Thrishantha Nanayakkara *Editor*

Handbook on Soft Robotics

Springer

Thrishantha Nanayakkara
Editor

Handbook on Soft Robotics

With Contributions by Cecilia Laschi, Barbara Mazzolai, Isuru Godage, Ian Walker, Manu Srivastava, Shinichi Hirai, Christian Duriez, Nicolas Rojas, Cynthia Sung, Jamie Paik

Springer

Editor
Thrishantha Nanayakkara
Imperial College London
London, UK

ISBN 978-3-031-68619-1 ISBN 978-3-031-68620-7 (eBook)
https://doi.org/10.1007/978-3-031-68620-7

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Chapter 4
Modeling Soft Robots

Isuru Godage and Hunter Gilbert

Abstract This chapter explores the evolving landscape of soft robotic kinematic and dynamic modeling, organized into four distinct sections to encompass a broad spectrum of methodologies. The first section investigate into Continuous Curvature Models, addressing the challenges associated with the inherently continuous and deformable nature of soft robots. Various approaches within continuum mechanics and finite element analysis are discussed, highlighting the complexities involved in capturing the intricate motion and shape changes exhibited by these systems. The second section focuses on Lumped Parametric Models, providing insights into techniques that discretize soft robots into simpler, interconnected elements. This section explores the advantages and limitations of such models, emphasizing their efficacy in simulating the dynamic behavior of soft robots with reduced computational complexity. The third section introduces Hybrid Models, which amalgamate the strengths of continuous curvature and lumped parametric models. This approach seeks to strike a balance between accuracy and computational efficiency, offering a versatile framework for modeling soft robotic systems in various applications. The fourth section explores Learning-Based Models, a burgeoning field leveraging machine learning and data-driven approaches to model the complex kinematics and dynamics of soft robots. The chapter provides an overview of neural networks, reinforcement learning, and other learning-based techniques, showcasing their potential in capturing intricate soft robotic behaviors and adapting to real-world scenarios. The chapter concludes by addressing the critical question of "How to Select Suitable Models" for soft robotic applications. It offers guidance on the criteria for model selection, taking into account factors such as system complexity, computational efficiency, and the availability of training data. By providing a comprehensive overview of these modeling approaches, this chapter aims to equip researchers, engineers, and practitioners with a nuanced understanding of the diverse methodologies available for soft robotic kinematic and dynamic modeling, paving the way for advancements in the field.

I. Godage (✉)
3367 TAMU, 466 Ross St, College Station, TX 77843, USA
e-mail: igodage@tamu.edu

H. Gilbert
3261 Patrick Taylor Hall, Lousiana State University, Baton Rouge, LA 70803, USA

T. Nanayakkara (ed.), *Handbook on Soft Robotics*,
https://doi.org/10.1007/978-3-031-68620-7_4

4.1 Introduction

In this chapter, we delve into the diverse landscape of soft and continuum robot modeling approaches, exploring four distinct paradigms that have shaped the field's evolution. We begin by unraveling the intricacies of continuous curvature models, which focus on capturing the gradual, seamless deformations of soft robots through mathematical representations of curvature and torsion. Next, we venture into the realm of lumped parametric models, where complex structures are approximated through discrete elements and simplified parameters, offering analytical insights into deformations and forces. Shifting gears, we delve into data-driven or learning models, which harness the power of machine learning and neural networks to predict the behavior of soft robots from empirical data, bridging the gap between theory and experimentation. Finally, we explore the synthesis of these modeling approaches in hybrid models, where analytical, data-driven, and empirical methods harmonize to provide a comprehensive understanding of soft robot dynamics and control.

Although there is no unanimous consensus on the precise definitions, the phrase "continuum robot" is generally used to indicate motion that occurs without identifiable kinematic pairs. Conversely, the term "soft robot" implies a greater degree of mechanical compliance, which is defined as the ratio of displacement to force. This compliance is more pronounced in soft robots than in traditional approaches to robotic interaction, as they are able to respond more effectively to environmental forces. Soft robots are typically composed of soft materials, which can be characterized by material parameters such as the modulus of elasticity. In contrast, continuum robots made of harder materials can be designed to exhibit high or low mechanical stiffness in response to external forces, depending on their specific design features.

Continuum robots are a type of flexible robotic manipulator composed of a long, flexible, and continuous structure, often modeled after biological organisms like elephant trunks or octopus tentacles, which can be bent, twisted, or elongated to perform a wide range of tasks in various environments. Unlike traditional rigid robots, continuum robots can adapt to their surroundings and conform to complex shapes, making them ideal for applications in areas such as medicine, manufacturing, and exploration. Some examples of continuum robots in the literature are shown in Fig. 4.1.

Often referred interchangeably, soft robots are a type of continuum robot that utilizes soft and flexible materials, such as elastomers, hydrogels, and fabrics, in their design and construction to achieve a range of functionalities and movements like those of biological organisms. In the literature, comparted to systems referred to as continuum robots, soft robots often refer to physical systems with lower stiffnesses. Soft robots can deform and change shape in response to external stimuli, such as changes in temperature or pressure, allowing them to interact with their environment in unique ways. Due to their flexibility and adaptability, soft robots have applications in fields such as healthcare, agriculture, and search and rescue, where they can perform delicate and complex tasks that are difficult for traditional rigid robots. Some examples of soft robots are shown in Fig. 4.2.

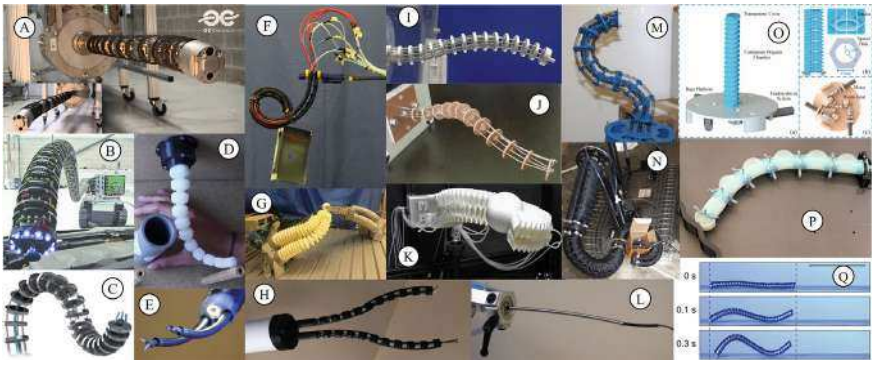

Fig. 4.1 Various Types of Continuum and Soft Robots. **a** OC Robotics—Series II, X125 snake-arm robot at UCL [3], **b** continuum manipulator [4], **c** tendon-driven continuum robot [15], **d** Tendon-actuated soft robot, **e** joystick-controlled concentric robot [29], **f** variable length multi-section pneumatic soft robot [9], **g** The Bionic Handling Assistant by Festo, showcasing a biomimetic design inspired by the elephant's trunk, **h** Concentric agonist-antagonist robot [25], **i** three-segment continuum robot [26], **j** continuum robot [31], **k** Honeycomb pneunets robot [37], **l** autonomous robotic catheter blazes trail [30], **m** discrete wire-driven continuum robot arm [41], **n** 'Octarm' continuum manipulator [23], **o** origami continuum robot [44], **p** 2D robotic manipulator [22], **q** magnetic soft robot [17]

Fig. 4.2 **a** Soft robotic glove [28], **b** 3D-printed soft robotic hand [32], **c** gecko-inspired soft robot [33], **d** soft elbow exosuit [38], **e** Meter-scale soft hexapod robot [18], **f** Soft biomimetic fish robot [35], **g** Soft tetrahedral robot [27], **h** Sorx: soft pneumatic hexapedal robot [19], **i** Electronics-free soft-legged robot [6], **j** Wheelless soft robotic snake [2], **k** Multigait soft robot [34], **l** pressure-operated soft robotic snake [20]

This chapter examine the current state-of-the-art in mathematical modeling of continuum manipulators that possess at least one "long" aspect in their shape. Such manipulators, also referred to as slender, are characterized by beam-like or bending

deformations that dominate their motion. The purpose of these models is to establish a relationship between the motion of the robot and the actuator variables, boundary conditions, and sensor measurements. However, they do not typically address other essential factors in robot design and analysis, such as repeatability, wear, and safety. Slender designs include "arms," "snakes," and individual "fingers" of a multi-fingered hand. Designs composed of individual components with this property, such as concentric tube robots or multi-backbone continuum robots, are natural extensions of this classification. Even for robots made of softer materials like the STIFF-FLOP designs, which exhibit localized deformations that may be complex, the dominant behavior remains beam-like. This chapter aims to discuss different modeling approaches at depth and present an assortment of methods reported in the literature using a common notation.

4.1.1 Preliminaries

Before embarking on the journey of understanding and exploring the modeling approaches for continuum and soft robots, it's essential to establish a foundational understanding of key terms and concepts that will pave the way for a deeper comprehension of the subject matter. This subsection serves as a crucial stepping stone, offering readers a concise yet comprehensive introduction to the preliminary terms that will frequently emerge throughout our exploration. These terms, ranging from elasticity and stress-strain relationships to kinematics and finite element analysis, provide the conceptual scaffolding upon which our discussions will rest. By familiarizing ourselves with this essential vocabulary, we equip ourselves with the tools necessary to navigate the intricate landscape of soft and continuum robot modeling with confidence and clarity.

Deformation: In soft and continuum robotics, deformation—often structural—refers to the changes in the shape and size of the robot's structure as it undergoes external forces or movements. This type of deformation is typically caused by the inherent compliance and flexibility of the materials used in soft and continuum robotics. Structural deformation can have a significant impact on the performance and behavior of soft and continuum robots. For example, in a soft gripper designed to grasp objects, the deformation of the gripper's structure can affect the force and grip strength applied to the object. Similarly, in a continuum robot designed to move through complex environments, the structural deformation of the robot's flexible structure can affect its ability to navigate and manipulate objects. To account for deformation in soft and continuum robotics, researchers often use appropriate models—often motivated by the robot design—that describe the robot's behavior as a function of its material properties and geometry. Understanding and modeling structural deformation is an essential aspect of soft and continuum robotics, as it enables researchers and engineers to design and control robots that can move and interact with their environment in a safe and effective manner.

Backbone Curve: The backbone curve in continuum robotics is a term used to describe the overall shape and geometry of a continuum robot [39]. It refers to the line that runs through the center of the robot's flexible structure and defines the robot's bending characteristics. The backbone curve is an important concept in continuum robotics because it determines the robot's motion and deformation properties. For example, the curvature of the backbone curve can affect the robot's ability to bend and twist, while the length and shape of the curve can influence the robot's reach and dexterity. Designing and modeling the backbone curve is a critical step in the development of continuum robots, as it requires careful consideration of factors such as the robot's intended application, the materials used in its construction, and the desired motion and deformation characteristics. A variety of techniques have been developed to model the backbone curve, including finite element analysis and optimization algorithms.

Framed Curve: The concept of framed curve in continuum robotics involves the attachment of a triad of reference vectors at each point along the curve. These vectors are used to describe the orientation and shape of the robot at that particular point, allowing for precise control over its motion and deformation. Typically, the three vectors attached to each point on the framed curve include one that is tangent to the curve itself, and two that span the cross sections of the robot at that point. This triad of vectors allows researchers and engineers to fully describe the orientation and shape of the robot at any given point, which is essential for controlling its motion and deformation. For instance, in the context of Homogeneous Transformation Matrices (HTM), the rotation matrix can be considered as a framed curve that serves to model deformation along the length of the robot. This framed curve helps track various factors such as bending, torsion, and other shear phenomena, ultimately contributing to a more accurate and comprehensive modeling approach.

Constant Curvature Shapes: Constant curvature modeling is a method used to model the kinematics of continuum robots [40]. This approach assumes that the robot's backbone curve has a constant curvature, meaning that the curve does not change its curvature along its length. Under the constant curvature model, the robot's motion and deformation can be described using a set of equations that relate the curvature of the backbone curve to the motion of the robot's end effector. This approach allows for precise control over the robot's motion, and it has been used in a wide range of applications, including medical robotics and industrial automation. However, the constant curvature model does have some limitations. In particular, it assumes that the robot's deformation is dominated by bending, rather than stretching or other types of deformation. This may not be the case for all types of continuum robots, particularly those made of very soft materials or those with complex geometries.

Finite Approximations: Finite approximations are a commonly used technique for modeling continuum robots [14]. Continuum robots are robots that use soft, flexible materials to achieve their motion, which makes them highly adaptable and able to move through complex, curved spaces. However, this flexibility also makes them challenging to model accurately using traditional mathematical methods. Finite approximations address this challenge by breaking the continuous motion of a continuum robot into a series of discrete segments. These segments are modeled using

finite element analysis, which is a numerical method for approximating solutions to partial differential equations. In practice, this means that the robot is divided into small sections, and the behavior of each section is modeled using a set of equations that describe its deformation and motion. These equations are solved numerically, and the solutions are then combined to provide a complete model of the robot's motion. Note that finite approximations are still an approximation, and there may be limitations to the accuracy of the model depending on the complexity of the robot

Configuration: Configuration refers to the positions that a mechanical system's parts can be in. Thus, configuration refers to the arrangement of all the robot's degrees of freedom (DOFs) that define its state. For example, in a simple planar robot arm, the configuration might include joint angles, while in a more complex robot, it could involve joint angles, translations, and orientations.

Configuration Space: Configuration space (C-space) is a mathematical space in which each point corresponds to a unique configuration of the robot. In a C-space, each dimension represents a different degree of freedom (DOF) of the robot. For instance, in a 3-DOF robot, the C-space would be three-dimensional. The entire C-space encompasses all possible combinations of joint values or robot states that the robot can achieve without violating any constraints. Also, the configuration space takes into account all kinematic, geometric, and other constraints that the robot must satisfy. For example, it considers joint limits, collision avoidance, and workspace boundaries. Constraints in C-space restrict the robot's motion to feasible and collision-free configurations.

Joint Space: In robot modeling, joint space refers to a specialized coordinate system used to describe the configuration or state of a robot. This coordinate system is particularly focused on representing the positions and orientations of the robot's individual joints. Each joint contributes to the overall configuration of the robot, and joint space provides a convenient way to define and control these configurations. In joint space, the parameters typically include joint angles, joint velocities, and possibly other joint-specific parameters, depending on the robot's design and complexity. By defining the robot's state in joint space, it becomes easier to plan and execute robot movements, perform kinematic and dynamic analyses, and develop control algorithms. Joint space representations are particularly useful in robotics for tasks such as path planning, inverse kinematics, and trajectory control, as they allow engineers and researchers to work directly with the robot's articulation.

Task Space: Task space provides a vital perspective for describing the actions and behaviors of robots. It represents a higher-level coordinate system focused on the position and orientation of a robot's end-effector, such as its gripper or tool, in relation to a reference frame. Task space simplifies the planning and control of robotic tasks by specifying where the end-effector should be and how it should be oriented to achieve particular objectives. This approach allows engineers and programmers to design robots that interact effectively with their environment, from picking and placing objects to performing complex assembly tasks. Solving for the joint angles necessary to achieve desired end-effector poses in task space, known as inverse kinematics, plays a central role in robot modeling and control.

Generalized Coordinates: Generalized coordinates are a vital concept in the realm of mechanics and dynamics. They represent a set of parameters that comprehensively describe the configuration of a system. These coordinates are not merely numerical values but serve as the coordinates of a singular point within an abstract space known as "configuration space." Generalized coordinates can take two primary forms: absolute and relative. Absolute coordinates are referenced with respect to an unchanging inertial frame, providing a fixed point of reference. In contrast, relative coordinates are defined concerning a co-moving frame, which moves in tandem with the system under consideration. This distinction in reference frames allows for a versatile and adaptable means of characterizing the configuration and motion of dynamic systems.

4.2 Continuous Curvature Models

Continuous curvature approaches are mathematical techniques used to model the shape and behavior of soft and continuum robots. These robots are made of flexible materials, allowing them to move and deform continuously rather than in discrete steps, which is characteristic of traditional rigid robots. Continuous curvature approaches aim to capture and describe the complex and continuous deformations that soft and continuum robots can achieve.

4.2.1 Curve Parametric Models

Curve parametric (CP) modeling is a fundamental approach that revolves around the utilization of mathematical curves and parameters to describe the form and actions of soft robots. One of the notable strengths of curve parametric modeling in the realm of soft robotics is its capacity to accurately capture the intricate deformations and motions exhibited by these robots. In addition, curve parametric approaches offer numerous other advantages in soft robot modeling. Firstly, they excel in accurately representing the intricate and nonlinear deformations of soft robots, even when subjected to substantial loads and constraints. This precision arises from their capability to capture the spatial distribution of material properties and interactions within the robot's structure. Secondly, these approaches exhibit computational efficiency, proving effective for complex soft robot geometries. Curve parametric models are flexible and adaptable, accommodating a wide range of soft robot geometries, including continuum, articulated, hybrid, and surface robots. In this section, we will utilize the pneuamtically actuated multisection continuum arm shown in Fig. 4.3.

Consider the schematic of any continuum arm shown in Fig. 4.4. Without losing generality, the three variable length actuator configuration is considered and the actuators are fixed to a circular rigid frame at a radius, r from the center and $\frac{2\pi}{3}\ rads$ apart. Hence, the actuators are operated at a distance r, aligned with the neutral

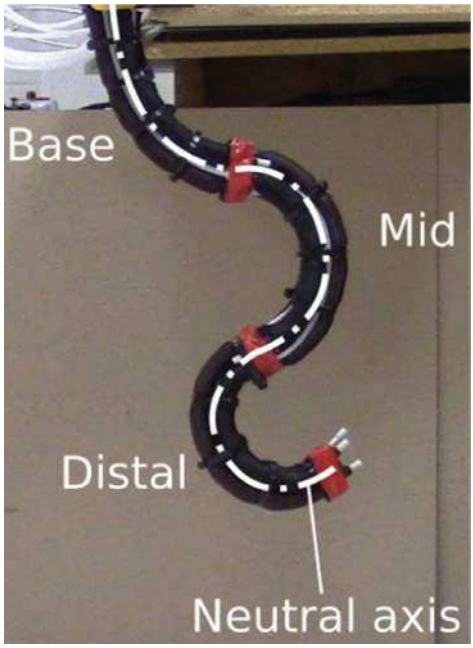

Fig. 4.3 Multisection Continuum Arm with Pneumatic Muscle Actuators [8]. The arm consists of three serially attached continuum sections, each actuated by three extending-mode numeric muscle actuators symmetrically attached. Differential pressures in the pneumatic muscle actuators create circular arc shapes in the sections, while common-mode pressure causes extension. Note that some prototypes may use more than three actuators, such as four, and may incorporate different types of actuators that can extend, compress, or both during operation

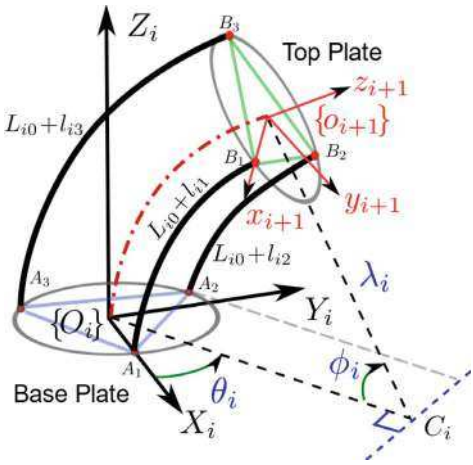

Fig. 4.4 Joint-space of a continuum/soft robotic section

axis, which is an imaginary line running from the center along the length of the continuum robot. Let the initial length of each actuator is L whose change is in $l_i \in \mathbb{R}$ where $l_{i:\min} \leq l_i(t) \leq l_{i:\max}$ for $i \in \{1, 2, 3\}$; i and t denote the actuator number and time, respectively. Therefore, $L_i(t) = L + l_i(t)$ calculates the actuator length at any time and the vector of joint variables of the continuum robot is defined as $\boldsymbol{q} = [l_1(t),\ l_2(t),\ l_3(t)]^T \in \mathbb{R}^3$.

Due to the constrained actuator arrangement, when the robot is actuated, it either demonstrates straightforward linear motions (expansion or contraction) or curves into a circular arc. Thus, assuming there are no substantial external forces, the spatial alignment of the robot can be entirely defined as a circular arc with variable curvature radius and length. The arc is defined by three spatial parameters; radius of curvature $\lambda \in (0, \infty)$ with instantaneous center C, angle subtended by the bending arc $\phi \in [0, 2\pi]$, and angle of the bending plane with respect to the $+X$ axis, $\theta \in [-\pi, \pi]$.

4.2.1.1 Deriving Curve Parameters in Joint Space Variables

Let the origin of the task-space coordinate frame $\{O_i\}$ coincide with the center of the base plate, where $\overrightarrow{O_i A_1}$ defines the X_i axis. The actuator attachment points form an equilateral triangle with sides of length $r_i\sqrt{3}$ at each end of the continuum section. The coordinates of the actuator attachment points are $A_1 = [r_i, 0, 0]^T$, $A_2 = \frac{r_i}{2}[-1, \sqrt{3}, 0]^T$, and $A_3 = -\frac{r_i}{2}[1, \sqrt{3}, 0]^T$. The instantaneous center of the bent arm's circular arc shape is represented as C_i. In Fig. 4.5, you can observe the actuator base points, A_1, A_2, and A_3, projected onto $\overrightarrow{O_i C_i}$, where they intersect at x_1^*, x_2^*, and x_3^*. The respective distances between O_i and these intersection points are

$$\begin{aligned} O_i x_1^* &= r_i \cos\theta_i \\ O_i x_2^* &= r_i \cos\left(\frac{2\pi}{3} - \theta_i\right) \\ O_i x_3^* &= r_i \cos\left(\frac{4\pi}{3} - \theta_i\right) \end{aligned} \tag{4.1}$$

The actuator lengths form radii of three concentric circular arcs at $O_i C_i$ (see Fig. 4.5). Employing the arc geometrical relationship where *arc length* is equal to the *curvature radius* times the *subtended angle*, the actuator lengths are related to curve parameters as follows:

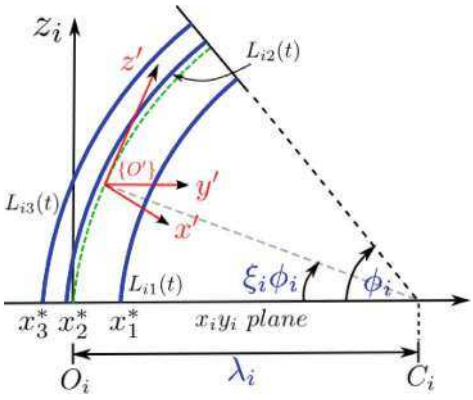

Fig. 4.5 Orthographic projection of the moving coordinate frame, $\{O'\}$ along the neutral axis and the normalizing scalar, ξ_i shown on the bending plane (i.e., $z_i O_i C_i$ plane)

$$\begin{aligned} L_{i0} + l_{i1} &= \left(\lambda_i - O_i x_1^*\right) \phi_i \\ &= \{\lambda_i - r_i \cos(\theta_i)\} \phi_i \end{aligned} \tag{4.2}$$

$$\begin{aligned} L_{i0} + l_{i1} &= (\lambda_i - O_i x_2^*) \phi_i \\ &= \left(\lambda_i + \frac{1}{2} r_i \cos\theta_i - \frac{\sqrt{3}}{2} r_i \sin\theta_i\right) \phi_i \end{aligned} \tag{4.3}$$

$$\begin{aligned} L_{i0} + l_{i1} &= (\lambda_i - O_i x_3^*) \phi_i \\ &= \left(\lambda_i + \frac{1}{2} r_i \cos\theta_i + \frac{\sqrt{3}}{2} r_i \sin\theta_i\right) \phi_i \end{aligned} \tag{4.4}$$

These relationships are now manipulated to derive curve parameters in joint space variables. Summing up Eqs. (4.2), (4.3), and (4.4) yields

$$\begin{aligned} 3\lambda_i \phi_i &= 3L_{i0} + l_{i1} + l_{i2} + l_{i3} \\ \phi_i &= \frac{1}{3\lambda_i} (3L_{i0} + l_{i1} + l_{i2} + l_{i3}) \end{aligned} \tag{4.5}$$

Subtracting Eq. (4.3) from Eq. (4.4) and rearranging the terms produces

$$\begin{aligned} l_{i3} - l_{i2} &= \left(\sqrt{3} r_i \sin\theta_i\right) \phi_i \\ \sin\theta_i &= \frac{l_{i3} - l_{i2}}{\sqrt{3} r_i \phi_i} \end{aligned} \tag{4.6}$$

Similarly, rearranging Eq. (4.2) gives

$$\cos\theta_i = \frac{\lambda_i \phi_i - (L_{i0} + l_{i1})}{r_i \phi_i} \tag{4.7}$$

Applying Eqs. (4.6) and (4.7) to the trigonometric identity $\sin^2\theta_i + \cos^2\theta_i = 1$, to remove θ_i from the relationship, results in

$$\left(\frac{l_{i3} - l_{i2}}{\sqrt{3} r_i \phi_i}\right)^2 + \left(\frac{\lambda_i \phi_i - (L_{i0} + l_{i1})}{r_i \phi_i}\right)^2 = 1 \tag{4.8}$$

Substituting ϕ_i from Eq. (4.5) into Eq. (4.8) and solving for $\lambda_i \in \mathbb{R}^+$ gives

$$\lambda_i\left(\boldsymbol{q}_i\right) = \frac{(3L_{i0} + l_{i1} + l_{i2} + l_{i3})\, r_i}{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}} \tag{4.9}$$

The result given by Eq. (4.12) is then substituted into Eq. (4.5) to solve ϕ_i as

$$\phi_i\left(\boldsymbol{q}_i\right) = \frac{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{3r_i} \tag{4.10}$$

Dividing Eqs. (4.6) by (4.7) yields θ_i as

$$\theta_i\left(\boldsymbol{q}_i\right) = \tan^{-1}\left(\frac{\sqrt{3}\left(l_{i3} - l_{i2}\right)}{l_{i2} + l_{i3} - 2l_{i1}}\right) \tag{4.11}$$

Substituting ϕ_i from Eq. (4.5) into Eq. (4.8) and solving for $\lambda_i \in \mathbb{R}^+$ gives

$$\lambda_i\left(\boldsymbol{q}_i\right) = \frac{\left(3L_{i0} + l_{i1} + l_{i2} + l_{i3}\right) r_i}{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}} \tag{4.12}$$

The result given by Eq. (4.12) is then substituted into Eq. (4.5) to solve ϕ_i as

$$\phi_i\left(\boldsymbol{q}_i\right) = \frac{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{3r_i} \tag{4.13}$$

Dividing Eqs. (4.6) by (4.7) yields θ_i as

$$\theta_i\left(\boldsymbol{q}_i\right) = \tan^{-1}\left(\frac{\sqrt{3}\left(l_{i3} - l_{i2}\right)}{l_{i2} + l_{i3} - 2l_{i1}}\right) \tag{4.14}$$

4.2.1.2 Deriving the Homogeneous Transformation Matrix (HTM) for a Single Continuum Section

For complete kinematic modeling, it is essential to accurately calculate both the position and orientation of all joints and links in a robotic system. Due to the substantial inherent mechanical flexibility, continuum sections exhibit varying orientations along their length, necessitating the representation of this variation using a continuous homogeneous transformation matrix (HTM). We define a moving coordinate frame denoted as $\{O'\}$ is established (refer to Fig. 4.4), and a scalar parameter $\xi \in [0, 1]$ is introduced. This parameter allows for the movement of $\{O'\}$ from the base ($\xi = 0$) to the tip ($\xi = 1$) along the neutral axis of the continuum section. Accordingly, homogeneous translational and rotational transformations based on curve parameters can be derived as

$$\begin{aligned}\mathbf{T}(\boldsymbol{q},\xi) &= \mathbf{R}_Z(\theta)\mathbf{P}_X(\lambda)\mathbf{R}_Y(\phi)\mathbf{P}_X(-\lambda)\mathbf{R}_Z(-\theta) \\ &= \begin{bmatrix} \mathbf{R}(\boldsymbol{q},\xi) & \boldsymbol{p}(\boldsymbol{q},\xi) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}\end{aligned} \tag{4.15}$$

where $\mathbf{R}_Z$ and $\mathbf{R}_Y$ are homogeneous rotation matrices about $+Z$ and $+Y$ axes and $\mathbf{P}_X$, is the homogeneous translation matrix along $+X$ axis. Additionally, $\boldsymbol{q} = [\lambda,\ \phi,\ \theta]^T \in \mathbb{R}^3$ is the curve parameter vector, $\mathbf{R} \in SO(3)$, $\boldsymbol{p} = [x,\ y,\ z]^T \in \mathbb{R}^3$ are the rotational and translational matrices of the robot.

4.2.1.3 Constant-Length System Modeling

Noting that the bending of the soft module is critical to the robot locomotion, understanding the relationship between the PMA lengths and the curve parameters is important as controlling the PMA lengths enables the control of module shape and robot locomotion. The PMA lengths can be related to the curve parameters as given in [8] as (Fig. 4.6).

$$\begin{aligned} L + l_{ji} &= \left\{ \frac{L}{\phi_j} - r\cos\left(\frac{2\pi}{3}(i-1) - \theta_j\right)\right\}\phi_j \\ l_{ji} &= -r_j\phi_j\cos\left(\frac{2\pi}{3}(i-1) - \theta_j\right) \end{aligned} \tag{4.16}$$

Note that the inextensibility of the soft module implies that the sum of PMA length changes becomes zero, i.e., $\sum_i l_{ji} = 0$. This kinematic constraint gives rise to a relationship between the three joint variables, i.e., $l_{j1} = -\left(l_{j2} + l_{j3}\right)$, which implies that the soft module forward kinematics can be obtained using just two degrees of freedom. Employing Eq. (4.16), we can derive the curve parameters in terms of the joint variables as:

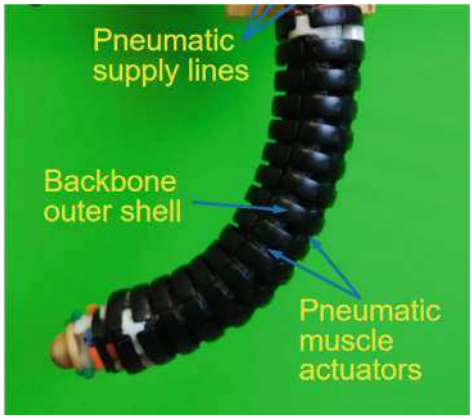

Fig. 4.6 A hybrid continuum robotic arm section constructed with a rigid inextensible backbone forming a kinematic chain, symmetrically actuated by numeric muscle actuators

$$\begin{aligned}\phi_j &= \frac{2}{3r}\sqrt{\sum_{i=1}^{3}\left(l_{ji}^2 - l_{ji}l_{j\,\mathrm{mod}(i,3)} + 1\right)} \\ \theta_j &= \arctan\left\{\sqrt{3}\left(l_{j3} - l_{j2}\right), l_{j2} + l_{j3} - 2l_{j1}\right\}\end{aligned} \tag{4.17}$$

4.2.1.4 Numerically Stable Modal Representation

Numerical instability can arise in the curve parametric approach for soft robot modeling under certain conditions. For instance, in cases where the robot's configuration leads to singularities, such as when all actuator lengths are nearly equal, mathematical operations involving division by small or zero values can result in numerical instability. Consider the following matrix element of Eq. (4.15).

$$[\mathbf{T}(\boldsymbol{q},\xi)]_{24} = \frac{\sqrt{3}\left(l_2 - l_3\right)\left(3L_0 + l_1 + l_2 + l_3\right)r}{4\left(l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3\right)}\left\{\cos\left(2\xi\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3}/3r\right) - 1\right\} \tag{4.18}$$

The modal approach for soft robot kinematics offers a promising avenue to avoid singularities and enhance the stability of robot modeling decomposing the robot's deformation into a set of modal or basis functions. These modal functions capture the fundamental shape and motion patterns of the robot, allowing for a more efficient and robust representation of its behavior. By expressing the robot's configuration as a linear combination of these modes, singularities associated with specific parameter values can be mitigated or even eliminated. However, in order to retain the physical insight and avoid nonlinear mapping problems, modal forms are preferred to retain joint space representation. A simple and straightforward method, such as using multivariate Taylor series approximation, has been presented in [8] for finding suitable modal functions. This approach allows us to derive modal forms of kinematics that effectively capture the complex deformations while mitigating singularities, making it a valuable tool for modeling and controlling soft robotic systems (Fig. 4.7).

4.2.1.5 Recursive Formulation of Complete Kinematics

A multisection continuum or soft robotic arm, a schematic is shown in Fig. 4.8, can be derived using the HTM derived for a single section in Sect. 4.2.1.2. Employing the continuum section HTM given in Eq. (4.15) and principles of kinematics of serial robot chains, the HTM of any i^{th} section with respect to the task-space coordinate system $\{O\}$, $\mathbf{T}^i : \left(\boldsymbol{q}^i, \xi_i\right) \mapsto \mathbb{SE}^3$, is given by

$$\mathbf{T}^i = \prod_{k=1}^{i}\mathbf{T}_i = \begin{bmatrix}\mathbf{R}^i & \boldsymbol{p}^i \\ \mathbf{0} & 1\end{bmatrix} \tag{4.19}$$

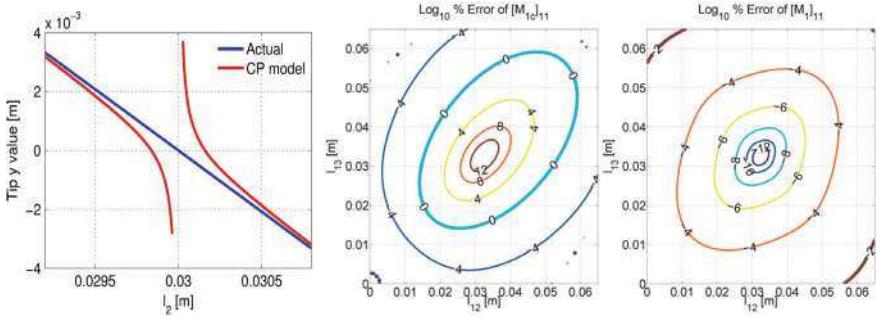

Fig. 4.7 (LEFT) Errors of $[\mathbf{T}]_{24}$, given by Eq. (4.18), within singular neighborhood against l_2 when $l_2 \to 0.03$ where $l_1 = l_3 = 0.03$. The error spans within a singularity neighborhood thus eliminating the possibility of conditional HTM's to counter the numerical instabilities. Comparison of numerical errors in the generalized inertia matrix for the base continuum section. (MIDDLE) Note the large errors (103%) towards the singularity at $l_{11} = 0.0325$ and its (expanded to around $\{l_{12}, l_{13}\} \in [0.02, 0.05]$ neighborhood. (RIGHT) In contrast to Fig. 4.15a, the error is negligible ($\leq 0.014\%$) within the entire actuation region. During both simulations, $l_{11} = 0.0325$

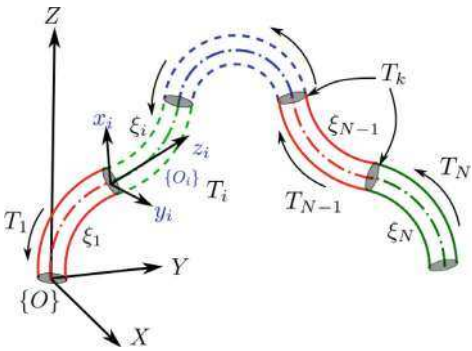

Fig. 4.8 Schematic of a general multisection continuum arm with N continuum sections

where $\mathbf{R}^i\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{p}_i\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^3$ define the position and orientation of $\left\{O_i'\right\}$ along the neutral axis at ξ_i of the i^{th} continuum section.

The homogeneous transformation matrix in Eq. (4.19) can be expanded to obtain the recursive form of the kinematics as

$$\begin{aligned} \mathbf{R}^i &= \mathbf{R}^{i-1}\mathbf{R}_i \\ \boldsymbol{p}^i &= \boldsymbol{p}^{i-1} + \mathbf{R}^{i-1}\boldsymbol{p}_i \end{aligned} \tag{4.20}$$

where $\mathbf{R}^{i-1}\left(\boldsymbol{q}^{i-1}\right) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{p}_i\left(\boldsymbol{q}^{i-1}\right) \in \mathbb{R}^3$ is the section tip rotation matrix and position vector of the preceding continuum section.

Utilizing the Eq. (4.20), the angular velocity of a thin disc at ξ_i with respect to $\left\{O_i'\right\}$, $\boldsymbol{\omega}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^3$ can be defined as

$$\begin{aligned}\boldsymbol{\Omega}_i &= \mathbf{R}^{i^T}\dot{\mathbf{R}}^i \\ &= \left(\mathbf{R}^{i-1}\mathbf{R}_i\right)^T\left(\dot{\mathbf{R}}^{i-1}\mathbf{R}_i + \mathbf{R}^{i-1}\dot{\mathbf{R}}_i\right) \\ &= \mathbf{R}_i\left\{\left(\mathbf{R}^{i-1^T}\dot{\mathbf{R}}^{i-1}\right)\mathbf{R}_i + \left(\mathbf{R}^{i-1^T}\mathbf{R}^{i-1}\right)\dot{\mathbf{R}}_i\right\} \\ &= \mathbf{R}_i^T\left(\boldsymbol{\Omega}_{i-1}\mathbf{R}_i + \dot{\mathbf{R}}_i\right)\end{aligned} \tag{4.21}$$

where $\boldsymbol{\omega_i} = \begin{bmatrix}\omega_{ix} & \omega_{iy} & \omega_{iz}\end{bmatrix}^T$ and $\boldsymbol{\Omega}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^{3\times 3} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0\end{bmatrix}$.

Similarly, Eq. (4.20) can be used to derive the linear body velocity of a thin disc at ξ_i with respect to $\left\{O_i'\right\}$, $\boldsymbol{v}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^3$ as

$$\begin{aligned}\boldsymbol{v}_i &= \mathbf{R}^{i^T}\dot{\boldsymbol{p}}^i \\ &= \left(\mathbf{R}^{i-1}\mathbf{R}_i\right)^T\left(\dot{\boldsymbol{p}}^{i-1} + \dot{\mathbf{R}}^{i-1}\mathbf{R}_i + \mathbf{R}^{i-1}\dot{\boldsymbol{p}}_i\right) \\ &= \mathbf{R}_i^T\left\{\left(\mathbf{R}^{i-1^T}\dot{\boldsymbol{p}}^{i-1}\right)\mathbf{R}_i + \left(\mathbf{R}^{i-1^T}\dot{\mathbf{R}}^{i-1}\right)\boldsymbol{p}_i + \left(\mathbf{R}^{i-1^T}\mathbf{R}^{i-1}\right)\dot{\boldsymbol{p}}_i\right\} \\ &= \mathbf{R}_i^T\left(\boldsymbol{v}_{i-1} + \boldsymbol{\Omega}_{i-1}\boldsymbol{p}_i + \dot{\boldsymbol{p}}_i\right)\end{aligned} \tag{4.22}$$

As shown in [9], Jacobians and Hessians play a critical role in recursive development of the EoM. Applying the standard techniques, the angular and linear velocity Jacobians, $\mathbf{J}_i^{\omega}\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{3\times 3n}$ and $\mathbf{J}_i^{\upsilon}\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{3\times 3n}$ respectively are derived. Here also, we use the property $\boldsymbol{\omega}_i = \boldsymbol{\Omega}_i^{\vee}$ to define $\mathbf{J}_i^{\Omega}\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{3\times 9n}$, as

$$\begin{aligned}\mathbf{J}_i^{\Omega} &= \boldsymbol{\Omega}_{i,\left(\dot{\boldsymbol{q}}^i\right)^T} \\ &= \mathbf{R}_i^T\left(\boldsymbol{\Omega}_{i-1}\mathbf{R}_i + \dot{\mathbf{R}}_i\right)_{,\left(\dot{\boldsymbol{q}}^i\right)^T} \\ &= \mathbf{R}_i^T\left[\boldsymbol{\Omega}_{i-1,\left(\dot{\boldsymbol{q}}^{i-1}\right)^T}\mathbf{R}_i \middle| \dot{\mathbf{R}}_{i,\dot{\boldsymbol{q}}_i^T}\right] \\ &= \mathbf{R}_i^T\left[\mathbf{J}_{i-1}^{\Omega}\mathbf{R}_i \middle| \mathbf{R}_{i,\boldsymbol{q}_i}\right]\end{aligned} \tag{4.23}$$

where $\mathbf{J}_i^{\omega} = \left(\mathbf{J}_i^{\Omega}\right)^{\vee}$ and $\mathbf{J}_{i-1}^{\Omega}\left(\boldsymbol{q}^{i-1}\right) \in \mathbb{R}^{3\times 9(n-1)}$.

Taking the partial derivative of Eq. (4.23) with respect to $\boldsymbol{q}^i$, the angular body velocity Hessian, $\mathbf{H}_i^{\Omega} = \mathbf{J}_{i,\boldsymbol{q}_i}^{\Omega}\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{9n\times 9n}$ is given by.

$$\begin{aligned}\mathbf{H}_i^{\Omega} &= \mathbf{J}_{i,\boldsymbol{q}^i}^{\Omega} \\ &= \left(\mathbf{R}_i^T\left[\mathbf{J}_{i-1}^{\Omega}\mathbf{R}_i \;\; \mathbf{R}_{i,\boldsymbol{q}_i^T}\right]\right)_{,\boldsymbol{q}^i} \\ &= \left[\begin{array}{c|c} \mathbf{R}_i^T\left(\mathbf{J}_{i-1,\boldsymbol{q}^{i-1}}^{\Omega}\right)\mathbf{R}_i & \mathbf{R}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}^{i-1}} \\ \hline \mathbf{R}_{i,\boldsymbol{q}_i}^T\mathbf{J}_{i-1}^{\Omega}\mathbf{R}_i\cdots & \mathbf{R}_{i,\boldsymbol{q}_i}^T\mathbf{R}_{i,\boldsymbol{q}_i^T}\cdots \\ +\mathbf{R}_i^T\mathbf{J}_{i-1}^{\Omega}\mathbf{R}_{i,\boldsymbol{q}_i} & +\mathbf{R}_i^T\mathbf{R}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i}\end{array}\right]\end{aligned}$$

$$= \left[\begin{array}{c|c} \mathbf{R}_i^T \mathbf{H}_{i-1}^{\Omega} \mathbf{R}_i & \mathbf{0} \\ \hline \mathbf{R}_{i,\boldsymbol{q}_i}^T \mathbf{J}_{i-1}^{\Omega} \mathbf{R}_i \cdots & \mathbf{R}_{i,\boldsymbol{q}_i}^T \mathbf{R}_{i,\boldsymbol{q}_i^T} \cdots \\ +\mathbf{R}_i^T \mathbf{J}_{i-1}^{\Omega} \mathbf{R}_{i,\boldsymbol{q}_i} & +\mathbf{R}_i^T \mathbf{R}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i} \end{array}\right] \tag{4.24}$$

where $\mathbf{J}_i^{\omega} = \left(\mathbf{J}_i^{\Omega}\right)^{\vee}$ and $\mathbf{J}_{i-1}^{\Omega}\left(\boldsymbol{q}^{i-1}\right) \in \mathbb{R}^{3\times 9(n-1)}$.

Similarly, the linear velocity Jacobian, $\mathbf{J}_i^{\upsilon}$, and Hessian, $\mathbf{H}_i^{\upsilon} = \mathbf{J}_{i,\boldsymbol{q}_i}^{\upsilon}\left(\boldsymbol{q}^i, \xi_i\right) \in \mathbb{R}^{9n\times 3n}$ are given by

$$\begin{aligned} \mathbf{J}_i^{\upsilon} &= \boldsymbol{\upsilon}_{i,(\dot{\boldsymbol{q}}^i)^T} \\ &= \mathbf{R}_i^T \left(\boldsymbol{\upsilon}_{i-1} + \boldsymbol{\Omega}_{i-1}\boldsymbol{p}_i + \dot{\boldsymbol{p}}_i\right)_{,(\dot{\boldsymbol{q}}^i)^T} \\ &= \mathbf{R}_i^T \left[\begin{array}{c|c} \boldsymbol{\upsilon}_{i-1,(\dot{\boldsymbol{q}}^{i-1})^T} + \boldsymbol{\Omega}_{i-1,(\dot{\boldsymbol{q}}^{i-1})^T}\boldsymbol{p}_i & \dot{\boldsymbol{p}}_{i,\dot{\boldsymbol{q}}_i^T} \end{array}\right] \\ &= \mathbf{R}_i^T \left[\begin{array}{c|c} \mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_i & \boldsymbol{p}_{i,\boldsymbol{q}_i^T} \end{array}\right] \end{aligned} \tag{4.25}$$

$$\begin{aligned} \mathbf{H}_i^{\upsilon} &= \mathbf{J}_{i,\boldsymbol{q}^i}^{\upsilon} \\ &= \left(\mathbf{R}_i^T \left[\begin{array}{c|c} \mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_i & \boldsymbol{p}_i \end{array}\right]\right)_{,\boldsymbol{q}^i} \\ &= \left[\begin{array}{c|c} \mathbf{R}_i^T \left(\mathbf{J}_{i-1,\boldsymbol{q}^{i-1}}^{\upsilon} + \mathbf{J}_{i-1,\boldsymbol{q}^{i-1}}^{\Omega}\boldsymbol{p}_i\right) & \left(\mathbf{R}_i^T \boldsymbol{p}_{i,\boldsymbol{q}_i^T}\right)_{,\boldsymbol{q}^{i-1}} \\ \hline \mathbf{R}_{i,\boldsymbol{q}_i}^T \left(\mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_i\right)\cdots & \mathbf{R}_{i,\boldsymbol{q}_i}^T \boldsymbol{p}_{i,\boldsymbol{q}_i^T}\cdots \\ +\mathbf{R}_i^T \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_{i,\boldsymbol{q}_i} & +\mathbf{R}_i^T \boldsymbol{p}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i} \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{R}_i^T \left(\mathbf{H}_{i-1}^{\upsilon} + \mathbf{H}_{i-1}^{\Omega}\boldsymbol{p}_i\right) & \mathbf{0} \\ \hline \mathbf{R}_{i,\boldsymbol{q}_i}^T \left(\mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_i\right)\cdots & \mathbf{R}_{i,\boldsymbol{q}_i}^T \boldsymbol{p}_{i,\boldsymbol{q}_i^T}\cdots \\ +\mathbf{R}_i^T \mathbf{J}_{i-1}^{\Omega}\boldsymbol{p}_{i,\boldsymbol{q}_i} & +\mathbf{R}_i^T \boldsymbol{p}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i} \end{array}\right] \end{aligned} \tag{4.26}$$

where $\mathbf{J}_{i-1}^{\upsilon}\left(\boldsymbol{q}^{i-1}, \xi_i\right) \in \mathbb{R}^{3\times 3(n-1)}$, $\mathbf{H}_{i-1}^{\upsilon}\left(\boldsymbol{q}^{i-1}\right) \in \mathbb{R}^{9(n-1)\times 3(n-1)}$.

4.2.1.6 Dual Quaternion Representation

Robotic systems often require accurate modeling and representation of their configurations to perform tasks efficiently. Dual quaternions are an extension of standard quaternions, a mathematical tool originally developed for spatial rotations. Dual quaternions provide a compact and efficient way to represent both the position and orientation of coordinate frames defining the pose of robotic elements.

Dual quaternions offer a valuable approach when dealing with the kinematics of robotic systems, especially in scenarios where stability in inverse kinematic solutions is crucial. This significance becomes particularly pronounced in the context of soft robotic arms. Soft robotic arms, owing to their inherent redundancy, often encounter numerical challenges when solving constrained inverse kinematics problems. It has

been demonstrated that employing dual quaternion-based kinematics can significantly enhance stability and accuracy in such situations. Therefore, understanding how to effectively utilize dual quaternions is a valuable asset for researchers and practitioners in the field.

Dual quaternions are an extension of dual numbers into the realm of quaternions. A dual quaternion is typically denoted as $Q = s + \varepsilon t$, where $s = s_0 + s_x i + s_y j + s_z k$ and $t = 0 + t_x i + t_y j + t_z k$ are standard quaternions. s represents the orientation of a rigid body, while εt captures its translation or position.

Mathematical Basis of Dual Quaternions

In dual quaternion addition, you add two dual quaternions component-wise. Each component of the resulting dual quaternion is the sum of the corresponding components of the two input dual quaternions. Mathematically, if you have two dual quaternions

$$D_1 = s_1 + \varepsilon t_1 \tag{4.27}$$

$$D_2 = s_2 + \varepsilon t_2 \tag{4.28}$$

The dual quaternion algebra follows specific rules for addition, multiplication, and conjugation, making it a closed and algebraically consistent system as follows.

$$\textbf{1. Addition}: \quad D_1 + D_2 = (s_1 + s_2) + \varepsilon(t_1 + t_2) \tag{4.29}$$

$$\textbf{2. Multiplication}: \quad D_1 \odot D_2 = (s_1 \circ s_2) + \varepsilon(s_1 \circ t_2 + t_1 \circ s_2) \tag{4.30}$$

$$\textbf{3. Conjugation}: \quad D^* = s - \varepsilon t \tag{4.31}$$

where $\odot$ is the dual quaternion multiplication, $\cdot$ is the quaternion multiplication, and $*$ is the quaternion conjugate.

Dual quaternions and homogeneous transformation matrices

A unique mapping exists between Homogeneous Transformation Matrices and Dual Quaternions. Understanding one representation enables the derivation of the other. This is particularly significant because deriving the Homogeneous Transformation is often more intuitive for robotic systems. Subsequently, this knowledge facilitates the derivation of the dual coordinate form of the representation, which, in turn, serves as the foundation for deriving the complete system kinematics using Dual Quaternions.

Consider the given homogeneous transformation $\mathbf{T} \in \mathbb{SE}(3)$, represented as follows.

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \boldsymbol{p} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{SE}(3) \tag{4.32}$$

Then, you can express the dual quaternion Q as $Q = s + \varepsilon t$ as follows.

$$\begin{aligned} s_0 &= \frac{1}{2}\sqrt{[\mathbf{R}]_{11} + [\mathbf{R}]_{22} + [\mathbf{R}]_{33} + 1} \\ s_x &= \frac{1}{4}\left([\mathbf{R}]_{32} - [\mathbf{R}]_{23}\right) \\ s_y &= \frac{1}{4}\left([\mathbf{R}]_{13} - [\mathbf{R}]_{31}\right) \\ s_z &= \frac{1}{4}\left([\mathbf{R}]_{21} - [\mathbf{R}]_{12}\right) \end{aligned} \tag{4.33}$$

$$\boldsymbol{t} = \frac{1}{2}\langle 0, \boldsymbol{p}\rangle \circ \boldsymbol{s} \tag{4.34}$$

Similarly, if you have knowledge of a dual quaternion, denoted as $Q = s + \varepsilon t$, you can reconstruct the Homogeneous Transformation Matrix (HTM), $T \in \mathbb{SE}(3)$, as follows

$$\mathbf{R} = \begin{bmatrix} 1 - 2(s_2^2 + s_3^2) & 2(s_1 s_2 - s_0 s_3) & 2(s_0 s_2 + s_1 s_3) \\ 2(s_1 s_2 + s_0 s_3) & 1 - 2(s_1^2 + s_3^2) & 2(s_2 s_3 - s_0 s_1) \\ 2(s_1 s_3 - s_0 s_2) & 2(s_0 s_1 + s_2 s_3) & 1 - 2(s_1^2 + s_2^2) \end{bmatrix}, \qquad \boldsymbol{p} = \begin{bmatrix} 2t_x \\ 2t_y \\ 2t_z \end{bmatrix} \tag{4.35}$$

Coordinate transformation in Dual Quaternion Systems

When it comes to coordinate transformations, the traditional approach involves multiplying transformation matrices, denoted as $T_1 \cdot T_2 \cdot T_3 \ldots \cdot T_n$. In the realm of dual quaternions, these transformations are represented as $Q_1 \odot Q_2 \odot Q_3 \ldots \odot Q_n$, showcasing the dual quaternion's unique capability to handle complex coordinate transformations. Readers are encouraged to refer to [10] that provides a detailed account of how a multi-section continuum can be represented using the dual quarternion system to improve accuracy in inverse kinematic solutions.

4.2.1.7 Inverse Kinematics

Given the complexity of the resulting kinematics of a sectioned continuum robot, closed-form solutions are generally not available. While there have been some attempts to derive closed-form solutions, these attempts often overlook the constraint coupling between the joint space variables, treating the curvature parameters $\boldsymbol{\lambda}$, $\boldsymbol{\phi}$, and $\boldsymbol{\theta}$ as independent variables. This simplification can lead to unfeasible and often physically inaccurate inverse kinematic solutions.

The most practical approach that has been explored involves numerical methods. One way to solve for the inverse kinematic solution is to formulate it as a constrained optimization problem. The goal is to find the joint space variables $\boldsymbol{q}$ that optimize a cost function while satisfying constraints:

$$\text{Minimize}: \quad f(\boldsymbol{q}) \tag{4.36}$$

$$\text{Subject to}: \quad \boldsymbol{g}(\boldsymbol{q}) \leq \mathbf{0} \tag{4.37}$$

$$\boldsymbol{h}(\boldsymbol{q}) = \mathbf{0} \tag{4.38}$$

where $f(\boldsymbol{q})$ is the cost function to be minimized, $\boldsymbol{g}(\boldsymbol{q})$ represents inequality constraints (e.g., actuator range constraints), and $\boldsymbol{h}(\boldsymbol{q})$ represents equality constraints.

To solve the optimization problem, numerical optimization routines available in software tools like MATLAB [1] and Python can be employed. These routines aim to find the optimal joint variables $\boldsymbol{q}^*$ that minimize the cost function while satisfying the constraints

$$\boldsymbol{q}^* = \text{argmin}_{\boldsymbol{q}} \, f(\boldsymbol{q}) \tag{4.39}$$

The solution $\boldsymbol{q}^*$ represents the joint configuration that allows the robot to track desired trajectories efficiently.

Alternatively, another numerical approach involves leveraging the robot's kinematics. By using the Jacobian matrix, denoted as $\mathbf{J}$, which describes the relationship between the task space (i.e., the end-effector's position and orientation) and the joint space (i.e., the curvature parameters), one can iteratively adjust the joint variables to move the robot towards its target location. This iterative process allows for real-time adjustments, making it suitable for dynamic and adaptive control of the continuum robot.

The relationship between the joint velocities $\dot{\boldsymbol{q}}$ and the end-effector velocities $\dot{\boldsymbol{x}}$ can be expressed using the Jacobian as follows:

$$\dot{\boldsymbol{x}} = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.40}$$

where $\dot{\boldsymbol{x}}$ signifies the end-effector's velocity in the task space, $\dot{\boldsymbol{q}}$ denotes the joint velocities in the joint space, and $\mathbf{J}(\boldsymbol{q})$ represents the Jacobian matrix, which depends on the current joint configuration $\boldsymbol{q}$.

To control the continuum robot's motion towards a desired target velocity $\dot{\boldsymbol{x}}_{\text{desired}}$, one can use an iterative scheme:

$$\dot{\boldsymbol{q}}_{k+1} = \dot{\boldsymbol{q}}_k + \mathbf{J}^{-1}(\boldsymbol{q}_k)\Delta\dot{\boldsymbol{x}} \tag{4.41}$$

where $\dot{\boldsymbol{q}}k + 1$ represents the updated joint velocities at iteration $k + 1$, while $\dot{\boldsymbol{q}}k$ denotes the joint velocities at iteration k. Additionally, $\mathbf{J}^{-1}(\boldsymbol{q}k)$ signifies the inverse Jacobian matrix at iteration k, and $\Delta\dot{\boldsymbol{x}}$ corresponds to the desired change in end-effector velocity required to reach $\dot{\boldsymbol{x}}$desired.

By iteratively updating the joint velocities, the continuum robot can adapt its configuration in real-time to achieve the desired task-space motion, making this approach well-suited for dynamic and adaptive control.

4.2.1.8 Derive Dynamics for Curve Parametric Models

Once the kinematics is established, we can derive the dynamics using the Lagrangian approach within the context of dynamic modeling based on curve parameters. It is assumed that the robot is made up of an infinite number of thin circular slices with constant mass and uniform linear density as shown in Fig. 4.9. Kinetic and potential energies are calculated for a slice at ξ. The total energy is then determined by integrating the energies from base to top ($\xi : 0 \rightarrow 1$).

$$\mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \left[\int_0^1 (\mathbf{J}_\xi^b)^T \delta M(\boldsymbol{q}) (\mathbf{J}_\xi^b) d\xi \right] \dot{\boldsymbol{q}} \tag{4.42}$$

where $\delta M = \left[\delta m I_{3\times 3}\ 0_{3\times 3}; 0_{3\times 3}\ I_{\delta m r^2}\right]$ and $\mathbf{J}_\xi^b \in \mathbb{R}^{3\times 6}$ is the body Jacobian matrix that contains the linear and angular velocities of a disc at ξ_i [24].

Potential energy is composed of gravitational and elastic potential energy. Therefore, the total potential energy can be written as

$$\mathcal{P}(\boldsymbol{q}) = m_i \left(\int_0^1 \boldsymbol{p}^T d\xi \right) \boldsymbol{g} + \frac{1}{2}\boldsymbol{q}^T \mathbf{K}_e \boldsymbol{q} \tag{4.43}$$

where m_i is the mass of the robot, $\boldsymbol{g} = [0, 0, g]^T$ is the gravitational acceleration vector, and $\boldsymbol{K}_e$ is the elastic stiffness matrix.

When total kinetic energy and the potential energy of the robot are known, the complete Lagrangian can be derived as $\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{P}(\boldsymbol{q})$. By applying the Lagrangian, the generalized equation of motion (EoM) can be expressed as

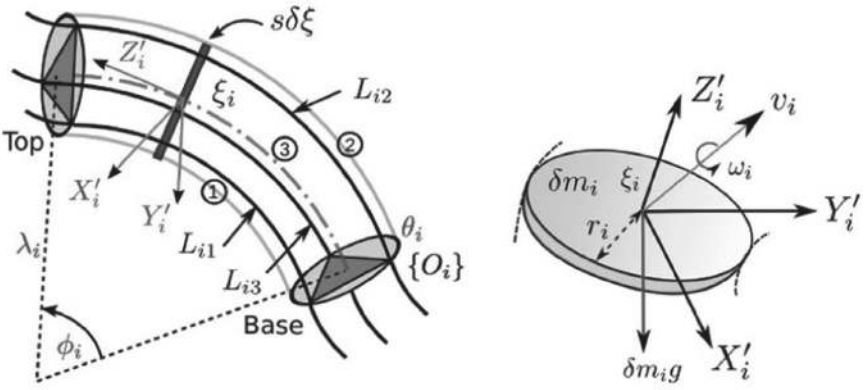

Fig. 4.9 **a** Schematic illustration of the infinitesimally thin slice at ξ_i on any ith continuum section along with curve parameters $\{\lambda_i, \phi_i, \theta_i\}$ (listed in Appendix B.1), actuator variables $\{O_i\}$, and $\{O_i'\}$. **b** Velocities and forces acting on the thin slice with respect to $\{O_i'\}$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial \dot{\boldsymbol{q}}}\mathcal{L} - \frac{\partial}{\partial \boldsymbol{q}}\mathcal{L} = \mathbf{F}_e \tag{4.44}$$

where $\mathbf{F}_e$ defines the input force vector in the curve parametric jointspace $\boldsymbol{q}$. The classical compact matrix form of Eq. (4.44) gives the complete EoM of the robot as

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) = \mathbf{F}_e \tag{4.45}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the generalized inertia matrix , $\mathbf{C} \in \mathbb{R}^{3\times3}$ is the centrifugal and Coriolis force matrix given in Eq. (4.46), $\mathbf{G} \in \mathbb{R}^3$ is the gravitational force matrix given in Eq. (4.47), and $\mathbf{F}_e \in \mathbb{R}^3$ is the external force vector in the jointspace $\boldsymbol{q}$.

$$\mathbf{C}_{k,j}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^{3} \frac{1}{2}\left[\frac{\partial M_{kj}}{\partial \boldsymbol{q}_i} + \frac{\partial M_{ki}}{\partial \boldsymbol{q}_j} - \frac{\partial M_{ij}}{\partial \boldsymbol{q}_k}\right]\dot{\boldsymbol{q}}_i \tag{4.46}$$

$$\mathbf{G}_{k,j}(\boldsymbol{q}) = \frac{\partial P(\boldsymbol{q})}{\partial \boldsymbol{q}_i} \tag{4.47}$$

4.2.2 Beam Theory

Beam theory is a mathematical framework used to model the behavior of slender, flexible structures like beams and rods, and it can also be applied to modeling soft and continuum robots with long, tubular segments. In the context of soft and continuum robots, beam theory approximates these robots as flexible beams subjected to various forces and deformations.

In the realm of soft and continuum robots, the application of beam theory proves to be an invaluable approach. This method simplifies the intricate structures of these robots into interconnected beams, with each beam meticulously representing a segment of the robot. One of the central tenets of beam theory is the assumption that these beams are slender; their length significantly surpasses their cross-sectional dimensions. This assumption aligns with the characteristic elongated, tubular shapes of many soft robots.

The pivotal role of cross-sectional properties comes to the forefront when employing beam theory. These properties encompass critical attributes such as the beam's area, moment of inertia, and stiffness. They are not uniform but instead fluctuate along the length of the robot due to its inherent compliance. These variations reflect the dynamic nature of soft materials, which change their mechanical responses under different circumstances.

Deformation analysis is the linchpin of beam theory. It delves deep into the intricate dance of a robot's segments as they yield to external forces and moments. Three primary types of deformations—axial, bending, and torsional—take center stage in this analysis. These deformations are elegantly elucidated through differential

equations derived from the fundamental principles of the theory. They offer profound insights into how soft and continuum robots adapt and respond to the complex interplay of forces.

The concept of boundary conditions adds a layer of sophistication to the modeling process. Accurate boundary conditions at the extremities of each robot segment are paramount. These conditions serve as the bridge between the robot and its external environment, delineating how the robot interacts with external constraints or even with neighboring segments. The correct specification of boundary conditions is pivotal in capturing the robot's overall deformation and behavior with precision.

Material properties constitute another essential facet of beam theory's applicability. The ability to account for various material properties, including elasticity, viscoelasticity, and nonlinear behavior under significant deformations, enables a nuanced understanding of the robot's response. Material choice and the fidelity of their modeling become decisive factors in determining the accuracy of the model itself.

The consideration of loads and forces finalizes the comprehensive scope of beam theory. This approach acknowledges the myriad external influences acting upon the robot, such as axial forces, distributed loads, and bending moments. These forces emanate from diverse sources, including external perturbations, actuation mechanisms, or intricate interactions with the surrounding environment. By incorporating these forces into the model, beam theory empowers researchers and engineers to predict and control the robot's behavior under various conditions.

4.2.2.1 Cosserat Rod Theory

Cosserat rod theory is a mathematical framework that treats a flexible robot as a one-dimensional continuum with intrinsic curvature and twist. This approach is particularly useful for modeling slender, flexible structures such as tentacles or snake-like robots. It takes into account the rod's curvature, torsion, and material properties to describe its behavior accurately.

Applying Cosserat rod theory to model a concentric tube robot involves a systematic approach to capture the robot's behavior accurately. Concentric tube robots are composed of multiple nested, flexible tubes, each of which can elongate, rotate, and bend. Cosserat rod theory, a mathematical framework for modeling slender, flexible structures, is well-suited for this purpose (Fig. 4.10).

Deformation within the rod is described by two key factors: strain and curvature. Strain refers to local elongation or compression occurring along the rod, while curvature characterizes the bending of the rod. Both strain and curvature exhibit variation along the robot's length, providing a means to account for its continuous curvature.

The equations of motion in Cosserat rod theory are derived from principles of continuum mechanics. These equations elucidate how external forces and torques applied to the robot influence its deformation and motion. They are expressed as partial differential equations (PDEs) that establish relationships between curvature, strain, and external loads. General equations used in the derivation can be expressed as

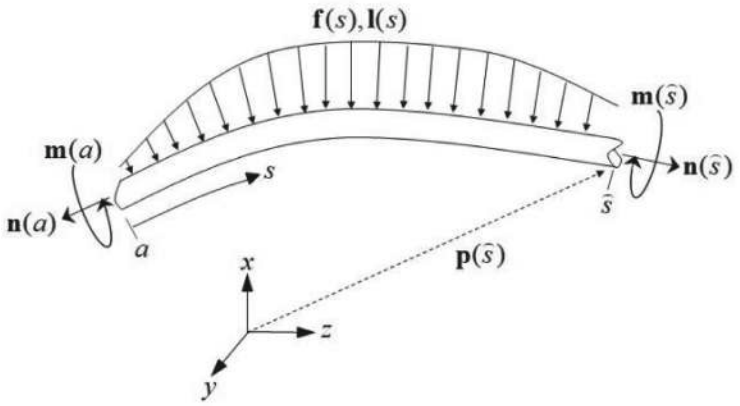

Fig. 4.10 An arbitrary section of the rod from a to bs, subject to distributed forces $f(s)$ and moments $l(s)$ at a given time t [16]

$$\frac{\partial \mathbf{N}}{\partial s} + (\mathbf{t} \cdot \nabla)\mathbf{t} - \mathbf{n} \cdot \nabla \mathbf{t} = \mathbf{0} \tag{4.48}$$

$$\frac{\partial \mathbf{M}}{\partial s} + (\mathbf{t} \cdot \nabla)\mathbf{m} - \mathbf{n} \cdot \nabla \mathbf{m} - \mathbf{m} \times \mathbf{t} = \mathbf{0} \tag{4.49}$$

where **N** is Force vector, **M** is Moment vector, **t** is Tangent vector along the rod's centerline, **n** is Normal vector, **m** is Director vector, κ is Curvature vector, $\boldsymbol{\kappa}$ is Rate of change of curvature vector, τ is Couple vector, s is arclength parameter, and ∇ is gradient operator.

By integrating the equations of motion throughout the rod's length, it becomes feasible to simulate how the robot's shape transforms over time in response to applied forces and torques. This process facilitates the anticipation of the robot's trajectory and configuration during various tasks.

4.2.2.2 Kirchhoff-Love Theory

The Kirchhoff-Love theory provides a mathematical framework for modeling the deformation of thin, flexible structures, such as continuum robots [42]. In this theory, we consider the deformation of a segment of the robot in a 2D plane. The key assumptions are that the deformation is primarily due to bending, with negligible stretching.

Let's denote the reference configuration of the segment as the undeformed state, and the deformed configuration as the state after bending. We will use a local coordinate system with x and y axes in the reference configuration and X and Y axes in the deformed configuration.

The fundamental mathematical equations of Kirchhoff-Love theory for continuum robot modeling are given by

$$\text{1. Bending Moment :} \quad M_x = -D\frac{d^2w}{dx^2} \tag{4.50}$$

$$\text{2. Shear Force :} \quad Q = -D\frac{d^2v}{dx^2} \tag{4.51}$$

$$\text{3. Bending-Extension Coupling :} \quad N = -D\frac{d^2u}{dx^2} \tag{4.52}$$

$$\text{4. Compatibility :} \quad \frac{d^2u}{dx^2} + \frac{d^2v}{dx^2} = 0 \tag{4.53}$$

where M_x is the bending moment about the x-axis, Q is the shear force, N is the bending-extension coupling, D is the flexural rigidity of the segment, $u(x)$ represents the axial displacement of the segment, $v(x)$ represents the transverse displacement (in-plane deformation) along the y-axis, and $w(x)$ represents the transverse displacement (out-of-plane deformation) along the z-axis.

Equations (4.50), (4.51), and (4.52) describe the equilibrium of moments and forces within the deformed segment, while Eq. (4.53) enforces the compatibility between axial and in-plane deformations. To fully specify the behavior of the continuum robot segment, appropriate boundary conditions must be applied. These conditions depend on the particular robot design and application. Solving the Kirchhoff-Love equations, subject to the boundary conditions, provides a mathematical description of the deformation of the continuum robot segment. The resulting displacement fields $u(x)$, $v(x)$, and $w(x)$ describe how the segment bends and deforms under applied loads.

4.2.3 Bezier Curves and Splines

Bezier curves and splines are mathematical representations of curves and surfaces. They are often used to design the shape of soft robots, allowing for smooth and continuous curvature changes. These curves can be controlled through control points, enabling the design of complex robot shapes.

4.2.3.1 Bezier Curves

Bézier curves [43] offer several advantages in geometric modeling including their simplicity and ease of control [36]. Bézier curves are defined by a small set of control points, typically with a fixed degree (e.g., quadratic or cubic). This simplicity makes them user-friendly and intuitive for artists and designers to create and manipulate curves. Moreover, Bézier curves are known for their smoothness, particularly in the

connections between curve segments. They provide visually pleasing and continuous transitions between control points, making them suitable for applications where aesthetic considerations are crucial.

The curve interpolates between the first and last control points, while the middle control points influence its shape. The following equation shows how to evaluate a Bézier curve at a given parameter value, t.

$$\mathbf{P}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i$$

where $\mathbf{P}(t)$ is the point on the Bézier curve at parameter value t and $\mathbf{P}_i$ are the control points of the Bézier segment.

To model a soft robotic arm using Bézier curves, one initial step involves identifying key points along the arm, which may be determined through experimentation or design considerations. These key points could include locations like the base of the arm, the elbow, and the wrist. Once these key points are identified, they can be utilized as control points for corresponding Bézier segments.

Having defined the control points for each Bézier segment, the subsequent step is to evaluate the Bézier curve at various parameter values. This evaluation process generates a point cloud that effectively represents the shape of the soft robotic arm. This approach allows for the precise modeling of the arm's kinematics and geometry using Bézier curves.

4.2.3.2 B-spline Curves

B-spline (Basis-spline) [13] curves offer a different set of advantages. One of the most significant advantages of B-spline curves is their flexibility. Unlike Bézier curves, B-splines can have variable degrees, allowing for more control over the curve's shape. This adaptability is particularly useful when modeling complex structures, such as soft robotic arms with varying curvatures [21]. Additionally, B-spline curves provide global control, meaning that each control point can influence a more extensive portion of the curve. This global influence gives designers greater freedom to shape curves with intricate details and deformations. B-splines are also versatile in handling both open and closed curves, making them suitable for a wide range of applications, including tasks where a curve needs to loop or connect back on itself. Finally, B-spline curves offer flexible interpolation options, allowing designers to interpolate any subset of control points, which can be advantageous when precise interpolation of key points is required for soft robotic arm modeling or other applications.

B-spline curves are defined using a set of control points and a knot vector. The degree of the B-spline curve determines the number of control points involved in shaping the curve. A B-spline curve interpolates between some of the control points, depending on its degree and knot vector.

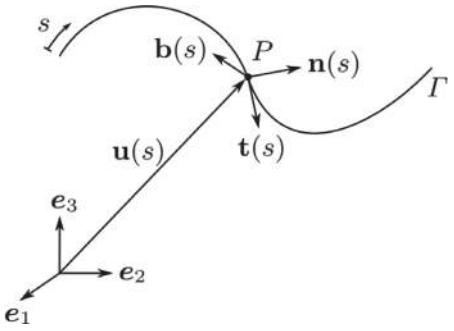

Fig. 4.11 Frenet-Serret definition of a spatial curve provides a comprehensive mathematical description of a spatial curve, characterizing its orientation, curvature, and torsion at each point along the curve. This definition is essential in differential geometry for analyzing the behavior of curves in three-dimensional space

The equation for a B-spline curve of degree n with control points $P_0, P_1, \ldots, P_n$ and a knot vector $[t_0, t_1, \ldots, t_{n+k+1}]$ is given by:

$$B(t) = \sum_{i=0}^{n} N_{i,k}(t) \cdot P_i \tag{4.54}$$

where $B(t)$ represents the point on the B-spline curve at parameter t (with $t_0 \leq t \leq t_{n+1}$), and $P_0, P_1, \ldots, P_n$ are the control points. The basis functions $N_{i,k}(t)$ depend on both the degree k of the B-spline curve and the knot vector $[t_0, t_1, \ldots, t_{n+k+1}]$.

4.2.4 Differential Geometry

Differential geometry provides a mathematical framework for describing the intrinsic and extrinsic properties of curved surfaces. Soft robots with continuous curvature can be analyzed using concepts from this field to understand their behavior and design.

4.2.4.1 Frenet-Serret Frame in Modeling Continuum Robots

The Frenet-Serret frame, also known as the moving trihedron or TNB frame, is a fundamental concept in differential geometry used to describe the local geometric properties of curves in three-dimensional space [5]. It consists of three mutually orthogonal unit vectors: the tangent vector (**T**), the normal vector (**N**), and the binormal vector (**B**). This frame is particularly useful in modeling the shape and behavior of continuum robots, which often exhibit complex, curved trajectories and deformations. The Frenet-Serret frame is defined as follows (Fig. 4.11).

$$\mathbf{T}(s) = \frac{d\mathbf{r}}{ds} \quad \text{(Tangent vector)} \tag{4.55}$$

$$\mathbf{N}(s) = \frac{d\mathbf{T}}{ds} \quad \text{(Normal vector)} \tag{4.56}$$

$$\mathbf{B}(s) = \mathbf{T} \times \mathbf{N} \quad \text{(Binormal vector)} \tag{4.57}$$

where $\mathbf{r}(s)$ represents the parametric equation of a curve in space, and s is the arc length parameter along the curve.

Frenet-Serret frames offer a compelling mathematical foundation for modeling spatial curves due to their intrinsic representation, capturing the local differential properties of curvature and torsion. These frames provide a continuous and coherent representation of orientation, which is particularly useful in modeling continuum and soft robots that often exhibit highly flexible and deformable structures. The Frenet-Serret frame's ability to capture curvature and torsion becomes crucial in characterizing the bending and twisting of the robot's structure, providing essential information for control and manipulation tasks. Their well-established numerical methods contribute to efficient algorithms, suitable for real-time simulations and control systems. Additionally, the natural parametrization based on arc length simplifies computations and mitigates parametrization-related challenges.

4.2.4.2 Euler Curves

Euler curves, also known as clothoids, are a type of parametric curve that offers several advantages for modeling soft robots. These advantages include their infinite smoothness, ensuring that soft robots deform smoothly without wrinkles or creases. Additionally, their flexibility allows for the modeling of various shapes, from simple linear segments to complex curves with multiple curvatures, accommodating a wide range of soft robot designs, from actuators to wearable devices. Furthermore, Euler curves are computationally efficient, making them suitable for real-time control of soft robots, which is essential for applications requiring rapid responses to changing conditions or environments.

Euler curves describe the equilibrium shapes of flexible structures with linearly varying curvature along their length. This concept can be extended to 3D and model the backbone of a continuum robot. The approach includes the calculation of position vectors and rotation matrices for each point along the curve, considering linear curvature and torsion variation.

An Euler curve is a parametric curve defined by the following equations

$$x(t) = at \cosh^3\left(\frac{t}{a}\right) \tag{4.58}$$

$$y(t) = at \sinh^3\left(\frac{t}{a}\right) \tag{4.59}$$

where t is a parameter and a is a scale factor.

The curvature of an Euler curve is given by the following equation

$$\kappa(t) = \frac{1}{a}\cosh\left(\frac{t}{a}\right) \tag{4.60}$$

The torsion of an Euler curve is given by the following equation:

$$\tau(t) = \frac{3\tanh\left(\frac{t}{a}\right)}{a\cosh^2\left(\frac{t}{a}\right)} \tag{4.61}$$

The arc length of an Euler curve from t_0 to t_1 is given by the following equation:

$$L = \int_{t_0}^{t_1} a\,\mathrm{sech}^2\left(\frac{t}{a}\right) dt \tag{4.62}$$

These equations can be used to model the behavior of Euler curves under various loading conditions. For example, the Euler curve curvature equation can be used to calculate the radius of curvature of the curve at any point. The Euler curve torsion equation can be used to calculate the rate of twist of the curve at any point. The Euler curve bending stiffness equation can be used to calculate the force required to bend the curve to a given curvature.

The incorporation of Euler curves in modeling soft robots can be compelling due to their simplicity and efficiency in preserving a constant rate of rotation, which is often a key aspect of the intricate bending and twisting motions exhibited by deformable bodies. Thus, Euler curves serve as a versatile tool for representing the spatial configuration of soft robots, whether navigating confined spaces, adapting to irregular surfaces, or performing delicate manipulation tasks.

4.3 Lumped Parametric Models

Lumped-parameter approaches are simplification techniques used to model soft and continuum robots by dividing them into discrete segments or elements, each with simplified properties and dynamics. These approaches provide a way to approximate the behavior of these complex systems using a reduced set of parameters.

Discrete kinematic modeling is a subset of kinematics that extends the principles of rigid body kinematics to robots with flexible, deformable structures of soft continuum robots. In discrete kinematics, the robot's continuous deformations are approximated using a series of discrete elements or segments. This discrete representation simplifies the modeling of soft robot motion and deformation, enabling fast computations and making it advantageous for robots that undergo large deformations and lack rigid joints.

Discrete kinematic modeling offers several advantages over continuous kinematic modeling for soft robots. They are more computationally efficient than continuous

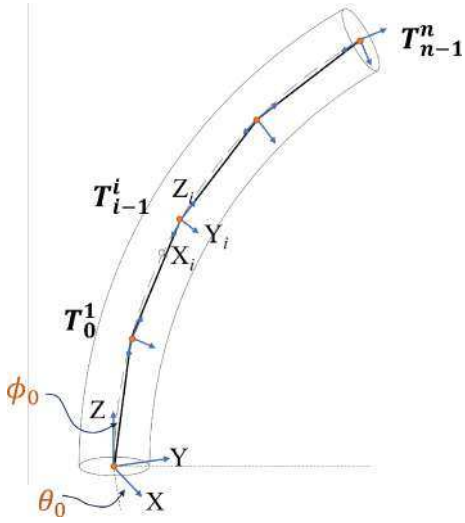

Fig. 4.12 Discrete link approximation method

kinematic models, especially in real-time applications. This is because discrete kinematic models avoid the need to solve complex differential equations that describes soft body deformations which can facilitate enabling real-time control with model-based approaches. Also, discrete models are more compatible with the discrete nature of sensors and actuators commonly used in robotics, which facilitates seamless integration. Further it provides a modular framework that allows for easy extension and adaptation to different robot designs and applications

Note that discrete kinematic modeling also has some limitations, primarily related to the accuracy of the discrete approximation. The accuracy of the discrete approximation depends on the number of discrete elements used to represent the robot and the complexity of the robot's deformations. For robots with complex deformations, a large number of discrete elements may be required to achieve sufficient accuracy. This can increase the computational complexity of the model and make it less suitable for real-time applications (Figs. 4.12, 4.13, 4.14 and 4.15).

4.3.1 Kinematics Modeling

Discrete link approximation involves representing the soft continuum robot as a chain of discrete, rigid links connected by joints. Each link is a rigid segment of the robot's body, and the joints represent the connection points between these segments. The robot's continuous deformation is approximated by considering the relative transformations between these links, which can be mathematically expressed using homogeneous transformation matrices.

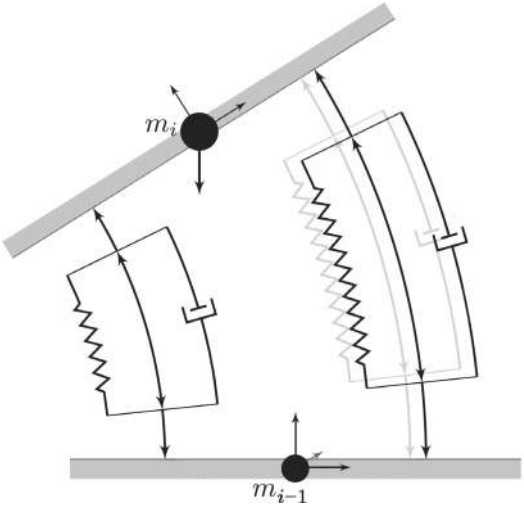

Fig. 4.13 Lumped-mass approximation with spring and dampers for material properties applied to a planar soft robotic arm. The segments are approximated as point masses, and in this case, there is no rotational energy considered. If a plate were used instead, rotational energy would be accounted for, resulting in a more accurate model. (Figure is adapted from [7])

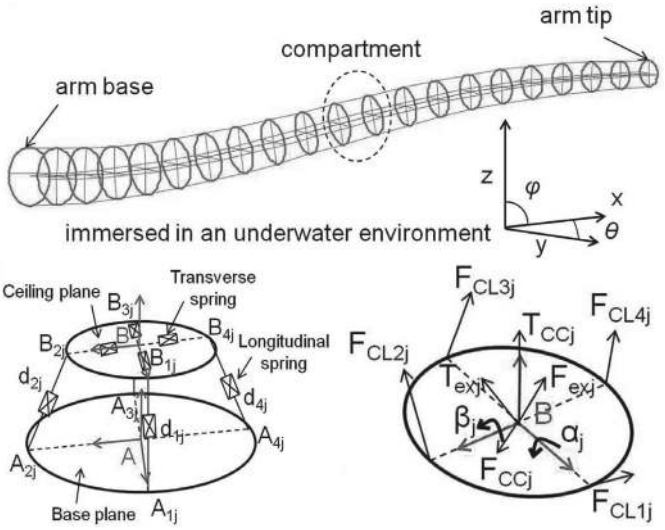

Fig. 4.14 Discretized representation of an octopus arm actuated parallel units actuated by two pairs of linear actuators. The arm consists of multiple discretized segments and this modular structure allows for flexible and coordinated motion, resembling the versatility of an octopus arm in real-world applications

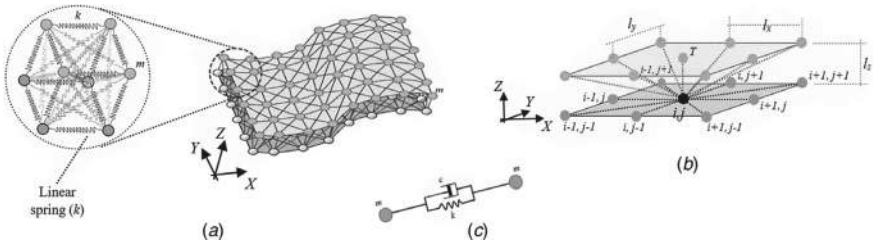

Fig. 4.15 **a** Illustration of the LDCS lumped-mass model with an enlarged view of one corner. **b** Depiction of the mass arrangement, showing the connections to the central masses i and j in the bottom layer of the two-layer model. **c** Detailed view of the spring-damper links connecting adjacent masses within the model

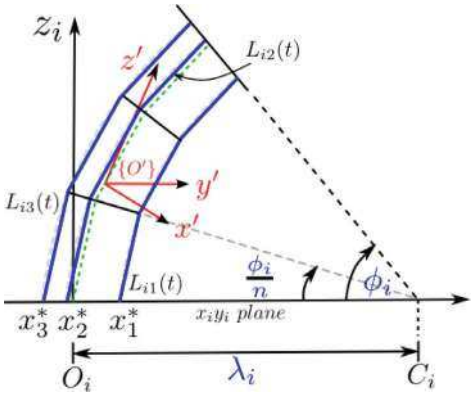

Fig. 4.16 **a** Schematic of a "segmented" cable-actuated continuum arm similar to prototypes reported in with three illustrated segments. Note that there can be an arbitrary number of segments depending on the design

The general equation of discrete link kinematics for a soft continuum robot is given by

$$\mathbf{T}_0^n = \mathbf{T}_0^1 \cdot \mathbf{T}_1^2 \cdot \mathbf{T}_2^3 \cdot \ldots \cdot \mathbf{T}_{i-1}^i \ldots \cdot \mathbf{T}_{n-1}^n \tag{4.63}$$

where $\mathbf{T}_0^n$ is the transformation matrix from the base frame (0) to the end-effector frame (n) and $\mathbf{T}_{i-1}^i$ is the transformation matrix from link i to link $(i+1)$. Each transformation matrix $\mathbf{T}_{i-1}^i$ captures the relative position and orientation of one link with respect to the next. By chaining these transformations together, we can describe the complete configuration of the robot in its continuous deformation.

The curve parameters λ_i, ϕ_i, and θ_i have now been expressed in joint space variables. This approach can be readily extended to discrete/segmented tendon-based continuum arms to derive the curve parameters, as illustrated in Fig. 4.16. Similar to the continuum arm case, for a continuum section with n segments, the corresponding length relationships are provided in Eq. (4.64).

$$L_{i0} + l_{i1} = 2n \sin\left(\tfrac{\phi_i}{2n}\right) \{\lambda_i - r_i \cos\theta_i\}$$

$$L_{i0} + l_{i2} = 2n \sin\left(\frac{\phi_i}{2n}\right)\left\{\lambda_i - r_i \cos\left(\frac{2\pi}{3} - \theta_i\right)\right\} \tag{4.64}$$
$$L_{i0} + l_{i3} = 2n \sin\left(\frac{\phi_i}{2n}\right)\left\{\lambda_i - r_i \cos\left(\frac{4\pi}{3} - \theta_i\right)\right\}$$

By solving for $\{\lambda_i, \phi_i, \theta_i\}$ using a similar approach, identical orientation parameters are derived except for ϕ_i, which is given as

$$\phi_i = 2n \sin^{-1}\left(\frac{\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{3nr_i}\right) \tag{4.65}$$

4.3.2 Dynamic Modeling

4.3.2.1 Newton-Euler Approach

The Newton-Euler equations provide a systematic framework for analyzing the dynamics of rigid bodies in both translational and rotational motion. This generalized approach outlines the key steps in applying Newton-Euler equations to model the dynamics of rigid bodies. The Newton-Euler equations can be divided into two sets of equations: the Newton equations for translational motion and the Euler equations for rotational motion.

The Newton equations describe how linear momentum changes over time for a rigid body, as given by

$$\sum F_i = m \cdot \ddot{x}_i \tag{4.66}$$

where $\sum F_i$ represents the net external forces acting in the mass, m_i is the mass, $\ddot{x}_i$ is the acceleration.

The Euler equations deal with rotational motion and describe how angular momentum changes over time for a discrete mass, is given by

$$\sum \tau_i = I_i \cdot \ddot{\theta}_i \tag{4.67}$$

where $\sum \tau_i$ represents the net external torques acting on the body, I_i is the moment of inertia, and $\ddot{\theta}_i$ is the angular acceleration.

Once the Newton-Euler equations are established for a specific rigid body, they result in a set of coupled differential equations. These equations can be solved numerically using integration methods such as the finite difference method or the Runge-Kutta method. Solving these equations provides insight into how the rigid body's motion evolves over time.

4.3.2.2 Euler-Lagrangian Approach

The Euler-Lagrangian approach, rooted in the broader Lagrangian mechanics, is a powerful framework employed in dynamic modeling for soft continuum robots. Its

primary appeal lies in its ability to systematically derive the equations of motion for complex systems while taking into account the system's configuration, forces, and constraints. This approach is especially valuable for soft continuum robots, which possess highly deformable and compliant structures.

The key assumptions and theories underpinning the Euler-Lagrangian approach include the principle of least action and the concept of generalized coordinates. The principle of least action posits that the path a system takes between two points in configuration space is the one that minimizes the action integral, where action is defined as the difference between kinetic and potential energy, denoted by the Lagrangian, $\mathcal{L}$ as

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \tag{4.68}$$

where $\mathcal{K}$ and $\mathcal{P}$ are the system kinetic and potential energy respectively.

The kinetic energy ($\mathcal{K}$) component of the Lagrangian accounts for the energy associated with the robot's motion. In discrete modeling approach one can approximate the entire body by lumping masses together, allowing for the calculation of kinetic energy for each lumped mass. These individual kinetic energies can then be summed to obtain the total kinetic energy of the system as

$$\mathcal{K} = \sum_{i=1}^{n} \left(\frac{1}{2} m_i \dot{x}_i^2 + \frac{1}{2} I_i \dot{\theta}_i^2 \right) \tag{4.69}$$

where, for the i-th discrete segment of system with n segments, m_i is mass, I_i is moment of inertia, x_i is translation, and θ_i is rotation.

The potential energy ($\mathcal{P}$) component of the Lagrangian represents the energy associated with the deformation of the robot due to stretching and bending. $\mathcal{K}$ also depends on the robot's configuration and the external forces acting on it. For instance, soft robots are often made of elastic materials that can undergo large deformations. Such deformations generates potential energy in terms of axial (extension/compression) and bending strain. The gravitational potential energy also contributes to the total potential energy. Thus, $\mathcal{K}$ can be mathematically denoted as

$$\mathcal{P} = \sum_{i=1}^{n} \left(\frac{1}{2} K_e \delta x_i^2 + \frac{1}{2} K_b \theta_i^2 + m_i g h_i \right) \tag{4.70}$$

where δx_i is the elastic strain, g is the gravitational acceleration, K_e is the elastic stiffness, K_b is the bending stiffness, m_i is the mass, h_i is the projection of x_i on to the axis representing the gravitational acceleration, and θ_i is the bending angle.

Note that Eq. (4.70) primarily addresses the elastic, bending potential energy, and gravitational potential energy, which are common and significant in most soft robots. However, there exist various other potential energy components such as fluid pressure, strain, electrostatic, and surface potential energies, among others.

With the Lagrangian defined in Eq. (4.68), the Euler-Lagrangian equations are employed to derive the equations of motion for the soft continuum robot. These equations describe how the robot's configuration and velocities change over time in response to applied forces and torques. The general form of the Euler-Lagrangian equations is given as:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \tag{4.71}$$

where $\mathcal{L}$ is the Lagrangian, $\dot{q}_i$ are the generalized velocities, q_i are the generalized displacements, Q_i represents the generalized forces.

While the Euler-Lagrangian approach is a powerful and widely used method for modeling the dynamics of robotic systems, including soft continuum robots, it has its limitations. One of the primary limitations is that it assumes a continuous and differentiable Lagrangian function, which may not always accurately represent the behavior of highly deformable and compliant soft robots. Soft robots often exhibit complex, nonlinear, and time-varying dynamics due to their flexibility, making it challenging to find an analytical Lagrangian that fully captures their behavior. Additionally, deriving the potential energy term ($\mathcal{P}$) in the Lagrangian can be particularly challenging for soft robots, as it requires accurately modeling the deformation of the robot's body under various external forces and constraints, which can be a complex task.

4.4 Hybrid Models

Hybrid models leverage a synergistic blend of diverse modeling approaches, such as discrete or lumped mass methods harmoniously integrated with constant curvature approaches. This strategic combination harnesses the strengths of each individual model, allowing for a comprehensive representation that captures both the overarching behavior of the robot and intricate details of deformation. This versatile approach not only enhances accuracy but also ensures computational efficiency, embodying the advantageous features of the integrated modeling techniques.

4.4.1 Discrete Constant Curvature Approximation

In this approach, the soft continuum robot curve is approximated as a chain of discrete segments, each characterized by a constant curvature. Figure 4.17 shows the schematic of this hybrid modeling approach. The curvature of each segment remains constant, simplifying the mathematical description of the robot's deformation. This simplification is particularly useful when dealing with robots that primarily exhibit

continuous large bending, such as snake-like or tentacle-like structures that undergo variable curvature shapes.

The general equation of discrete constant curvature-based kinematic modeling involves representing the robot's configuration as a sequence of homogeneous transformation matrices, similar to the discrete link approximation method. However, the transformation matrix T_{i-1}^{i}, of the curve s_i needs to be obtained. The general approach is as follows.

$$\mathbf{T}_{i-1}^{i}(s_i) = \mathbf{T}_y(\lambda_i) \cdot \mathbf{R}_x\left(\frac{s_i}{\lambda_i}\right) \cdot \mathbf{T}_y(-\lambda_i) \tag{4.72}$$

where λ_i is the radius of the constant curvature arc s_i. The angle $\phi_i = \frac{s_i}{\lambda_i}$ is the arc angle, which is considered to be the configuration variable of the kinematic. $Ty(\cdot)$ is the homogeneous translation matrix along the *y-axis*. The $Rx(\cdot)$ is the homogeneous rotation matrix around *x-axis*.

Then we can use HTM's of separate segments, T_{i-1}^{i}, to obtain the complete forward kinematics HTM of the hybrid model, from base frame to the end of manipulator, T_0^n, as

$$\mathbf{T}_0^n(s) = \mathbf{T}_0^1(s_1) \cdot \mathbf{T}_1^2(s_2) \cdot \mathbf{T}_2^3(s_3) \ldots \mathbf{T}_{i-1}^{i}(s_i) \ldots \mathbf{T}_{n-1}^{n}(s_n) \tag{4.73}$$

Discrete constant curvature-based kinematic modeling offers several advantages. It provides an intuitive representation of the robot's motion, aligning with its natural behavior of continuous bending. This makes it well-suited for robots that primarily undergo variable curvature bending. Additionally, it simplifies control strategies by breaking down the robot's motion into manageable segments, facilitating real-time control and trajectory planning.

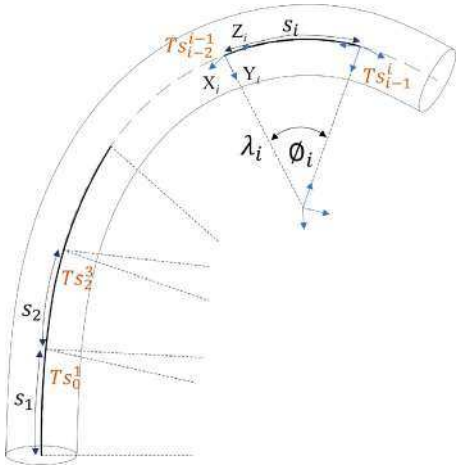

Fig. 4.17 This illustration showcases a hybrid model that integrates the discrete modeling approach with the constant curvature approximation method. The notable advantage of this methodology is its capacity to model variable curvature within soft robotic arms without resorting to computationally expensive alternatives

However, like discrete link approximation, this method also has limitations. The accuracy of the kinematic model depends on the number of segments used, and increasing the number of segments can lead to increased computational complexity. Therefore, the discrete constant curvature approach is not suitable for soft robots with long continuum arms.

4.4.2 Center-of-Gravity Based Approach

The primary challenge in continuous curvature approaches is their limited computational efficiency, stemming from the continuous nature of these robots. Calculating motion equations requires integration along the entire length of the robot, posing a significant computational burden. This has hindered the widespread adoption of continuous curvature dynamic models for controller design in continuum or soft robots. A potential solution involves a novel approach focusing on describing deformation and spatial movement using a center-of-gravity-based methodology. This alternative allows for deriving a mapping, resembling the traditionally integrated energy, but with the simplification of considering a single mass disc located at the center of gravity.

Similar to the approach outlined in Sect. 4.2.1.5, and to maintain general applicability, we proceed to derive the kinematics for the CoG of any i-th section. We establish a coordinate system at the CoG, denoted as $\left\{\overline{O}_i\right\}$, and introduce a homogeneous transformation matrix (HTM), denoted as $\overline{\mathbf{T}}_i : \left(\boldsymbol{q}_i\right) \mapsto \mathbb{SE}^3$, with respect to $\{O_i\}$, defined as

$$\overline{\mathbf{T}}_i = \int \mathbf{T}_i = \begin{bmatrix} \overline{\mathbf{R}}_i & \overline{\boldsymbol{p}}_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.74}$$

where $\overline{\mathbf{R}}_i = \int \mathbf{R}_i \left(\boldsymbol{q}_i\right) \in \mathbb{R}^{3\times 3}$ is the resultant rotation matrix and $\overline{\boldsymbol{p}}_i = \int \boldsymbol{p}_i \left(\boldsymbol{q}_i\right) \in \mathbb{R}^3$ is the position vector [?]. Note that the CoG is a function of $\boldsymbol{q}_i$ and therefore varies as the continuum section deforms.

To establish the kinematics of the CoG coordinate frame, denoted as $\left\{\overline{O}_i\right\}$, with respect to $\{O\}$, we integrate $\overline{\mathbf{T}}_i$ with the general HTM provided in (4.19). Following the definition, $\left\{O'_{i-1}|_{\xi_{i-1}=1}\right\} \equiv \{O_i\}$ (Fig. 4.18). Consequently, the CoG of the i-th section relative to $\{O\}$, denoted as $\overline{\mathbf{T}}^i : \left(\boldsymbol{q}^i\right) \mapsto \mathbb{SE}^3$, is defined as

$$\overline{\mathbf{T}}^i = \int \mathbf{T}^{i-1}\mathbf{T}_i = \left(\prod_{k=1}^{i-1} \mathbf{T}_k\right)\left(\int \mathbf{T}_i\right) = \begin{bmatrix} \overline{\mathbf{R}}^i & \overline{\boldsymbol{p}}^i \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.75}$$

where $\overline{\mathbf{R}}^i \left(\boldsymbol{q}^i\right) \in \mathbb{R}^{3\times 3}$ is orientation and $\overline{\boldsymbol{p}}_i \left(\boldsymbol{q}^i\right) \in \mathbb{R}^3$ are position matrices of the CoG coordinate frame.

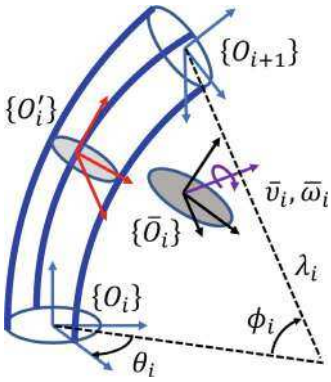

Fig. 4.18 Schematic of an infinitesimally thin slice at the CoG of any ith continuum section

Analogous to Eq. (4.20), the recursive form of $\overline{\mathbf{R}}^i$ and $\overline{\boldsymbol{p}}^i$ is given by

$$\begin{aligned} \overline{\mathbf{R}}^i &= \mathbf{R}^{i-1}\overline{\mathbf{R}}_i \\ \overline{\boldsymbol{p}}^i &= \boldsymbol{p}^{i-1} + \mathbf{R}^{i-1}\overline{\boldsymbol{p}}_i \end{aligned} \tag{4.76}$$

where $\mathbf{R}^{i-1}$ and $\boldsymbol{p}^{i-1}$ are formulated from Eq. (4.20).

Similar to Eqs. (4.21) and (4.22), the angular and linear body velocities of the CoG (relative to $\{\overline{O}_i\}$), $\overline{\boldsymbol{\omega}}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^3$ and $\overline{\boldsymbol{v}}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^3$, can be derived as

$$\begin{aligned} \overline{\boldsymbol{\Omega}}_i &= \overline{\mathbf{R}}_i^T\left(\boldsymbol{\Omega}_{i-1}\overline{\mathbf{R}}_i + \dot{\overline{\mathbf{R}}}_i\right) \\ \overline{\boldsymbol{v}}_i &= \overline{\mathbf{R}}_i^T\left(\boldsymbol{v}_{i-1} + \boldsymbol{\Omega}_{i-1}\overline{\boldsymbol{p}}_i + \dot{\overline{\boldsymbol{p}}}_i\right) \end{aligned} \tag{4.77}$$

where $\boldsymbol{v}_{i-1}$ and $\boldsymbol{\Omega}_{i-1}$, defined in Eqs. (4.22) and (4.21), are linear and angular velocities at the tip of the $(i-1)^{th}$ continuum section. Here too, we employ the relationship $\overline{\boldsymbol{\omega}}_i = \overline{\boldsymbol{\Omega}}_i^{\vee}$ to compute $\overline{\boldsymbol{\Omega}}_i\left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \in \mathbb{R}^{3\times 3}$.

Similar to the expressions in Eqs. (4.23), (4.24), (4.25), and (4.26), the angular body velocity Jacobian of the Center of Gravity (CoG), denoted as $\overline{\boldsymbol{J}}i^{\Omega}\left(\boldsymbol{q}^i\right) \in \mathbb{R}^{3\times 9n}$, its Hessian $\overline{\mathbf{H}}i^{\Omega}\left(\boldsymbol{q}^i\right) \in \mathbb{R}^{9n\times 9n}$, the linear body velocity Jacobian, $\overline{\boldsymbol{J}}i^{\upsilon}\left(\boldsymbol{q}^i\right) \in \mathbb{R}^{3\times 3n}$, and its Hessian $\overline{\mathbf{H}}i^{\upsilon}\left(\boldsymbol{q}^i\right) \in \mathbb{R}^{9n\times 3n}$ are provided respectively by Eqs. (4.78), (4.79), (4.80), and (4.81) as

$$\overline{\mathbf{J}}_i^{\Omega} = \overline{\mathbf{R}}_i^T\left[\mathbf{J}_{i-1}^{\Omega}\overline{\mathbf{R}}_i \middle| \overline{\mathbf{R}}_{i,\boldsymbol{q}_i^T}\right] \tag{4.78}$$

$$\overline{\mathbf{H}}_i^{\Omega} = \left[\begin{array}{c|c} \overline{\mathbf{R}}_i^T\mathbf{H}_{i-1}^{\Omega}\overline{\mathbf{R}}_i & \mathbf{0} \\ \hline \overline{\mathbf{R}}_{i,\boldsymbol{q}_i}^T\mathbf{J}_{i-1}^{\Omega}\overline{\mathbf{R}}_i\cdots & \overline{\mathbf{R}}_{i,\boldsymbol{q}_i}^T\overline{\mathbf{R}}_{i,\boldsymbol{q}_i^T}\cdots \\ +\overline{\mathbf{R}}_i^T\mathbf{J}_{i-1}^{\Omega}\overline{\mathbf{R}}_{i,\boldsymbol{q}_i} & +\overline{\mathbf{R}}_i^T\overline{\mathbf{R}}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i} \end{array}\right] \tag{4.79}$$

$$\overline{\mathbf{J}}_i^{\upsilon} = \overline{\mathbf{R}}_i^T \left[\mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega} \overline{\boldsymbol{p}}_i \middle| \overline{\boldsymbol{p}}_{i,\boldsymbol{q}_i^T} \right] \tag{4.80}$$

$$\overline{\mathbf{H}}_i^{\upsilon} = \left[\begin{array}{c|c} \overline{\mathbf{R}}_i^T \left(\mathbf{H}_{i-1}^{\upsilon} + \mathbf{H}_{i-1}^{\Omega} \overline{\boldsymbol{p}}_i \right) & \mathbf{0} \\ \hline \overline{\mathbf{R}}_{i,\boldsymbol{q}_i}^T \left(\mathbf{J}_{i-1}^{\upsilon} + \mathbf{J}_{i-1}^{\Omega} \overline{\boldsymbol{p}}_i \right) \cdots & \overline{\mathbf{R}}_{i,\boldsymbol{q}_i}^T \overline{\boldsymbol{p}}_{i,\boldsymbol{q}_i^T} \cdots \\ + \overline{\mathbf{R}}_i^T \mathbf{J}_{i-1}^{\Omega} \overline{\boldsymbol{p}}_{i,\boldsymbol{q}_i} & + \overline{\mathbf{R}}_i^T \overline{\boldsymbol{p}}_{i,\boldsymbol{q}_i^T,\boldsymbol{q}_i} \end{array} \right] \tag{4.81}$$

4.4.2.1 Derive Energy Balance of Center of Gravity-Based System

Without losing generality, we derive the kinetic energies (angular and linear) for any i^{th} continuum section. We then compare the terms to formulate the energy scaling conditions. Analogous to [9], to find the kinetic energy of the continuum section using an integral approach, we consider an infinitesimally thin disc of radius r_i along the length of the continuum section. By applying the body velocities given by Eq. (4.77), the energy computed for a disc is then integrated with respect to ξ_i to compute the section energy.

The angular kinetic energy, $\mathcal{K}_i^{\omega} : \left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \mapsto \mathbb{R}$, is given by

$$\begin{aligned} \mathcal{K}_i^{\omega} &= \int \left(\frac{1}{2} \boldsymbol{\omega}_i^T \mathcal{M}_i^{\omega} \boldsymbol{\omega}_i \right) = \frac{1}{2} I_{xx} \mathbb{T}_2 \left(\int \boldsymbol{\Omega}_i^T \boldsymbol{\Omega}_i \right) \\ &= \frac{1}{2} I_{xx} \mathbb{T}_2 \left(\int \mathbf{R}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \mathbf{R}_i \cdots \right. \\ &\quad \left. + 2 \int \dot{\mathbf{R}}_i^T \boldsymbol{\Omega}_{i-1} \mathbf{R}_i + \int \dot{\mathbf{R}}_i^T \dot{\mathbf{R}}_i \right) \end{aligned} \tag{4.82}$$

where $I_{xx} = \frac{1}{4} m_i r_i^2$ is the moment of inertia about the X axis of $\left\{O_i'\right\}$.

Using the angular velocity given in Eq. (4.77), finding the angular kinetic energy of the disc at the CoG, $\overline{\mathcal{K}}_i^{\omega} : \left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \mapsto \mathbb{R}_0^+$, results in

$$\begin{aligned} \overline{\mathcal{K}}_i^{\omega} &= \frac{1}{2} \overline{\boldsymbol{\omega}}_i^T \mathcal{M}_i^{\omega} \overline{\boldsymbol{\omega}}_i = \frac{1}{2} I_{xx} \mathbb{T}_2 \left(\overline{\boldsymbol{\Omega}}_i^T \overline{\boldsymbol{\Omega}}_i \right) \\ &= \frac{1}{2} I_{xx} \mathbb{T}_2 \left(\overline{\mathbf{R}}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\mathbf{R}}_i + 2 \dot{\overline{\mathbf{R}}}_i^T \boldsymbol{\Omega}_{i-1} \overline{\mathbf{R}}_i + \dot{\overline{\mathbf{R}}}_i^T \dot{\overline{\mathbf{R}}}_i \right) \end{aligned} \tag{4.83}$$

Similarly, using the linear body velocity in Eq. (4.77), the linear kinetic energy of the continuous model, $\mathcal{K}_i^{\upsilon} : \left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \mapsto \mathbb{R}_0^+$, can be computed as

$$\begin{aligned} \mathcal{K}_i^{\upsilon} &= \int \left(\frac{1}{2} \boldsymbol{\upsilon}_i^T \mathcal{M}_i^{\upsilon} \boldsymbol{\upsilon}_i \right) \\ &= \frac{1}{2} m_i \left(\boldsymbol{\upsilon}_{i-1}^T \boldsymbol{\upsilon}_{i-1} + 2 \boldsymbol{\upsilon}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\boldsymbol{p}}_i + 2 \boldsymbol{\upsilon}_{i-1}^T \dot{\overline{\boldsymbol{p}}}_i \cdots \right. \\ &\quad \left. + \int \boldsymbol{p}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \boldsymbol{p}_i + 2 \int \boldsymbol{p}_i^T \boldsymbol{\Omega}_{i-1}^T \dot{\boldsymbol{p}}_i + \int \dot{\boldsymbol{p}}_i^T \dot{\boldsymbol{p}}_i \right) \end{aligned} \tag{4.84}$$

where $\mathcal{M}_i^{\upsilon} = m_i \mathbf{I}_3$. Additionally, the CoG model's linear kinetic energy, $\overline{\mathcal{K}}_i^{\upsilon} : \left(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i\right) \mapsto \mathbb{R}_0^+$, is derived as

$$\begin{aligned}\overline{\mathcal{K}}_i^{\upsilon} &= \frac{1}{2}\overline{\boldsymbol{v}}_i^T \mathcal{M}_i^{\upsilon} \overline{\boldsymbol{v}}_i = \frac{1}{2} m_i \left(\boldsymbol{v}_{i-1}^T \boldsymbol{v}_{i-1} + 2\boldsymbol{v}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\boldsymbol{p}}_i \cdots \right. \\ &\left. + 2\boldsymbol{v}_{i-1}^T \dot{\overline{\boldsymbol{p}}}_i + \overline{\boldsymbol{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\boldsymbol{p}}_i + 2\overline{\boldsymbol{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \dot{\overline{\boldsymbol{p}}}_i + \dot{\overline{\boldsymbol{p}}}_i^T \dot{\overline{\boldsymbol{p}}}_i \right)\end{aligned} \tag{4.85}$$

4.4.2.2 Minimize Energy Difference Between the Integral and CoG-Based Models

In this section, we systematically derive scalars to match the kinetic energy of the CoG models to that of the integral model, utilizing the energies derived in Sect. 4.4.2.1. Unlike the single-section case [?], however, the kinetic energy is dependent on the velocities of the i^{th} section as well as the previous sections. Consider the angular energy difference between the models, derived for the i^{th} continuum section, given by

$$\begin{aligned}\mathcal{K}_i^{\omega} - \overline{\mathcal{K}}_i^{\omega} = \frac{1}{2} I_{xx} \mathbb{T}_2 \left(\int \dot{\mathbf{R}}_i^T \dot{\mathbf{R}}_i - \beta_3^{\omega} \dot{\overline{\mathbf{R}}}_i^T \dot{\overline{\mathbf{R}}}_i \cdots \right. \\ + 2 \int \mathbf{R}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \mathbf{R}_i - 2\beta_1^{\omega} \overline{\mathbf{R}}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\mathbf{R}}_i \cdots \\ \left. + \int \dot{\mathbf{R}}_i^T \boldsymbol{\Omega}_{i-1} \mathbf{R}_i - \beta_2^{\omega} \dot{\overline{\mathbf{R}}}_i^T \boldsymbol{\Omega}_{i-1} \overline{\mathbf{R}}_i \right)\end{aligned} \tag{4.86}$$

where β_k^{ω} for all $k \in \{1, 2, 3\}$ are the energy shaping coefficients that we apply to the Center of Gravity (CoG) energy terms to match the energies.

Note that, in this case, unlike the single-section case [12], we have three terms that do not get canceled when taking the difference. Likewise, the linear kinetic energy difference is computed as

$$\begin{aligned}\mathcal{K}_i^{\upsilon} - \overline{\mathcal{K}}_i^{\upsilon} = \frac{1}{2} m_i \left(\int \boldsymbol{p}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \boldsymbol{p}_i - \beta_1^{\upsilon} \overline{\boldsymbol{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \overline{\boldsymbol{p}}_i \cdots \right. \\ + \int \boldsymbol{p}_i^T \boldsymbol{\Omega}_{i-1}^T \dot{\boldsymbol{p}}_i - \beta_2^{\upsilon} \overline{\boldsymbol{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \dot{\overline{\boldsymbol{p}}}_i \cdots \\ \left. + \int \dot{\boldsymbol{p}}_i^T \dot{\boldsymbol{p}}_i - \beta_3^{\upsilon} \dot{\overline{\boldsymbol{p}}}_i^T \dot{\overline{\boldsymbol{p}}}_i \right)\end{aligned} \tag{4.87}$$

Notice that some terms are canceled due to the absence of products of integrable terms, resulting in three remaining terms. We introduce the energy shaping coefficients, β_k^{υ} for all $k \in \{1, 2, 3\}$, for each of those terms.

The coefficients, as introduced in Eqs. (4.86) and (4.87), can be determined in the subsequent part of this section through a multivariate optimization routine. Incorporating the physical robot parameters, such as L_{i0}, l_i, and r_i, the energy differences

described by Eqs. (4.86) and (4.87) become functions of $(\alpha_l, \alpha_r, \boldsymbol{q}i, \dot{\boldsymbol{q}}i, \boldsymbol{\Omega}i - 1) \in \mathbb{R}^{11}$. Here, $\alpha l = \frac{\max(l_i)}{L_{i0}}$ and $\alpha_r = \frac{r_i}{L_{i0}}$ represent the normalized length and radius of the continuum section.

The modified energy-based Center of Gravity (CoG) discs can subsequently serve as a basis for deriving the equations of motion through standard Newton-Euler or other rigid-bodied dynamic algorithms similar to [11].

4.4.2.3 Computing the Energy Shaping Coefficients

For the random combinations of joint-space variables and physical parameters, we compute the corresponding kinetic energy differences between the integral and CoG-based models as presented in Eqs. (4.86) and (4.87). To facilitate a straightforward comparison of corresponding terms, we calculate the three residual terms of each kinetic energy difference separately. For example, in the case of $\mathcal{K}i^{\omega}$, we separately compute the terms $\mathbb{T}2(\int \mathbf{R}i^T \boldsymbol{\Omega}i - 1^T \boldsymbol{\Omega}i - 1\mathbf{R}i)$, $\mathbb{T}2(\int \dot{\mathbf{R}}i^T \boldsymbol{\Omega}i - 1\mathbf{R}i)$, and $\mathbb{T}2(\int \dot{\mathbf{R}}i^T \dot{\mathbf{R}}_i)$.

Similarly, for $\overline{\mathcal{K}}i^{\omega}$, we compute the terms $\mathbb{T}2(\overline{\mathbf{R}}i^T \boldsymbol{\Omega}i - 1^T \boldsymbol{\Omega}i - 1\overline{\mathbf{R}}i)$, $2\mathbb{T}2(\dot{\overline{\mathbf{R}}}i^T \boldsymbol{\Omega}i - 1\overline{\mathbf{R}}i)$, and $\mathbb{T}2(\dot{\overline{\mathbf{R}}}i^T \dot{\overline{\mathbf{R}}}_i)$ separately. The energy difference, $\mathcal{K}i^{\omega} - \overline{\mathcal{K}}i^{\omega}$, is obtained by summing these terms and scaling the result by $\frac{1}{2} I_{xx}$. The same approach is applied to the linear kinetic energy difference given by Eq. (4.87), scaled by $\frac{m_i}{2}$.

4.4.2.4 Potential Energy of Continuum Sections

As reported in [9], a continuum arm is subjected to gravitational and elastic potential energies. Elastic potential energy, given by $\mathcal{P}^e = \frac{1}{2}\boldsymbol{q}^T \mathbf{K}_e \boldsymbol{q}$, only depends on $\boldsymbol{q}$ and is therefore independent of the modeling approach herein. The gravitational potential energy for the integral and CoG-based model can be defined as $\mathcal{P}_i^g = \int m_i \boldsymbol{g}^T \boldsymbol{p}^i$ and $\overline{\mathcal{P}}_i^g = m_i \boldsymbol{g}^T \overline{\boldsymbol{p}}^i$ respectively. Note that, $\mathcal{P}_i^g$ does not contain products of integrable terms. Therefore, $\mathcal{P}_i^g$ can be simplified to $\mathcal{P}_i^g = m_i \boldsymbol{g}^T \left(\int \boldsymbol{p}^i\right)$ and from the definition in Eq. (4.75), then becomes $\mathcal{P}_i^g = m_i \boldsymbol{g}^T \left(\overline{\boldsymbol{p}}^i\right) = \overline{\mathcal{P}}_i^g$. Thus, the gravitational potential energy is identical in both models.

4.5 Learning-Based Models

4.5.1 Artificial Neural Networks (ANN)

Neural networks have emerged as a powerful tool in the modeling of soft robots, offering the capability to capture complex and nonlinear behaviors. These models leverage artificial neural networks inspired by the structure and function of the human brain. Neural network-based approaches have found extensive applications in soft

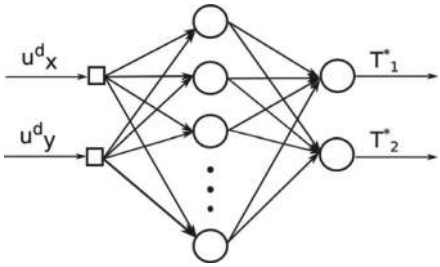

Fig. 4.19 ANN

robot modeling due to their ability to learn from data, adapt to changing conditions, and represent intricate relationships within the system. Artificial neural networks (ANNs), including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are powerful machine learning algorithms that can be used to model complex relationships in soft robot behavior.

ANNs have shown remarkable success in improving the accuracy and versatility of soft robot modeling, enabling applications ranging from robot design and control to perception and interaction with the environment. The choice of the neural network architecture and training strategy depends on the specific modeling task and available data, making them a valuable tool in advancing the field of soft robotics.

In ANNs, neurons are organized into layers within a neural network where the first layer is known as the input layer, and the final layer is referred to as the output layer with hidden layers in between yielding a high-dimensional set of nested functions given by

$$y = g_M\left(A_M, \ldots, g_2\left(A_2, g_1\left(A_1, x\right)\right)\ldots\right) \tag{4.88}$$

where x represents the input values, A_i stands for the edge weights, g_i are the activation functions, and y denotes the output values.

In many soft robotics applications, the input and output layers correspond to the actuation variables or jointspace inputs u and the shape parameters q. The training of the ANN or in other words, the learning process, involves optimizing the network weights, typically achieved through back-propagation.

CNNs are well-suited for image-based modeling of robot deformations. CNNs are able to learn spatial features in images, which can be used to model the deformation of a soft robot in response to actuator inputs and external loads. For example, CNNs have been used to model the deformation of soft grippers, soft actuators, and soft robots with complex shapes.

Recurrent neural networks (RNNs) are well-suited for modeling the temporal dynamics of soft robots. RNNs are able to learn sequential patterns in data, which can be used to model the dynamic behavior of a soft robot over time. For example, RNNs have been used to model the dynamics of soft robots during locomotion, manipulation, and interaction with the environment (Figs. 4.19 and 4.20).

Fig. 4.20 Octopus robot arm

4.5.2 Reinforcement Learning (RL)

Reinforcement learning (RL) is a machine learning technique that allows agents to learn how to behave in an environment by trial and error. RL agents are rewarded for taking actions that lead to desired outcomes and penalized for taking actions that lead to undesired outcomes. Over time, the agent learns to select actions that maximize its expected reward. RL algorithms can be used for robot modeling, control, and optimization. RL algorithms are well-suited for soft robot control and optimization because they can learn to control complex systems with nonlinear dynamics. Soft robots can be particularly challenging to control because they are often deformable and have many degrees of freedom.

4.5.3 Physics-Informed Neural Networks (PINNs)

Physics-informed neural networks (PINNs) are a type of neural network that is trained to enforce physical equations. PINNs are well-suited for modeling soft robots because they can learn to model the complex deformation and dynamics of soft robots while satisfying physical constraints. PINNs combine deep learning with physical equations to model the behavior of soft robots. They enforce physical constraints and can learn from sparse data, making them suitable for modeling soft robots' deformation and dynamics.

PINNs are trained on a dataset of input-output pairs, where the inputs are the actuator forces and external loads, and the outputs are the robot's deformations and dynamics. The PINN is trained to minimize the difference between the predicted outputs and the actual outputs, while also satisfying the physical equations.

One of the advantages of PINNs is that they can learn from sparse data. This is important for soft robots because it can be difficult to collect dense data on soft robots due to their deformable nature. PINNs can also learn to model complex relationships in the data, which makes them well-suited for modeling soft robots' nonlinear dynamics.

4.5.4 Data-Driven Modeling

Data-driven modeling is a robust and effective approach in the realm of soft robotics. Soft robots, characterized by their complex and nonlinear behaviors, often pose challenges for traditional physics-based modeling. Data-driven models leverage experimental or sensor data to establish relationships between inputs (e.g., control inputs) and outputs (e.g., deformations), making them well-suited for capturing the intricate behavior of soft robots. Data-driven modeling holds significant promise in advancing our understanding and control of soft robots, enabling the capture of their intricate and nonlinear behaviors. The choice of modeling technique depends on the specific characteristics of the soft robot and the nature of the available data, making it a versatile approach in the soft robotics domain. Several common data-driven modeling techniques find application in the context of soft robotics.

4.5.4.1 Regression Models

Regression models aim to learn a functional mapping from inputs to outputs. Within soft robot modeling, various regression models such as linear regression, polynomial regression, and support vector regression are employed to capture the relationships between control inputs and resulting deformations.

Regression models play a crucial role in modeling various aspects of soft robot behaviors. They are employed for tasks such as deformation modeling, where they capture the intricate relationship between control inputs and the resulting deformations, facilitating the design and precise control of soft robots. Additionally, these models can be applied to force and torque modeling, enabling the prediction of forces and torques exerted by soft robots, making them valuable for tasks like grasping and manipulation. Furthermore, regression models can be instrumental in contact modeling, allowing the depiction of interactions between soft robots and their surroundings. This capability proves beneficial in designing and controlling soft robots for applications such as navigation and obstacle avoidance, enhancing their adaptability and utility in diverse scenarios.

Regression models aim to learn a functional mapping from inputs to outputs. Within soft robot modeling, various regression models such as linear regression, polynomial regression, and support vector regression are employed to capture the relationships between control inputs and resulting deformations.

Mathematically, a regression model can be represented as follows:

$$y = f(x) + \epsilon \tag{4.89}$$

where y is the output of the model, x is the input to the model, $f(x)$ is the learned functional mapping from inputs to outputs, and ϵ is an error term.

Linear regression is a simple regression model that assumes a linear relationship between the input and output variables. The learned functional mapping for linear regression is given by:

$$f(x) = \beta_0 + \beta_1 x \tag{4.90}$$

where β_0 and β_1 are the model parameters.

Polynomial regression is a more complex regression model that can capture non-linear relationships between the input and output variables. The learned functional mapping for polynomial regression is given by

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n \tag{4.91}$$

where $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are the model parameters.

Support vector regression (SVR) is a non-linear regression model that uses a kernel function to project the input data into a higher-dimensional space. The learned functional mapping for SVR is given by

$$f(x) = \sum_i \alpha_i K(x, x_i) + b \tag{4.92}$$

where α_i are the support vectors, $K(x, x_i)$ is the kernel function, and b is the bias term.

4.5.4.2 Gaussian Processes (GPs)

Gaussian processes are a class of Bayesian non-parametric models capable of discerning intricate data relationships. GPs are particularly advantageous in soft robot modeling due to their ability to handle noisy and incomplete data, allowing for the accurate representation of complex behaviors.

Mathematically, a GP can be defined as a random process over functions, where the function values at any two input points are jointly Gaussian distributed. The mean and covariance function of a GP can be specified, which allows the model to be tailored to the specific problem at hand. One of the key advantages of GPs is that they can be used to make predictions at new input points without requiring any explicit training. This is because the GP learns a distribution over functions, rather than a specific function.

For instance, deformation modeling using GPs allows for precise control and design optimization, as demonstrated by Smith et al. who used GPs to predict real-time deformations of a pneumatic soft robot arm. Stiffness estimation, another application, aids in tasks like object manipulation, where GPs, as shown by Jones and Brown, can estimate stiffness variations along a soft gripper for effective grasp force modulation. Additionally, GPs serve as the foundation for learning-based control strategies, exemplified by Roberts et al.'s framework, enabling soft robots to adapt

their behavior in response to changing conditions and tasks, particularly in obstacle avoidance scenarios. These applications showcase the versatility and impact of Gaussian Processes in advancing the capabilities of soft robots in various domains.

The mathematical definition of deformation modeling using Gaussian processes (GPs) can be formulated as follows. Let x be a vector of control inputs to a soft robot, and let y be the resulting deformation of the robot. We can model the relationship between x and y using a GP as follows:

$$y = f(x) + \epsilon \tag{4.93}$$

where $f(x)$ is a latent function that represents the relationship between x and y, ϵ is a Gaussian noise term.

In order to use GPs for deformation modeling, we need to specify a prior distribution over $f(x)$. This can be done using a kernel function, which is a function that measures the similarity between two input points. The kernel function determines how smoothly the latent function $f(x)$ varies over the input space.

Once we have specified a kernel function, we can use the GP to make predictions about the deformation of the robot at new control inputs. To do this, we first need to train the GP on a set of training data, which consists of pairs of control inputs and deformations.

Once the GP is trained, we can use it to predict the deformation of the robot at a new control input x^* as follows:

$$y_* = f(x_*) + \epsilon_* \tag{4.94}$$

where y_* is the predicted deformation, and ϵ_* is a Gaussian noise term.

Various programming languages offer libraries tailored for Gaussian Process (GP) modeling, each with its strengths and capabilities. In Python, an extensively used language for machine learning and data science, there are several GP modeling libraries such as GPy, GPflow, and scikit-learn. These libraries provide a user-friendly and versatile interface for GP modeling, making it accessible to a wide range of users. On the other hand, Julia, known for its high-performance capabilities in scientific computing, offers libraries like Gen and GPModels, which leverage the language's speed and flexibility for more complex GP models. Lastly, R, a popular choice for statistical analysis, features libraries like DiceKriging and GPfunctions, making it convenient to implement and utilize GP models within the R environment. The availability of these libraries in multiple languages empowers practitioners to choose the one that best suits their specific modeling needs and language preferences.

4.5.4.3 Support Vector Machines (SVMs)

SVMs, known for their robustness in classification and regression tasks, find utility in soft robot modeling. They can effectively handle noise and outliers in data, making them a valuable tool for establishing relationships between control inputs and soft

robot responses and flexibility in modeling a wide range of robot behaviors, including deformation, force generation, and contact interactions.

SVMs offer several advantages over GPs in terms of computational efficiency and model interpretability. SVMs are computationally efficient, particularly for large datasets, as they only require storage and operations on support vectors, whereas GPs need to process the entire training set. Moreover, SVMs are more interpretable than GPs, as they can be expressed as a set of linear equations, making it easier to grasp their inner workings. In contrast, GPs are a more intricate model, posing challenges in terms of model interpretation.

In addition, compared to regression models, SVMs excel in modeling non-linear relationships between input and output variables, a critical capability for capturing the often non-linear behavior of soft robots. They are also less susceptible to overfitting due to their built-in regularization parameter, ensuring a balanced trade-off between model complexity and accuracy. Additionally, SVMs demonstrate superior generalization ability, making them more reliable when applied to new, unseen data, whereas regression models are more prone to overfitting the training data and might exhibit suboptimal performance on new data.

SVMs can be formulated as a quadratic optimization problem, as follows.

$$\begin{aligned} \text{minimize}: \quad & \frac{1}{2} w^T w + C \sum (\xi_i + \xi_i^*) \\ \text{subject to}: \quad & y_i (w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \xi_i^* \geq 0 \end{aligned}$$

where w is the weight vector of the SVM, b is the bias term of the SVM, C is a regularization parameter that controls the trade-off between model complexity and accuracy, and ξ_i and ξ_i^* are slack variables that allow the SVM to tolerate errors in the training data.

SVMs use kernel functions to project the input data into a higher-dimensional space, where it is easier to learn non-linear relationships. Some common kernel functions include the linear kernel $K(x, x') = x^T x'$, the polynomial kernel $K(x, x') = (x^T x' + c)^d$, and the radial basis function (RBF) kernel $K(x, x') = \exp(-\gamma ||x - x'||^2)$. Here c and γ are hyperparameters that need to be tuned. These kernels play a pivotal role in SVMs, enabling them to handle complex data patterns by transforming them into higher-dimensional spaces where linear separations or relationships become more apparent. Once the SVM has been trained, it can be used to predict the output for a new input vector x^* as $y_* = \text{sign}(w^T x_* + b)$.

4.6 How to Select a Suitable Modeling Approach

Selecting an appropriate modeling approach for soft continuum robots is a crucial step in the development and analysis of these complex systems. Table 4.1 provides a concise overview of different modeling approaches for soft robots and their respective pros and cons. It serves as a valuable reference to guide the selection of an appropriate modeling framework based on the complexity of the robot's deformations, computational resource availability, and the specific requirements of various applications in soft robot modeling.

4.6.1 Robot Geometry and Structure

When considering the robot's geometry, the choice of modeling approach becomes crucial. For instance, if the robot has a linear structure, such as a multi-section continuum of concentrically attached units resembling an elephant trunk or an octopus arm, a classical continuous curvature model can be suitable for accurately representing its deformations. This approach excels in capturing the robot's continuous and complex shape changes. However, for scenarios where computational efficiency is a

Table 4.1 Comparison of soft robot modeling approaches

Modeling approach	Pros	Cons	Examples of applications
Continuous	High accuracy, can model complex deformations	Computationally expensive, difficult to implement	Soft robotic grippers, soft robotic manipulators, soft robotic locomotors
Lumped	Computationally efficient, easy to implement	Less accurate than continuous models, may not be suitable for robots with complex deformations	Soft robotic actuators, soft robotic sensors, soft robotic wearables
Hybrid	Combines the advantages of continuous and lumped models	More complex to implement than lumped models	Soft robotic arms, soft robotic hands
Learning-based	Can learn complex nonlinear relationships between inputs and outputs	Requires a large amount of training data, may be computationally expensive to train and deploy	Soft robotic grippers that need to grasp objects with complex shapes, soft robotic locomotors that need to move through complex environments

priority, a discretized approach, like the center of gravity method, can be employed to model the same robot while simplifying the computational demands.

Conversely, if the robot is designed as a wearable system intended to interact with the environment, experience distributed forces, and undergo significant deformation in a 2D plane, a more generalized modeling framework may be necessary. In such cases, discrete mass-spring-damper models or reduced-order finite element methods provide the flexibility and accuracy needed to represent the robot's behavior adequately.

4.6.2 Deformation Characteristics

The choice of modeling technique, whether it be continuous, lumped, hybrid, or AI-based, is intricately tied to comprehending how the soft robot deforms when subjected to various loads and environmental conditions. Robots exhibit diverse deformation modes, which include bending, stretching, twisting, or even combinations of these deformations. Furthermore, the speed at which these deformations occur, their uniformity, degree of nonlinearity, and memory effects contribute to the complexity of the modeling process. Each of these factors plays a pivotal role in determining the most suitable modeling approach for accurately capturing the robot's behavior. For example, continuous models excel at representing complex, nonlinear deformations but might be computationally intensive, while lumped models offer computational efficiency but may struggle with intricate deformations. Hybrid approaches attempt to strike a balance, and AI-based methods leverage extensive data to learn complex deformation patterns. Thus, understanding these deformation characteristics is crucial for selecting the most appropriate modeling technique for soft robots, ensuring accurate representation of their behavior under varying conditions. When choosing a modeling approach, it is important to consider the robot's dominant deformation modes. For soft robots that primarily undergo bending, a continuous or hybrid model may be the best choice. For soft robots that undergo stretching, twisting, or a combination of these deformations, an AI model may be the best choice.

4.6.3 Material Properties

Soft robots are inherently unique due to their construction from compliant materials that possess distinctive mechanical properties. These materials often exhibit behaviors like nonlinear elasticity, viscoelasticity, or other material-specific characteristics that need to be accurately captured by the modeling techniques employed. In this context, the choice of modeling approach becomes paramount. Continuous models, such as those based on the theory of elasticity, offer the capability to describe the complex nonlinearities inherent to these materials, making them suitable for soft robots with intricate material behaviors. Lumped parameter models, on the other hand, may

provide a simplified representation that neglects certain material complexities, offering computational efficiency but at the expense of some accuracy. Hybrid models aim to combine elements of both approaches to strike a balance between accuracy and efficiency. AI methods, leveraging vast datasets, can excel in capturing intricate material behaviors, but their performance depends heavily on the quality and quantity of available training data. Therefore, understanding the specific mechanical properties of the compliant materials used in soft robots is crucial for selecting the most appropriate modeling technique to ensure an accurate representation of their behavior.

4.6.4 Complexity of Deformation

The complexity of a soft robot's deformation is a critical factor when selecting the most suitable modeling technique. Soft robots often exhibit intricate and nonlinear deformation behaviors, which can be influenced by factors like large strains and complex interactions between their segments or modules. In scenarios where these deformations are particularly challenging to capture accurately, more advanced modeling techniques come into play.

For instance, finite element analysis (FEA) is a sophisticated method that excels at simulating highly nonlinear deformations in soft robots. FEA divides the soft robot into smaller, interconnected elements and calculates their deformations and interactions. This approach allows for a detailed and accurate representation of the robot's behavior, making it ideal for scenarios where precision is paramount.

Continuum mechanics is another advanced technique that models soft robots as continuous deformable bodies. It offers the advantage of accurately representing complex deformations, even when large strains are involved. However, continuum mechanics models can be computationally intensive, demanding substantial computational resources.

In cases where the soft robot's deformation behavior is less complex or computational efficiency is a primary concern, other modeling techniques like lumped parameter models or hybrid models may be more appropriate. These methods offer a trade-off between accuracy and computational cost, making them suitable for various soft robot applications.

4.6.5 Computational Resources

The availability of computational resources is a pivotal factor to weigh when selecting an appropriate modeling technique. If you have access to robust, high-performance computers capable of handling large-scale learning-based systems, then learning-based approaches become viable options. Conversely, if your plan involves running

Table 4.2 Summary of modeling approaches and computational resource requirements

Modeling approach	Computational resource requirements
Learning-based	High
Parametric	Low to medium
Reduced-order FEA	Medium to high

controllers on peripheral, low-power computational processing systems, the emphasis shifts towards computational efficiency and the lightweight implementation of a modeling framework. In such scenarios, opting for parametric approaches with a limited number of degrees of freedom can be advantageous. On the other hand, if your application demands higher precision, exploring reduced-order finite element analysis, such as those provided by the SOFA modeling framework, becomes feasible. However, this still necessitates access to reasonably powerful computational resources. In cases where the soft robot is deployed in the field, as is the case with legged or snake-like robots, the modeling and control approach must prioritize lightweight implementation due to the constraints of field deployment and real-time operation. Table 4.2 summarizes the different modeling approaches and their computational resource requirements.

4.6.6 Accuracy and Precision

Accuracy and precision requirements play a pivotal role in the selection of a suitable modeling approach for soft robots, especially in macro-scale applications. Soft robots, owing to their unique ability to conform to their surroundings, often prioritize their adaptability and interaction with physical environments over strict accuracy and precision. In such scenarios, their operation may not demand pinpoint accuracy. However, in applications where precision is of utmost importance, such as minimally invasive surgeries, meticulous consideration must be given to both the modeling approach and the subsequent controllers derived from these models.

Continuous curve models represent soft robots as continuous curves in space, offering high accuracy but at the cost of computational complexity. Lumped parameter models, on the other hand, portray soft robots as networks of masses, springs, and dampers, providing a more computationally efficient alternative, albeit potentially sacrificing accuracy for intricate geometries. Learning-based models leverage machine learning to establish input-output relationships, delivering exceptional accuracy but necessitating substantial training data. Lastly, hybrid models combine elements from different approaches, striking a balance between accuracy and computational efficiency.

In cases where accuracy and precision are paramount, opting for a continuous curve model is typically advisable. However, for scenarios involving highly complex

robots or constrained computational resources, alternatives such as lumped parameter models or hybrid models may present more practical choices. Learning-based models also offer promise in accuracy-critical applications but demand substantial volumes of training data. The selection among these modeling options hinges on a careful evaluation of factors like the soft robot's complexity, available computational resources, and the precise requirements of the application at hand.

With this diverse toolbox at your disposal, you have the flexibility to select the modeling technique that best suits your specific needs, whether they involve optimizing conformable interactions with the environment or meeting the exacting requirements of precision-demanding applications like medical procedures. Your choice of modeling approach should be guided by the unique constraints and objectives of your particular use case, ensuring that your soft robot performs flawlessly within its intended application context.

4.6.7 Dynamic Behavior

It's crucial to recognize that not all robots require complexity to be highly practical in real-world applications. Soft robots, with their unique ability to conform to their environment without causing harm to either themselves or the spaces they operate in, exemplify this concept. Their inherent simplicity, adaptability, and gentle interaction with surroundings make them invaluable in scenarios such as delicate object or food item handling.

However, when the robot's intended tasks involve interactions with dynamic objects or necessitate continuous motion, it becomes imperative to incorporate dynamic modeling into the robot's design. Without dynamic modeling, the controllers may encounter instabilities during operation, leading to unpredictable and potentially unsafe behavior. Similarly, if the robot is intended for environmental sensing applications, particularly through deflection-based force estimation, dynamic modeling assumes a pivotal role. Thus, within the confines of application-specific criteria and constraints, it becomes essential to carefully evaluate the requirements and potential challenges. By doing so, one can make an informed decision about selecting the most suitable modeling approach that aligns with their objectives and effectively addresses the demands of the application.

4.6.8 Control and Actuation

Different modeling approaches for soft and continuum robots have varying implications for control and actuation. Continuous curve models, which represent soft robots as continuous curves, are apt for intricate geometries but can be computationally intensive, limiting real-time control feasibility. In contrast, lumped parameter

models, depicting robots as networks of masses, springs, and dampers, offer computational efficiency suitable for real-time control, albeit at the potential expense of accuracy, particularly for complex geometries. Learning-based models, driven by machine learning, excel in precision but demand copious training data and computational resources. Hybrid models, amalgamating elements of diverse approaches, strike a balance between accuracy and computational efficiency. The selection hinges on the specific control objectives, computational resources, and the desired trade-off between accuracy and efficiency within the soft robotic system.

4.6.9 Application-Specific Requirements

It is important to note that there is no single "best" modeling approach for soft and continuum robots. The best approach will vary depending on the specific application-specific requirements. Therefore, it is important to thoroughly evaluate the requirements of your application before choosing a modeling approach.

Thus, choosing the right modeling approach in soft robotics hinges on the specific application domain. In medical contexts like surgical robots and rehabilitation devices, precision and safety are paramount, making Finite Element Analysis (FEA) an ideal choice. Soft robotics research, characterized by rapid prototyping, benefits from computationally efficient lumped parameter models. On the other hand, real-time control in soft robots leans toward reduced-order models, ensuring fast and accurate predictions. Each domain necessitates a tailored modeling strategy that aligns with its unique requirements, emphasizing the adaptability and versatility of soft robotics modeling.

4.6.10 Existing Literature and Tools

Review existing literature and available modeling tools. Leveraging established techniques and software can expedite the modeling process and ensure compatibility with existing research and development efforts.

The necessity of adopting common, user-friendly, well-supported, and freely available tools for modeling soft robots cannot be overstated. Such tools not only lower the entry barriers for researchers and engineers but also foster collaboration and knowledge exchange within the soft robotics community. By utilizing widely accepted software and platforms, we ensure the reproducibility and transparency of research outcomes, making it easier for others to validate and build upon existing work. Moreover, freely accessible tools democratize access to cutting-edge soft robot modeling capabilities, driving innovation and advancing the field collectively.

Continuous Curvature Models: Leveraging symbolic computation software tools such as Maple, Mathematica, Maxima (an open-source alternative), SageMath (another open-source option), and MATLAB Symbolic Toolbox is paramount in

deriving symbolic equations for continuous curvature parametric models for soft and continuum robots. These tools excel in simplifying complex mathematical expressions, significantly reducing the risk of errors in analytical modeling. Moreover, the ability to directly port these derived symbolic equations to common computational frameworks like MATLAB, Python, and C streamlines the integration of mathematical models into practical applications. Researchers and engineers benefit from the seamless transition between symbolic analysis and practical implementation, accelerating the development and validation of soft and continuum robot designs while ensuring accuracy and reliability in their performance predictions.

Lumped Parametric Models: A variety of software tools are available for developing lumped parametric models for soft robots. SOFA (Simulation Open Framework Architecture), known for its versatility, offers an open-source framework suitable for modeling deformable objects, including soft robots with intricate geometries. Gazebo, a widely embraced robotics simulation environment, can be tailored to simulate soft robots employing lumped parametric models. It equips users with robust physics engines and analytical tools for assessing the behavior of soft robots within complex environments. FEBio, as a finite element analysis software, emerges as a robust choice for modeling soft robots using lumped parametric models. OpenAI Gym, developed as a reinforcement learning toolkit, allows simulation and control of soft robots employing lumped parametric models.

Acknowledgements his work is supported in part by the National Science Foundation (NSF) Grants IIS–2008797, CMMI–2048142, CMMI–2132994, and CMMI-2133019.

References

1. MATLAB. https://www.mathworks.com/products/matlab.html, 2023. MathWorks, Inc., Natick, MA
2. Arachchige DDK, Chen Y, Godage IS (2021) Soft robotic snake locomotion: modeling and experimental assessment. In: 2021 IEEE 17th international conference on automation science and engineering (CASE), pp 805–810. IEEE
3. Buckingham RO, Graham AC (2010) Dexterous manipulators for nuclear inspection and maintenance, case study. In: 2010 international conference on applied robotics for the power industry, pp 1–6. IEEE
4. Chiaverini S, Oriolo G, Walker ID (2008) Kinematically redundant manipulators. Springer, Berlin, Heidelberg
5. Chirikjian GS, Burdick JW (1994) A modal approach to hyper-redundant manipulator kinematics. IEEE Trans Robot Autom 10(3):343–354
6. Drotman D, Jadhav S, Sharp D, Chan C, Tolley MT (2021) Electronics-free pneumatic circuits for controlling soft-legged robots. Sci Robot 6(51):eaay2627
7. Falkenhahn V, Mahl T, Hildebrandt A, Neumann R, Sawodny O (2015) Dynamic modeling of bellows-actuated continuum robots using the eulerlagrange formalism. IEEE Trans Robot 31(6):1483–1496
8. Godage IS, Medrano-Cerda GA, Branson DT, Guglielmino E, Caldwell DG (2015) Modal kinematics for multisection continuum arms. Bioinspiration Biomimetics 10(3):035002
9. Godage IS, Medrano-Cerda GA, Branson DT, Guglielmino E, Caldwell DG (2016) Dynamics for variable length multisection continuum arms. Int J Robot Re 35(6):695–722

10. Godage IS, Walker ID (2015) Dual quaternion based modal kinematics for multisection continuum arms. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 1416–1422. IEEE
11. Godage IS, Webster RJ, Walker ID (2019) Center-of-gravity-based approach for modeling dynamics of multisection continuum arms. IEEE Trans Robot 35(5):1097–1108
12. Godage IS, Wirz R, Walker ID, Webster III RJ (2015) Accurate and efficient dynamics for variable-length continuum arms: a center of gravity approach. Soft Robot 2(3):96–106
13. Gordon WJ, Riesenfeld RF (1974) B-spline curves and surfaces. In: Computer aided geometric design, pp 95–126. Elsevier
14. Grazioso S, Di Gironimo G, Siciliano B (2019) A geometrically exact model for soft continuum robots: the finite element deformation space formulation. Soft Robot 6(6):790–811
15. Hansen S (2020) Continuum robot offers brand new possibilities
16. Janabi-Sharifi F, Jalali A, Walker ID (2021) Cosserat rod-based dynamic modeling of tendon-driven continuum robots: a tutorial. IEEE Access 9:68703–68719
17. Lee Y, Koehler F, Dillon T, Loke G, Kim Y, Marion J, Antonini MJ, Garwood IC, Sahasrabudhe A, Nagao K et al. (203) Magnetically actuated fiber based soft robots. Adv Mater, p2301916
18. Li S, Awale SA, Bacher KE, Buchner TJ, Santina CD, Wood RJ, Rus D (2022) Scaling up soft robotics: a meter-scale, modular, and reconfigurable soft robotic system. Soft Robot 9(2):324–336
19. Liu Z, Lu Z, Karydis K (2020) Sorx: a soft pneumatic hexapedal robot to traverse rough, steep, and unstable terrain. In: 2020 IEEE international conference on robotics and automation (ICRA), pp 420–426. IEEE
20. Luo M, Agheli M, Onal CD (2014) Theoretical modeling and experimental analysis of a pressure-operated soft robotic snake. Soft Robot 1(2):136–146
21. Luo S, Edmonds M, Yi J, Zhou X, Shen Y (2020) Spline-based modeling and control of soft robots. In: 2020 IEEE/ASME international conference on advanced intelligent mechatronics (AIM), pp 482–487. IEEE
22. Marchese AD, Katzschmann RK, Rus D (2014) Whole arm planning for a soft and highly compliant 2d robotic manipulator. In: 2014 IEEE/RSJ international conference on intelligent robots and systems, pp 554–560. IEEE
23. McMahan W, Chitrakaran V, Csencsits M, Dawson D, Walker ID, Jones BA, Pritts M, Dienno D, Grissom M, Rahn CD (2006) Field trials and testing of the octarm continuum manipulator. In: Proceedings 2006 IEEE international conference on robotics and automation, 2006. ICRA 2006, pp 2336–2341. IEEE
24. Murray RM, Li Z, Sastry SS (2017) A mathematical introduction to robotic manipulation. CRC Press
25. Oliver-Butler K, Epps ZH, Rucker DC (2017) Concentric agonist-antagonist robots for minimally invasive surgeries. In: Medical imaging 2017: image-guided procedures, robotic interventions, and modeling, vol 10135, pp 270–278. SPIE
26. Ouyang B, Liu Y, Sun D (2016) Design of a three-segment continuum robot for minimally invasive surgery. Robot Biomimetics 3:1–4
27. Perera DM, Arachchige DDK, Mallikarachchi S, Ghafoor T, Kanj I, Chen Y, Godage IS (2023) Teleoperation of soft modular robots: Study on real-time stability and gait control. In: 2023 IEEE international conference on soft robotics (RoboSoft), pp 01–07. IEEE
28. Polygerinos P, Wang Z, Galloway KC, Wood RJ, Walsh CJ (2015) Soft robotic glove for combined assistance and at-home rehabilitation. Robot Auton Syst 73:135–143
29. Price K, Peine J, Mencattelli M, Chitalia Y, Pu D, Looi T, Stone S, Drake J, Dupont PE (2023) Using robotics to move a neurosurgeon's hands to the tip of their endoscope. Sci Robot 8(82):eadg6042
30. Richard R (2019) Autonomous robotic catheter blazes trail
31. Rucker DC, Webster III RJ (2011) Statics and dynamics of continuum robots with general tendon routing and external loading. IEEE Trans Robot 27(6):1033–1044
32. Scharff RBN, Doubrovski EL, Poelman WA, Jonker PP, Wang CCL, Geraedts JMP (2017) Towards behavior design of a 3d-printed soft robotic hand. In: Soft robotics: trends, applications and challenges: proceedings of the soft robotics week, April 25–30, 2016, Livorno, Italy

33. Schiller L, Seibel A, Schlattmann J (2019) Toward a gecko-inspired, climbing soft robot. Frontiers Neurorobotics 13:106
34. Shepherd RF, Ilievski F, Choi W, Morin SA, Stokes AA, Mazzeo AD, Chen X, Wang M, Whitesides GM (2011) Multigait soft robot. Proceedings of the national academy of sciences, 108(51):20400–20403
35. Shintake J, Cacucciolo V, Shea H, Floreano D (2018) Soft biomimetic fish robot made of dielectric elastomer actuators. Soft Robot 5(4):466–474
36. Song S, Li Z, Meng MQ-H, Yu H, Ren H (2015) Real-time shape estimation for wire-driven flexible robots with multiple bending sections based on quadratic bézier curves. IEEE Sensors J 15(11):6326–6334
37. Sun H, Chen X-P (2014) Towards honeycomb pneunets robots. In: Robot intelligence technology and applications 2: results from the 2nd international conference on robot intelligence technology and applications, pp 331–340. Springer
38. Thalman CM, Lam QP, Nguyen PH, Sridar S, Polygerinos P (2018) A novel soft elbow exosuit to supplement bicep lifting capacity. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 6965–6971. IEEE
39. Walker ID (2013) Continuous backbone "continuum" robot manipulators. International scholarly research notices, 2013
40. Webster III RJ, Jones BA (2010) Design and kinematic modeling of constant curvature continuum robots: a review. Int J Robot Res 29(13):1661–1683
41. Yeshmukhametov A, Koganezawa K, Yamamoto Y (2019) A novel discrete wire-driven continuum robot arm with passive sliding disc: design, kinematics and passive tension control. Robotics 8(3):51
42. Yuan H, Zhou L, Wenfu X (2019) A comprehensive static model of cable-driven multi-section continuum robots considering friction effect. Mechanism Mach Theory 135:130–149
43. Zhang J (1999) C-bézier curves and surfaces. Graphical Models Image Process 61(1):2–15
44. Zhang Z, Tang S, Fan W, Xun Y, Wang H, Chen G (2022) Design and analysis of hybrid-driven origami continuum robots with extensible and stiffness-tunable sections. Mechanism Mach Theory 169:104607