have demonstrated the effectiveness of our framework through extensive real-world experiments, comparing it to both conventional and state-of-the-art paper folding strategies. Furthermore, we have implemented an efficient, nearly real-time visual feedback algorithm that further reduces folding errors. Our closed-loop sensorimotor control algorithm successfully handled challenging scenarios, including folding stiff cardboard with repeatable accuracy.

For future work, we aim to address the complex problem of creating arbitrary creases along sheets of paper with non-symmetric centerlines. Such non-symmetric papers cannot be represented as reduced-order 2D elastic rod models, necessitating more advanced formulations. Additionally, folding along paper regions with preexisting creases and folds will be crucial in achieving intricate folding tasks, such as robotic origami. Looking ahead, we plan to explore solutions to these challenges that leverage generalized problem formulations with data-driven control schemes, including reinforcement learning.

CHAPTER 5

Sim2Real Physically Informed Neural Controllers for Robotic Deployment of Deformable Linear Objects

The chapter focuses on the challenging task of manipulating deformable linear objects (DLOs), such as rods, cables, and ropes. These objects are subject to complex geometrically nonlinear deformations during manipulation, and different deformation modes (e.g., stretching, bending, and twisting) can lead to elastic instabilities. The goal is to accurately deploy a solid deformable linear object onto a rigid substrate along various prescribed patterns.

To tackle this problem, we present a novel framework that combines machine learning, scaling analysis, and physical simulations to develop a physics-based neural controller for DLO deployment. This approach accounts for the interplay between gravitational and elastic energies in the manipulated DLO and is designed to be robust against factors like friction and material properties. The paper demonstrates that the complex deployment process can be described with just three non-dimensional parameters, simplifying the control model and significantly improving computation speed.

The effectiveness of the proposed optimal control scheme is validated through a comprehensive robotic case study, which includes comparisons with an intuitive control method for deploying rods across various patterns. Additionally, the paper showcases the practicality of the control scheme by demonstrating the robot's ability to accomplish challenging high-level tasks like mimicking human handwriting, cable placement, and tying knots.

The underlying motivation and related literature is in § 5.1. Then, we fully analyze the system in § 5.2. Next, we formulate an optimal control policy for DLO deployment in § 5.3. Next, we introduce our designed robotic system in § 5.4. Finally, we give the conclusions and future avenues in § 5.6. The content of this chapter is from Ref. [107].

5.1 Motivation

The intelligent manipulation of deformable objects, such as ropes and fabric, is significant in robotics. In a world where most objects are non-rigid, equipping robots with the ability to manipulate deformable objects carries immense humanitarian and economic potential. This capability finds applications in various domains, ranging from robotic surgical suturing [108, 109] to wire management [110], laundry folding [111], and caregiving for the elderly and disabled [112, 113, 114, 115, 116]. However, the manipulation of deformable objects presents unique challenges. These objects undergo substantial geometrically nonlinear deformations, making distinguishing from observed deformations to concrete robotic manipulation strategies difficult. Consequently, developing precise and effective techniques for manipulating deformable objects remains an ongoing research challenge.

Among deformable objects, deformable linear objects (DLOs), including elastic rods and similar structures like cables, ropes, and wires [6], have attracted significant research interest. These objects have diverse industrial and domestic applications. In this chapter, we concentrate on rod-like DLOs and use the term “DLO” to refer to these solid elongated objects. Manipulating DLOs is a challenging endeavor due to their inherent complexity. These objects exhibit nonlinear behaviors arising from the interplay of multiple deformation modes, including stretching, bending, and twisting. Given the practical significance and difficulty associated with manipulating DLOs, there is a growing demand for robust and effective methods to tackle this task.

Prior works on manipulating DLOs can be divided into two categories. The first cate-

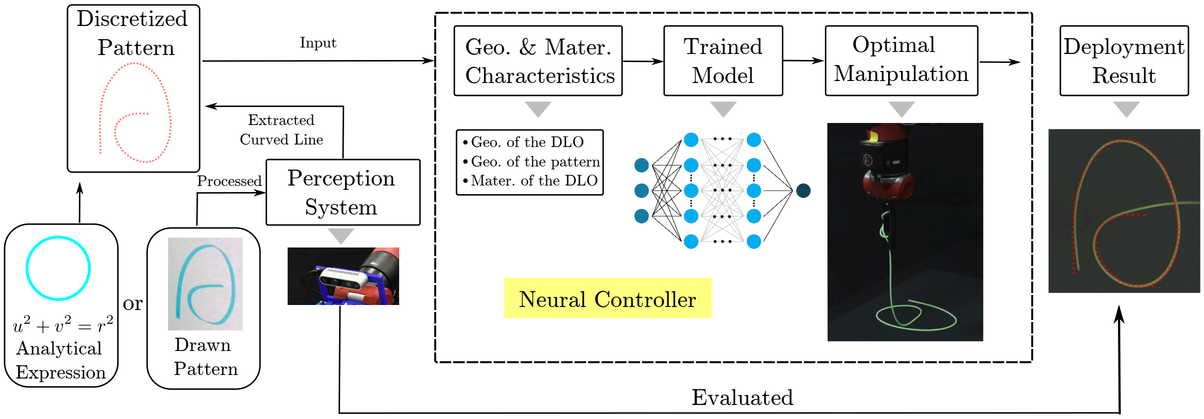
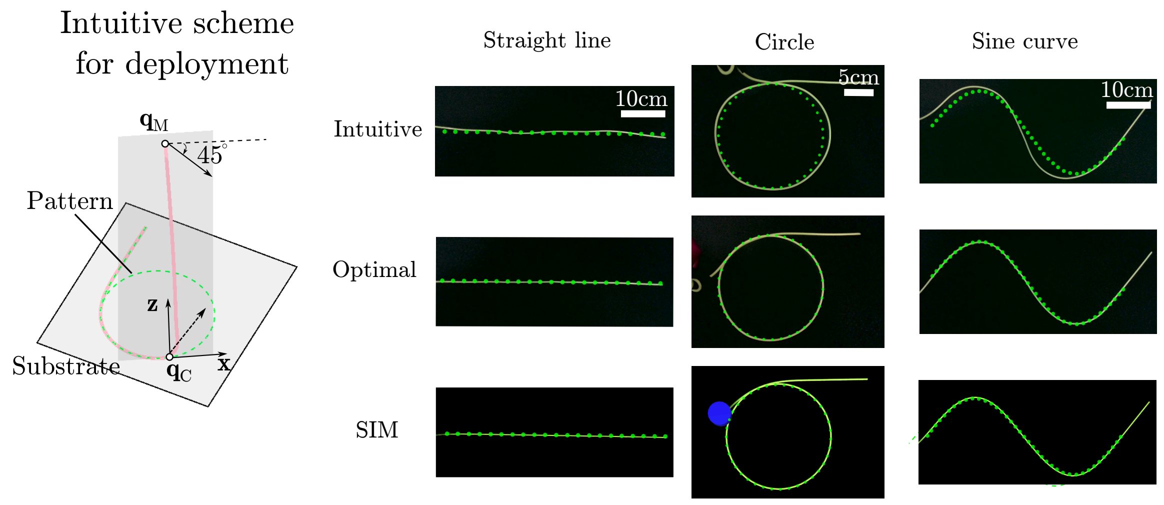


Figure 5.1: A full end-to-end pipeline for deploying a DLO with a sim2real physics-based deployment scheme. The pipeline begins by discretizing the DLO pattern, which can be obtained through user input via an analytical expression or a hand-drawn pattern scanned by a perception system [117]. A neural controller trained entirely from simulation then generates an optimal manipulation path for deploying the pattern, taking into account the pattern's shape and the geometrical and material properties of the DLO. Finally, the deployment result is evaluated using an Intel RealSense camera positioned to provide a top-down view of the pattern to assess the accuracy of the deployment.

gory involves robots manipulating DLOs to meet high-level objectives without precise control over the exact shapes of the DLOs. Examples of such tasks include knot tangling and untangling [118, 119], obstacle avoidance [120, 121], adhering to predefined paths and insertion [110, 122], among others. The second category focuses on robots exerting exact control over the specific shapes of DLOs. In this context, a key challenge lies in mapping the robot's actions and the resulting shape of the manipulated DLO [9, 123, 124].

In this chapter, we look into how to design a manipulation scheme for controlling the shape of elastic rods through deployment. Deployment entails controlling one end of the DLO to gradually place the DLO onto a substrate in a desired pattern with extraordinary precision, efficiency, and robustness. The complete end-to-end framework of our physics-based deployment approach is depicted in Fig. 5.1.



(a) Schematic of the intuitive scheme

(b) Deployment results for intuitive scheme, our optimal schemes and simulation results.

Figure 5.2: (a) Schematic of the intuitive control method from [123]. A DLO is being deployed along a circular pattern shown in dashed green. During the deployment process, the manipulated position $\mathbf{q}_M$ deploys along the tangent of the pattern $\mathbf{x}$ in a downward 45-degree angle with respect to the y-axis. The x-z-plane is shown in opaque gray. In addition, a comparison of experimental results between the intuitive control method, our designed optimal control method, and simulation results using the optimal control method for the patterns of straight line, circle, and sine curve are shown in (b).

5.1.1 Deployment of DLOs

Deploying DLOs finds applications in various practical applications, including cake icing decoration [19], marine cable laying [20], depositing carbon nanotubes [21], and melting electrospinning for advanced manufacturing [22]. Consequently, developing a concrete and applicable deployment scheme is critical to solving the shape control problem of DLOs in real-world scenarios.

A natural question arises: how can one precisely deploy a DLO along a specified pattern on a substrate? The intuitive approach might assume that during deployment, the manipulated end $\mathbf{q}_M$ is directly above the contact point $\mathbf{q}_C$, and the gripper's decrease in the distance along the negative z-axis corresponds to the added deployed length on the substrate.

However, this simplistic deployment strategy overlooks the nonlinear geometric deformations of the manipulated DLO, resulting in suboptimal deployments, as later experimental results demonstrate. A schematic of the intuitive deployment method, inspired by Ref. [123], is depicted in Fig. 5.2a.

This chapter proposes a framework that combines physically accurate simulation, scaling analysis, and machine learning to generate an optimized control scheme capable of deploying solid rod-like structures, which we refer to as DLOs, along any feasible pattern. While our control scheme does not currently incorporate energy dissipation from DLO manipulations, such as viscous threads, as our physics-based simulation relies on the rod model, the control scheme can be adapted by modifying the physical-based simulation within our integrated framework to account for these factors. Our robotic experiments validate the scheme using various DLOs, including elastic rods, ropes, and cables. We infuse physics into our manipulation scheme by leveraging physically accurate numerical simulations, facilitating full sim-to-real realization. Scaling analysis enables us to formulate the problem generously, leveraging non-dimensional parameters to ensure robustness against the material properties of the manipulated rods. Additionally, machine learning enables us to train a neural network to model the deployment control rules in a data-driven manner. The high inference speed of our neural controller enables real-time operation.

Our primary contributions include: (1) Formulating a solution to the DLO shape control problem through deployment with a physically robust scheme that leverages scaling analysis, ensuring generality against material, geometric, and environmental factors (e.g., friction); (2) Training a neural network using non-dimensional simulation data to serve as a fast and accurate neural controller for optimal deployment manipulations. The mechanics-based neural network solver exhibits remarkable efficiency and adequate accuracy compared to a numerical solver and (3) Demonstrating full sim-to-real realization through an extensive robotic case study, showcasing the success of our control method in various practical deployment patterns with different DLOs on diverse substrates. Furthermore, we illustrate the utility of our control scheme in complex high-level applications, such as replicating human handwriting, managing cables, and tying various knots.

Moreover, we have released our source code and supplementary videos[1].

5.1.2 Related Work

Constructing a mapping relationship from observations of a manipulated DLO to the robot's action space is the primary basis of controlling DLOs. To uncover this mapping relationship, prior works usually implemented models to predict or perception systems to observe the deformations of DLOs under various manipulations. Manipulation schemes are then generated based on the predicted or sensed data. Therefore, model-based and perception-based methods can be considered two of the main categories for tackling manipulation problems of deformable objects. Due to the outstanding performance of machine learning algorithms for processing and generalizing data from models and perceptions, learning-based approaches have become another mainstream solution. In fact, many prior works take advantage of a combination of these three methods to develop hybrid schemes for different manipulation tasks. Here, we systematically review prior scholarly contributions that have utilized techniques based on the three delineated categories to manipulate DLOs and other deformable objects.

Perception-based approaches involve utilizing sensors such as tactile sensors [110] and cameras [125, 126, 127, 80] to generate motions based on detected deformations. While sensors can capture the deformations as the manipulation proceeds, perception-based methods are usually not robust against the material and geometrical differences of the manipulated objects. In Ref. [125], a learning-based perception framework is presented based on the Coherent Point Drift algorithm, which can register states of manipulated DLOs with captured images. Ref. [126] developed state estimation algorithms for DLOs based on images

[1] Refer to https://github.com/StructuresComp/rod-deployment.

so that a robot can perform pick-and-place manipulation on the detected configuration. However, those perception systems based on cameras fail to extract accurate results when occlusions happen. To overcome this shortcoming, tactile sensors have become prevalent in robotics. For example, She et al. [110] implemented GelSight, a force feedback tactile sensor, to perform robotic cable management. Since sensing data by itself cannot predict future deformations of the manipulated objects, pure perception-based methods are typically insufficient for complex deformable material manipulation tasks.

Model-based methods usually construct a physically accurate model to predict the behavior of manipulated DLOs. Multiple methods exist for modeling DLOs [8, 6]. A simple and widely-used model, mass-spring systems, are often used to model deformable objects including ropes [83, 77, 128], fabrics [129, 130], etc. However, due to the simplification of mass-spring systems, such models usually suffer from inaccuracies when undergoing large deformations and lack of physical interpretability. Position-based dynamics is another type of modeling method that usually represents DLOs as chains of rigid bodies [131, 132, 133] and introduces constraints between the positions of those rigid bodies to simulate deformations. Though this method is straightforward and fast, physical interpretability is also lacking.

Finite element methods (FEM) are also popular for modeling deformable objects [134, 135, 136]. However, FEM usually requires considerable computation resources and is hardly suitable for online predictions. More recently, fast simulation tools from the computer graphics community have attracted researchers' attention. For example, Discrete Elastic Rods (DER) [13, 14] has become a robust and efficient algorithm for simulating flexible rods. Ref.[124] used DER as a predictive modeling tool and achieved promising performance in DLO manipulation tasks. Though various ways to model deformable objects exists, each has its respective strengths and weaknesses and often possesses a trade-off between computational efficiency and accuracy.

Finally, learning-based approaches have become prevalent as they can predict not only the deformable object's shape but also higher-level information such as forces [137]. Most prior works use human demonstrations or robot explorations to train controlling policies for different tasks. Refs.[9, 10, 11] fed human-made demonstrations to robots for learning control policies for shape control and knot-tying. Due to the tedium of constructing manual demonstrations, some researchers use the robots' automation to learn a policy purely from robotic exploration [138, 139]. Researchers have also looked into training policies purely from simulation [98] to acquire training data more efficiently. Although learning-based methods have shown promising performance for manipulating deformable objects, the trained policies are often only valid for specific tasks whose state distribution matches the training sets. In other words, learning-based approaches often fail when parameters such as the material and geometrical properties of the manipulated object change.

More relevant to the deployment task itself, Ref. [123] implemented the intuitive control method shown in Fig. 5.2a for controlling the shape of a rope to make a clove hitch knot. They achieve a success rate of 66% but require empirical hardcoded adjustments to their controlling scheme, indicating the intuitive approach's unsuitability for extreme precision deployment. Additionally, Ref. [124] uses a precise physical numerical model to predict the DLO's configuration during deployment. However, they use a trial-and-error method to exhaustively solve the optimal deployment path, which is computationally expensive and slow.

Although the three discussed types of methods are suitable to be combined when solving deformable manipulation problems given the complementariness of their pros and cons, how to develop a combined approach to take advantage of different types of approaches is still an open problem in the robotic community. We find that combining physically accurate simulations and machine learning can endow the learned model with excellent accuracy from the simulations and real-time performance because of the inference speed of the neural network. In addition, scaled physics analysis, which is a vital tool from the mathematical physics community, is valuable for augmenting the model with high generality.

In this study, we show how physical analysis can extract the true contributing factors of

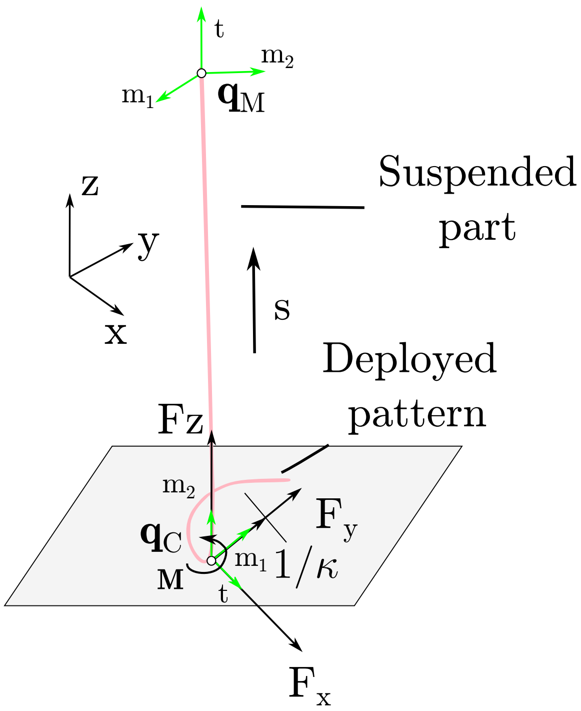


Figure 5.3: Schematic of deploying a DLO along a prescribed pattern.

the deployment problem and how a learning-based approach can generalize the information from physics to offer real-time computation speed for the manipulation task.

5.2 Physical Analysis and Controlling Rule Construction

This section examines the numerical framework used to investigate the nonlinear behaviors of the deformable linear object (DLO) during deployment. Following that, we will employ Buckingham's π theorem to identify the primary controlling parameters for the analyzed system.

When manipulating DLOs, we should consider their geometrically nonlinear deformations. For clarity, let's establish some notations: $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ will represent the unit vectors of the coordinate system centered at the contact end $\mathbf{q}_C$, as depicted in Figs. 5.2a and 5.2b.

As shown in Fig. 5.3, when a DLO is being deployed along a prescribed pattern on a rigid substrate, it can be divided into two distinct parts: the deployed section resting on the substrate and the suspended part that remains in contact with the substrate. We assume that the pattern on the substrate remains fixed since the DLO should ideally conform to the prescribed pattern. Consequently, the unknown deformations occur primarily within the suspended portion.

5.2.1 Solving the Suspended Part

To capture the deformations of the suspended part, we introduce some quantities to aid our analysis. First, we define $\mathbf{q}(s)$ to describe the position of the manipulated DLO's centerline, where s is the arc length along the DLO's centerline. Then, a material frame $\mathbf{m}(s) = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{t}] \in SO(3)$ is attached along the DLO to capture its rotation, where $\mathbf{t} = \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}s}$ is the tangent of the DLO. With the help of $\mathbf{q}(s)$ and $\mathbf{m}(s)$, we can fully describe the deformed configuration of the suspended part.

To solve for the configuration of the suspended part, we can treat it as an independent DLO segment starting from the connection node $\mathbf{q}_C$ to the manipulated node $\mathbf{q}_M$. Here, $\mathbf{q}_C = \mathbf{q}(0)$ denotes the connective node that links the deployed part and the suspended part. Maintaining the continuity of the manipulated DLO, the curvature vector $\boldsymbol{\kappa}$ at $\mathbf{q}_C$ can be obtained from the prescribed pattern, where the magnitude of $\boldsymbol{\kappa}$ is denoted as κ. The manipulated end grasped by the robot is denoted as $\mathbf{q}_M = \mathbf{q}(l_s)$, where l_s represents the total curve length of the suspended part. Deployment of the pattern is achieved solely by controlling $\mathbf{q}_M$. Since Eq. 1.8 implies that the DLO's configuration $\mathbf{q}(s)$ and $\mathbf{m}(s)$ can be solved when boundary conditions are determined, we can write down the governing equations for the suspended part as

$$
\begin{aligned}
& \mathcal{R}(\mathbf{q}) = 0, \\
\text{s.t.} \quad & \mathbf{q}(l_s) = \mathbf{q}_M, \quad \mathbf{R} = \mathbf{m}(l_s)\mathbf{m}(0)^T, \\
& \mathbf{q}(0) = \mathbf{q}_C, \quad \frac{\mathrm{d}\mathbf{q}(0)}{\mathrm{d}s} = \mathbf{t}(0), \quad \frac{\mathrm{d}\mathbf{t}(0)}{\mathrm{d}s} = \kappa\hat{\mathbf{y}},
\end{aligned}
\tag{5.1}
$$

where $\mathbf{q}_M$ is the position and $\mathbf{R}$ is the orientation of the manipulated end with respect to the connective node $\mathbf{q}_C$. Note that the position of the connective node $\mathbf{q}_C$, tangent $\mathbf{t}(0)$, and curvature vector $\kappa\hat{\mathbf{y}}$ can be determined from the deployed pattern, where $\hat{\mathbf{y}}$ is the unit vector illustrated in Fig. 5.3. By solving Eq. 5.1, we can obtain the configuration of the suspended part for any predefined pattern and manipulated end pose.

5.2.2 Influence of Forces and Friction

Once the deformed configuration is known, we can proceed to calculate the forces acting on the suspended part, which is crucial for achieving hyper-accurate control of the DLO. We represent the external forces as $\mathbf{F}^{\text{ext}} = F_x\hat{\mathbf{x}} + F_y\hat{\mathbf{y}} + F_z\hat{\mathbf{z}}$, applied to the suspended part from the connection node $\mathbf{q}_C$. Here, the moment M is a function of the arc length s, with, for example, $M(s)$ representing the twisting moment applied to the manipulated end. The quantities $\mathbf{F}^{\text{ext}}$ and $M(0)$ are associated with the friction coefficient μ between the substrate and the rod, and μ represents an unknown and uncontrollable environmental factor. Additionally, the quantities $\mathbf{F}^{\text{ext}}$ and $M(0)$ influence the tangent vector $\mathbf{t}(0)$ at the connection node $\mathbf{q}_C$ due to Newton's third law. Therefore, it is imperative to minimize the values of $\mathbf{F}^{\text{ext}}$ and $M(0)$ to establish an optimal control strategy.

Despite the optimal controlling rule minimizing the influence of friction, it is still worth clarifying the significance of friction within this context. Though we make the strong assumption that the deployed pattern remains fixed during deployment, this is only upheld if the following relation is satisfied for the deployed segment:

$$k_b \kappa'' \leq \mu_s \rho A g, \tag{5.2}$$

where k_b is the bending stiffness of the rod, κ'' is the second derivative of κ with respect to the arc length s ($\kappa'' = \frac{\mathrm{d}^2\kappa}{\mathrm{d}s^2}$), μ_s is the static friction coefficient, ρ is the volumetric density of the rod, A is the cross-sectional area, and g is the gravitational acceleration. Eq. 5.2 is derived by analyzing an arbitrary finite element of the deployed pattern with a clamped-end Euler-Bernoulli beam model. Clearly, friction plays a crucial role in the deployment process.

As a result, our designed optimal deployment strategy relies on adequate friction for effective execution while the scheme mitigates external tangential forces apart from the essential friction on the substrate. Consequently, the scheme necessitates only a modest static friction coefficient between the substrate and the manipulated DLO.

5.2.3 Computing Optimal Grasp

In addition to minimizing the external forces $\mathbf{F}^{\text{ext}}$ and the twisting moment $M(0)$ applied to the suspended part, we establish a rule for the manipulated end. Specifically, we aim to minimize the deformations induced by the robot end-effector on the manipulated node $\mathbf{q}_M$, ensuring that the curvature (bending deformations) at the manipulated end remains close to 0. This results in the following optimization problem to compute the optimal grasp:

$$\begin{aligned} (\mathbf{q}_M^*, \mathbf{R}^*) &= \arg\min \left(\|\mathbf{F}^{\text{ext}}\|^2 + \left(\frac{\|\mathbf{M}(0)\|}{h} \right)^2 \right) \\ \text{s.t.} \quad \frac{\mathrm{d}\mathbf{q}(0)}{\mathrm{d}s} &= \mathbf{t}(0), \qquad \frac{\mathrm{d}\mathbf{t}(0)}{\mathrm{d}s} = \kappa \hat{\mathbf{y}}, \\ \frac{\mathrm{d}^2\mathbf{q}(l_s)}{\mathrm{d}s^2} &= 0, \qquad \mathcal{R}(\mathbf{q}) = 0. \end{aligned} \tag{5.3}$$

By solving Eq. 5.3, we can obtain the optimal grasp represented by $\mathbf{q}_M^*$ and $\mathbf{R}^*$. Physical analysis reveals the existence of a direct mapping relationship between the contributing factors and the optimal grasp. Here, we denote the stiffness of the DLO by stretching stiffness k_s, bending stiffness k_b, and twisting stiffness k_t, which governs the geometric changes of the manipulated DLO. In addition, the density ρ and rod radius h contribute to the analyzed system. By adding in additional geometric properties such as suspended length l_s and curvature κ, the mapping relationship

$$(\mathbf{q}_M^*, \mathbf{R}^*) = f(l_s, \kappa, k_s, k_b, k_t, h, \rho), \tag{5.4}$$

can be constructed where $f(\cdot)$ is a highly nonlinear (and unknown) function that describes the controlling rule.

However, it's essential to acknowledge the high input dimensionality of Eq. 5.4. In practical terms, accurately learning the mapping $f(\cdot)$ would require conducting an exhaustive exploration of a wide range of material and geometric parameters within simulations. This data collection process can quickly become unmanageable due to the curse of dimensionality. To overcome this challenge, we can employ scaling analysis to derive an equivalent reduced-order mapping that simplifies the problem.

5.2.4 Scaling Analysis via Buckingham's π Theorem

In this project, we leverage Buckingham's π theorem to reduce the dimensions of the mapping $f(\cdot)$. Buckingham's π theorem is a fundamental principle in dimensional analysis, which posits that a physically meaningful equation involving n physical parameters can be expressed using a reduced set of $p = n - k$ dimensionless parameters derived from the original parameters. In this context, k denotes the number of physical dimensions. By applying Buckingham's π theorem, we can obtain a reduced-order, non-dimensionalized mapping $\mathcal{F}(\cdot)$

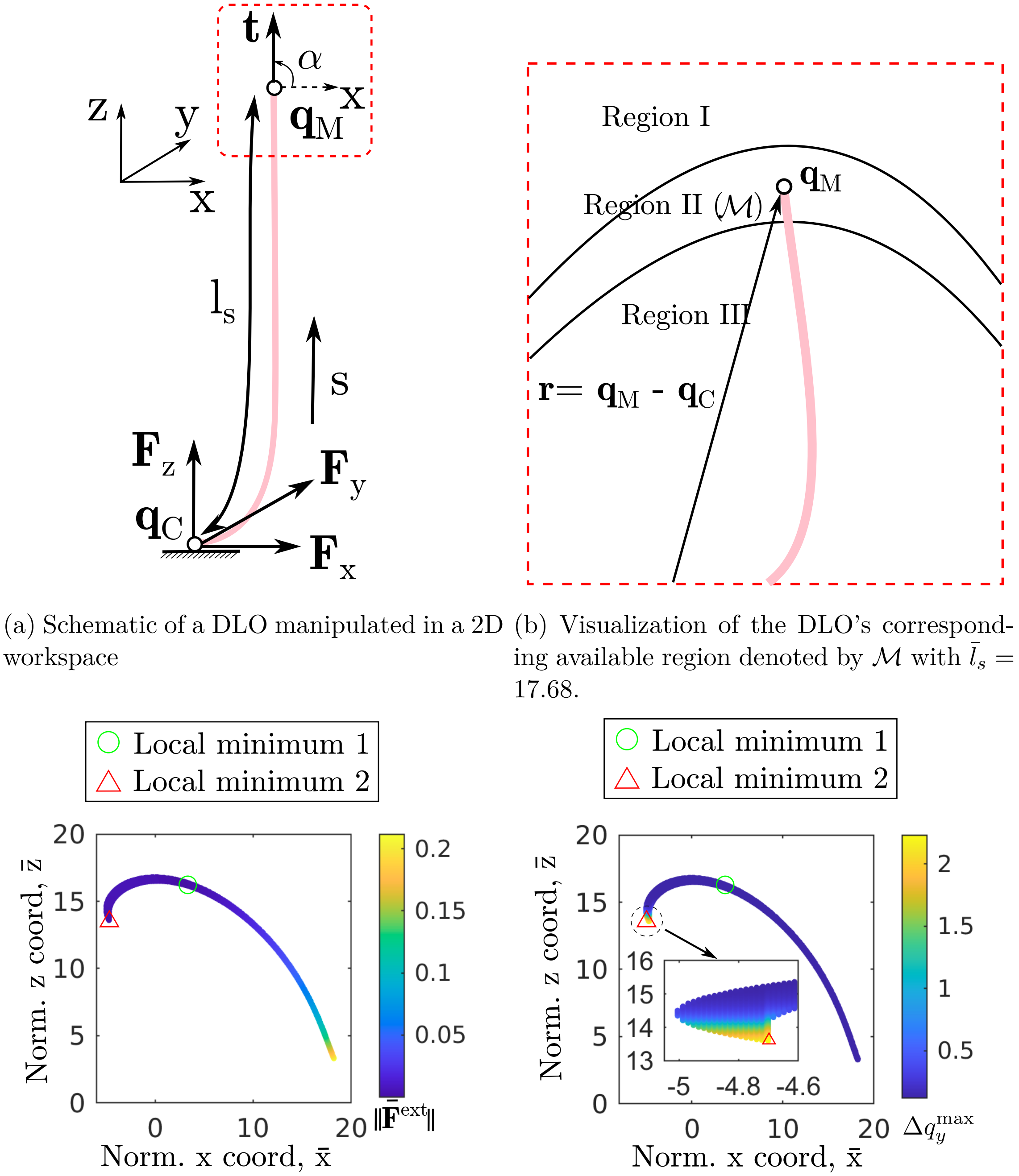
from the original function f:

$$\begin{aligned}
(\bar{\mathbf{q}}_M^*, \mathbf{R}^*) &= \mathcal{F}(\bar{l}_s, \bar{\kappa}, \bar{k}_s), \\
L_{gb} &= \left(\frac{k_b}{2\pi h^2 \rho g} \right)^{1/3}, \\
\bar{k}_s &= \frac{k_s L_{gb}^2}{k_b}, \\
\bar{\mathbf{q}}_M^* &= \frac{\mathbf{q}_M^* - \mathbf{q}_C}{L_{gb}}, \\
\bar{l}_s &= \frac{l_s}{L_{gb}}, \\
\bar{\kappa} &= \kappa L_{gb}.
\end{aligned} \tag{5.5}$$

Hereafter, all quantities with $\bar{()}$ indicate normalized quantities. In Eq. 5.5, all quantities are unitless so that the mapping relationship $\mathcal{F}(\cdot)$ maps from the unitless groups encapsulated the geometric and material properties to the unitless optimal robotic grasp. The benefit of doing such is that we reduce the dimensions of the mapping function $\mathcal{F}(\cdot)$ in Eq. 5.4 and eliminate the dependence of $\mathcal{F}(\cdot)$ on the units. Note that in Eq. 5.5, we do not consider the influence of the twisting stiffness k_t in this chapter since twisting energies are minimal compared to bending and stretching. However, the influence of k_t can also be analyzed with our proposed analysis. The following paragraph will show how to establish the nonlinear mapping function in Eq. 5.5.

5.3 Optimization and Machine Learning

In this section, we will delve deeper into the optimization of the system to derive the nonlinear mapping function presented in Eq .5.5. Given the significant nonlinearity of the system, we will initially solve Eq .5.5 using a numerical optimization solver in a data-driven manner. During this process, we will also analyze the system's elastic instability to determine the optimal robotic grasp for the deployment task. Subsequently, we will reconstruct Eq. 5.5



(a) Schematic of a DLO manipulated in a 2D workspace

(b) Visualization of the DLO's corresponding available region denoted by $\mathcal{M}$ with $\bar{l}_s = 17.68$.

(c) Force distribution of $\mathcal{M}$ with $\bar{l}_s = 17.68$

(d) The maximum geometric deformation of the suspended part under a disturbance of $\Delta\bar{y} = 0.12$ along the y-axis.

Figure 5.4: Avaialble region of the robotic grasp in 2D deployment

utilizing a neural network to leverage its high inference speed. This neural controller will serve as the controlling law for our robotic system to execute various deployment tasks in § 5.5.

5.3.1 Elastic Instability in Deployment along a Straight Line

In this section, we explore an intriguing physical phenomenon known as elastic instability. Elastic instability occurs when boundary condition changes lead to a deformed structure's instability. Visually, it can be observed as a small geometric perturbation of the system, resulting in a significant configuration change [140]. An example of this phenomenon can be seen when a robot employs an intuitive control method to deploy a DLO along a straight line, but the rod unexpectedly adopts a curved shape on the substrate. This observation is counterintuitive because the intuitive method only manipulates the DLO in the 2D plane (specifically, the x-z plane), as illustrated in Fig. 5.4a. Therefore, the suspended part should ideally experience only 2D deformations within that plane, avoiding significant deformations along the y-axis. However, the observed phenomenon results from the unaccounted elastic instability of the manipulated DLO.

Given this, it is crucial to consider elastic instability when designing an optimal deployment scheme to ensure that the robot's grasp and possible jittering of the manipulator does not introduce large undesired deformations in the DLO. To achieve this, we will thoroughly analyze all potential robot grasps for manipulating a DLO in the x-z plane to achieve a straight-line deployment. We aim to identify an optimal grasp that satisfies Eq. 5.3 while effectively preventing the manipulated DLO from buckling due to elastic instability.

Algorithm 6: Bisection Method for Obtaining $\partial\mathcal{M}$

```
Input: $\bar{l}_s, \bar{k}_s, \nu$
Output: $\partial\mathcal{M}$
Func DiscoverManifoldBoundary($\bar{l}_s, \bar{k}_s$):
    $\theta \leftarrow$ a small positive value
    $\beta \leftarrow$ a small positive value
    $\partial M \leftarrow$ initialize an empty list
    $\delta \leftarrow$ a small positive value as tolerance
    $\mathcal{R} \leftarrow$ initialized rod solver with $\bar{l}_s, \bar{k}_s, \nu$
    while $\theta \leq \pi$ do
        $\mathbf{r} \leftarrow (\bar{l}_s \cos(\theta), \bar{l}_s \sin(\theta))$
        do
            $\mathbf{r} \leftarrow (1+\beta)\mathbf{r}$
            $\bar{F}_z \leftarrow \mathcal{R}(\mathbf{r})$
        while $\bar{F}_z < 0$
        $\mathbf{r}_c \leftarrow \mathbf{r}$
        while $\mathcal{C}$ is not satisfied do
            $\mathbf{r} \leftarrow \mathbf{r} - \delta\hat{\mathbf{r}}$
            $\bar{\mathbf{q}}, \bar{F}_z \leftarrow \mathcal{R}(\mathbf{r})$
            if $\|\mathbf{r}\| < 0$ then
                break
        $\mathbf{r}_f \leftarrow \mathbf{r}$
        while $\|\mathbf{r}_c - \mathbf{r}_f\| \geq \delta$ do
            $\bar{\mathbf{q}}, \bar{F}_z \leftarrow \mathcal{R}(\mathbf{r})$
            if $\mathcal{C}$ is satisfied then
                $\mathbf{r}_f \leftarrow \mathbf{r}$
            else
                $\mathbf{r}_c \leftarrow \mathbf{r}$
            $\mathbf{r} \leftarrow (\mathbf{r}_c + \mathbf{r}_f)/2$
        $\partial\mathcal{M}$.append($(\mathbf{r}\cos\theta, \mathbf{r}\sin\theta)$)
        $\mathbf{r}_c \leftarrow \mathbf{r}$
        $\mathbf{r}_f \leftarrow (0, 0)$
        while $\|\mathbf{r}_c - \mathbf{r}_f\| \geq \delta$ do
            $\bar{\mathbf{q}}, \bar{F}_z \leftarrow \mathcal{R}(\mathbf{r})$
            if $\mathcal{C}$ is satisfied then
                $\mathbf{r}_c \leftarrow \mathbf{r}$
            else
                $\mathbf{r}_f \leftarrow \mathbf{r}$
            $\mathbf{r} \leftarrow (\mathbf{r}_c + \mathbf{r}_f)/2$
        $\partial\mathcal{M}$.append($(\mathbf{r}\cos\theta, \mathbf{r}\sin\theta)$)
        $\theta \leftarrow \theta + \delta\theta$
return $\partial\mathcal{M}$
```

5.3.2 Discovering Potential Grasp Region

Given the suspended part's geometric properties and boundary conditions, we can write down the constraints $\mathcal{C}$ which should be satisfied:

$$\begin{aligned}&\bar{\mathbf{q}}(\bar{s}) \cdot \hat{\mathbf{z}} \geq 0 \;\forall\; \bar{s} \in [0, \bar{l}_s],\\&\bar{\mathbf{F}}^{\text{ext}} \cdot \hat{\mathbf{z}} \geq 0.\end{aligned} \tag{5.6}$$

These constraints serve to ensure two important conditions: (i) The suspended part of the DLO should remain above the substrate to prevent unwanted contact. (ii) The external contact forces along the z-axis should always be greater than or equal to 0 to avoid stretching beyond a certain threshold.

By solving Eq. 5.1 while adhering to constraints $\mathcal{C}$, we can determine all potential robot grasps for the manipulated end, forming a closed manifold $\mathcal{M}$ for a fixed normalized suspended length $\bar{l}_s$. At the connective node $\bar{\mathbf{q}}_C$, we set the boundary conditions as $\mathbf{t}(0) = (1, 0, 0)$ and $\bar{\kappa} = 0$. Each point in the manifold $\mathcal{M}$ corresponds to a position $\bar{\mathbf{q}}_M$ and a rotation $\mathbf{R}$ of the manipulated end. Given that the deformed configuration is located within the 2D x-z plane, we can represent the position of $\bar{\mathbf{q}}M$ using a 2×1 vector $\bar{\mathbf{q}}M = (\bar{x}\text{Top}, \bar{z}\text{Top})$ and a scalar value α to denote the rotation information. For example, the tangent $\mathbf{t}(\bar{l}_s) = (\cos(\alpha), \sin(\alpha))$ is shown in Fig. 5.4a. Since the manifold $\mathcal{M}$ is a closed set, we only need to obtain the boundary of the manifold $\partial\mathcal{M}$.

To find the boundary $\partial\mathcal{M}$, we explore along a ray $\mathbf{r}$ from the connective node $\bar{\mathbf{q}}_C$ to the manipulated node $\bar{\mathbf{q}}_M$. The robot grasp along the ray can be divided into three regions, as shown in Fig. 5.4b. When the robot grasp exists in regions I and III, constraints $\mathcal{C}$ are unsatisfied. In region I, the external force $\bar{F}_z = F_z h^2/k_b$ is smaller than 0, violating the constraints as stretching occurs. In region III, the manipulated end is too low, leading to contact between the suspended part and the substrate. Thus, region II, existing between regions I and III, represents the manifold $\mathcal{M}$ area that satisfies the constraints $\mathcal{C}$. The pseudocode for the bisection method is given in Algo. 6.

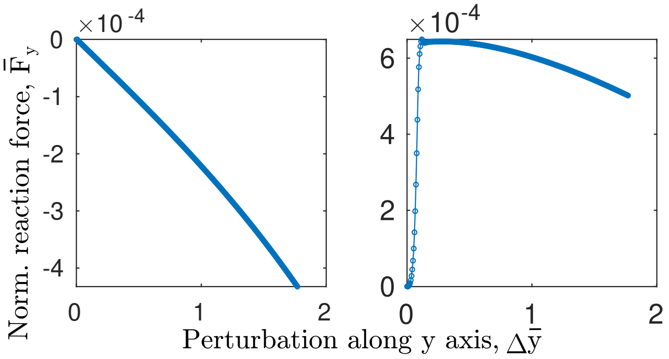
Note that θ in Algo. 6 is the angle between the x-axis and ray $\mathbf{r}$. A specific case for $\bar{l}_s = 17.68$ is visualized in Fig. 5.4c. Since deformations only occur in the x-z plane, the twisting moment $\bar{\mathbf{M}}(0) = \mathbf{M}(0)h/k_b$ applied on the connective node $\bar{\mathbf{q}}_C$ is always 0. To achieve the optimal pose of the manipulated end for $\bar{l}_s = 17.68$, we need to find the poses in $\mathcal{M}$ that minimizes $\|\bar{\mathbf{F}}^{\text{ext}}\|$. Two local minima are found in the case shown in Fig. 5.4c, corresponding to two solutions of Eq. 5.3. As stated before, we must select the local minima corresponding to the stable deformed suspended part.

5.3.3 Checking Elastic Instability via Perturbations

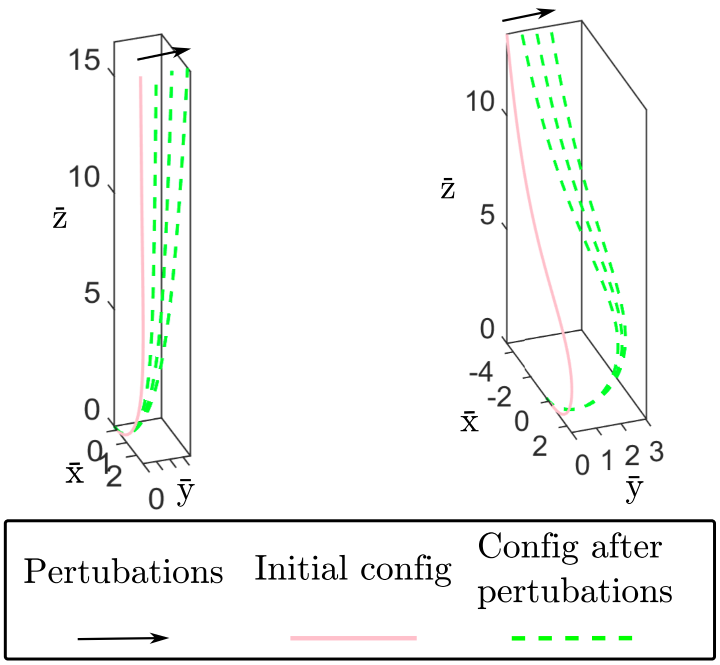
To test the elastic stability of these local minima, we introduce a disturbance $\Delta\bar{y} = y/L_{gb}$ along the y-axis while the manipulated end $\bar{\mathbf{q}}_M$ is at each local minimum. Fig. 5.5a demonstrates the variations in $\bar{F}_y = F_y h^2/k_b$ and the resulting configurations due to these perturbations for each local optimum.

For local minimum 2, we observe a sudden snapping process, marked by an abrupt change. Conversely, the disturbance applied to local minimum 1 leads to a continuous and gradual change. As a result, we can conclude that the optimal configuration for deploying the DLO is at local minimum 1 since this minimum corresponds to a configuration with gentler bending deformations of the suspended part, providing greater elastic stability.

Here, we also demonstrate that the neighboring region around the elastic instability points has a higher tendency for significant deformations when the jittering of the manipulator occurs. In simulation, we introduce a slight disturbance of $\Delta\bar{y} = 0.12$ along the y-axis for all potential robot grasps on the manifold $\mathcal{M}$. Fig. 5.4d shows the maximum displacement of the suspended part along the y-axis, denoted as $\Delta q_y^{\max} = \max_{0 \le s \le l_s}(\mathbf{q}(s) \cdot \hat{\mathbf{y}})$, resulting from this small disturbance. The results clearly indicate that the neighboring region around local minimum 2 exhibits a higher tendency for significant deformations along the y-axis.



(a) Change of the magnitude of normalized force $\bar{F}_y$ when adding a perturbation along the y-axis at local minimum 1 and local minimum 2 is shown.



(b) Change of the configurations of the rod when adding the perturbations at local minimum 1 and local minimum 2 for $\bar{l}_s = 17.68$ is shown.

Figure 5.5: Stability of two local minimum during 2D deployment.

Consequently, robot grasps within this region are more likely to induce instability in the manipulated DLO.

We can now define the optimal deployment rule for a straight line using the method introduced in this section. In the next section, our focus will shift to optimal 3D manipulation, which involves deploying patterns with nonzero curvature. In the following contexts, we focus on optimal 3D manipulation, i.e., deploying patterns with nonzero curvature. The following section discusses how to use a first-order optimization algorithm to solve Eq. 5.3 for deploying any arbitrary prescribed pattern, where the optima for straight-line deployment is used as seeds when searching for the optima of more complex patterns.

5.3.4 Deployment in 3D Workspace

As discussed in § 5.2, constructing the mapping relationship $\mathcal{F}(\cdot)$ in Eq. 5.5 is essential for achieving optimal deployment in the 3D workspace. For the connective node of any prescribed pattern, since the pattern deformations are only in the x-y plane, we can ensure that the twisting moment $\mathbf{M}(0)$ is always 0. Therefore, the optimal pose of the manipulated end can be obtained by minimizing $\|\bar{\mathbf{F}}^{\text{ext}}\|$ through the following optimization:

$$\boldsymbol{\nabla}_{\bar{\mathbf{q}}_M} \|\bar{\mathbf{F}}^{\text{ext}}\| = \frac{\partial \bar{\mathbf{F}}^{\text{ext}}}{\partial \bar{\mathbf{q}}_M} \bar{\mathbf{F}}^{\text{ext}} = 0. \tag{5.7}$$

As the deploying rod is a continuous system, $\bar{\mathbf{F}}^{\text{ext}}$ must change when $\mathbf{q}_M$ changes. Therefore, we can convert Eq. 5.7 to be a root finding problem

$$\bar{\mathbf{F}}^{\text{ext}} = 0. \tag{5.8}$$

Since we stated previously that solving the configurations of the deploying DLO is a boundary value problem and the boundary conditions on the connective end are determined by the pattern's shape, the external forces $\bar{\mathbf{F}}^{\text{ext}}$ are influenced solely by the manipulated end pose $\mathbf{q}_M$, with a unique corresponding $\mathbf{R}$ for describing the rotation of the manipulated end.

Due to the high nonlinearity of the DLO, it is challenging to analytically solve the root-

```
Algorithm 7: Gradient Descent for Optimal Grasp
  Input: $\bar{l}_s, \bar{\kappa}\hat{\mathbf{y}}, \bar{k}_s, \nu$
  Output: $\mathbf{q}_M^{\star}$
  Func OptimalGrasp($\bar{l}_s, \bar{\kappa}, \bar{k}_s$):
      $k \leftarrow 0$
      $\delta \leftarrow$ a small value as tolerance
      $\bar{\mathbf{q}}_M^{(k)} \leftarrow$ initialize a random pose of end-effector
      $\mathcal{R}(\cdot) \leftarrow$ initialize the rod solver with $\bar{l}_s, \bar{\kappa}, \bar{k}_s$
      do
          $\bar{\mathbf{F}}^{\text{ext}} \leftarrow \mathcal{R}(\bar{\mathbf{q}}_M^{(k)})$
          $\mathbf{J}^{\text{ext}} \leftarrow$ Eq. 5.9
          $\Delta\bar{\mathbf{q}} \leftarrow (\mathbf{J}^{\text{ext}})^{-1}\bar{\mathbf{F}}^{\text{ext}}$
          $\alpha \leftarrow$ LineSearch($\bar{\mathbf{q}}_M^{(k)}, \Delta\mathbf{q}, \|\bar{\mathbf{F}}^{ext}\|, \mathcal{R}$)
          $\bar{\mathbf{q}}_M^{(k+1)} \leftarrow \bar{\mathbf{q}}_M^{(k)} - \alpha\Delta\bar{\mathbf{q}}$
          $k \leftarrow k + 1$
      while $\|\bar{\mathbf{F}}^{ext}\| \geq \delta$
      $\bar{\mathbf{q}}_M^{*} \leftarrow \bar{\mathbf{q}}_M^{(k)}$
      return $\bar{\mathbf{q}}_M^{*}$
```

finding problem in Eq. 5.8. Therefore, we employ a finite difference approach to calculate the numerical Jacobian of $\mathbf{F}^{\text{ext}}$. We perturb the manipulated end along the x, y, and z-axes by a small distance δ and use finite differences to compute the numerical Jacobian:

$$\mathbf{J}^{\text{ext}} = \frac{1}{\delta}\begin{bmatrix} \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M + \delta\hat{\mathbf{x}}) - \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M), \\ \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M + \delta\hat{\mathbf{y}}) - \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M), \\ \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M + \delta\hat{\mathbf{z}}) - \bar{\mathbf{F}}^{\text{ext}}(\bar{\mathbf{q}}_M) \end{bmatrix}^T, \tag{5.9}$$

where T is the transpose operator and $\delta\hat{\mathbf{x}}, \delta\hat{\mathbf{y}}$, and $\delta\hat{\mathbf{z}}$ are small perturbations along x, y, and z-axes, respectively, i.e., $\delta\hat{\mathbf{x}} = [\delta, 0, 0]^T$.

Here, $\mathbf{J}^{\text{ext}}$ is a 3×3 matrix and can be used to calculate the Newton search step so that Eq. 5.8 can be solved with a gradient descent method. Further details of this solving process are stated in Algo. 7. Additionally, we also implement a line search algorithm to help determine the appropriate step size for the Newton search step $\Delta\bar{\mathbf{q}}$ as shown in Algo. 8.

Algorithm 8: Line Search Algorithm

```
Input: $\bar{\mathbf{q}}_M, \Delta\mathbf{q}, f_0, \mathcal{R}$
Output: $\alpha$
Func LineSearch($\bar{\mathbf{q}}_M, \Delta\mathbf{q}, f_0, \mathcal{R}, \alpha_0 = 1, m = 0.5$):
    $\alpha \leftarrow \alpha_0$
    $k \leftarrow 0$
    success $\leftarrow$ False
    do
        $\bar{\mathbf{q}}_M^{(k)} \leftarrow \bar{\mathbf{q}}_M - \alpha\Delta\mathbf{q}$
        $\bar{\mathbf{F}}^{\text{ext}} \leftarrow \mathcal{R}(\bar{\mathbf{q}}_M^{(k)})$
        $f^{(k)} \leftarrow \|\bar{\mathbf{F}}^{\text{ext}}\|$
        if $f^{(k)} \geq f_0$ then
            $\alpha \leftarrow m\alpha$
            $k \leftarrow k + 1$
        else
            success $\leftarrow$ True
    while not success
    return $\alpha$
```

In this project, both position $\bar{\mathbf{q}}_M$ and rotation $\mathbf{e}$ of the manipulated end are represented as 3×1 vectors: $\bar{\mathbf{q}}_M = (\bar{x}^{\text{Top}}, \bar{y}^{\text{Top}}, \bar{z}^{\text{Top}})$ and $\mathbf{e} = (e_x, e_y, e_z)$. The rotation vector $\mathbf{e}$ can be translated to a rotation matrix through an axis-angle representation $(\hat{\mathbf{e}}, \|\mathbf{e}\|)$, where $\|\mathbf{e}\|$ is the rotation angle along the rotation axis $\hat{\mathbf{e}} = \mathbf{e}/\|\mathbf{e}\|$. For an input tuple $(\bar{l}_s, \bar{\kappa}, \bar{k}_s)$, we can now solve for the optimal pose of the manipulated end $(\mathbf{q}_M^*, \mathbf{e}^*)$. Visualizations of the discretely solved optimal poses obtained from simulation are shown as red hollow circles in Fig. 5.6.

We now know how to obtain the optimal manipulation pose given the input $(\bar{l}_s, \bar{\kappa}, \bar{k}_s)$ with simulations. A numeric solver based on simulations for generating the optimal trajectory for various prescribed patterns is released (see 1). However, solving for the optimal poses with the numeric solver makes real-time manipulation infeasible as trajectory generation can take several hours. Instead, the following section introduces using a neural network to learn the optimal controlling rule for fast real-time inference.

5.3.5 Training the Neural Controller

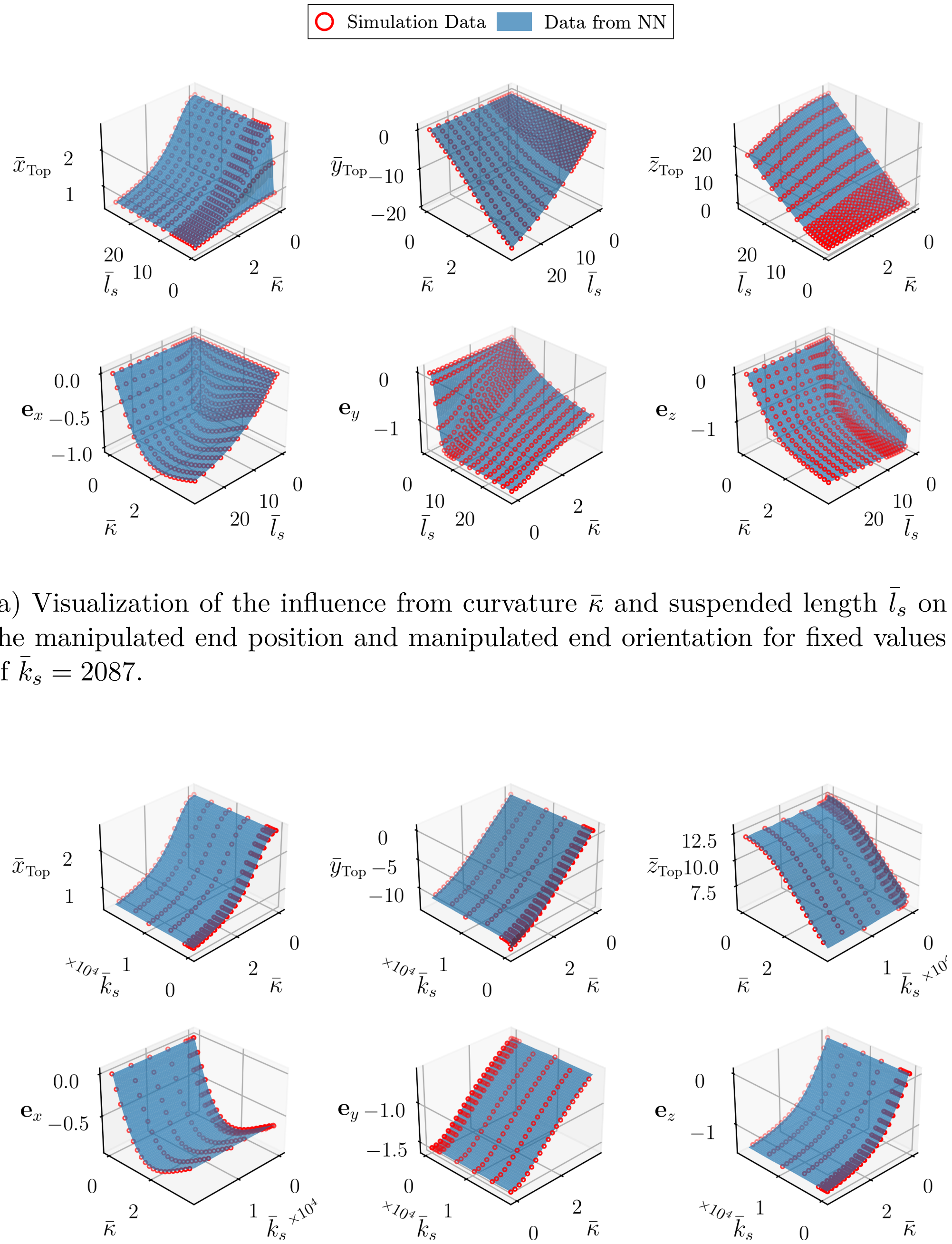
Rather than obtaining the optimal grasp through the numerical solver detailed in the previous section, we train a neural network to learn an analytical approximation of $\mathcal{F}(\cdot)$ similar to the approach in [137]. We use a simple fully-connected feed-forward nonlinear regression network consisting of 4 hidden layers, each with 392 nodes, as the network architecture. Aside from the output, each layer is followed by a rectified linear unit (ReLU) activation.

We frame the neural controller to have an input $\mathbf{i} \in \mathbb{R}^3$ and an output $\mathbf{o} \in \mathbb{R}^6$, where the input consists of the three non-dimensional values $\bar{l}_s$, $\bar{\kappa}$, and $\bar{k}_s$ and the output consists of two concatenated 3×1 vectors: the optimal position $\bar{\mathbf{q}}_M^*$ and rotation $\mathbf{e}^*$ of the manipulated end. Using our simulation framework, we construct a dataset $\mathcal{D}$ consisting of 6358 training samples.

When training the neural controller, we first preprocess all inputs $\mathbf{i}$ through the standardization.

$$\mathbf{i}' = \frac{\mathbf{i} - \bar{\mathbf{i}}_{\mathcal{D}}}{\boldsymbol{\sigma}_{\mathcal{D}}},$$

where $\bar{\mathbf{i}}_{\mathcal{D}}$ and $\boldsymbol{\sigma}_{\mathcal{D}}$ are the mean and standard deviation of the input portion of the dataset $\mathcal{D}$. Afterward, we use an initial 80-20 train-val split on the dataset $\mathcal{D}$ with a batch size of 128. We use mean absolute error (MAE) as our loss and use a training strategy that alternates between stochastic gradient descent (SGD) and Adam whenever training stalls. In addition, the batch size is gradually increased up to a max size of 2048, and the entire dataset is used to train the controller once MAE reaches < 0.003. With this scheme, we achieve a final MAE of < 0.0015. The neural network's approximation of $\mathcal{F}(\cdot)$ can be seen visualized in Fig. 5.6.



(a) Visualization of the influence from curvature $\bar{\kappa}$ and suspended length $\bar{l}_s$ on the manipulated end position and manipulated end orientation for fixed values of $\bar{k}_s = 2087$.

(b) visualization of the influence from stretching stiffness $\bar{k}_s$ and curvature $\bar{\kappa}$ on the manipulated end position and manipulated end orientation with fixed values of $\bar{l}_s = 13.68$.

Figure 5.6: Visulization of the computed optimal grasps with various $\bar{l}_s$, $\bar{\kappa}$, and $\bar{k}_s$

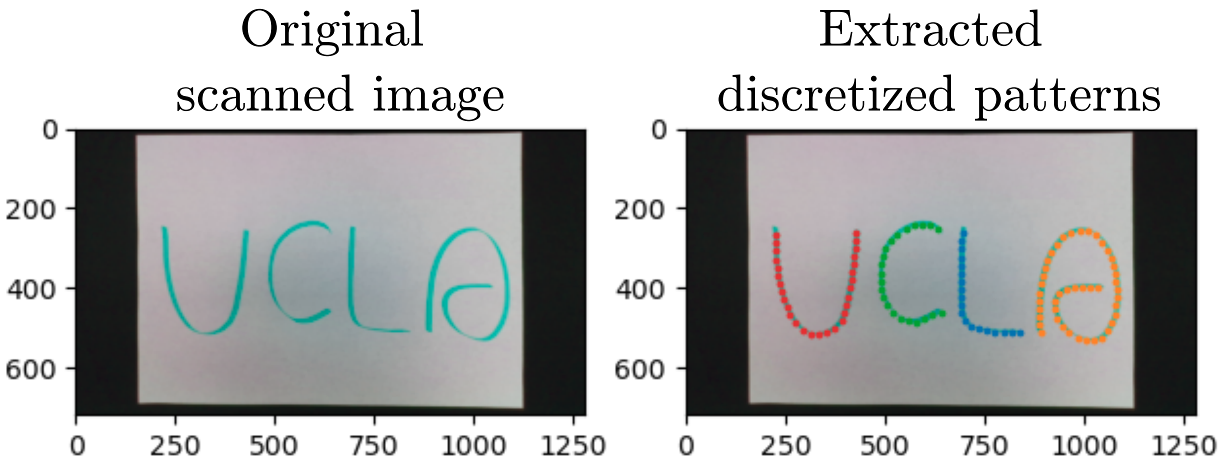


Figure 5.7: Handwritten letters and the corresponding extracted discretized patterns using mBEST [117].

5.4 Robotic System

5.4.1 Perception System

To obtain the Cartesian centerline coordinates of the deployed DLO (or drawn patterns), we use the DLO perception algorithm mBEST [117]. This algorithm obtains the centerline coordinates of DLOs within an image by traversing their skeleton pixel representations. The ambiguity of path traversal at intersections is handled by an optimization objective that minimizes the cumulative bending energy of the DLOs during the pixel traversal. One case of extracting discretized patterns from the hand-writing pattern is shown in Fig. 5.7. RGB images of the deployed DLO are obtained through an Intel RealSense D435 camera as shown in Fig. 5.9. Further details of the perception algorithm itself can be found in the referenced paper.

5.4.2 Motion Planning with the Neural Controller

In Fig. 5.1, we showcase the full end-to-end pipeline of our proposed deployment scheme. Here, we give a full description of how to integrate the trained neural controller into a robot motion planner.

The first step of the deployment process is to specify the desired pattern. This pattern can be defined by either an analytical function or detected as a drawn curve as shown in Fig. 5.1. Note that the pattern $\mathbf{P}(s)$ is treated as a function of the curve length s. Based on the configuration of the pattern, we can compute the required inputs for the neural controller when the connective node $\mathbf{q}_C$ achieves each point in the pattern $\mathbf{P}(s)$. The details of generating an optimal trajectory based on the pattern $\mathbf{P}(s)$ and the properties of the manipulated rod are given in Algo. 9.

In Algo. 9, κ and $\mathbf{T}$ are all functions of the arc length s of the pattern, where $\mathbf{T}$ is the tangent along the pattern. With Algo. 9, we obtain the optimal grasp trajectory τ and then use OMPL [141] to generate a valid motion planning sequence on real robot system.

One highlight of our overall robotic system is its realtime capability. The realtime efficiency of the perception algorithm has been validated by [117] while the average end-to-end time to generate a full optimal deployment motion plan is less than 1 second. Therefore, our approach is also efficient enough for sensorimotor closed-loop control. However, as offline control has achieved excellent deployment accuracy in our experiments, online control is not carried out in this work.

5.5 Results

5.5.1 Measurement of Material Parameters

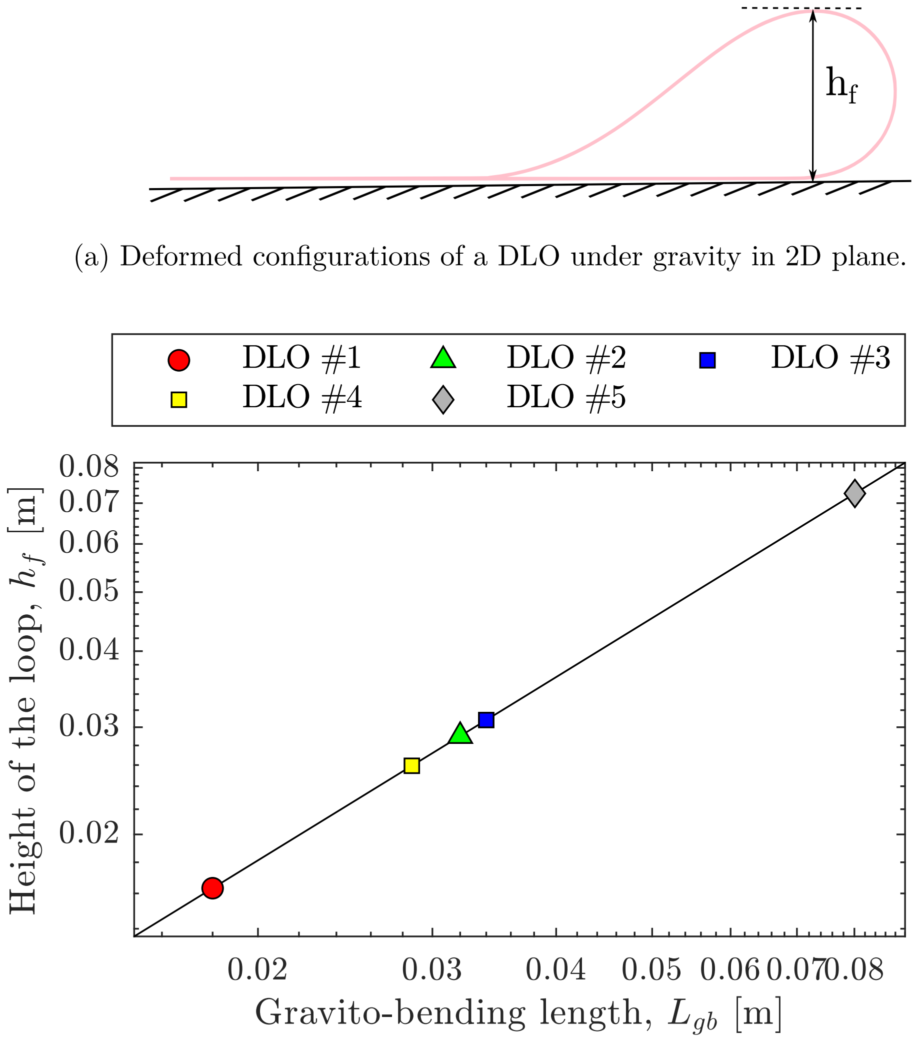
To carry out deployment with our proposed scheme, we must validate its efficacy with comprehensive experiments. In this project, we choose to deploy various DLOs on different

```
Algorithm 9: Optimal Deployment Trajectory
  Input: $\mathbf{P}, L, L_{gb}, \bar{k}_s$
  Note that material parameters $L_{gb}, \bar{k}_s$ must be measured in advance (Fig. 5.8 and
  Eq. 5.10).
  Output: $\tau$
  Func OPT($L, \mathbf{P}, L_{gb}, \bar{k}_s$):
      $S, \kappa, \mathbf{T} \leftarrow$ ProcessPattern $(\mathbf{P})$
      $\Delta s \leftarrow$ step size of deployment
      $\tau \leftarrow$ initialize an empty list
      $\hat{\mathbf{z}} \leftarrow$ director along vertical direction
      $s \leftarrow 0$
      while $s \leq S$ do
          $\mathbf{q}_C \leftarrow \mathbf{P}(s)$
          $\hat{\mathbf{x}} \leftarrow \mathbf{T}(s)$
          $\bar{\kappa} \leftarrow \kappa(s) L_{gb}$
          $\bar{l}_s \leftarrow (L - s)/L_{gb}$
          $(\bar{\mathbf{q}}_M^*, \mathbf{e}^*) \leftarrow \mathcal{F}(\bar{l}_s, \bar{\kappa}, \bar{k}_s)$
          $\mathbf{R} \leftarrow$ AxangtoRot $(\hat{\mathbf{e}}^*, \|\mathbf{e}^*\|)$
          $\mathbf{R}_t \leftarrow (\hat{\mathbf{x}}, \hat{\mathbf{z}} \times \hat{\mathbf{x}}, \hat{\mathbf{z}})$
          $\mathbf{q}_M^* \leftarrow \mathbf{q}_C + \mathbf{R}_t \bar{\mathbf{q}}_M^* L_{gb}$
          $\mathbf{R}^* \leftarrow \mathbf{R}_t \mathbf{R}$
          Append $(\mathbf{q}_M^*, \mathbf{R}^*)$ to $\tau$
          $s \leftarrow s + \Delta s$
      return $\tau$
```

substrates for multiple tasks so that we can look into the robustness of the proposed scheme against the material difference and friction.

First, we need to find the geometric and material properties of the manipulated DLO. The geometry of the manipulated rod, e.g., total length L and rod radius h, is trivial to measure. Measuring the material properties of the DLO is less clear. Overall, we need to develop a way to find the following material properties: gravito-bending length L_{gb} and normalized stretching stiffness $\bar{k}_s$.

Here, we presume the material is linearly elastic and incompressible. The incompressible material means the volume of the rod will not change when deformations happen. Therefore, Poisson's ratio is set as $\nu = 0.5$. In addition, bending stiffness is $k_b = \frac{E\pi h^4}{4}$ where



(a) Deformed configurations of a DLO under gravity in 2D plane.

(b) Relationship between the height of the loop h_f and the gravito-bending length L_{gb}.

Figure 5.8: Measurement of material properties L_{gb} of a DLO.

E is Young's modulus, and the expression for gravito-bending length L_{gb} and normalized stretching stiffness $\bar{k}_s$ is

$$\begin{aligned} L_{gb} &= \left(\frac{Eh^2}{8\rho g}\right)^{1/3}, \\ \bar{k}_s &= \frac{k_s L_{gb}^2}{k_b} = \frac{4L_{gb}^2}{h^2}. \end{aligned} \tag{5.10}$$

When observing Eq. 5.10, we find that the only parameter we must obtain is L_{gb}. It is

Table 5.1: Material and geometric properties of the DLOs used in the experiments.

DLO	Material & Geometric Parameters							
	Material	L_{gb} [cm]	h [mm]	L [m]	ν	μ_{fabric}	μ_{steel}	μ_{foam}
#1	Pink VPS	1.8	1.6	0.875	0.5	Low	Medium	High
#2	Green VPS	3.2	1.6	0.885	0.5	Low	Medium	High
#3	Rope	3.4	2.0	0.89	0.5	Medium	Low	High
#4	Pink VPS	2.86	3.2	0.84	0.5	Low	Medium	High
#5	Cable	8.01	1.8	0.87	0.5	Medium	Low	High

still unclear how to compute this as L_{gb} is relevant to Young's Modulus E and the density ρ of the rod, which is usually hard to measure. Here, we propose a simple method that is able to measure L_{gb} by observing the geometry of the rod. When we form a loop in a rod naturally using gravity in a 2D plane, we can observe the geometry of the rod becomes what is shown in Fig. 5.8a. Indeed, the height h_f of the loop has a linear relationship with L_{gb}. Therefore, we can obtain L_{gb} for different rods by simply measuring h_f. According to prior work [142] and our validation shown in Fig. 5.8b, $h_f = 0.9066L_{gb}$.

5.5.2 Experiment Setup

5.5.2.1 Materials and Robot Hardware

In this study, we conducted experiments involving five distinct types of DLOs. Among these, three are silicone-based rubber fabricated by vinyl polysiloxane (VPS); the fourth is a commercially available rope; and the fifth is a stiff USB cable. Note that we also validate the robustness of the deployment scheme against different substrates. The friction between the DLOs and substrates is also qualitatively measured. Comprehensive details regarding the parameters for each of these DLOs can be found in Tab. 5.1.

For our experiments, we used two Rethink Robotics' Sawyer manipulators as shown in Fig. 5.9. One arm is attached with a gripper for manipulating the rod. The other arm holds

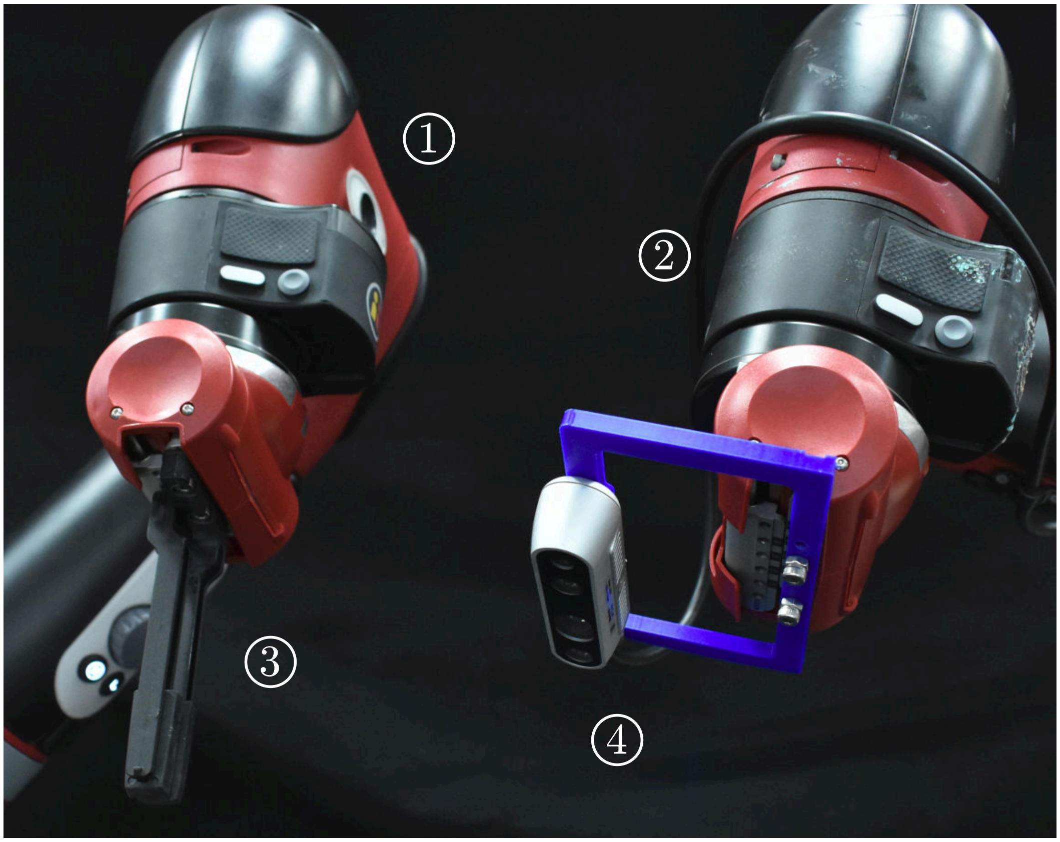


Figure 5.9: Experimental apparatus: Two robot manipulators, one for manipulation of the deploying rod (1) and the other for holding the camera for perception (2). A gripper (3) is used for grabbing the manipulated end of the rod. A camera (4) is used for extracting patterns from the drawn patterns and evaluating the deployment results.

an Intel RealSense D435 camera which is used to scan drawn patterns as well as obtain a top-down view of the deployment result for evaluations. A workstation with an AMD Ryzen 7 3700X CPU and an NVIDIA RTX 2070 SUPER GPU was used for all experiments.

5.5.2.2 Experiment Tasks

We implement our proposed deployment scheme across four distinct tasks. First, we deploy a rod along some canonical cases obtainable through analytical expressions such as a line, circle, and sine curve. The rod is deployed using the robotic arm with the gripper. Once the deployment is finished, the other arm with the camera moves to scan the deployment result.

The second task involves deploying patterns drawn on paper. Users draw patterns, subsequently scanned by the camera to obtain ordered discretized pattern coordinates. The robot then manipulates the rod to replicate the drawn pattern. This study showcases deployment results for the letters “U”, “C”, “L”, and “A”, with the precise shapes detailed in Fig. 5.10(a). The third task is to validate the deployment scheme’s application in cable placement, a vital aspect of cable management. The scheme’s efficacy is demonstrated by placing cables along constrained paths with the help of pre-installed fixtures on the substrate. Lastly, the deployment scheme’s application for tying knots is verified. Both robotic arms are equipped with grippers for this task.

Patterns are evaluated using intuitive and optimal control methods for the first two tasks. Additionally, three different rods (DLOs #1, #2, and #3) are deployed on substrates of various materials (fabric, steel, foam) to assess the method’s robustness against material disparities and friction. In the third task, both algorithms employ DLO #5 (USB cable) for cable placement. Finally, DLOs #2 and #4 are used to tie distinct knots for the fourth task. Each experimental case is subjected to ten trials for each control method, culminating in a total of 1340 experimental trials.

5.5.3 Metrics

We now formulate the metrics used to evaluate the performance of the deployment scheme. When deploying a pattern $\mathbf{P}$, we need to assess the accuracy of the deployment result. We first discretize the pattern $\mathbf{P}$ into N points and denote the i-th point of the prescribed pattern as $\mathbf{P}^i$. The actual deployment pattern obtained from perception is denoted as $\mathbf{P}_{\text{exp}}$. With this discretization scheme, we compute the average error e_{mean} and standard deviation σ as

$$\begin{aligned} e_{\text{mean}} &= \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{P}^i_{\text{exp}} - \mathbf{P}^i\|, \\ \sigma &= \sqrt{\frac{\sum_{i=1}^{N}(\|\mathbf{P}^i_{\text{exp}} - \mathbf{P}^i\| - e_{\text{mean}})}{N}}, \end{aligned} \tag{5.11}$$

Table 5.2: Evaluation of deployment accuracy for various patterns, DLOs, and substrates.

DLO	SUB	Control Scheme	Pattern Type and Accuracy $e_{\text{mean}} \pm \sigma$ [cm] (Eq. 5.11)						
			Line	Circle	Sine curve	Letter "U"	Letter "C"	Letter "L"	Letter "A"
#1	Fabric	INT	0.40 ± 0.22	0.61 ± 0.36	1.66 ± 0.74	1.39 ± 0.63	2.21 ± 0.92	1.00 ± 0.59	4.81 ± 2.27
		OPT	0.14 ± 0.09	0.15 ± 0.07	0.27 ± 0.10	0.22 ± 0.07	0.22 ± 0.10	0.35 ± 0.18	0.47 ± 0.23
	Steel	INT	1.42 ± 0.66	2.34 ± 1.24	2.69 ± 1.69	3.59 ± 2.39	3.67 ± 1.93	0.87 ± 0.55	3.64 ± 2.09
		OPT	0.22 ± 0.12	0.22 ± 0.08	0.27 ± 0.10	0.24 ± 0.13	0.27 ± 0.09	0.42 ± 0.16	0.58 ± 0.37
	Foam	INT	1.03 ± 0.21	1.23 ± 0.45	2.84 ± 1.52	3.33 ± 1.93	3.89 ± 1.29	1.13 ± 0.74	4.09 ± 2.19
		OPT	0.25 ± 0.15	0.18 ± 0.06	0.29 ± 0.16	0.24 ± 0.15	0.41 ± 0.20	0.35 ± 0.12	0.54 ± 0.24
#2	Fabric	INT	0.52 ± 0.13	1.64 ± 0.95	1.60 ± 0.83	3.74 ± 2.89	4.58 ± 1.15	1.74 ± 1.11	4.95 ± 2.63
		OPT	0.13 ± 0.07	0.16 ± 0.07	0.20 ± 0.09	0.17 ± 0.11	0.19 ± 0.22	0.29 ± 0.11	0.32 ± 0.18
	Steel	INT	1.72 ± 0.63	2.52 ± 1.02	3.30 ± 2.08	4.78 ± 4.15	6.66 ± 2.53	2.14 ± 1.26	5.23 ± 3.38
		OPT	0.17 ± 0.08	0.22 ± 0.09	0.54 ± 0.20	0.21 ± 0.09	0.74 ± 0.31	0.66 ± 0.24	0.36 ± 0.17
	Foam	INT	1.38 ± 0.60	2.24 ± 0.97	4.17 ± 2.57	5.42 ± 4.47	6.14 ± 3.08	1.70 ± 1.32	5.09 ± 3.39
		OPT	0.27 ± 0.13	0.20 ± 0.09	0.37 ± 0.14	0.17 ± 0.08	0.39 ± 0.18	0.37 ± 0.15	0.43 ± 0.19
#3	Fabric	INT	1.56 ± 0.81	1.13 ± 0.53	5.09 ± 1.35	4.22 ± 3.10	3.36 ± 1.58	2.37 ± 1.56	4.59 ± 2.54
		OPT	0.49 ± 0.28	0.29 ± 0.15	0.47 ± 0.23	0.36 ± 0.18	0.35 ± 0.19	0.50 ± 0.24	0.56 ± 0.29
	Steel	INT	4.53 ± 2.80	1.85 ± 0.45	4.43 ± 2.82	4.53 ± 2.80	3.35 ± 1.55	2.57 ± 1.62	4.30 ± 1.73
		OPT	0.47 ± 0.20	0.29 ± 0.13	0.46 ± 0.20	0.47 ± 0.20	0.56 ± 0.20	0.51 ± 0.24	0.81 ± 0.30
	Foam	INT	2.00 ± 0.88	1.94 ± 0.84	3.80 ± 1.96	3.67 ± 2.46	6.03 ± 3.11	3.32 ± 1.80	4.47 ± 2.50
		OPT	0.78 ± 0.34	0.27 ± 0.15	0.46 ± 0.20	0.32 ± 0.16	0.56 ± 0.26	0.33 ± 0.14	0.52 ± 0.20

for both the intuitive and optimal control results.

The accuracy evaluation is not applicable for the two application tasks: cable placement and knot tying as they are high-level tasks. Therefore, we simply use the success rate of those application tasks to evaluate the performance of the deployment scheme. In addition to accuracy, we also report a detailed comparison of runtimes and errors between the numerical and NN-based solvers. Details of the relevant results and analysis are discussed in the next section.

5.5.4 Results and Analysis

5.5.4.1 Accuracy

All experimental results can be seen in Tab. 5.2. To compute the error metrics in Eq. 5.11, we used a discretization of $N = 50$. From all results, we can observe a noticeable improvement in our optimal control method over the intuitive method for various geometrical, material, and environmental parameters.

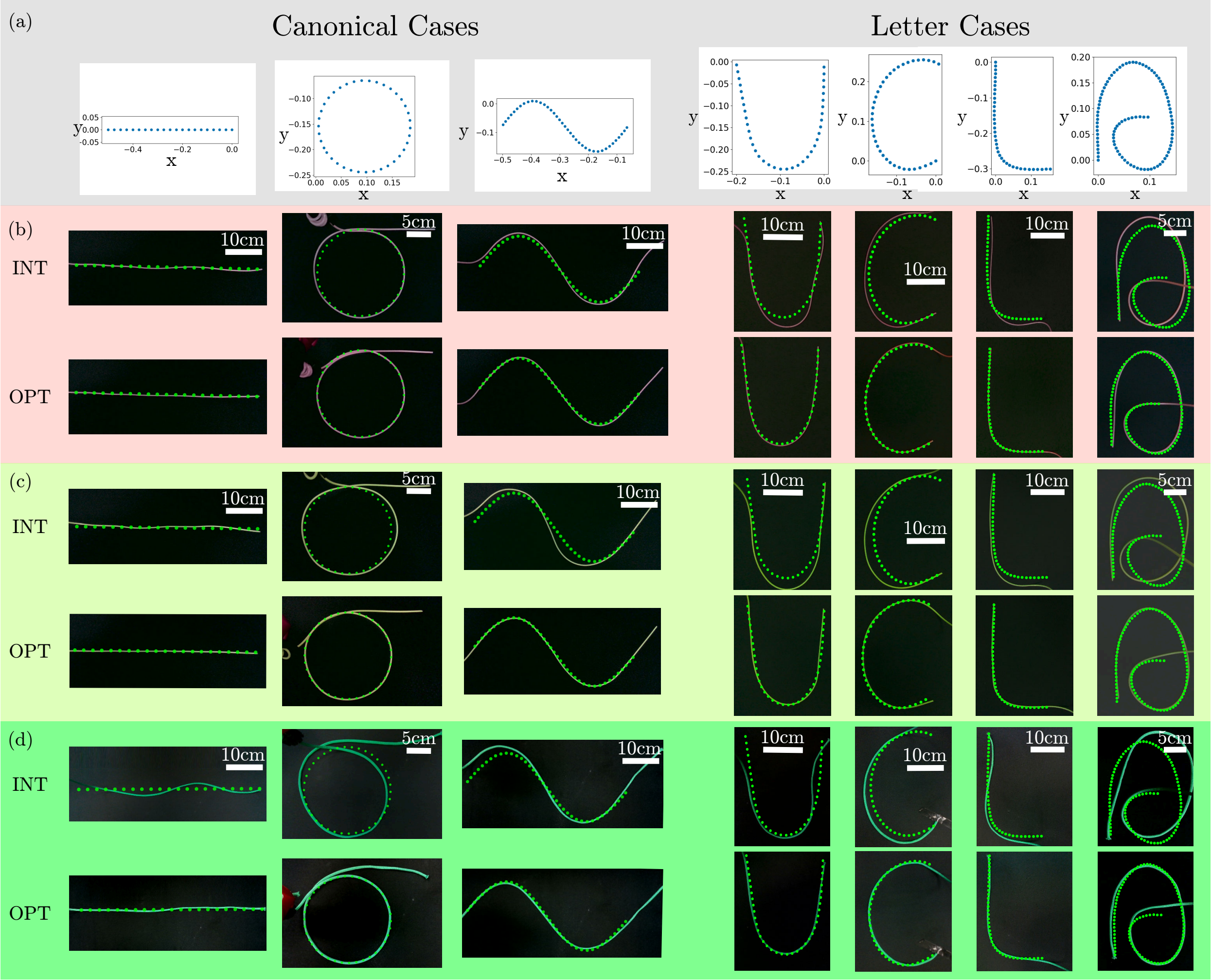


Figure 5.10: Experiment results of deployment along various patterns. (a) All used prescribed patterns are discretized and plotted. Deployment results for (b) DLO #1 (pink VPS), (c) DLO #2 (green VPS), and (3) DLO #3 (rope) are shown for each prescribed pattern. Results for the intuitive control method and optimal control method are shown for each rod.

To better visualize our method's generality, we visually depict deployment outcomes across different DLOs on the fabric surface in Fig. 5.10. In addition, a comparative visual representation of deployment results for a single DLO (#2) on varying substrates is shown in Fig. 5.11. Readers seeking comprehensive visual comparisons of all deployment outcomes can refer to the supplementary video for detailed insights (see Footnote 1).

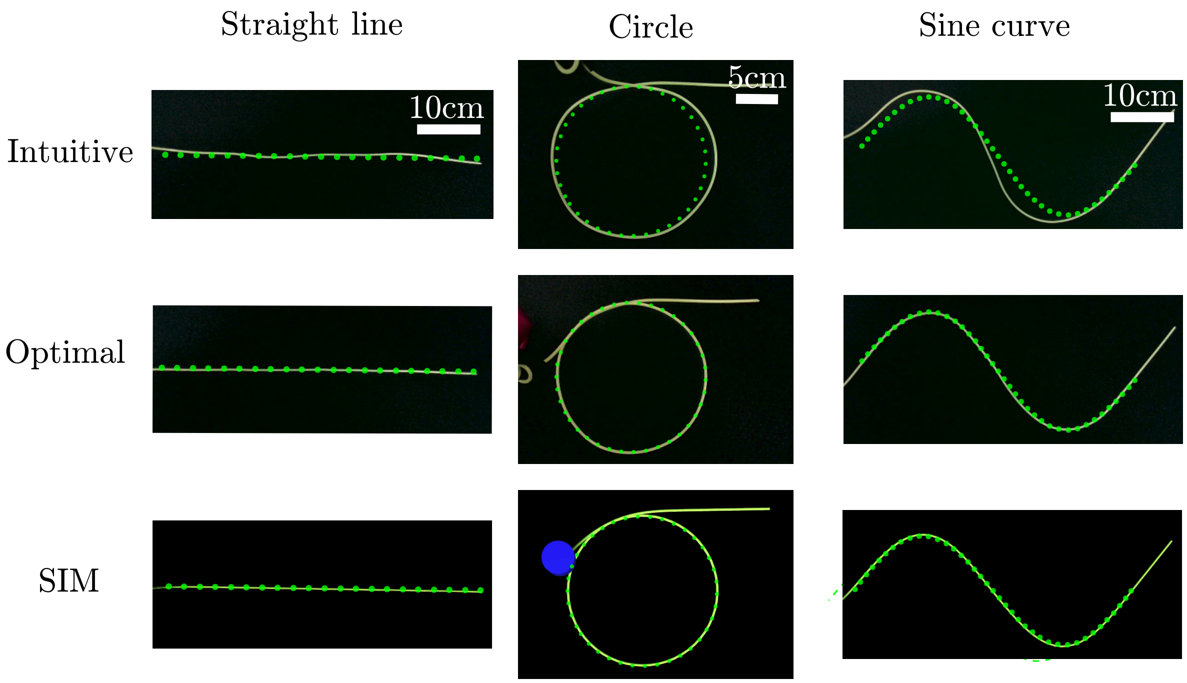


Figure 5.11: Experiment results of deployment with DLO #2 (green VPS) along various patterns on different substrates.

Among the seven deployed patterns, the first three (straight line, circle, and sine curve) are canonical cases, i.e., their shapes have explicit analytical expressions. Note that when deploying the circle and sine curve patterns, a small "remainder" section is first deployed. This is necessary as the circle and sine curve patterns have a non-zero curvature at the start of their pattern. We compensate for this by deploying a remainder part whose curvature gradually evolves from a straight line with 0 curvature to the prescribed curvature of the pattern's first point. The remainder can improve the deployment task's accuracy as the deployed pattern will require slight friction based on Eq. 5.2.

We have omitted the designed remainder for the four remaining patterns denoted by the letters "U", "C", "L", and "A" for better visualization. Among these, patterns "U", "L", and "A" exhibit a relatively low κ'' value during the beginning stage of the deployment, resulting in the deployment accuracy being minimally affected by surface friction.

Conversely, the “C” pattern demonstrates a comparatively higher κ'' value initially, leading to a possible noticeable mismatch between the deployed DLO and the intended pattern in the beginning. The impact of friction becomes more pronounced during the rope deployment corresponding to DLO #3 since the rope has higher bending stiffness k_b and experiences lower friction with the substrate. Fixing the free end is essential to precisely replicate the “C” pattern with the rope as shown in Fig. 5.10 (d). Despite this limitation, our optimized deployment strategy consistently outperforms the intuitive approach.

5.5.4.2 Computational Efficiency

Next, we also evaluated the computational efficiency of our neural controller. Tab. 5.3 compares time costs between the neural network solver (NN-solver) and the numeric solver based on simulations. When calculating a single optimal robot grasp for a given parameter tuple ($\bar{l}_s$, $\bar{\kappa}$, $\bar{k}_s$), the numeric solver takes approximately 10 to 20 seconds, while our NN-solver takes roughly 0.4 seconds.

The difference of time costs becomes more significant when generating a series of optimal robot grasps for a discretized pattern. Note that a discretized pattern typically consists of 100 to 200 nodes and that the numeric solver needs to compute the robot trajectory in sequence as the optimal grasp for the previous step is needed as the seed for computing the next optimal grasp. Therefore, the time costs quickly accumulate for the numeric solver, which substantially elongates the overall computation time. In contrast, the NN-solver leverages vectorization to solve multiple robot grasps simultaneously, resulting in a speed advantage of several orders of magnitude compared to the numeric solver when generating optimal deployment trajectories.

Table 5.3: Evaluation of computation times of various patterns for the numerical and NN-solvers with error metrics.

DLO	Solver Times [s] & MAEs	Patterns with Number of Nodes						
		Line 101 nodes	Circle 156 nodes	Sine curve 138 nodes	Letter “U” 190 nodes	Letter “C” 190 nodes	Letter “L” 190 nodes	Letter “A” 194 nodes
#1	Numeric-Solver	1572.68	2036.11	2897.17	3954.12	4015.24	4777.30	4666.55
	NN-Solver	0.402	0.393	0.395	0.431	0.431	0.400	0.417
	Position Error [m]	0.0008	0.0007	0.0009	0.0008	0.0007	0.0008	0.0008
	Orientation Error	0.0012	0.0010	0.0032	0.0025	0.0020	0.0020	0.0021
#2	Numeric-Solver	776.56	1213.14	1769.66	2286.66	2226.73	2720.08	2933.90
	NN-Solver	0.397	0.391	0.396	0.419	0.408	0.404	0.406
	Position Error [m]	0.0016	0.0016	0.0019	0.0018	0.0016	0.0020	0.0017
	Orientation Error	0.0012	0.0078	0.0050	0.0042	0.0020	0.0058	0.0030
#3	Numeric-Solver	666.01	1041.71	1561.12	1984.63	1972.39	2405.71	2639.44
	NN-Solver	0.400	0.407	0.395	0.407	0.420	0.405	0.411
	Position Error [m]	0.0016	0.0017	0.0020	0.0020	0.0016	0.0021	0.0018
	Orientation Error	0.0010	0.0087	0.0052	0.0055	0.0023	0.0054	0.0032

5.5.4.3 Precision of the Neural Controller

Finally, Tab. 5.3 also presents the precision of the NN-solver. The solutions from the numeric solver serve as the ground truth. Mean Absolute Error (MAE) is employed to evaluate the optimal trajectories the NN-solver generates against the ground truth. Remarkably, the MAE consistently remains below 0.003m for position error and 0.009 for differences in rotation quaternions. Importantly, it’s noteworthy that none of the solved trajectories in this analysis were part of the training dataset. Thus, we can confidently assert that our NN-solver exhibits robustness, efficiency, and accuracy, rendering it well-suited for real-time control applications.

5.5.5 Application #1: Cable Placement

In this section, we showcase the application of the deployment scheme for cable placement. The importance of cable management has surged, particularly in engineering contexts involving tasks like wire harnessing, infrastructure development, and office organization [6, 143]. Given cables’ inherent high bending stiffness, shaping them to specific forms can be chal-

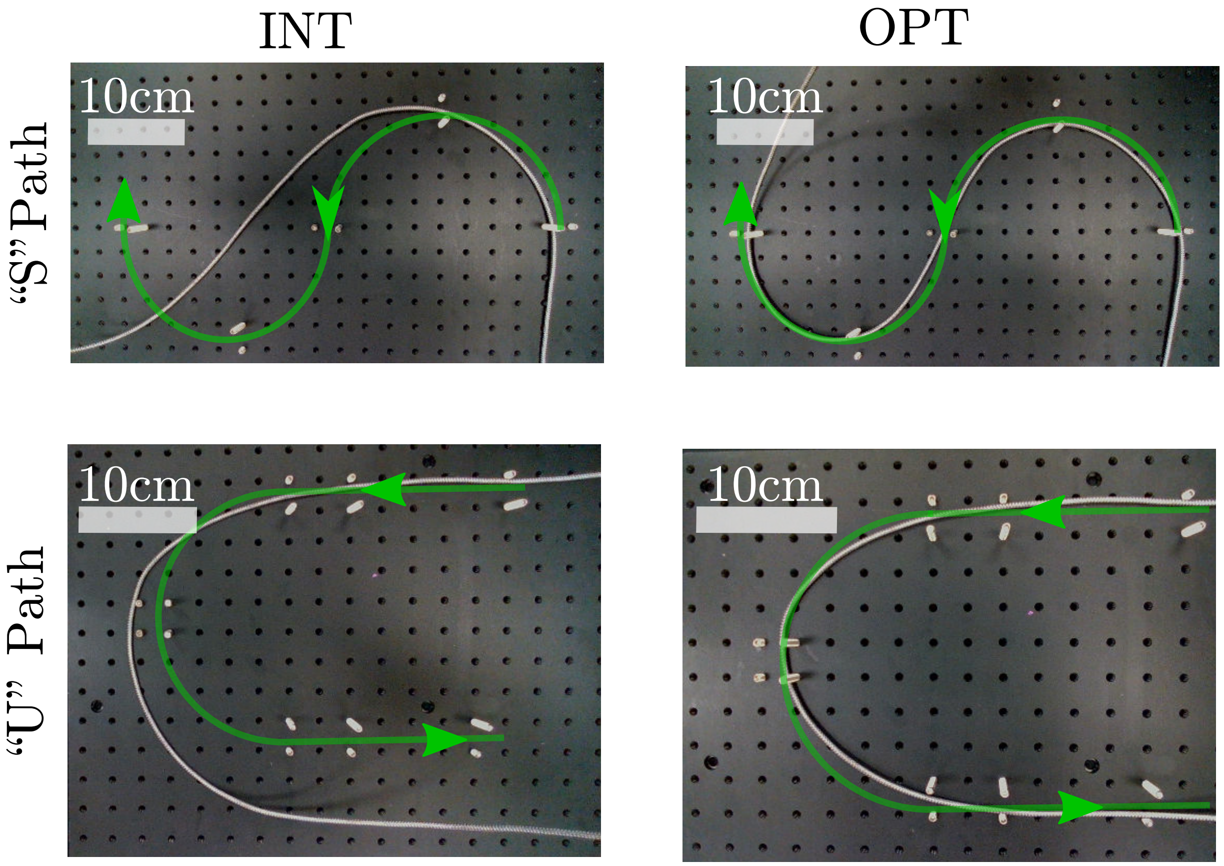


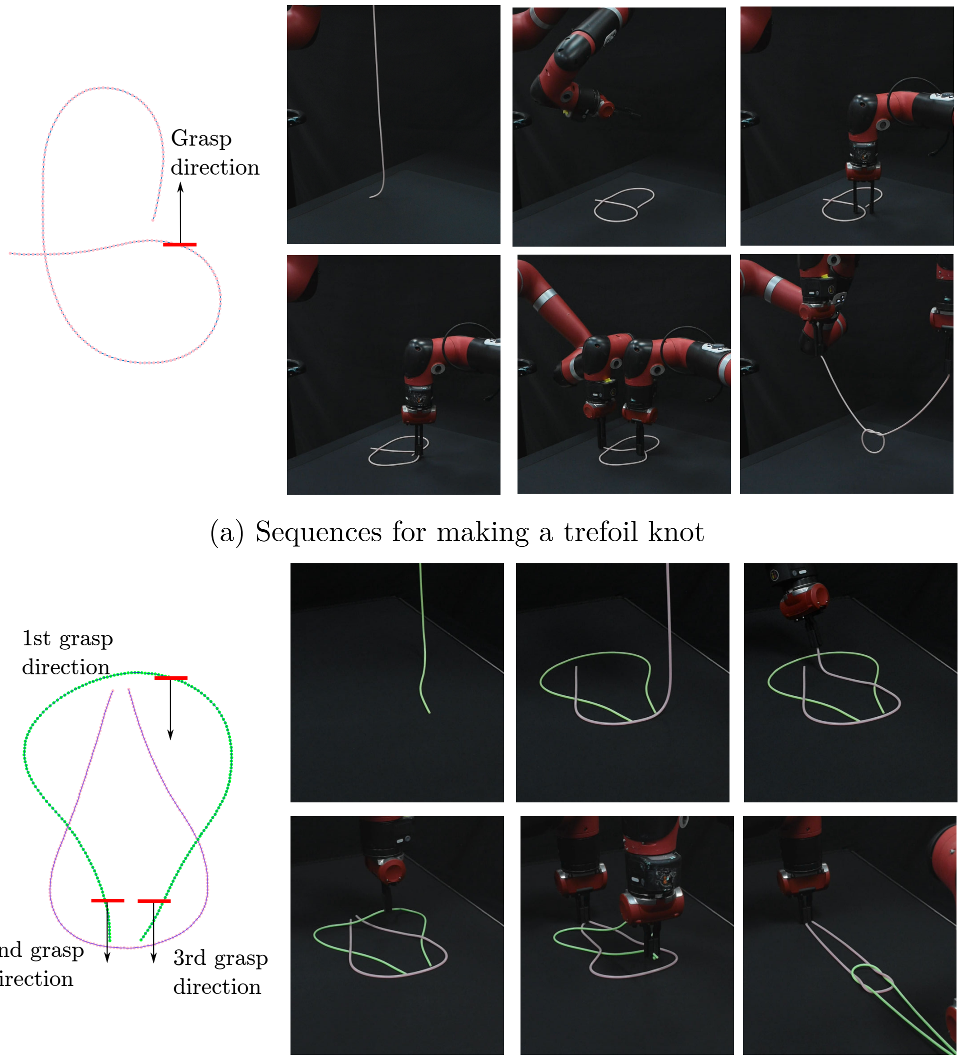
Figure 5.12: A demonstration of cable placement along different prescribed patterns with both intuitive and optimal control schemes.

lenging, often necessitating external fixtures to maintain the desired configuration. When humans perform cable management manually, meticulous placement along the designated pattern is essential, coupled with the use of fixtures to secure the cable in place. However, a robotic system can autonomously execute cable placement with our designed optimal deployment strategy.

In our experimental setup, we preinstalled external fixtures into the stainless steel breadboard to delineate the intended patterns. These fixtures also counteract the cable's rigid nature, preventing it from reverting to its original shape. The deployment results can be visualized in Fig. 5.12. Compared to the failure placement results with the intuitive scheme, our optimal deployment scheme can place the cable along the prescribed pattern "U" and

"S" on the substrate. We did 10 experimental trials for each deployment task illustrated in Fig. 5.12. Notably, the optimal deployment approach achieved an impressive 90% (9/10) success rate for both patterns, whereas the intuitive method failed in all trials (0/10) as shown in Tab. 5.4.

5.5.6 Application #2: Knot Tying



(a) Sequences for making a trefoil knot

(b) Sequences for making a reef knot

Figure 5.13: A demonstration of two knot-tying cases using the DLO deployment scheme. The sequences are shown for tying (a) a trefoil knot and (b) a reef knot.

Since our optimal deployment scheme can control the shape of various DLOs with excellent accuracy, we can use this scheme to tie knots. First, the manipulated rod is deployed along a predesigned pattern on the substrate. Users can draw the predesigned pattern so that only a few extra manipulations are required. Then, the camera will scan the drawn pattern and send it as input to our designed scheme. The deployed pattern is designed in a way that only a few simple pick-and-place operations on certain knot segments is required to complete the tying sequence. Since the prescribed pattern's shape is known in advance, we can let the robot execute the pick-and-place procedure without perception feedback. So long as the initial deployment is accurate and repeatable, the subsequent pick-and-place procedure should succeed most of the time.

We showcase two knot-tying sequences in Fig. 5.13. The top row showcases a trefoil knot, one of the most fundamental knots in engineering [144]. For this knot, we used DLO #4. Another case is a reef knot, a prevalent knot widely used in for various applications including shoelaces, packaging, sewing, etc. When tying the reef knot, we used DLOs #2 and #4. Although these two DLOs have totally different material properties, our generalizable neural controller allows two robots to deploy both DLOs accurately along the designed patterns. With the help of the deployed patterns, reef knots can be tied with simple pick-and-place procedures. Such knot-tying cases strongly support the potential of our deployment scheme in various engineering applications.

We show the results of the knot-tying tasks in Tab. 5.4. The successful rate of knot-tying is remarkable. We achieved a success rate of 90 % (9 successful trials out of 10) for tying a trefoil knot and a success rate of 70 % (7 successful trials out of 10) with the optimal control method. Based on our observations, all the failure cases were caused by the rod slipping out of the gripper. In contrast, the intuitive control method achieves a success rate of 0% for both cases as the initially deployed pattern does not match the intended pattern.

Therefore, the intuitive control method would require some visual feedback to choose the pick-and-place motion adaptively for the trefoil knot case. As for the reef knot case, due to

Table 5.4: Real-world application experiment results.

Experiment Type	Scheme	Success rate
“S” Cable Placement	INT	0/10
	OPT	9/10
“U” Cable Placement	INT	0/10
	OPT	9/10
Trefoil Knot	INT	0/10
	OPT	9/10
Reef Knot	INT	0/10
	OPT	7/10

the deployment results are totally wrong, even though the visual feedback is applied, it is still hard to achieve a complete reef knot with intuitive method.

Therein, we can see the potential of the deployment scheme in high-level robotic tasks like knot tying. In future work, the optimal deployment scheme will be incorporated with the perception system to automatically tie any prescribed knots with the robotics system.

5.6 Summary and Outlook

In this chapter, we have developed a novel deployment scheme that allows for robust and accurate control of the shape of DLOs using a single manipulator. Our framework integrates techniques from different realms, including physical simulation, machine learning, and scaling analysis, and has been demonstrated to be highly effective in robotic experiments. Our results highlight the advantages of incorporating physics into robotic manipulation schemes and showcase impressive performance on complex tasks such as writing letters with elastic rods, cable placement, and tying knots.

Looking to the future, we plan to leverage the precision and efficiency of our deployment scheme to tackle some high-level robotic tasks systematically, for example, robotic knot tying. While exact shape control is not strictly required during such manipulations, our deployment scheme offers sufficient accuracy and efficiency to design the configurations of the middle states of a manipulated DLO, which is essential for robots to tie complex knots successfully. We also aim to explore the use of generalized problem formulations and data-driven control schemes, such as reinforcement learning, to develop more flexible and adaptive solutions to the challenges of robotic manipulation. By continuing to push the boundaries of robotic manipulation, we hope to advance the state-of-the-art in this field and enable new and exciting applications of robotic technology.

CHAPTER 6

Learning Buckling of a Rod with Helical Centerline with Robotics

This chapter explores the complexities of experimental mechanics, focusing on the challenging task of studying the stability of slender elastic rods with helical centerlines. It offers valuable insights and contributions to this area of research.

First, let us revisit the historical work of Kirchhoff in 1859, who made significant observations about the equilibrium configurations of elastic rods. He identified circular helices as potential equilibria for an inextensible, unshearable, isotropic, and uniform elastic rod [145, 146]. A natural question arises: When an elastic rod with a helical centerline is stable? Although the rod configurations with helical centerlines can be described using only three parameters, it is challenging to study its stability experimentally since it requires manipulation of both the position and orientation at one end of the rod, which is not possible using traditional experimental methods that only actuate a limited number of degrees of freedom. In addition, many possible equilibriums of a helical rod exist, making the exploration of all the equilibriums becomes an impossible mission with human labor.

In recent times, the growth of automation in scientific investigations has drawn more and more researcher's interest. In Ref. [147], a "robotic scientist" is designed to explore yeast functions. It highlights the transformative potential of automation and robotics in advancing scientific understanding. The recent advancements in the geometric characterization of stability for helical rods told us only three parameters should be controlled when manipulating a helical rod. However, the geometric properties of the helical rod still require delicate manipulations on both position and orientation. Therein, we developed an automated stability testing platform empowered by robotics, enabling the systematic exploration of stable helical configurations. Additionally, a vision system is integrated to detect the onset of instabilities during experimental trials.

Experimental results obtained through the automated testing system are rigorously validated by comparing them to numerical simulations. This comparison demonstrates a commendable level of agreement, reinforcing the effectiveness of the proposed automation-driven approach in studying helical rod stability. The broader implication of this chapter lies in its pioneering role in the realm of experimental mechanics. It serves as a foundational step toward automating complex experimental setups, ultimately facilitating the comprehensive study of deformable objects' mechanics.

The underlying motivation is in § 6.1. The modeling and physical guidance for automated experiments is detailed in § 6.2. Then, the automated experimental system is given in § 6.3. Finally, we give the conclusions in § 6.5. The content of this chapter is from Ref. [16].

6.1 Motivation

Elastic rods possess the fascinating ability to adopt an array of equilibrium configurations, even when their boundary conditions remain fixed. Discerning the stability of these configurations distinguishing between stability and instability is a fundamental challenge in mechanics. However, experimental assessments of these rod configurations entail repetitive testing and specialized setups capable of manipulating up to six degrees of freedom at the rod's ends. Prior work has primarily been constrained by traditional experimental platforms that controlled only a limited number of degrees of freedom (often just one or two) at the rod's ends (e.g., Refs. [148, 149]). For example, Lazarus et al.[148] employed a platform with two degrees of freedom to observe rod behavior during contortion, while Thompson et al. [149] used a platform with two controlled degrees of freedom to investigate the post-buckling behavior of flexible rods with torsion. The restricted control over degrees of freedom in these platforms often limited the range of rod configurations that experiments could explore.

This chapter addresses these limitations by developing an automated testing methodology that employs robotic systems to determine the stability of elastic rods. The focus is mainly on configurations where the rod's centerline forms a helix. This choice is motivated by recent research that characterizes the set of stable helical rod configurations [150]. However, conducting experimental analyses of rod stability in previous studies proved challenging due to the repetitive testing procedures and the complexity involved in simultaneously controlling both the position and orientation of the rod's end. The solution to these challenges lies in applying robotic automation to conduct comprehensive stability experiments.

Fig. 6.1 showcases a practical experiment conducted using our automated testing system. In this experiment, an elastic rod, identifiable by blue markers along its length, undergoes deformation while adopting a helical centerline. A collaborative robot manipulates one end of the rod, as depicted in Fig.6.1(a-e). Notably, between Fig.6.1(e) and (f), an instability emerges, leading to a transition to another configuration that lacks a helical centerline. The experiment concludes in Fig.6.1(g), offering valuable insights into the mechanics of elastic rods. Simulated rod configurations with identical boundary conditions, based on the Discrete Elastic Rod formulation [14, 151], are presented alongside the experimental images for comparison.

Manipulating a deformable object, as illustrated in Fig. 6.1, presents unique challenges that diverge from traditional rigid object manipulation. Deformable object manipulation has garnered significant attention in robotics [152], often focusing on tasks like deforming an object, such as an elastic rod, from an initial configuration to a specific target configuration (e.g., Refs. [153, 154]). These tasks frequently entail additional constraints, such as limiting the object's deformation, avoiding unstable configurations, and preventing self-collisions(e.g., Refs. [155, 156, 157]). Elastic rods find applications across various engineering domains, such as robot-assisted hot wire cutting [158], magnetically guided rods for medical procedures [159], and the development of concentric tube robots [160]. While the methods discussed in this chapter draw inspiration from existing robotic manipulation research, our approach distinguishes itself by intentionally inducing instabilities to gain insights into the mechanics of deformable objects.

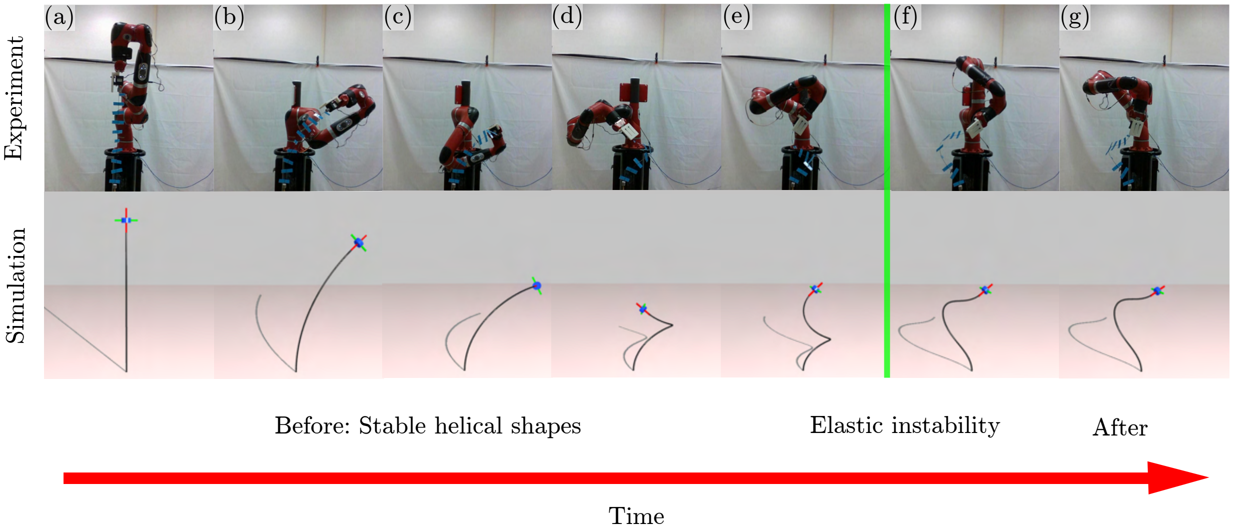


Figure 6.1: Snapshots of an elastic rod manipulated by a robot from stable helical configurations to non-helical configurations after an instability. (a-e) Stable helical configurations; (f-g) non-helical configurations after an instability; elastic instability happens at the vertical green line.

Our automated stability testing method includes not only the robotic system but also a vision system crucial for detecting instances of rod instability. Visual tracking of deformable objects has been a subject of prior research in robotics [161], with notable applications in fields like robotic surgery [162]. However, the primary focus of previous studies has been to estimate the current configuration of deformable objects, typically from images or sensor data, to aid in tasks like surgical navigation. In contrast, our approach combines visual tracking with instability detection. We begin by estimating the rod's current configuration using imagery captured by a camera. Subsequently, we calculate the error between this detected configuration and the expected or desired configuration. A sudden and significant increase in this error serves as an automated indicator of instability.

The elastic rods under investigation in this paper belong to a broader category of flexible slender structures commonly encountered in our daily lives. Understanding the equilibrium configurations of such structures holds practical relevance in various fields, including materials science (polymers), biology (bacterial fibers, DNA), and botany (plant growth) [163, 164, 32, 165]. Beyond equilibrium, distinguishing between stable and unstable configurations has been a central topic in mechanics [67, 166]. Configurations characterized by a helical centerline have received particular attention, with roots tracing back to Kirchhoff's seminal work in 1859. Kirchhoff demonstrated that, under specific boundary conditions, an initially straight, inextensible, unshearable, isotropic, and uniform rod can adopt a helical centerline. Subsequent research established that the stability of these helical configurations can be characterized by just three parameters: the centerline's curvature, torsion, and the applied twisting moment [150]. Moreover, within this three-dimensional parameter space, the set of stable helical configurations exhibits a star-convex structure.

Our manipulation scheme, employed to induce instabilities in elastic rods, draws inspiration from the geometric properties of this set of stable helical configurations. To validate the results obtained through our automated testing method, we compare the stability measurements collected by our robotic system with simulations of helical elastic rods. These simulations rely on the Discrete Elastic Rod formulation [14, 151]. The mechanical insights garnered from these robotic experiments have broader implications and could potentially be used to refine existing mechanical models. For instance, recent research [167] has explored the application of machine learning to construct advanced constitutive models for materials. Integrating machine learning with the robotic system described in this paper could enable the use of mechanical experiment data to enhance our understanding of complex structures and inform finite element analyses and simulations.

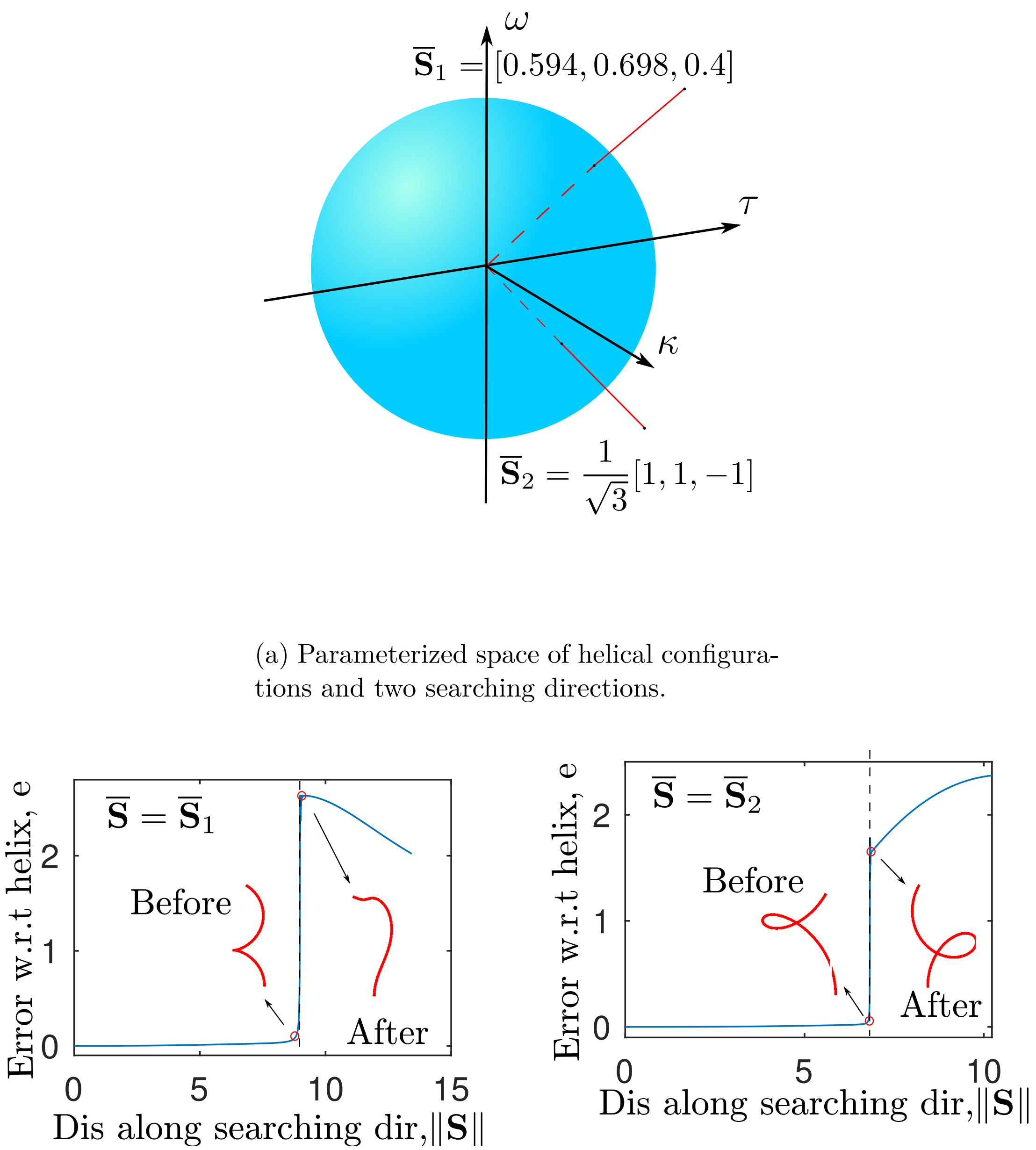
6.2 Exploration and Simulation of Helical Rods

This section describes a parameterized space for helical rod configurations and introduces a scheme to navigate this space, focusing on the subset of stable configurations. We begin by outlining the coordinates within this parameter space, encompassing the rod's curvature, torsion, and twisting moment. Following that, we delve into the specifics of our numerical simulation, a crucial tool for predicting the stability of helical rod configurations. Lastly, we unveil a manipulation strategy designed to provoke instabilities in helical rods. This strategy will be later realized in the robotic system in § 6.3.

6.2.1 Parameter space of helical rod configurations

In this section, we provide essential background information based on the findings from Ref. [150]. This research revealed a parameterization method for all stable helical configurations of an initially straight, inextensible, unshearable, isotropic, and uniform elastic rod with a length of L. These configurations can be defined by three key parameters: the centerline's curvature $\kappa \geq 0$, torsion τ, and the twisting moment ω applied to the rod. Understanding these parameters is fundamental to determining the stability of helical rod configurations. Consequently, we can represent the stability of all helical configurations within a three-dimensional parameter space, utilizing axes κ, τ, and ω.

Each point within this parameter space corresponds to an equilibrium state of a helical rod configuration. However, not all points in this space denote stable helical configurations, which minimize elastic potential energy. This potential energy comprises two components: bending energy and twisting energy. Notably, potential energy arising from axial stretching and external forces like gravity are neglected. The balance between bending and twisting energy is influenced by the stiffness ratio $c = k_t/k_b = 1/(1 + \nu)$, where k_b stands for bending stiffness, k_t represents twisting stiffness, and ν denotes Poisson's ratio. Interestingly, the stiffness ratio c doesn't impact stability within the κ-τ-ω parameter space.



(a) Parameterized space of helical configurations and two searching directions.

(b) Error between the simulated rod and the expected helical shapes vs. distance along the searching direction $\|\mathbf{S}\|$ when $\overline{\mathbf{S}} = [0.594, 0.698, 0.4]$.

(c) Error between the simulated rod and the expected helical shapes vs. distance along the searching direction $\|\mathbf{S}\|$ when $\overline{\mathbf{S}} = \frac{1}{\sqrt{3}}[1, 1, -1]$.

Figure 6.2: Scheme to incite an instability in a helical rod.

A geometric property concerning the subset of stable helical configurations, as established in Ref. [150], is that within the κ-τ-ω parameter space, the set of points corresponding to stable configurations exhibits star-convexity. This property implies that each ray originating from the origin in the κ-τ-ω parameter space intersects the boundary that separates stable and unstable helical configurations precisely once. This unique geometric characteristic forms the basis for our manipulation scheme to provoke instabilities in our automated testing process, which will be explained in subsequent sections.

6.2.2 Manipulation scheme to explore stable configurations

We now propose a scheme for manipulating a helical rod to induce instabilities, enabling us to explore points in the κ-τ-ω parameter space corresponding to stable configurations. As previously explained, it has been established that every ray originating from the origin within the κ-τ-ω parameter space intersects the boundary distinguishing stable from unstable helices exactly once. Moving along such a ray within the parameter space corresponds to continuously altering the shape of the helical rod. Therefore, our manipulation scheme commences with a straight, untwisted rod (corresponding to the origin in the κ-τ-ω parameter space) and then proceeds along a search direction denoted as $\mathbf{S}$ within this space. Here, $\|\mathbf{S}\|$ represents the distance from the origin along the search direction, and $\overline{\mathbf{S}}$ is a unit vector in this direction. Starting at the origin, we move along $\mathbf{S}$ within the κ-τ-ω space until we encounter an instability.

Two examples of this procedure are shown in Fig. 6.2. Fig. 6.2a shows the κ-τ-ω parameter space along with a half-sphere of radius 1. Each point on this half-sphere corresponds to a different search direction $\overline{\mathbf{S}}$, two of which are shown in the figure. Moving along either search direction corresponds to manipulating a helical rod and continuously changing its curvature, torsion, and twisting moment. This manipulation process was completed using the numerical simulation framework described in the previous section, and the results for the search directions $\overline{\mathbf{S}}_1$ and $\overline{\mathbf{S}}_2$ are shown in Figs. 6.2b and 6.2c. To measure the error between the simulated rod and the predicted helical shape, we calculate the average deviation of the rod configuration from the predicted helix, normalized by the helix's radius. The expression for this error e is given by

$$e = \frac{1}{n+1} \sum_{i=0}^{n} \frac{\kappa^2 + \tau^2}{\kappa} \|\mathbf{q}_i - \mathbf{q}_i^{\text{helix}}\|, \tag{6.1}$$

where $n+1$ is the number of discrete nodes representing the centerline of the rod, $\mathbf{q}_i$ is the position of the node, and $\mathbf{q}_i^{\text{helix}}$ is the position of the node when the rod is assumed to be helical. In Figs. 6.2band 6.2c, you can observe that this error initially remains small and then undergoes a sharp increase at a critical distance along the search direction. This sudden increase corresponds to an instability event in which the rod transitions to another configuration that is not helical. This process was systematically repeated for 58,352 search directions using numerical simulation, and the resulting points $\mathbf{S}$ at which instabilities occurred were meticulously recorded. These points were then utilized to construct the surface depicted in Fig. 6.6, which represents the predicted boundary distinguishing stable and unstable configurations within the κ-τ-ω parameter space.

6.3 Robotic System

In this section, we describe a robotic system that implements the manipulation scheme from the previous section within the κ-τ-ω parameter space. This system allows for automated stability testing and unsupervised collection of relevant data. We first provide an overview of the robotic system. We then discuss how boundary conditions for the rod corresponding to points in the κ-τ-ω parameter space are implemented. Finally, we describe the vision system used to detect instabilities, and we discuss the effects of disturbances in the robotic system.

6.3.1 Overview of the robotic system

A flowchart of the robotic manipulation scheme is shown in Fig. 6.3. The robotic system is composed of three parts: a collaborative robot, an externally mounted motor, and a camera. The collaborative robot imposes the prescribed position and tangent on one end of the manipulated rod while the mounted motor imposes the required torsion by rotating one end of the rod. In other words, the collaborative robot and the mounted motor work together to apply the required clamped boundary conditions on the manipulated rod. A camera is used to image the configuration of the rod undergoing manipulation. The images are used to calculate the difference between the experimental rod and the predicted helical shape when searching for the boundary of the set of stable helices. As described in Fig. 6.2a, a large increase in the error between the manipulated rod and the predicted configuration indicates that a point on the boundary between stable and unstable configurations has been found. The robotic system is able to explore the boundary of the set of stable helices by repeating this experimental procedure along different directions $\mathbf{S}$.

6.3.2 Boundary conditions and path planning

In this study, the boundary conditions on the two ends of the rod are clamped, which require fixing both the position and the tangent. The position can be expressed by three variables $[x, y, z]$ in the world coordinate system – a reference frame fixed to the environment – and the tangent – a unit vector – can be specified using a rotation matrix. First, we discuss our implementation of the conditions on the position using the robot. As shown in Fig. 6.4a, we regard one clamped end (the mounting end) of the rod as the origin of the world frame without any loss in generality. This position $\mathbf{q}_0$ is the first node in the discrete representation. For a helical shape, the last node $\mathbf{q}_n$ held by the end-effector of the robot can be expressed

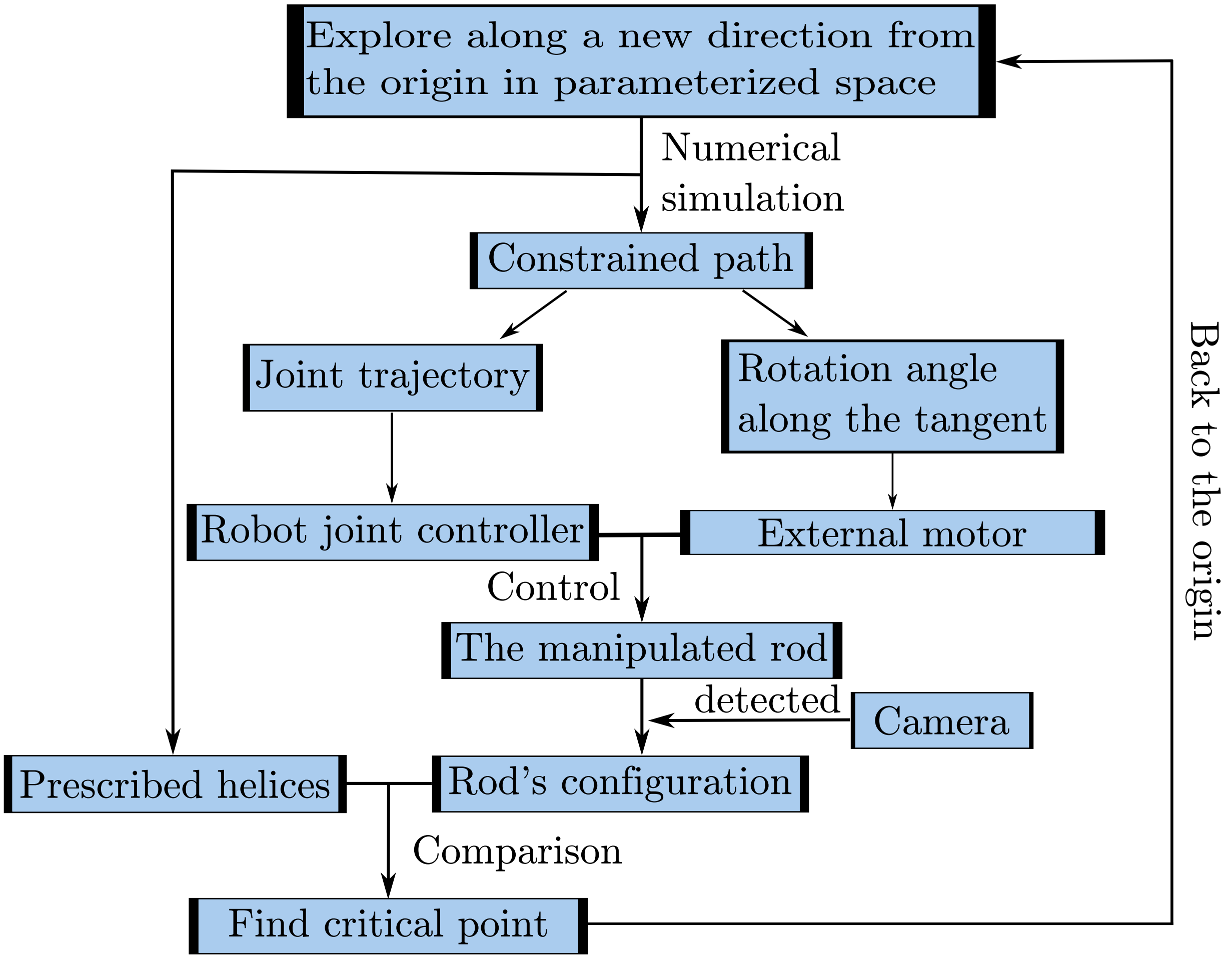


Figure 6.3: Flow chart of the manipulation scheme.

using the geometrical properties (κ and τ) of the helix such that

$$\mathbf{q}_n = \begin{bmatrix} -\frac{\kappa\tau\left(-1+(\kappa^2+\tau^2)^{3/2}\sin\frac{1}{(\kappa^2+\tau^2)^{3/2}}\right)}{\kappa^2+\tau^2} \\ \kappa\left(\cos\left(\frac{1}{(\kappa^2+\tau^2)^{3/2}}\right)-1\right)(\kappa^2+\tau^2) \\ \frac{\tau^2}{\kappa^2+\tau^2}+\kappa^2\sqrt{(\kappa^2+\tau^2)}\sin\left(\frac{1}{(\kappa^2+\tau^2)^{3/2}}\right) \end{bmatrix}. \tag{6.2}$$

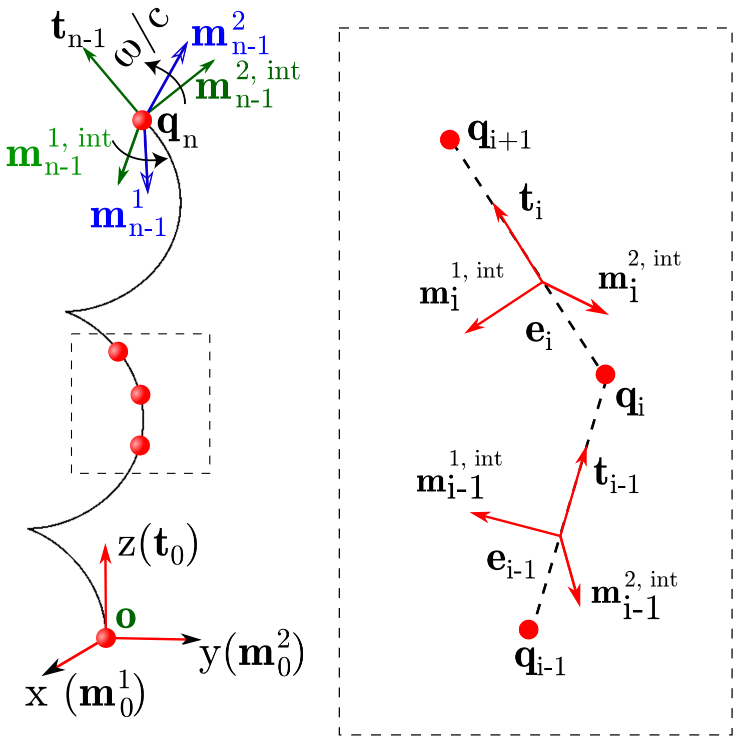
Next, a method to implement the tangent or orientation boundary condition has to be developed. The material frame $[\mathbf{m}^1_{n-1}, \mathbf{m}^2_{n-1}, \mathbf{t}_{n-1}]$ on the last edge (manipulated end) gives the required rotation matrix. Referring to Fig. 6.3, this material frame is imposed by the joint controller of the end-effector and the external motor mounted on the end-effector. When the twisting moment is zero ($\omega = 0$), we denote the material frame of the last edge to be $[\mathbf{m}_{n-1}^{1,\text{int}}, \mathbf{m}_{n-1}^{2,\text{int}}, \mathbf{t}_{n-1}]$. The joint controller of the robot imposes this "intermediate" material frame. The intermediate material frame can be computed using parallel transport, which allows us to move the material frame from one edge to another without twisting about the tangent. Referring to Fig. 6.4b, given the material frame on edge $\mathbf{e}_{i-1}$, the material frame on $\mathbf{e}_i$ that does not generate any twist can be computed from the following steps.

$$\begin{aligned}
\mathbf{b} &= \frac{\mathbf{t}_{i-1} \times \mathbf{t}_i}{\|\mathbf{t}_{i-1} \times \mathbf{t}_i\|} \text{ where } \mathbf{t}_{i-1} = \frac{\mathbf{e}_{i-1}}{\|\mathbf{e}_{i-1}\|}, \mathbf{t}_i = \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|}, \\
\mathbf{m}_i^{1,\text{int}} &= (\mathbf{m}_{i-1}^{1,\text{int}} \cdot (\mathbf{t}_{i-1} \times \mathbf{b}))(\mathbf{t}_i \times \mathbf{b}) + (\mathbf{m}_{i-1}^{1,\text{int}} \cdot \mathbf{b})\mathbf{b}, \\
\mathbf{m}_i^{1,\text{int}} &= \frac{\mathbf{m}_i^{1,\text{int}}}{\|\mathbf{m}_i^{1,\text{int}}\|}, \\
\mathbf{m}_i^{2,\text{int}} &= \mathbf{t}_i \times \mathbf{m}_i^{1,\text{int}},
\end{aligned} \tag{6.3}$$

where $[\mathbf{m}_i^{1,\text{int}}, \mathbf{m}_i^{2,\text{int}}, \mathbf{t}_i]$ is the intermediate frame on the i-th edge with zero twist compared with the material frame $[\mathbf{m}_0^1, \mathbf{m}_0^2, \mathbf{t}_0]$ on the fixed end. By sequentially parallel transporting the material frame from the first edge to the last one, we can obtain the intermediate frame $[\mathbf{m}_{n-1}^{1,\text{int}}, \mathbf{m}_{n-1}^{2,\text{int}}, \mathbf{t}_{n-1}]$.

This intermediate frame and the prescribed material frame $[\mathbf{m}_{n-1}^1, \mathbf{m}_{n-1}^2, \mathbf{t}_{n-1}]$ share the tangent $\mathbf{t}_{n-1}$ as the third director. Therefore, only a scalar quantity – the rotation angle – is needed to obtain the prescribed material frame from the intermediate frame. As indicated in Fig. 6.3, an external motor that is mounted on the end-effector rotates the last edge by a rotation angle ω/c, where c is the ratio between twisting stiffness and bending stiffness. Fig. 6.4a schematically shows the two frames and the rotation angle. The reason behind using an external motor to impose the rotation is that the rotation angle in this study can be so large that it falls outside the joint limits of the collaborative robot.

Knowing the required boundary conditions of a series of helical shapes to be explored, we can construct a constrained path in Cartesian space. During motion planning, singu-



(a) Illustration of the reference frames on the manipulated end

(b) Schematic diagram of parallel transport

Figure 6.4: Illustration of the boundary conditions and the parallel transport.

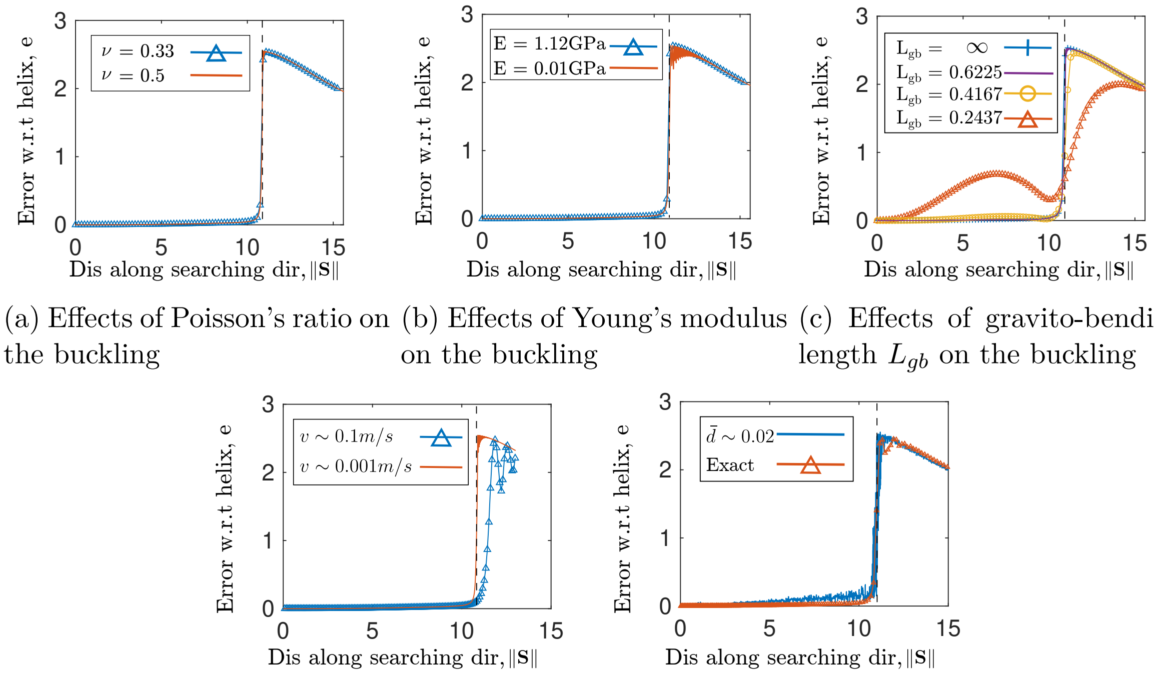
lar configurations of the robot should be avoided since these singularities could result in a high speed of the manipulator, thereby influencing the stability of the rod. To minimize the likelihood of encountering these singularities, the desired path of the manipulator was divided into a series of discrete points and the Descartes planner [168] was used to plan a corresponding path in the robot's joint space. The collaborative robot used in our experiments has seven DOF, and there are multiple joint solutions for a specified pose along the desired path. Between discrete points along the path, the Descartes planner minimizes the function $f = \|\theta^r_{i+1} - \theta^r_i\|$, where θ^r is the robot joint solution and subscripts denote the index of the corresponding discrete point on the path. The minimization of f reduces the likelihood of encountering large and sudden changes in the robot's joint angles due to singularities. During motion planning, we also account for self-collisions and joint limits of the robotic system, resulting in a joint path that does not have sudden jumps associated with singularities. We note, however, that for a robotic system with fewer DOFs (e.g., 6 DOFs), avoiding singularities might be more challenging during motion planning. In this circumstance, methods such as those described in [169] can be used during motion planning.

6.3.3 Perception system

The perception system is completed with a camera (Intel Realsense D435) that images the rod. Lightweight markers made of paper are attached along the rod (see Fig. 6.1) to track its configuration. Using the extrinsic and intrinsic matrices of the camera, the expected helical shapes from numerical simulations can be projected into the image domain of the experiments. The intrinsic camera matrix is provided by the vendor, and the extrinsic matrix is measured with robot hand-eye calibration. The difference between the expected helical shape and the detected manipulated rod in the image domain is used to evaluate if elastic instability occurs. When the elastic rod with a helical centerline reaches the critical point, it will snap into a non-helical shape and induce a large difference between the experimental and prescribed helical shapes. A representative example of this experiment vs. simulation comparison is shown in the supplementary video. The corresponding curvature κ, torsion ω, and twist τ at the onset of elastic instability is a point on the boundary of the set of stable helical configurations. In this work, a detailed comparison between the 3D simulated and experimental configurations is not necessary; we are only interested in capturing the onset of instability.

6.3.4 Effects of disturbances

We now discuss potential sources of disturbances in the robotic system and their effects on the stability measurements. First, in § 6.2, we described how the parameters κ, τ, ω, and c can be used to describe the set of all helical rod configurations, where curvature κ and



(a) Effects of Poisson's ratio on the buckling

(b) Effects of Young's modulus on the buckling

(c) Effects of gravito-bending length L_{gb} on the buckling

(d) Effects of velocity v on the buckling

(e) Effects of jittering d on the buckling

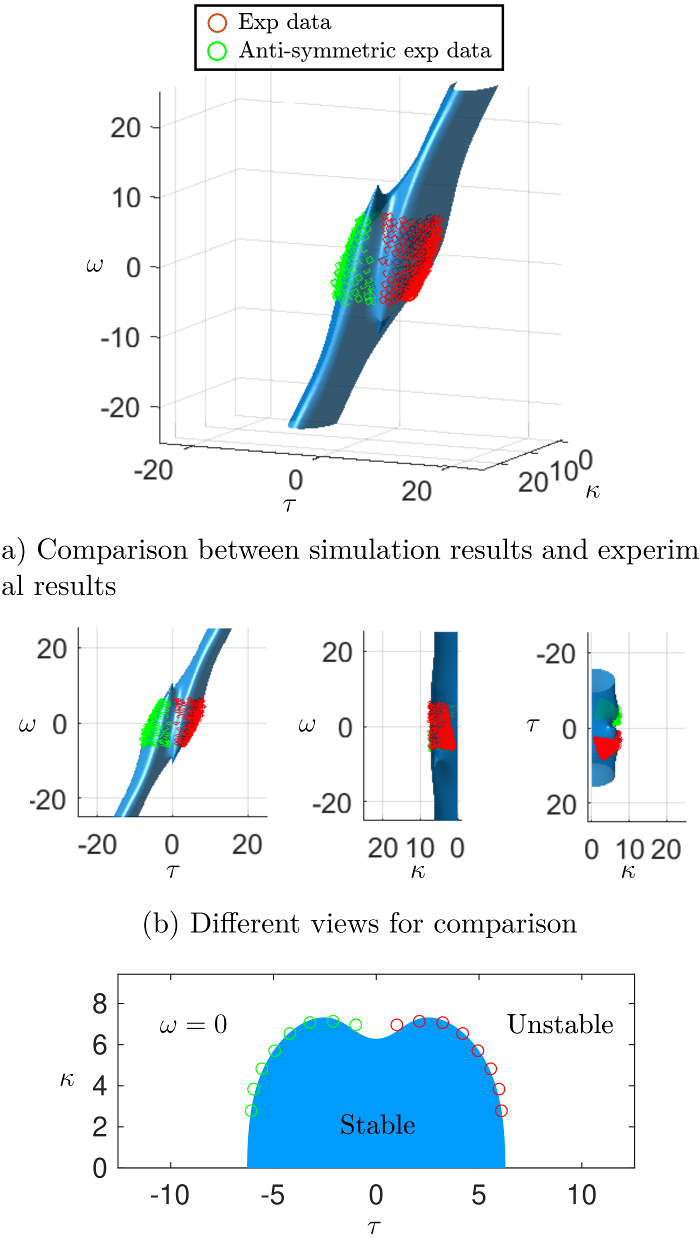
Figure 6.5: The effects of various contributing factors on the point of elastic instability. In each figure, the horizontal axis is the distance along the search direction $\overline{\mathbf{S}} = 1/\sqrt{3}[1\ 1\ 1]$, and the vertical axis is the error with respect to the expected helical shape. The predicted point of instability is denoted by the vertical dashed line.

torsion τ are geometrical parameters of the rod that are independent of the rod's material. Furthermore, the ratio of twisting stiffness to bending stiffness, $c = k_t/k_b = 1/(1+\nu)$, is only dependent upon the Poisson's ratio of the material. Varying c changes the rod's twisting strain, ω/c, and a change of the rod's twisting energy. However, these two changes together result in no change in the rod's stability [150]. To validate this result, we simulated a helical rod using the DER formulation along the search direction $\overline{\mathbf{S}} = 1/\sqrt{3}[1\ 1\ 1]$. The default parameters for these simulations were: Young's modulus $E = 1.12$ Gpa, Poisson's ratio $\nu = 0.33$, density $\rho = 1180$ kg/m^3, length $L = 1$ m, and radius $h = 0.781$ mm. In Fig. 6.5 (a)-(b), we show the effect of varying Poisson's ratio and Young's modulus on the instability. We see that the rod's material properties have a minimal effect on the instability point (the value of which is indicated by the vertical dashed line for the default parameters). Therefore, the material of the rod should have minimal influence on our stability results, and it is sufficient to use a single rod in the experiments. In future work, our robotic system can be used to automate the experimental validation of the theoretical result regarding c obtained in [150].

In addition to the rod's material, the effect of gravity on the rod must also be considered. This paper assumes that the rod is sufficiently stiff so that gravity can be neglected. To quantify this assumption, we use the gravito-bending length $L_{gb} = \left(\frac{h^2 E}{8\rho g}\right)^{1/3}$, where g is the acceleration of gravity [7]. L_{gb} describes the balance between gravitational and bending energy, and the effects of gravity diminish as L_{gb} increases. Fig. 6.5c shows the effect of varying L_{gb} on the instability, and we see that gravity becomes negligible when $L_{gb}/L > 0.6$.

The main external disturbances (i.e., external to the rod) can be divided into two components: the robotic manipulator's speed and the manipulator's jittering. We again conducted simulations with the default parameters described above to assess these effects. Fig. 6.5d shows how varying the manipulator's speed v affects the instability point by introducing inertial effects. We conclude that the manipulator's speed should be sufficiently small to minimize the influence of these inertial disturbances. We also used the simulator to explore the effects of jittering, i.e., small deviations from the desired path. As shown in Fig. 6.5e, we found that jittering of magnitude less than $d \backsim 0.02L$, where L is the rod's length, has minimal influence on the rod's stability.

6.4 Results



(a) Comparison between simulation results and experimental results

(b) Different views for comparison

(c) Comparison between simulation and experimental results when $\omega = 0$.

Figure 6.6: Comparison between simulation data (blue surface) and experiment data: red circles denote the experimental data; green circles denote the anti-symmetric experimental data.

The robotic system, manipulation scheme, and vision system described in § 6.3 were implemented to conduct automated experimental testing of stability for helical elastic rods. We used a collaborative robot (Sawyer, Rethink Robotics) with seven degrees of freedom. The motor used for applying external torque is a stepper motor (NEMA 17) controlled with a microcontroller (Arduino Uno Rev3), and the camera used in the perception system is an RGBD camera (Intel Realsense D435). The elastic rod used was a superelastic nitinol wire with length $L = 0.5$ m, diameter $2r = 1.5875$ mm, Poisson's ratio $\nu = 0.33$, density $\rho = 6450$ kg/m^3, and Young's modulus $E = 67.5$ GPa. For this rod, we have $L_{gb}/L = 1.3816$, and we can therefore neglect the effects of gravity. Furthermore, the reported accuracy of the Sawyer robot is less than 0.01 cm, which is within the jittering tolerance established in the previous section.

As described in Ref. [150], there is a symmetry in the κ-τ-ω parameter space between points located at (κ, τ, ω) and $(\kappa, -\tau, -\omega)$. We therefore only consider search directions with $\tau > 0$, and we reflect these points on the stability boundary to generate data for $\tau < 0$. In future work, experiments will be conducted for all values of τ to validate this symmetry. The rod was manipulated along a total of 328 search directions until an instability occurred, and the resulting values of κ, τ, and ω at the instability were recorded.

The blue surface in Fig. 6.6a shows the predicted boundary between stable and unstable helical configurations within the κ-τ-ω parameter space based on the DER numerical simulations. The experimental data collected by our automated testing procedure is shown in red, and the anti-symmetric data generated from the symmetry is shown in green. Views along each coordinate axis are shown in Fig. 6.6b, and Fig. 6.6c shows the comparison for the two-dimensional section with $\omega = 0$, i.e., twist-free helices. Our results show good agreement between the simulated and experimental data, suggesting that our automated testing procedure can reproduce the stability boundary accurately.

We introduce an error that measures the relative distance between points on the simulated and experimental stability boundaries along the same search direction to quantify the difference between the simulated and experimental data. Along a given search direction $\overline{\mathbf{S}}$, we let $\mathbf{S_s} = [\kappa_s, \tau_s, \omega_s]$ denote the resulting point on the stability boundary based on numerical simulations, and similarly we let $\mathbf{S_e} = [\kappa_e, \tau_e, \omega_e]$ denote the resulting point on the stability boundary based on the experimental data. The relative error along the search direction $\overline{\mathbf{S}}$ is then defined as $err = |(\|S_e\| - \|S_s\|) / \|S_s\||$. The average error of our experimental results over all 328 search directions was 0.0272. The maximum error found was 0.1298, and the standard deviation of the error was 0.0238. Based on these results, we can conclude that our automated experimental testing method can accurately determine when an elastic rod in a helical configuration loses stability.

6.5 Summary and Outlook

This chapter developed an automated testing procedure for determining the stability of a helical elastic rod. Due to the repetitive nature of the test and the need to simultaneously manipulate both the position and orientation at one end of the rod, the robotic system was a key component of our testing procedure. Experimental observations were compared with results from numerical simulations based on the DER algorithm, and their agreement suggests our method can accurately capture the onset of instabilities in helical rods. Although our manipulation scheme relied on specific properties of helical rods, the other components of our testing method can be applied to analyze the stability of non-helical rod configurations.

While our automated testing system was able to accurately reproduce the boundary between stable and unstable helical rods, there are areas in which the method could be improved. First, a primary error source in our system was the stepper motor due to its inability to apply a consistently linear rotation and latency when communicating with the robotic system. Furthermore, since the position of one end of the rod as fixed in our experiment, self-collisions of the robot prevented us from exploring certain regions of the κ-τ-ω parameter space. Using two robots to manipulate the rod collaboratively would provide additional dexterity, allowing us to explore a larger region of the parameter space. Our future work will explore how the robot's workspace constrains the rod's mechanical parameter space. Despite these areas for improvement, the methods described in this paper provide a foundation for using robotic systems to research experimental mechanics. In future work, a robotic system could be used to perform mechanics experiments and apply machine learning based on the collected data to improve the mechanical model of the object being manipulated.

CHAPTER 7

Conclusions

We have presented extensive research into the synergy between computational mechanics and robotics for slender structures. Our work has led to the development of a sim2real framework aimed at generating optimal manipulation strategies for tasks involving deformable objects. Our framework leverages insights gained from computational mechanics, harnessing the accuracy and dexterity of robotics to conduct automated experiments that collect valuable experimental data through interactions with deformable slender structures. These interactions provide valuable insights that contribute to advancing our understanding of slender structures.

Our contributions to computational mechanics are primarily showcased in Chs. 2 and 3. Ch. 2 introduced a novel frictional contact handling framework called IMC. This framework incorporates discrete differential geometry principles to address frictional contact interactions. We then applied the IMC frictional contact model to enhance the capabilities of a popular simulation algorithm, Discrete Elastic Rods (DER)[13], widely used in computer graphics communities. The core concept of the IMC framework involves deriving artificial penalty energy directly from the topological characteristics of the simulated objects. IMC was integrated with DER and the regularized stokeslet segment (RSS) to investigate a crucial biological phenomenon, flagella bundling. In this context, we demonstrated the influence of frictional contact by conducting a comprehensive numerical sweep in simulations. Furthermore, we conducted an full comparison between IMC and a state-of-the-art frictional contact handling framework, IPC [24]. Our results clearly indicate that IMC outperforms

IPC regarding accuracy and computational efficiency, particularly when simulating flagella bundling. Additionally, we validated the physical accuracy of the IMC framework by comparing its results with established knot theories found in prior literature [25, 26]. These achievements lay a strong foundation for our subsequent investigations into the study and manipulation of slender structures in frictional contact scenarios.

In Ch. 3, we studied a fascinating phenomenon associated with the tightening of an overhand knot. The process of tightening an overhand knot leads to an occurrence where the closed loop of the knot "inverts", giving rise to the formation of additional tangles within the structure. This phenomenon is of great interest due to its universal characteristics in rod-like structures. Our investigation into this intriguing knot inversion process was conducted through a comprehensive study encompassing simulations, real-world experiments, and theoretical analysis. We showcased the influences of three intricate parameters, rod radius, unknotting number, and friction coefficient, on the inversion process. In addition, theoretical analyses reveal that the knot inversion process is essentially an interplay of energy between the knotted braids and the knotted loop within the structure. This interplay ultimately results in a snapping process, shedding light on the underlying mechanics of this phenomenon. This study uncovered a novel observation in knots and holds significant implications for manipulating knotted structures and the broader understanding of tangles observed in nature.

With the guidance of computational mechanics, a sim2Real framework was proposed to develop optimal manipulation schemes for slender structures. Combining computational mechanics-based simulations, scaling analysis, and machine learning, this framework leverages the precision of simulations, the generality of scaling analysis, and the rapid inference capabilities of machine learning to tackle complex deformable manipulation tasks. Ch. 4 tackled a challenging deformable manipulation task, folding a paper with only one manipulator. A physics-informed model was learned from scaled simulation data to execute optimal motion planning with the model's prediction. We compared our physics-informed folding with a state-of-the-art folding algorithm [102]. We demonstrate our folding scheme is globally optimal, effectively avoiding the vexing issue of buckling that had proven challenging for the previously established method (STOA method).

In Ch. 5, we learned an optimal policy from the scaled simulation data to facilitate the precise deployment of solid deformable linear objects onto various rigid substrates, following various feasible patterns. The trained policy is validated to be robust, efficient, and accurate with an extensive real robotic experiment. In addition, we also showcased the versatility of the deployment scheme in engineering domains including cable management and knot tying.

Beyond manipulation, we ventured into the realm of robotics and automation to experimentally study mechanics. By designing an automated experimental platform based on a 7 DOF robot, we successfully unveiled the buckling behavior of slender structures with helical centerlines in Ch. 6. This pioneering work has paved the way for the precise and automated exploration of mechanics using robotic platforms, overcoming the limitations of traditional experimental approaches.

REFERENCES

[1] Basile Audoly and Yves Pomeau. Elasticity and geometry. In *Peyresq Lectures on Nonlinear Phenomena*, pages 1–35. World Scientific, 2000.

[2] Hannes G Kenngott, Beat P Müller-Stich, Michael A Reiter, Jens Rassweiler, and Carsten N Gutt. Robotic suturing: technique and benefit in advanced laparoscopic surgery. *Minimally Invasive Therapy & Allied Technologies*, 17(3):160–167, 2008.

[3] Vishal P Patil, Joseph D Sandt, Mathias Kolle, and Jörn Dunkel. Topological mechanics of knots and tangles. *Science*, 367(6473):71–75, 2020.

[4] Majid Taghavi, Tim Helps, and Jonathan Rossiter. Electro-ribbon actuators and electro-origami robots. *Science Robotics*, 3(25):eaau9795, 2018.

[5] Yichao Tang, Yinding Chi, Jiefeng Sun, Tzu-Hao Huang, Omid H Maghsoudi, Andrew Spence, Jianguo Zhao, Hao Su, and Jie Yin. Leveraging elastic instabilities for amplified performance: Spine-inspired high-speed and high-force soft robots. *Science advances*, 6(19):eaaz6912, 2020.

[6] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018.

[7] Mohammad K Jawed, Fang Da, Jungseock Joo, Eitan Grinspun, and Pedro M Reis. Coiling of elastic rods on rigid substrates. *Proceedings of the National Academy of Sciences*, 111(41):14663–14668, 2014.

[8] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021.

[9] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2146–2153. IEEE, 2017.

[10] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E Gonzalez, and Ken Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9418. IEEE, 2020.

[11] Robert Lee, Masashi Hamaya, Takayuki Murooka, Yoshihisa Ijiri, and Peter Corke. Sample-efficient learning of deformable linear object manipulation in the real world through self-supervision. *IEEE Robotics and Automation Letters*, 7(1):573–580, 2021.

[12] Weicheng Huang, Xiaonan Huang, Carmel Majidi, and M Khalid Jawed. Dynamic simulation of articulated soft robots. *Nature communications*, 11(1):2233, 2020.

[13] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. In *ACM transactions on graphics (TOG)*, volume 27, page 63. ACM, 2008.

[14] Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. Discrete viscous threads. In *ACM Transactions on Graphics (TOG)*, volume 29, page 116. ACM, 2010.

[15] Mohammad K Jawed, Pierre-Thomas Brun, and Pedro M Reis. A geometric model for the coiling of an elastic rod deployed onto a moving substrate. *Journal of Applied Mechanics*, 82(12):121007, 2015.

[16] Dezhong Tong, Andy Borum, and Mohammad Khalid Jawed. Automated stability testing of elastic rods with helical centerlines using a robotic system. *IEEE Robotics and Automation Letters*, 7(2):1126–1133, 2021.

[17] Weicheng Huang, Yunbo Wang, Xuanhe Li, and Mohammad K Jawed. Shear induced supercritical pitchfork bifurcation of pre-buckled bands, from narrow strips to wide plates. *Journal of the Mechanics and Physics of Solids*, 145:104168, 2020.

[18] Vladimír Petrík, Vladimír Smutný, Pavel Krsek, and Václav Hlaváč. Physics-based model of a rectangular garment for robotic folding. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 951–956, 2016.

[19] Jie Sun, Zhuo Peng, Weibiao Zhou, Jerry YH Fuh, Geok Soon Hong, and Annette Chiu. A review on 3d printing for customized food fabrication. *Procedia Manufacturing*, 1:308–319, 2015.

[20] Louis L Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 709–716. IEEE, 2000.

[21] Noam Geblinger, Ariel Ismach, and Ernesto Joselevich. Self-organized nanotube serpentines. *Nature nanotechnology*, 3(4):195–200, 2008.

[22] Wee E Teo and Seeram Ramakrishna. A review on electrospinning design and nanofibre assemblies. *Nanotechnology*, 17(14):R89, 2006.

[23] Veronika Magdanz, Samuel Sanchez, and Oliver G Schmidt. Development of a sperm-flagella driven micro-bio-robot. *Advanced materials*, 25(45):6581–6588, 2013.

[24] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics (TOG)*, 39(4), 2020.

[25] B Audoly, N Clauvelin, and S Neukirch. Elastic knots. *Physical Review Letters*, 99(16):164301, 2007.

[26] Mohammad K Jawed, P Dieleman, B Audoly, and Pedro M Reis. Untangling the mechanics and topology in the frictional response of long overhand elastic knots. *Physical review letters*, 115(11):118302, 2015.

[27] Dezhong Tong, Andrew Choi, Jungseock Joo, and M Khalid Jawed. A fully implicit method for robust frictional contact handling in elastic rods. *Extreme Mechanics Letters*, 58:101924, 2023.

[28] Mohammad K Jawed and Pedro M Reis. Pattern morphology in the elastic sewing machine. *Extreme Mechanics Letters*, 1:76–82, 2014.

[29] Changyeob Baek, Andrew O Sageman-Furnas, Mohammad K Jawed, and Pedro M Reis. Form finding in elastic gridshells. *Proceedings of the National Academy of Sciences*, 115(1):75–80, 2018.

[30] Julian Panetta, MINA Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)*, 38(4):83, 2019.

[31] Changyeob Baek and Pedro M Reis. Rigidity of hemispherical elastic gridshells under point load indentation. *Journal of the Mechanics and Physics of Solids*, 124:411–426, 2019.

[32] Mohammad K Jawed, Noor K Khouri, Fang Da, Eitan Grinspun, and Pedro M Reis. Propulsion and instability of a flexible helical rod rotating in a viscous fluid. *Physical review letters*, 115(16):168101, 2015.

[33] Mohammad K Jawed and Pedro M Reis. Deformation of a soft helical filament in an axial flow at low reynolds number. *Soft Matter*, 12(6):1898–1905, 2016.

[34] MK Jawed and Pedro M Reis. Dynamics of a flexible helical filament rotating in a viscous fluid near a rigid boundary. *Physical Review Fluids*, 2(3):034101, 2017.

[35] Jonas Spillmann and Matthias Teschner. An adaptive contact model for the robust simulation of knots. In *Computer Graphics Forum*, volume 27, pages 497–506. Wiley Online Library, 2008.

[36] Andrew Choi, Dezhong Tong, Mohammad K. Jawed, and Jungseock Joo. Implicit Contact Model for Discrete Elastic Rods in Knot Tying. *Journal of Applied Mechanics*, 88(5):051010, 03 2021.

[37] Michel Jean and Jean J Moreau. Dynamics in the presence of unilateral contacts and dry friction: a numerical approach. In *Unilateral Problems in Structural Analysis—2*, pages 151–196. Springer, 1987.

[38] Michel Jean and Jean Jacques Moreau. Unilaterality and dry friction in the dynamics of rigid body collections. In *1st Contact Mechanics International Symposium*, pages 31–48, 1992.

[39] Pierre Alart and Alain Curnier. A mixed formulation for frictional contact problems prone to newton like solution methods. *Computer methods in applied mechanics and engineering*, 92(3):353–375, 1991.

[40] Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.

[41] Danny M Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[42] Gilles Daviet. Simple and scalable frictional contacts for thin nodal objects. *ACM Transactions on Graphics (TOG)*, 39(4):61–1, 2020.

[43] Vladimir J. Lumelsky. On fast computation of distance between line segments. *Inf. Process. Lett.*, 21:55–61, 1985.

[44] Heather Flores, Edgar Lobaton, Stefan Méndez-Diez, Svetlana Tlupova, and Ricardo Cortez. A study of bacterial flagellar bundling. *Bulletin of mathematical biology*, 67(1):137–168, 2005.

[45] Luis H Cisneros, John O Kessler, Ricardo Ortiz, Ricardo Cortez, and Martin A Bees. Unexpected bipolar flagellar arrangements and long-range flows driven by bacteria near solid boundaries. *Physical review letters*, 101(16):168102, 2008.

[46] Shang Yik Reigh, Roland G Winkler, and Gerhard Gompper. Synchronization and bundling of anchored bacterial flagella. *Soft Matter*, 8(16):4363–4372, 2012.

[47] Ranjith Maniyeri and Sangmo Kang. Numerical study on bacterial flagellar bundling and tumbling in a viscous fluid using an immersed boundary method. *Applied Mathematical Modelling*, 38(14):3567–3590, 2014.

[48] Marius Hintsche, Veronika Waljor, Robert Großmann, Marco J Kühn, Kai M Thormann, Fernando Peruani, and Carsten Beta. A polar bundle of flagella can drive bacterial swimming by pushing, pulling, or coiling around the cell body. *Scientific reports*, 7(1):1–10, 2017.

[49] Frank TM Nguyen and Michael D Graham. Impacts of multiflagellarity on stability and speed of bacterial locomotion. *Physical Review E*, 98(4):042419, 2018.

[50] Wanho Lee, Yongsam Kim, Boyce E Griffith, and Sookkyung Lim. Bacterial flagellar bundling and unbundling via polymorphic transformations. *Physical Review E*, 98(5):052405, 2018.

[51] Weicheng Huang and M Khalid Jawed. Numerical simulation of bundling of helical elastic rods in a viscous fluid. *Computers & Fluids*, 228:105038, 2021.

[52] Thomas R Powers. Role of body rotation in bacterial flagellar bundling. *Physical Review E*, 65(4):040903, 2002.

[53] Ricardo Cortez. Regularized stokeslet segments. *Journal of Computational Physics*, 375:783–796, 2018.

[54] MunJu Kim, James C Bird, Annemarie J Van Parys, Kenneth S Breuer, and Thomas R Powers. A macroscopic scale model of bacterial flagellar bundling. *Proceedings of the National Academy of Sciences*, 100(26):15481–15485, 2003.

[55] Zhou Ye, Stéphane Régnier, and Metin Sitti. Rotating magnetic miniature swimming robots with multiple flexible flagella. *IEEE Transactions on Robotics*, 30(1):3–13, 2013.

[56] Nicolas Beyrand, Laurent Couraud, Antoine Barbot, Dominique Decanini, and Gilgueng Hwang. Multi-flagella helical microswimmers for multiscale cargo transport and reversible targeted binding. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1403–1408. IEEE, 2015.

[57] Kwangmin Son, Jeffrey S Guasto, and Roman Stocker. Bacteria can exploit a flagellar buckling instability to change direction. *Nature physics*, 9(8):494–498, 2013.

[58] Roberto Rusconi, Jeffrey S Guasto, and Roman Stocker. Bacterial transport suppressed by fluid shear. *Nature physics*, 10(3):212–217, 2014.

[59] Luis H Cisneros, Ricardo Cortez, Christopher Dombrowski, Raymond E Goldstein, and John O Kessler. Fluid dynamics of self-propelled microorganisms, from individuals to concentrated populations. In *Animal Locomotion*, pages 99–115. Springer, 2010.

[60] Saverio E Spagnolie and Eric Lauga. Hydrodynamics of self-propulsion near a boundary: predictions and accuracy of far-field approximations. *Journal of Fluid Mechanics*, 700:105–147, 2012.

[61] Sarah D Olson, Susan S Suarez, and Lisa J Fauci. Coupling biochemistry and hydrodynamics captures hyperactivated sperm motility in a simple flagellar model. *Journal of theoretical biology*, 283(1):203–216, 2011.

[62] Julie Simons, Lisa Fauci, and Ricardo Cortez. A fully three-dimensional model of the interaction of driven elastic filaments in a stokes flow with applications to sperm motility. *Journal of biomechanics*, 48(9):1639–1651, 2015.

[63] Bruce Rodenborn, Chih-Hung Chen, Harry L Swinney, Bin Liu, and HP Zhang. Propulsion of microorganisms by a helical flagellum. *Proceedings of the National Academy of Sciences*, 110(5):E338–E347, 2013.

[64] Dezhong Tong, Andrew Choi, Jungseock Joo, Andy Borum, and Mohammad Khalid Jawed. Snap buckling in overhand knots. *Journal of Applied Mechanics*, 90(4):041008, 2023.

[65] Eston Schweickart, Doug L James, and Steve Marschner. Animating elastic rods with sound. *ACM Transactions on Graphics (TOG)*, 36(4):1–10, 2017.

[66] Sylwester Przybyl and Piotr Pieranski. Tightening of the elastic overhand knot. *Physical Review E*, 79(3):031801, 2009.

[67] Derek E Moulton, Paul Grandgeorge, and Sébastien Neukirch. Stable elastic knots with no self-contact. *Journal of the Mechanics and Physics of Solids*, 116:33–53, 2018.

[68] Yoël Forterre, Jan M Skotheim, Jacques Dumais, and Lakshminarayanan Mahadevan. How the venus flytrap snaps. *Nature*, 433(7024):421–425, 2005.

[69] E Kebadze, SD Guest, and S Pellegrino. Bistable prestressed shell structures. *International Journal of Solids and Structures*, 41(11-12):2801–2820, 2004.

[70] Anupam Pandey, Derek E Moulton, Dominic Vella, and Douglas P Holmes. Dynamics of snapping beams and jumping poppers. *EPL (Europhysics Letters)*, 105(2):24001, 2014.

[71] Tian Chen, Osama R Bilal, Kristina Shea, and Chiara Daraio. Harnessing bistability for directional propulsion of soft, untethered robots. *Proceedings of the National Academy of Sciences*, 115(22):5698–5702, 2018.

[72] Bernard D Coleman, David Swigon, and Irwin Tobias. Elastic stability of dna configurations. ii. supercoiled plasmids with self-contact. *Physical Review E*, 61(1):759, 2000.

[73] J Michael T Thompson. Single-molecule magnetic tweezer tests on dna: bounds on topoisomerase relaxation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464(2099):2811–2829, 2008.

[74] N Clauvelin, B Audoly, and S Neukirch. Matched asymptotic expansions for twisted elastic knots: a self-contact problem with non-trivial contact topology. *Journal of the Mechanics and Physics of Solids*, 57(9):1623–1656, 2009.

[75] Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, and Vincent Acary. A nonsmooth newton solver for capturing exact coulomb friction in fiber assemblies. *ACM Transactions on Graphics (TOG)*, 30(1):6, 2011.

[76] Dezhong Tong, Andrew Choi, Demetri Terzopoulos, Jungseock Joo, and M Khalid Jawed. Deep learning of force manifolds from the simulated physics of robotic paper folding. *arXiv preprint arXiv:2301.01968*, 2023.

[77] Yasuyo Kita, Fumio Kanehiro, Toshio Ueshiba, and Nobuyuki Kita. Clothes handling based on recognition by strategic observation. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 53–58. IEEE, 2011.

[78] Andreas Doumanoglou, Jan Stria, Georgia Peleka, Ioannis Mariolis, Vladimir Petrik, Andreas Kargakos, Libor Wagner, Václav Hlaváč, Tae-Kyun Kim, and Sotiris Malassiotis. Folding clothes autonomously: A complete pipeline. *IEEE Transactions on Robotics*, 32(6):1461–1478, 2016.

[79] Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James F O'Brien, and Pieter Abbeel. Bringing clothing into desired configurations with limited perception. In *2011 IEEE international conference on robotics and automation*, pages 3893–3900. IEEE, 2011.

[80] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315. IEEE, 2010.

[81] Lukas Twardon and Helge Ritter. Interaction skills for a coat-check robot: Identifying and handling the boundary components of clothes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3682–3688. IEEE, 2015.

[82] Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, and Sotiris Malassiotis. Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In *2014 IEEE international conference on robotics and automation*, pages 987–993. IEEE, 2014.

[83] John Schulman, Ankush Gupta, Sibi Venkatesan, Mallory Tayson-Frederick, and Pieter Abbeel. A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4111–4117. IEEE, 2013.

[84] Wen Hao Lui and Ashutosh Saxena. Tangled: Learning to untangle ropes with rgb-d perception. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 837–844. IEEE, 2013.

[85] Tapomayukh Bhattacharjee, Gilwoo Lee, Hanjun Song, and Siddhartha S Srinivasa. Towards robotic feeding: Role of haptics in fork-based food manipulation. *IEEE Robotics and Automation Letters*, 4(2):1485–1492, 2019.

[86] Weifu Wang, Dmitry Berenson, and Devin Balkcom. An online method for tight-tolerance insertion tasks for string and rope. In *2015 IEEE International Conference on Robotics and Automation*, pages 2488–2495. IEEE, 2015.

[87] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm. In *2012 IEEE International Conference on Robotics and Automation*, pages 2249–2254. IEEE, 2012.

[88] Shunsuke Kudoh, Tomoyuki Gomi, Ryota Katano, Tetsuo Tomizawa, and Takashi Suehiro. In-air knotting of rope by a dual-arm multi-finger robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6202–6207. IEEE, 2015.

[89] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 49–54. IEEE, 2010.

[90] Harry K. H. Kim, David Bourne, Satyandra Gupta, and S S. Krishnan. Automated process planning for robotic sheet metal bending operations. *Journal of Manufacturing Systems*, 17(5):338 – 360, September 1998.

[91] Devin J Balkcom and Matthew T Mason. Robotic origami folding. *The International Journal of Robotics Research*, 27(5):613–627, 2008.

[92] Stephen Miller, Jur van den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research*, 31(2):249–267, 2012.

[93] Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 177–184. IEEE, 2015.

[94] Alex X. Lee, Abhishek Gupta, Henry Lu, Sergey Levine, and Pieter Abbeel. Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5265–5272, 2015.

[95] Matthias Rambow, Thomas Schauß, Martin Buss, and Sandra Hirche. Autonomous manipulation of deformable objects based on teleoperated demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2809–2814. IEEE, 2012.

[96] Pin-Chu Yang, Kazuma Sasaki, Kanata Suzuki, Kei Kase, Shigeki Sugano, and Tetsuya Ogata. Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robotics and Automation Letters*, 2(2):397–403, 2017.

[97] Vladimír Petrík and Ville Kyrki. Feedback-based fabric strip folding. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 773–778, 2019.

[98] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 734–743. PMLR, 2018.

[99] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. *arXiv preprint arXiv:2011.07215*, 2020.

[100] Yi Zheng, Filipe Fernandes Veiga, Jan Peters, and Veronica J Santos. Autonomous learning of page flipping movements via tactile feedback. *IEEE Transactions on Robotics*, 2022.

[101] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.

[102] Vladimír Petrík, Vladimír Smutný, and Ville Kyrki. Static stability of robotic fabric strip folding. *IEEE/ASME Transactions on Mechatronics*, 25(5):2493–2500, 2020.

[103] Yinxiao Li, Yonghao Yue, Danfei Xu, Eitan Grinspun, and Peter K. Allen. Folding deformable objects using predictive simulation and trajectory optimization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6000–6006, 2015.

[104] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, et al. Challenges and outlook in robotic manipulation of deformable objects. *IEEE Robotics & Automation Magazine*, 29(3):67–77, 2022.

[105] David Hilbert and Stephan Cohn-Vossen. *Geometry and the Imagination*, volume 87. American Mathematical Soc., 2021.

[106] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[107] Dezhong Tong, Andrew Choi, Longhui Qin, Weicheng Huang, Jungseock Joo, and M Khalid Jawed. Sim2real physically informed neural controllers for robotic deployment of deformable linear objects. *arXiv preprint arXiv:2303.02574*, 2023.

[108] Siddarth Sen, Animesh Garg, David V Gealy, Stephen McKinley, Yiming Jen, and Ken Goldberg. Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 4178–4185. IEEE, 2016.

[109] Dimitrios Stefanidis, Fikre Wang, James R Korndorffer, J Bruce Dunne, and Daniel J Scott. Robotic assistance improves intracorporeal suturing performance and safety in the operating room while decreasing operator workload. *Surgical endoscopy*, 24(2):377–382, 2010.

[110] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021.

[111] Stephen Miller, Jur Van Den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research*, 31(2):249–267, 2012.

[112] Ariel Kapusta, Zackory Erickson, Henry M Clever, Wenhao Yu, C Karen Liu, Greg Turk, and Charles C Kemp. Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Autonomous Robots*, 43(8):2183–2207, 2019.

[113] Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[114] Wenhao Yu, Ariel Kapusta, Jie Tan, Charles C Kemp, Greg Turk, and C Karen Liu. Haptic simulation for robot-assisted dressing. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 6044–6051. IEEE, 2017.

[115] Zackory Erickson, Henry M Clever, Greg Turk, C Karen Liu, and Charles C Kemp. Deep haptic model predictive control for robot-assisted dressing. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4437–4444. IEEE, 2018.

[116] Emmanuel Pignat and Sylvain Calinon. Learning adaptive dressing assistance from human demonstration. *Robotics and Autonomous Systems*, 93:61–75, 2017.

[117] Andrew Choi, Dezhong Tong, Brian Park, Demetri Terzopoulos, Jungseock Joo, and Mohammad Khalid Jawed. mbest: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals. *IEEE Robotics and Automation Letters*, 8(8):4863–4870, 2023.

[118] Hidefumi Wakamatsu, Eiji Arai, and Shinichi Hirai. Knotting/unknotting manipulation of deformable linear objects. *The International Journal of Robotics Research*, 25(4):371–395, 2006.

[119] Mitul Saha and Pekka Isto. Manipulation planning for deformable linear objects. *IEEE Transactions on Robotics*, 23(6):1141–1150, 2007.

[120] Dale McConachie, Andrew Dobson, Mengyao Ruan, and Dmitry Berenson. Manipulating deformable objects by interleaving prediction, planning, and control. *The International Journal of Robotics Research*, 39(8):957–982, 2020.

[121] Peter Mitrano, Dale McConachie, and Dmitry Berenson. Learning where to trust unreliable models in an unstructured world for deformable object manipulation. *Science Robotics*, 6(54):eabd8170, 2021.

[122] Jihong Zhu, Benjamin Navarro, Robin Passama, Philippe Fraisse, André Crosnier, and Andrea Cherubini. Robotic manipulation planning for shaping deformable linear objects withenvironmental contacts. *IEEE Robotics and Automation Letters*, 5(1):16–23, 2019.

[123] Masaru Takizawa, Shunsuke Kudoh, and Takashi Suehiro. Method for placing a rope in a target shape and its application to a clove hitch. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 646–651. IEEE, 2015.

[124] Naijing Lv, Jianhua Liu, and Yunyi Jia. Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations. *IEEE Transactions on Robotics*, 2022.

[125] Te Tang, Changhao Wang, and Masayoshi Tomizuka. A framework for manipulating deformable linear objects by coherent point drift. *IEEE Robotics and Automation Letters*, 3(4):3426–3433, 2018.

[126] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE robotics and automation letters*, 5(2):2372–2379, 2020.

[127] Alex X Lee, Sandy H Huang, Dylan Hadfield-Menell, Eric Tzeng, and Pieter Abbeel. Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects. In *2014 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pages 4402–4407. IEEE, 2014.

[128] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[129] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016.

[130] Puren Guler, Karl Pauwels, Alessandro Pieropan, Hedvig Kjellström, and Danica Kragic. Estimating the deformability of elastic materials using optical flow and position-based dynamics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 965–971. IEEE, 2015.

[131] Martin Servin and Claude Lacoursiere. Rigid body cable for virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):783–796, 2008.

[132] Demetri Terzopoulos and Hong Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics (TOG)*, 13(2):103–136, 1994.

[133] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[134] Nazim Haouchine, Winnie Kuang, Stephane Cotin, and Michael Yip. Vision-based force feedback estimation for robot-assisted surgery using instrument-constrained biomechanical three-dimensional maps. *IEEE Robotics and Automation Letters*, 3(3):2160–2165, 2018.

[135] Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. Flexible simulation of deformable models using discontinuous galerkin fem. *Graphical Models*, 71(4):153–167, 2009.

[136] Brad Buckham, Frederick R Driscoll, and Meyer Nahon. Development of a finite element cable model for use in low-tension dynamics simulation. *J. Appl. Mech.*, 71(4):476–485, 2004.

[137] Andrew Choi, Dezhong Tong, Demetri Terzopoulos, Jungseock Joo, and M. Khalid Jawed. Deep learning of force manifolds from the simulated physics of robotic paper folding. *arXiv*, 2023.

[138] Mingrui Yu, Kangchen Lv, Hanzhong Zhong, Shiji Song, and Xiang Li. Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach. *IEEE Transactions on Robotics*, 2022.

[139] Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. *arXiv preprint arXiv:1905.04411*, 2019.

[140] Stephen P Timoshenko and James M Gere. *Theory of elastic stability*. Courier Corporation, 2009.

[141] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[142] Kui Pan, A Srikantha Phani, and Sheldon Green. Periodic folding of a falling viscoelastic sheet. *Physical Review E*, 101(1):013002, 2020.

[143] David Lattanzi and Gregory Miller. Review of robotic infrastructure inspection systems. *Journal of Infrastructure Systems*, 23(3):04017004, 2017.

[144] Richard H Crowell and Ralph Hartzler Fox. *Introduction to knot theory*, volume 57. Springer Science & Business Media, 2012.

[145] G Kirchhoff. J. reine angew. *Math.*, 70:289, 1869.

[146] Augustus Edward Hough Love. *A treatise on the mathematical theory of elasticity*. University press, 1927.

[147] Kirill Tarassov, Vincent Messier, Christian R Landry, Stevo Radinovic, Mercedes M Serna Molina, Igor Shames, Yelena Malitskaya, Jackie Vogel, Howard Bussey, and Stephen W Michnick. An in vivo map of the yeast protein interactome. *Science*, 320(5882):1465–1470, 2008.

[148] Arnaud Lazarus, Jay T Miller, Matthew M Metlitz, and Pedro M Reis. Contorting a heavy and naturally curved elastic rod. *Soft Matter*, 9(34):8274–8281, 2013.

[149] John Michael Tutill Thompson and AR Champneys. From helix to localized writhing in the torsional post-buckling of elastic rods. In *Localization and solitary waves in solid mechanics*, pages 111–132. World Scientific, 1999.

[150] Andy Borum and Timothy Bretl. When is a helix stable? *Phys. Rev. Lett.*, 125(8):088001, 2020.

[151] M Khalid Jawed, Alyssa Novelia, and Oliver M O'Reilly. *A primer on the kinematics of discrete elastic rods*. Springer, 2018.

[152] Dominik Henrich and Heinz Wörn. *Robot manipulation of deformable objects*. Springer Science & Business Media, 2012.

[153] Florent Lamiraux and Lydia E Kavraki. Planning paths for elastic objects under manipulation constraints. *Int. J. Rob. Res.*, 20(3):188–208, 2001.

[154] Mark Moll and Lydia E Kavraki. Path planning for deformable linear objects. *IEEE Trans. Robot.*, 22(4):625–636, 2006.

[155] Herbert G Tanner. Mobile manipulation of flexible objects under deformation constraints. *IEEE Trans. Robot.*, 22(1):179–184, 2006.

[156] Avishai Sintov, Steven Macenski, Andy Borum, and Timothy Bretl. Motion planning for dual-arm manipulation of elastic rods. *IEEE Robot. Autom. Lett.*, 5(4):6065–6072, 2020.

[157] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *Int. J. Rob. Res.*, 33(1):48–68, 2014.

[158] Simon Duenser, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros. Robocut: hot-wire cutting with robot-controlled flexible rods. *ACM Trans. Graph.*, 39(4):98–1, 2020.

[159] Louis B Kratchman, Trevor L Bruns, Jake J Abbott, and Robert J Webster. Guiding elastic rods with a robot-manipulated magnet for medical applications. *IEEE Trans. Robot.*, 33(1):227–233, 2016.

[160] Hunter B Gilbert, Richard J Hendrick, and Robert J Webster III. Elastic stability of concentric tube robots: A stability measure and design test. *IEEE Trans. Robot.*, 32(1):20–35, 2015.

[161] Shervin Javdani, Sameep Tandon, Jie Tang, James F O'Brien, and Pieter Abbeel. Modeling and perception of deformable one-dimensional objects. In *2011 IEEE International Conference on Robotics and Automation*, pages 1607–1614. IEEE, 2011.

[162] Russell C Jackson, Rick Yuan, Der-Lin Chow, Wyatt S Newman, and M Cenk Çavuşoğlu. Real-time visual tracking of dynamic surgical suture threads. *IEEE Trans. Autom. Sci. Eng.*, 15(3):1078–1090, 2017.

[163] Hirofumi Wada. Hierarchical helical order in the twisted growth of plant organs. *Phys. Rev. Lett.*, 109(12):128104, 2012.

[164] Raymond E Goldstein, Alain Goriely, Greg Huber, and Charles W Wolgemuth. Bistable helices. *Phys. Rev. Lett.*, 84(7):1631, 2000.

[165] Alexander V Vologodskii, Vadim V Anshelevich, Alexander V Lukashin, and Maxim D Frank-Kamenetskii. Statistical mechanics of supercoils and the torsional stiffness of the dna double helix. *Nature*, 280(5720):294–298, 1979.

[166] Ajeet Kumar and Timothy J Healey. A generalized computational approach to stability of static equilibria of nonlinearly elastic rods in the presence of constraints. *Comput. Methods. Appl. Mech. Eng.*, 199(25-28):1805–1815, 2010.

[167] Hernan J Logarzo, German Capuano, and Julian J Rimoli. Smart constitutive laws: Inelastic homogenization through machine learning. *Comput. Methods Appl. Mech. Eng.*, 373:113482, 2021.

[168] Shaun Edwards, R Madaan, and J Meyer. The descartes planning library for semi-constrained cartesian trajectories. *ROSCon 2015*, 2015.

[169] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Trans. Robot. Autom.*, 17(1-19):16, 2004.