

Thrishantha Nanayakkara *Editor*

Handbook on Soft Robotics

Handbook on Soft Robotics

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>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

If disposing of this product, please recycle the paper.

We dedicate this book to our families for their support and patience to finish this work

Foreword

Soft deformation is pervasive in our daily lives. From our clothes, beds, pillows, sofas, and toys—virtually everything we touch—is soft. Even our foods, such as bread, rice, vegetables, and meat, tend to be soft, with rigid substances being less common in our diets. Moreover, our physical bodies, including our skin, hair, muscles, and most other tissues, exhibit soft characteristics. Consequently, soft deformation has captured the interest of scientists for many centuries. Its early history includes the formulation of Hooke’s law of elasticity in the seventeenth century, which models how materials deform and store or release energy when subjected to external forces. Since then, the fields of mechanics and material science have further developed various deformation models in theoretical terms, leading to a significant research community focused on soft matter physics. Additionally, disciplines such as chemistry, biology, and medical sciences have explored soft material deformation for their respective applications.

Despite the centuries-long history of soft deformation across various disciplines, robotics engineers have only recently “discovered” the fascinating potential of soft materials in their field. One of the earlier soft robotics research endeavors, led by Koichi Suzumori, proposed an actuator mechanism composed of silicone rubber, the so-called “rubber-tuators,” in the 1980s. However, this research did not immediately gain traction as a mainstream robotics endeavor. Soft materials were primarily utilized in various components of rigid robots, such as the foot-ground contacts of legged robots, the finger surfaces of robotic grippers, and the interfaces of human–user interactions. Yet, these applications of soft materials were predominantly aimed at mitigating failures or defects within rigid systems, rather than being actively exploited for the core functionalities of these systems.

Robot designers, developers, and researchers had reservations for a long time about incorporating soft elements into robotic systems for several reasons. Rigid materials were perceived as offering greater stability, predictability, and ease of mathematical modeling and manufacturing. In contrast, soft materials were prone to degrade more easily and quickly. Additionally, the mathematical complexity of continuum bodies posed challenges. Moreover, soft structures were unable to efficiently transmit forces over long distances, unlike metal gears, for example, thereby

not leading to the development of stronger, faster, more precise, and larger robots—qualities that were paramount in the robotics industry. Consequently, there was initially limited motivation to actively explore the use of soft materials in robotic research.

It took several decades after Suzumori's seminal work for Soft Robotics to emerge as a major research area within robotics. I consider myself fortunate to have been a researcher during this dynamic period of emergence. The journey commenced with a small workshop organized at the University of Tokyo in 2010, where approximately 50 innovative robotics researchers from Asia and Europe convened to discuss the challenges and prospects of soft robotics. Two years later, we organized the inaugural international workshop on Soft Robotics in Ascona, Switzerland, which attracted around 120 leading researchers worldwide to deliberate on the significance and impacts of Soft Robotics. In 2013, we established the first worldwide Soft Robotics community, the IEEE Robotics and Automation Systems Technical Committee on Soft Robotics, initially comprising about 100 members, which has since grown to over 1000 members. The same year saw the launch of the Journal of Soft Robotics. Following these dynamic years, we successfully inaugurated the first IEEE International Conference on Soft Robotics (RoboSoft Conference) in 2018, which has since evolved into an annual conference series, attracting over 300 participants each year.

What precipitated this exciting evolution in the field of soft robotics? Why did it happen in this timing, and how did it unfold? The pivotal moment in this movement occurred when a significant number of interdisciplinary researchers from diverse fields beyond robotics began engaging with soft robotics due to its cross-disciplinary nature. Material scientists and chemists, previously focused primarily on discovering new materials, started integrating these materials into systems of actuators, sensors, and controls. Computer scientists, who had hitherto worked predominantly on simulated robots, ventured into the physical realm by constructing physical robots. Simultaneously, robotics engineers began earnestly exploring materials and chemical substances to enhance the performance of their robots. Collectively, they were enthusiastic about developing new types of robots and embraced the diverse approaches of interdisciplinary researchers, deepening our understanding of the research challenges in the process.

These interdisciplinary collaborations were made possible by significant technological advancements. Low-cost 3D printing devices, commercially accessible soft functional materials, high-performance computers, and simulation tools were among the enabling technologies that facilitated the entry of new researchers into this exciting research field. Consequently, as a united soft robotics community, we collectively tackled the diverse challenges in this field, spanning design, fabrication, modeling, applications, materials, and innovative concepts such as growth, evolution, and bio-hybrid systems. Reflecting on the past decade or so, while the starting point was the concept of soft deformation, the underlying driver of research evolution lay in the recognition that material development was essential in robotics research. This shared view among interdisciplinary researchers enabled impressive progress in this field. While not all ideas proved successful, several crucial branches of research

emerged as the mainstays and cornerstones of the field of Soft Robotics, many of which are meticulously detailed in this book.

When a new technology is introduced, it is often overestimated initially in a short timescale. Indeed, soft robotics was expected to yield numerous human-friendly killer applications and incredibly adaptive robots capable of interacting with unstructured environments like humans and animals. However, many of these envisioned dreams have yet to materialize despite the exciting seminal decades in this field. Nevertheless, a new technology is also often underestimated in the longer term. Softness is intrinsic to nature, particularly in human-oriented environments. Consequently, we cannot afford to overlook this technology for long. While considerable challenges remain before soft robots achieve the impact originally envisioned, I am confident that we will overcome these challenges for broader impacts.

This book represents a historical milestone toward the bright future of this research field. I hope that students and researchers alike will gain insight into the historical trajectory of soft robotics, as well as the challenges and perspectives shaping this field, by studying this book. With the shared knowledge and experiences shared in this book, we together can make our world a better place.

February 2024

Fumiya Iida
Professor of Bio-Inspired Robotics
Lab, Department of Engineering
University of Cambridge
Cambridge, UK

Preface I

The foundation of embodied intelligence is a ghost circuit of different dynamic systems. Ghost circuits come together when conditions are right and dissolve otherwise. Intelligence is a temporary perception rendered by a ghost circuit while it exists.

Let me unpack it a little more:

The metaphorical term “ghost” is used to mean that the circuit can come together and dissolve.

A “circuit” has various pathways, nodes, and loops for signal propagation.

An important “condition” for a ghost circuit to come together is the smooth propagation of signals through the boundaries of different dynamic systems. The idea of “softness” plays a crucial role to mediate this smooth signal flow.

Embodied “ghost circuits” are everywhere. They come together on their own, continue their dynamic interactions on their own, dissolve on their own, and evolve on their own. The assessment of their “intelligence” remains subjective, contingent upon the intricacies perceived by a conscious observer—a judgment that resists replication in identical form, as the observer becomes an integral part of the circuit.

This book provides useful analytical and conceptual frameworks to design useful ghost circuits and their dynamic systems.

London, UK

Thrishantha Nanayakkara

Preface II

With the emergence of robotics, scientists began drawing inspiration from biology to create robotic counterparts aimed at emulating the behaviors and survival strategies observed in living organisms. However, it became evident that in numerous instances, these robotic counterparts fell far short of replicating the complexity and efficiency seen in their biological counterparts. Within the robotics community, it was frequently hypothesized that constraints such as computational processing speed, actuator bandwidth, sensor noise, and other factors might be the primary factors contributing to this disparity.

In the present day, we benefit from highly sophisticated control theories, accurate and precise sensors, advanced actuator technologies, and powerful computers. These building blocks enable the execution of intricate sensor-driven control algorithms, facilitating precise positioning of robots for tasks like pick-and-place operations within industrial settings. Moreover, symbolic artificial intelligence, exemplified by deep learning, now rivals human capabilities in areas ranging from memorization, pattern recognition, to generative AI.

However, even with these remarkable advancements, robots continue to encounter significant challenges when it comes to interaction scenarios involving physical contact dynamics. The field of soft robotics aims to bridge the gap between robotic systems and their biological counterparts, particularly in situations involving physical contact.

It is important to acknowledge that soft robots occupy a distinct realm of applications separate from the domain of rigid body robots, which excel in precision and force capabilities. Consequently, soft robots should not be perceived as a progression beyond rigid robots. Instead, they should be regarded as a distinct category of robots ideally suited for specific tasks that leverage the advantages of mechanical structures that can emerge compliance given a certain range of forces and their frequencies.

Furthermore, this book suggests that soft robots offer a promising platform for investigating biological hypotheses concerning the nexus of embodiment and computation as a crucial phenomenon for survival in natural environments.

In this book, we observe that biological beings and their typical ecological niches are well poised to emerge coupled dynamic behaviors without much need for a central

arbitrator. For instance, the passive dynamics of a fish that swims against a turbulent current and the vortexes in the stream of water tend to complement each other to emerge swimming actions against the current even in a dead fish. A mountain goat hoof would exhibit passive reshaping and vibrations against a steep cliff to generate forces against a slip. This book introduces three new terms.

One is the term called “ghost circuits” to explain meaningful networked dynamic phenomena when the right parts come together with physical coupling. A ghost circuit is a coupled mechanical system that can emerge useful coupled dynamics among systems in a system such as in the fish and mountain goat hoof examples. The important observation is that parts of a meaningful mechanical circuit to accomplish an intended dynamic behavior are distributed among the agent(s) and the environment. An arbitrator such as a central nervous system (CNS) maybe useful to tune and maintain a useful coupling. The term “ghost” is used to imply that it is hard to see the grand meaning of these circuits by taking parts in isolation and the grand mechanical circuit functions when the conditions are right and dissolves when the conditions are no longer present. For instance, the automatic braking of a mountain goat may not be successful if the joints in the hoof had a wrong range of stiffness or if the hoof was in the wrong orientation.

Akin to ghost circuits, we introduce the term “kinematic tuning” to explain how biological beings use a class of motor commands to tune the shape and stiffness of the body to support a concurrent class of motor commands given to control movements and interaction forces.

The third term we introduce is “behavioral lensing” to embody the phenomenon of movement and force control for better perception. For instance, when tasked with gauging the weight of an object, individuals instinctively engage in a gentle up-and-down motion before reaching a conclusion. Similarly, in the context of soft tissue palpation, the use of finger wiggling, as opposed to just applying pressure, aids in the assessment of tissue conditions. These dynamic interactions, integral to state estimation tasks, are underpinned by the underlying mechanisms that intertwine perception and action at a shared embodiment level.

This handbook on soft robots gives essential tools a reader can use to build and control soft robots. In Chap. 1, readers are introduced to the foundational concepts of soft robotics. Chapter 2 offers valuable insights into the derivation of design principles from biological counterparts, facilitating the creation of innovative soft robotic systems. The concept of embodied intelligence, illustrated with practical examples, is elaborated upon in Chap. 3. This chapter also proposes soft robots as a useful paradigm to test biological hypotheses due to the advantages in soft material to represent a controllable embodiment.

Moving forward, Chap. 4 presents four broad classes of methods available for kinematics and dynamics modeling of soft robots. Chapter 5 delves into approaches for modeling soft structures, providing readers with a robust foundation for understanding their behavior. Chapter 7 furnishes essential design guidelines tailored to soft grippers, enabling the development of versatile and adaptable grasping solutions. In Chap. 6, readers gain insights into designing soft fingers and sensors, essential components for enhancing dexterity and sensory capabilities in soft robotic systems.

Chapter 8 delves into the technical details of modeling and controlling continuum robots, offering specialized knowledge relevant to the construction sector. Finally, in Chap. 9, we show how origami-inspired approaches can be used to design soft robots.

London, UK

Thrishantha Nanayakkara

Contents

1	Introduction	1
	Thrishantha Nanayakkara	
2	Bioinspiration	13
	Cecilia Laschi and Barbara Mazzolai	
3	Soft Robots as a Platform to Understand Embodied Intelligence	35
	Thrishantha Nanayakkara, Barry Mulvey, Shehara Perera, Yukun Ge, Zhenhua Yu, and Parvathi Sunilkumar	
4	Modeling Soft Robots	85
	Isuru Godage and Hunter Gilbert	
5	Mechanical Compliance: From Soft Robot Modeling Theory to FEM Computation	141
	Christian Duriez	
6	Soft Hands for Grasping and Manipulation	179
	Shinichi Hirai	
7	Malleable Robots	227
	Angus B. Clark, Xinran Wang, Alex Ranne, and Nicolas Rojas	
8	Continuum Soft Robots with Applications in the Construction Industry	281
	Manu Srivastava and Ian D. Walker	
9	Origami Robots	315
	Cynthia Sung and Jamie Paik	
Index		345

List of Figures

Fig. 1.1	Tō-ji is a Shingon Buddhist temple in the Minami-ku ward of Kyoto, Japan. Founded in 796—survived earthquakes for hundreds of years. Its layers can slide against each other mimicking a serpentine skeleton	2
Fig. 1.2	Some soft robot examples; a The tendons, joints, and finger pads of this underactuated adaptive gripper are 3D printed on an entry-level Prusa i3 printer with soft TPU filament [31], b A robotic patient with a soft abdomen and controllable organs to train physicians [9, 14], c A 6-DOF hybrid manipulator, combining a tendon-driven segment [17] with two pneumatically driven soft segments, is developed for Lower GI interventions. It merges the benefits of soft and tendon-driven robotic manipulators to achieve intrinsic softness, compactness, and high payload. Notably, this design enables contact forces estimation by monitoring the actuation pressure of the soft segments [8, 18], d A medical examination system for the lower gastrointestinal tract featuring a soft robot for lower GI tract inspections, using an evertting motion and pneumatic control [7] for enhanced patient comfort and precise autonomous navigation [4]. The robot's design aims to minimize discomfort, making the procedure more comfortable for the patient	4
Fig. 1.3	An example of a soft continuum manipulator segment with tendon and pneumatic channels	5
Fig. 1.4	a Hall effect sensors placed orthogonally centering a cylindrical permanent magnet. b Gaussian ellipsoids showing the influence of different texture exploration speeds	7

Fig. 1.5	A example is a soft sensor made with FDM printing technology using flexible conductive TPU. This wearable sensor can be used to monitor finger bending	7
Fig. 2.1	A method for bioinspiration	15
Fig. 2.2	Hill-type muscle model, reproduced with permission from [2]	16
Fig. 2.3	Octopus muscular hydrostat, from [3]	18
Fig. 2.4	The table highlights the biological features, measurements, and characterization methods relevant to self-burying seeds, along with their associated specifications for soft robotic solutions [19]	23
Fig. 2.5	Phases of biped walking, from [20]	24
Fig. 2.6	SLIP model for biped walking, from [21]. a Model of the leg. b Sequence of one leg positions in a full cycle of stance and swing phases	24
Fig. 2.7	A scheme of CPG, reproduced with permission from [23]. The model is composed of 20 oscillators which receive a drive signal from the brain representing the descending stimulation. The outputs of the CPG are desired joint angle positions that, in a robot, can be reached with a PID controller	26
Fig. 2.8	Bioinspired crawling models for soft robots, reproduced from [24]	27
Fig. 2.9	Anatomy of earthworm segment cross-section showing constant volume coelomic chambers and setae and the working principles of the antagonistic muscles	29
Fig. 2.10	Plant root structures and their functions. Four main regions can be identified, each with different roles. Root cap cells exude mucilage and slough off, which decreases friction during penetration by lubricating the surrounding soil. New cells are created by mitosis in the meristematic region. These cells are then elongated by osmotic pressure and move to the elongation zone. Cell division and cell elongation provide the force to penetrate into the soil; asymmetries with respect to the root axis results in bending. The mature root consists of elongated cells that are stationary and provide a strong anchor to the soil, thus supporting tip penetration	30
Fig. 2.11	Human mechanoreceptors have different functions, depending on their class, and their combination enable the detection of contact characteristics and manipulation control events. 90% of type SA-I and FA-I receptors respond to a stimulus of 5 mN, corresponding to 87 mN/mm ² , applied with a Von Frey hair of diameter 0.27 mm	31
Fig. 3.1	A force field	40

Fig. 3.2	Mountain goats on steep slopes in Spain. Courtesy, Akila Anuradha Ekanayake	42
Fig. 3.3	The octopus tentacle rolls out along a straight line to reach a target. Courtesy, Cecilia Laschi	44
Fig. 3.4	A network of springs interacting together to accomplish a computational goal	46
Fig. 3.5	A spider in a web. Courtesy, Parvathi Sunilkumar	47
Fig. 3.6	Interaction between turbulent water and a fish	48
Fig. 3.7	Controllable stiffness tube waving	51
Fig. 3.8	The meaning of eigenvalues and eigenvectors in matrix multiplication	53
Fig. 3.9	Muscle co-contraction during palpation to estimate the depth of a nodule in a soft tissue. a The experimental condition with two electromyography (EMG) sensors placed on antagonistic muscles on the lower arm, b raw data of muscle EMG and co-contraction pattern, and c the state transition probability matrices for nodules buried at different depths	54
Fig. 3.10	How a vector in the state space rotates towards the absorbing state across iterations	55
Fig. 3.11	A general framework for design ideation to improve physical hardware designs of robots based on their interaction with the environment	57
Fig. 3.12	A design iteration framework to delegate control responsibility to hardware. The u_{MC_T} commands to tune the proportional (P) and derivative (D) gains of certain joints can provide useful statistical distributions of their required impedance in a given environment. This information can then be used to design spring-damper networks to provide the same functionality the motors provide with variable gains	58
Fig. 3.13	An alternative design iteration framework to delegate control responsibility to hardware. The u_{MC_T} commands to tune the mechanical impedance of certain joints are visualised using colors (with dark colors being very stiff and light colors being very soft). The gradient of $V(u_{MC_T}, u_{MC_M}, s) \geq \beta$ against u_{MC_T} is proportional to the radius of a transparent blue circle projected onto each joint. The memory consolidation process of such graphical information can then lead to design spring-damper networks to provide the same functionality given by the motors with variable gains	59

Fig. 3.14	The ability of animals—such as cats—to morph their shape in order to traverse spaces smaller than their natural body states is the bio-inspiration for the design of DeforMoBot, a deformable mobile robot	60
Fig. 3.15	Design of the bio-inspired deformable mobile robot DeforMoBot, detailing sensors and components	61
Fig. 3.16	DeforMoBot, a bio-inspired deformable mobile robot, changes its shape depending on its surroundings to squeeze through gaps and traverse obstacles in its path	62
Fig. 3.17	CAD Model: simplified robotic leg including the damper, spring, leg segments and drop weight	65
Fig. 3.18	Snapshots of a drop test for three constant dampers and an angle-dependent variable damper (A) $t = 0$, (B) $t = \text{maximum compression of leg}$, (C) $t = \text{settling minimum of the leg}$, (D) $t = \infty$ (settled position). The angle-dependent variable damper settles with a smaller variation	66
Fig. 3.19	The micromotor in the endoscopy capsule can pull the Nylon thread which runs through the origami structure to control the shape of the origami structure	68
Fig. 3.20	The mechanism of the origami structure is pushed backwards by intestinal peristalsis	70
Fig. 3.21	The mechanism of the origami structure is pushed forwards by intestinal peristalsis	70
Fig. 3.22	After being squeezed by intestinal peristalsis in the intestinal phantom, the origami structure in shape A moves to the left, and the origami structure in shape B moves to the right	71
Fig. 3.23	a The tapered whisker sensor designed for the mobile robot terrain identification and navigation. b Schematic of the reservoir computing system, showing the internal nonlinear reservoir and the linear readout function. In this research, the nonlinear tapered spring works as a reservoir, and the tapered spring induces vibration components of different frequencies along the axial positions of the whisker when there is a time-series stimulus from the base in the vertical direction of the whisker sensor. A whisker-based reservoir computing system can accommodate an arbitrary number of outputs, wherein only the matrix W_t requires training via linear logistic regression. However, in our specific experiment focusing on terrain recognition, we have achieved sufficient accuracy by utilizing only three outputs	73

Fig. 3.24	A differential-drive mobile robot with a bio-inspired tapered whisker sensor vertically installed on the front to achieve reservoir computing-based terrain classification and texture-guided navigation	74
Fig. 3.25	The average classification outcomes of the TWRC system were evaluated over 25 random trials, with the robot moving at a velocity of 0.2 m/s and utilizing a time window duration of 1.5 s	75
Fig. 3.26	Reservoir computing average prediction accuracy of eight terrains when the whiskered mobile robot receives different MC_M commands to change its speeds at 0.1, 0.2, 0.3, 0.4, and 0.5 m/s	76
Fig. 3.27	Experimental setup with the fingertip model affixed to the XY Table with the help of a spring attachment to generate the motion of a finger stroke	76
Fig. 3.28	Fingertip models used for the experiment	77
Fig. 3.29	Hall effect sensors placed orthogonally centering a cylindrical permanent magnet	78
Fig. 3.30	Different textures used for the experiment	79
Fig. 3.31	Heatmaps representing the Frobenius distances between the covariance matrices	81
Fig. 3.32	Gaussian Ellipsoids showing the influence of MC_T	82
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]	87
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]	87

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	92
Fig. 4.4	Joint-space of a continuum/soft robotic section	92
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)	93
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	96
Fig. 4.7	(LEFT) Errors of $[T]_{24}$, given by Eq. (4.18), within singular neighborhood against l_2 when $l_2 \rightarrow 0.03$ 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, $l_{11} = 0.0325$	98
Fig. 4.8	Schematic of a general multisection continuum arm with N continuum sections	98
Fig. 4.9	a Schematic illustration of the infinitesimally thin slice at ξ_i on any i th 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 $\{O'_i\}$	104
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]	107

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	110
Fig. 4.12	Discrete link approximation method	113
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])	114
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	114
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	115
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	115
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	119
Fig. 4.18	Schematic of an infinitesimally thin slice at the CoG of any i th continuum section	121
Fig. 4.19	ANN	125
Fig. 4.20	Octopus robot arm	126
Fig. 5.1	Use mechanical compliance to find the value of a spring equivalent k_{eq} to springs connected in series	143

Fig. 5.2	Kinematics through the mechanical compliance of soft robots: To describe the kinematics of a soft robot, the mechanical compliance of the manipulator structure is projected into actuator space with \mathbf{W}_{aa} and into effector-actuator space with \mathbf{W}_{ea} . Thanks to these two operators, we can obtain the displacements of the actuators ($\Delta\delta_a$) and of the effector ($\Delta\delta_e$) when increments of forces are applied to the actuators ($\Delta\lambda_a$). To illustrate, the figure shows a dummy soft robot, actuated by two linear pistons. The robot's behavior is decomposed in three steps with two different sets of incremental actuator forces on the top and bottom lines. We show that the value of compliances is not constant due to non-linear deformations. In this example, the kinematics of the soft robot can be calculated using the Eq. 5.1	144
Fig. 5.3	Lagrangian view: for any point \mathbf{p} in the reference domain, the application $\mathbf{A}(\mathbf{p})$ provides \mathbf{p}' the unique corresponding point in the deformed domain	146
Fig. 5.4	Illustration of the deformation gradient	147
Fig. 5.5	Right and Left polar decomposition of the deformation gradient	148
Fig. 5.6	Stress corresponds to the internal force exerted on unit areas around the points of the domain. It has to be balanced with the forces at the boundary	149
Fig. 5.7	Examples of elements used in FEM	154
Fig. 5.8	Modeling the stiffening of a part of a deformable model through reduction of DOFs	161
Fig. 5.9	Reduction of the degrees of freedom: In this figure, we suppose that the motion of the soft robot can be captured by a linear combination of 6 modes. The intensity of these modes are the new degrees of freedom	162
Fig. 5.10	The different spaces in which the deformable mechanics of soft robots have been described	164
Fig. 5.11	Pressure based actuator	165
Fig. 5.12	Tendon (Cable) actuator	166
Fig. 5.13	Direct actuation on rigid part	167
Fig. 5.14	Signorini's unilateral law for multicontact case: the three principles of this law are (i) to prevent penetration at contact points, (ii) to have a complementarity between distance and contact force and (iii) to impose repulsive forces. The drawn forces are applied to the yellow object but opposite forces are applied to the green solid	173

Fig. 5.15	The goal position is illustrated by a green circle in both figures. On the left, the obstacle is not modeled in the inverse model. The obstacle prevents the real robot from reaching the goal. On the right, however, contact with the obstacle is taken into account in the inverse model, enabling the real robot to find a solution to the inverse problem of reaching the goal position while sliding over the obstacle	175
Fig. 6.1	Human manipulation	180
Fig. 6.2	Hertzian contact	181
Fig. 6.3	Soft pad hand [8]	183
Fig. 6.4	Jamming hand	185
Fig. 6.5	Bending finger hand [24]	185
Fig. 6.6	Molds for bending fingers	186
Fig. 6.7	Expanding membrane hand	187
Fig. 6.8	Remotely deformed hand	188
Fig. 6.9	Soft pad hand grasping rectangle rigid body	189
Fig. 6.10	Simulation of soft pad hand grasping rectangle rigid body	192
Fig. 6.11	Expansion of an elastic membrane	193
Fig. 6.12	Simulation results of expanding membrane hand	195
Fig. 6.13	Minimum coefficient of friction to pick up a circular object	197
Fig. 6.14	Deformation of PneuNet actuator	197
Fig. 6.15	Orientation of object grasped by soft pad hand [8]	199
Fig. 6.16	Binding hand	201
Fig. 6.17	Binding hand grasping and picking up a paper cup with beans [40]	202
Fig. 6.18	Scooping binding hand	203
Fig. 6.19	Scooping-binding hand grasping food materials a octopus leg, b pollock roe, c raw oyster, d meat, e fried shrimp, f green pepper [41]	203
Fig. 6.20	Finger of planar shell hand [42]	204
Fig. 6.21	Planar shell hands picking cucumbers [42]	204
Fig. 6.22	Circular shell hand	205
Fig. 6.23	Circular shell hand pick and place a rice cake	205
Fig. 6.24	Curved finger hand [24]	206
Fig. 6.25	Curved finger hand grasping food items [24]	206
Fig. 6.26	Wrapping hand	207
Fig. 6.27	Wrapping hand picking up cones [43]	207
Fig. 6.28	Pouring water into a paper cup grasped by human hand	208
Fig. 6.29	Resistive sensor for extension	209
Fig. 6.30	Function $2\epsilon_L/(1 - \epsilon_L)$	210
Fig. 6.31	Resistive sensor for bending	211
Fig. 6.32	Circuits to convert resistance to voltage	212
Fig. 6.33	Capacitive sensor	213
Fig. 6.34	Relationship between ΔC and Δd in capacitive sensor	214

Fig. 6.35	Capacitive extension sensor	215
Fig. 6.36	Principle in proximity sensing	216
Fig. 6.37	Voltage divider circuit for capacitance sensing	216
Fig. 6.38	Circuit converting capacitance to voltage	217
Fig. 6.39	Response of a circuit converting capacitance to voltage	218
Fig. 6.40	Magnetic flux density based sensing	219
Fig. 6.41	Inductance based sensing	219
Fig. 6.42	Voltage divider circuit for inductance sensing	220
Fig. 6.43	Fabric sensor	221
Fig. 6.44	Soft capacitive force sensor [57]	221
Fig. 7.1	A two-degree-of-freedom (DOF) malleable robot arm, showing various topology configurations it can achieve. The positioning of the revolute axes for each configuration are highlighted	229
Fig. 7.2	Double-sided flap pattern specifications for layer jamming sheath with guide holes and slots	232
Fig. 7.3	Cut-away malleable link highlighting individual components	233
Fig. 7.4	Neutral spine position (a), compressed spine (b), and extended spine (c)	234
Fig. 7.5	Sectional view of 2-segment spine with integrated flexible ligaments (square planar ligament connections)	235
Fig. 7.6	Motion of the flexible spine. Compression (a), extension (b), bending (c), and twist (d)	236
Fig. 7.7	Diagram highlighting the key elements that compose the layer jamming structure	239
Fig. 7.8	Cleanup options when exporting a DXF from SOLIDWORKS	240
Fig. 7.9	Diagram showing the sewing route (green) for the layer jamming. External flaps are not shown to increase visibility	241
Fig. 7.10	The parts of the spine: a the rigid component, b the flexible components, and c the assembled spine	242
Fig. 7.11	A dihedral angle ϕ of the tetrahedron $\triangle_{i,j,k,l}$ defined by the two triangles $\triangle_{i,j,k}$ and $\triangle_{i,k,l}$. Base vectors $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,k}$ and output vectors $\mathbf{p}_{i,l}$, $\mathbf{p}_{j,l}$, and $\mathbf{p}_{k,l}$ are shown. In this case, $V_{i,j,k,l} > 0$	245

Fig. 7.12	The 2-DOF malleable robot arm can be modelled as a bar-and-joint framework formed by connecting 5 points: P_1 and P_2 , which define the first axis; P_3 and P_4 , which define the second axis; and P_5 , which corresponds to the centre of the end effector. An additional offset point P_0 is added, which is used to define the angle of the first axis. The distances between points can then be categorised as constant distances that do not change (black), distances that vary based on the topology configuration of the robot (red), and distances that vary based on the joint positioning of the robot (blue)	248
Fig. 7.13	Simulated example workspaces for each type of robot topology achievable by a 2-DOF malleable robot: a Spherical, b PUMA-like, c SCARA, and d General Articulated [7]	250
Fig. 7.14	The developed augmented reality-assisted reconfiguration system of a malleable robot. The user is shown interacting with the AR assisted points placement scene overlaid on top of the robot base, with the presence of a transparent configuration space. The lectern to the right prints out the instructions for the task the user needs to conduct	254
Fig. 7.15	The previously developed 2-DOF malleable robot to be reconfigured, highlighting the point ($P_1 - 5$) representation and interpoint distances ($D_{13}, D_{14}, D_{23}, D_{24}$) that define the malleable link topology	257
Fig. 7.16	Key AR components	257
Fig. 7.17	The Coordinate system of the scene, where $\{H\}, \{B\}, \{L\}, \{W\}, \{E\}$ represents the frames for the Hololens, the robot base, the lectern, the world origin and the end-effector, respectively	259
Fig. 7.18	Involute and Cycloid curves traced by the malleable link in both maximum and minimum lengths, demonstrating how the reconfigurable space of $P_3/4$ (blue) and P_5 (green) were theoretically calculated	261
Fig. 7.19	A summary of the reconfiguration algorithm	264
Fig. 7.20	a Design of flexible spine. b Tendon path and cross-section area in flexible spine	266
Fig. 7.21	a Feedforward neural network. b Recurrent neural network. c Unfolded recurrent neural network	271
Fig. 7.22	a Control policy A. b Control policy B. c Control policy C	275
Fig. 8.1	Classification of 3DCP techniques [1]	283
Fig. 8.2	A continuum robot grasping and manipulating a fragile (glass) object, using its compliant backbone to adapt to the object's shape [10]	285
Fig. 8.3	Concentric tube continuum robot concept [11]	288

Fig. 8.4	Intrinsically actuated three-section, nine-degree-of-freedom pneumatic “Octarm” continuum manipulator [10]	289
Fig. 8.5	Examples of continuum hoses (clockwise) 1. Refueling a boat [71] 2. Cleaning the dust on top of solar panels [72], 3. Continuum sink faucet, 4. Vacuum gripper [32]	292
Fig. 8.6	Continuum hose (green element) in the construction environment [74]	293
Fig. 8.7	Industrial cement hose (black/green) modified to continuum robot section. Note external tendons (not shown in this image) pass through specially designed spacers (silver) [74]	298
Fig. 8.8	Detail of spacer design and tendon routing for continuum cement hose. Tendons pass through bearing assembly within collars (a), which are fixed to the cement hose backbone with hose clamps (b). Each section is actuated by three steel tendons, equally spaced apart around the circumference of the hose [74]	299
Fig. 8.9	Drive assembly. Motors (two shown, at left and bottom of image), connected through gearboxes, rotate capstans around which the actuating tendons are wound. One capstan is shown center top, on which a tendon and its termination point on the capstan can be seen [74]	299
Fig. 8.10	Effect of implementation of purely kinematic modeling on a printed structure. The desired structure was a cylinder [74]	300
Fig. 8.11	Extent of axial compression of the continuum hose shown by the protruding ball suspended on a string routed through the center of the hose. The picture on the left is at initial configuration with tendons barely under any tension and the picture on the right shows the extent of axial compression [82]	301
Fig. 8.12	Notation for single section robot. A section of nominal length s is actuated by three tendons with lengths l_i , to bend in a plane ϕ , with bending angle θ . Reference frame at the base (proximal) end of the section is shown, with z axis coincident with the tangent to the section backbone at the base [83]	302
Fig. 8.13	Section bending in plane of one of the tendons with the effects of axial compression illustrated. The tendon is retracted by an amount $\Delta l = R\theta_{des}$ and tension, T , develops in it as the backbone compresses. This tension is lesser than the tension developed if the backbone were incompressible [83]	304

Fig. 8.14	Compressed backbone state while bending in the midplane between two tendons. If tendon 3's length is controlled using compression unaware kinematics, it would be slack $\forall \phi \in [0, 2\pi/3]$ including at the midplane, $\phi = \pi/3$ [83]	305
Fig. 8.15	Radial offset of the tendons while bending at the midplane [83]. Note that the moment arm through which the tendon tensions here create the moment at the center is half that of the moment arm when the bending is directly in the plan of a tendon	306
Fig. 8.16	3D printing a cylinder using the EC model. We see that the actual trajectory is similar to the desired circular trajectory. The continuum section is able to maintain relatively constant EE pitch and speed at all bend planes which can be inferred from the circular shape and evenness of material deposition [83]	309
Fig. 9.1	Overview of engineering subfields that contribute to origami robotics	317
Fig. 9.2	Basic folds and origami patterns used in mechanisms and robot design	318
Fig. 9.3	Representation of a fold pattern. The fold pattern is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_{n_v}\}$ are the vertices of the fold pattern in the 2D sheet, and \mathcal{E} are the boundary and crease edges. The creases \mathbf{c} separate the sheet into faces \mathcal{F} . The pattern can also be simplified into a graph of faces and creases in order to compute its kinematics	319
Fig. 9.4	Curved crease pattern with ruling lines drawn in gray. In order to compute the folded state, the angles between the tangent to the curved crease \mathbf{t} and the ruling lines on the left and right of the curve must be known	321
Fig. 9.5	Origami structures consist of faces and folds, which impose a natural discretization of the structure. Common models for these structures either a treat the structure as rigid faces connected at perfect spring-like hinges or b model facial deformation using a combination of linear and torsional springs for stretch, shear, and bending	324
Fig. 9.6	Example of a bellows origami pattern with programmable stiffness. Left: Force vs. displacement of REBO bellows [80] folded from 0.13 mm thick PET with 16 sides, 1 segment, and an outer radius of 60 mm. Fold pattern unit is shown as a gray inset. Right: Photo of 12-sided bellows with 9 segments. The compliance of the bellows structure allows it to bend into an S-shape under tendon-driven actuation [51]	327

Fig. 9.7	Laminate fabrication process developed to attach detailed circuitry on folded structures. Layers are fused into a single electromechanical sheet. These layers may include electronics, passive plastic or metal substrates, or active materials such as shape-memory polymers, hydrogels, dielectric elastomer actuators, and more. Depending on the geometry, gaps are cut into the different layers to expose material and create hinges. Right: Cross-section of the laminate showing gaps in material layers and the resulting folding direction	330
----------	---	-----

List of Tables

Table 3.1	Frobenius distances between covariance matrices of Model F1	79
Table 3.2	Frobenius distances between covariance matrices of Model F2	80
Table 3.3	Frobenius distances between covariance matrices of Model F3	80
Table 3.4	Frobenius distances between covariance matrices of Model F4	81
Table 4.1	Comparison of soft robot modeling approaches	131
Table 4.2	Summary of modeling approaches and computational resource requirements	134
Table 6.1	Calculated grasping forces and friction parameters	196
Table 7.1	Variable stiffness technologies	231
Table 7.2	Specifications of layer jamming pattern parameters	239
Table 8.1	Symbol definitions	303

Chapter 1

Introduction



Thrishantha Nanayakkara

1.1 The Notion of Softness/Adaptive Softness

1.1.1 *Historical Applications of Softness*

Softness has a meaning relative to the forces and their frequencies an object experiences. More specifically, an object is called soft if it can deform given the level of external forces and their frequencies. A structure can appear to be stiff, but show softness at a certain level of forces or in a certain range of frequencies of forces to survive. This idea has been used by ancient engineers who built the Tō-ji temple in Japan shown in Fig. 1.1 to survive earthquakes. The temple has survived for more than 400 years of earthquakes due to its design that allows layers shown in the highlighted area of Fig. 1.1 to slide against the other like the skeleton of a serpentine. Unless there is a lateral force large enough to overcome the friction between layers, the building appears to be a solid structure because $F_s = \mu F_n$, where F_s is the static friction force, μ is the friction coefficient, and F_n is the weight above the layer. Once the structure experiences a lateral force $F > \mu F_n$, the layer would slide and other forms of dynamic friction forces will come into play. The common model for dynamic friction is the Coulomb friction model given by $F_c = cv$, where F_c is the Coulomb friction force, c is the Coulomb friction coefficient, and v is the velocity of sliding. Therefore, high frequency oscillations will have larger friction forces to stop the movement. Since work done against friction is dissipated, the energy given by the earthquake does not get stored in the form of elastic energy in the building to break it.

The ancient Chinese philosophy on *Yin and Yang* also points at the nature of opposite phenomena co-existing in a balanced system. How grass and tree branches bend (*Yin*, the compliance) in the wind (*Yang*, the force) letting it through than resisting and

T. Nanayakkara (✉)
Imperial College London, London, UK
e-mail: t.nanayakkara@imperial.ac.uk

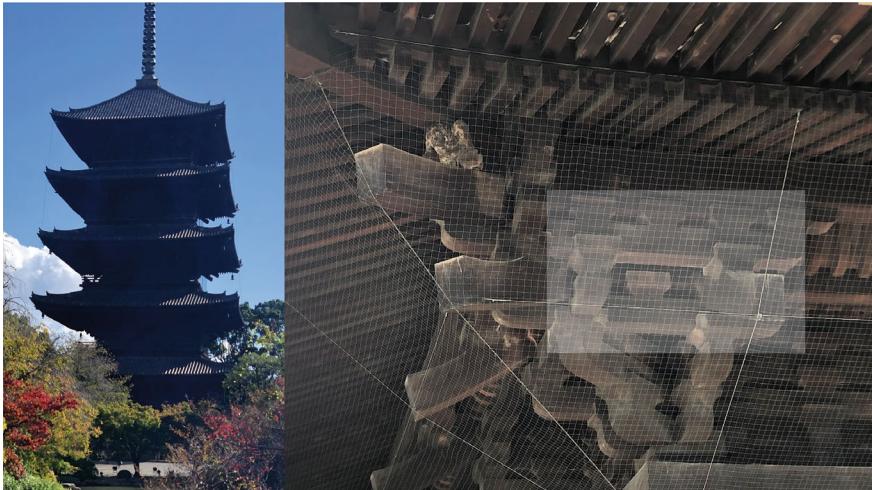


Fig. 1.1 Tō-ji is a Shingon Buddhist temple in the Minami-ku ward of Kyoto, Japan. Founded in 796—survived earthquakes for hundreds of years. Its layers can slide against each other mimicking a serpentine skeleton

breaking is one example of Yin and Yang. Great martial artists such as Bruce Lee who developed new forms of fighting based on philosophical foundations also explained how constant control of the body from being relaxed and compliant to avoid a kick or a punch land on the body (Yin) to a stiff structure just before a one’s own kick or a punch lands on the opponent’s body (Yang) [16]. Therefore, the same system should have the capacity to flow in a continuum between compliance and enforcement as two mutually supporting phenomena to meaningfully couple the system with others. The ideas such as “ghost circuits”, “kinematic tuning”, and “behavioral lensing” discussed in Chap. 3 of this book are based on these philosophical roots. The modern analytical methods such as impedance control [12] provide formal proof to such ancient approaches to build stable systems that encounter uncertain external forces.

1.1.2 *The Notion of Programming in a Physical Embodiment*

A robot is conventionally defined as a mechanical system capable of being programmed to execute specific tasks. In this book, we extend the concept of “programming” beyond its typical interpretation of software programs running on a microprocessor. Here, we encompass what are known as “morphological programs”—mechanical configurations that exhibit purposeful structural responses for solving real-time problems. A *purposeful structural response* depends on the task requirements. It can be bending/elongating/shortening when the forces are there in the right range and restoring in their absence.

To illustrate, consider the analogy of a tapered tube, akin to the cochlear membrane within the ear. This membrane effectively converts a time-domain pressure wave into its constituent frequency components at predefined locations along its length. In essence, this tapered tube serves as a functional role analogous to a software program that would sample an incoming pressure wave and generate its frequency components through the execution of a suitable algorithm, such as the Fast Fourier Transform (FFT).

1.1.3 A Working Definition of a Soft Robot

Soft robots are explained in different ways often referring to material properties. For instance, they are described to be “inherently compliant and exhibit large strains in normal operations” [28], “robots having mechanically compliant joints with variable stiffness” [1]. We refer the reader to [15] for a good coverage of various definitions and ideas of soft robotics given by co-authors of this book, Cecilia Laschi and Barbara Mazzolai.

In this book, we define a soft robot as follows: a soft robot is characterized by its capability to generate a purposeful structural response when subjected to a specific range of external forces, their associated frequency components, or a combination of both, all with the primary objective of executing a designated task.

However, the general meaning of soft robotics is very much centred around the range of forces humans can experience and exert. Some examples are shown in Fig. 1.2.

In this book, we use the term “soft” specific to a certain context. In most cases, the context will be a work environment that involves coming into contact with humans.

1.2 Introduction to Soft Robot Actuation Techniques

1.2.1 Pneumatic Actuators

Pneumatic actuators work based on air pressure applied in elastic tubular structures. A classic example is a silicone rubber continuum robot with two or more internal air channels. Usually, fiber reinforcement is used to avoid ballooning in the pneumatic channels. When pressurized, the continuum segment would bend, twist, or elongate depending on the relative pressure in the channels, the shape of channels, or the pattern of reinforcement. In some cases, reinforcement of the channel is hand crafted to give desired shapes of bending and ballooning [20]. The challenge in pneumatic

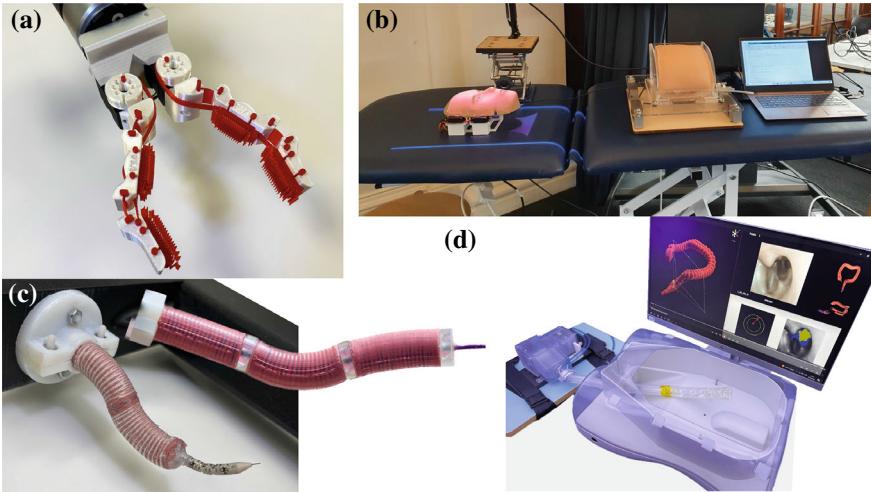


Fig. 1.2 Some soft robot examples; **a** The tendons, joints, and finger pads of this underactuated adaptive gripper are 3D printed on an entry-level Prusa i3 printer with soft TPU filament [31], **b** A robotic patient with a soft abdomen and controllable organs to train physicians [9, 14], **c** A 6-DOF hybrid manipulator, combining a tendon-driven segment [17] with two pneumatically driven soft segments, is developed for Lower GI interventions. It merges the benefits of soft and tendon-driven robotic manipulators to achieve intrinsic softness, compactness, and high payload. Notably, this design enables contact forces estimation by monitoring the actuation pressure of the soft segments [8, 18], **d** A medical examination system for the lower gastrointestinal tract featuring a soft robot for lower GI tract inspections, using an evertting motion and pneumatic control [7] for enhanced patient comfort and precise autonomous navigation [4]. The robot's design aims to minimize discomfort, making the procedure more comfortable for the patient

actuation has been the need to have a compressor. However, the recent FlowIo system [26] is a good example of a portable pneumatic actuator that can provide pressures more than 27 psi.

1.2.2 Tendons

Tendons are wires routed to deform or change the stiffness of a soft robot [25]. Unlike pneumatic actuators that can have both positive and negative pressures, tendons can only be pulled. The common methods of tendon control are to use servo motors to pull the tendons by winding around pulleys or to use Shape Memory Alloys (SMA) to control the tension by heating the SMA wires using an electric current. While tendons can potentially exert large forces, it has challenges in curvature control of soft structures because wires under tension try to straighten up. However, tendons have been successfully used in hybrid robots with joints such as articulated grippers and wearable soft robots [13] (Fig. 1.3).

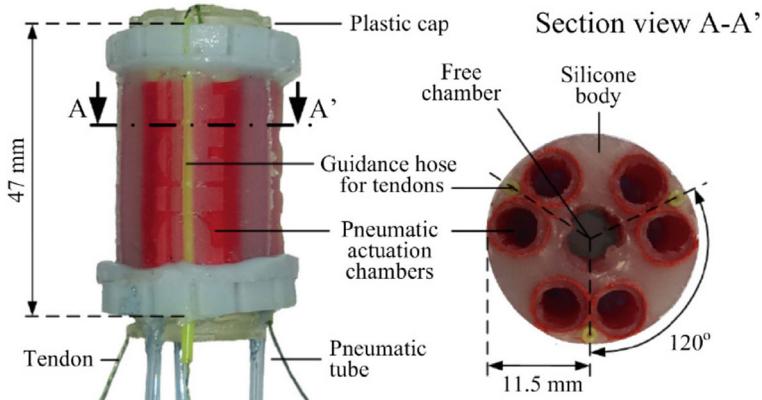


Fig. 1.3 An example of a soft continuum manipulator segment with tendon and pneumatic channels

1.2.3 Hydraulic

Hydraulic actuators in soft robots are very similar to pneumatic actuators except that a liquid is used instead of air in the channels. The non-compressibility of liquid gives several advantages for force control and remote haptic feedback. Hydraulic channels usually require more reinforcement than pneumatic actuators to avoid ballooning. In most applications, fiber reinforcement is used with winding patterns crafted to suit desired movements of the soft robot [24].

1.2.4 Other Actuation Methods

Depending on the size and application of the soft robot, other common actuation methods are dielectric elastomers [30], piezoelectric [29], and hydrogels [3].

Dielectric elastomers work based on the principle that two sheets of electrodes with opposite polarity (+ve and -ve) will squeeze a soft elastomer between them due to their electrostatic attraction. When an elastomer is squeezed, it would elongate to preserve the volume. This principle can be used to design various patterns of elastomer actuators to render different desired patterns of shape change when a Voltage difference is applied between the electrode sheets.

Piezoelectric actuators work based on the principle that they use electromechanical coupling to make a shape change when a Voltage difference is applied across the material. Given their ultrasonic actuation speeds but in micrometer elongation range, they are often used to control valves or in micro robots such as insect robots.

Hydrogels work based on the principle that water molecules getting into a 3D mesh of hydrophilic polymers (polymers that like water) swell the mesh. Then hydration and dehydration can make volume changes. Hydrogels are fragile and cannot produce

large forces. Therefore, they are used in micro or cellular robots for applications such as targeted chemical delivery, tissue engineering, and regenerative medicine.

1.3 Soft Sensors

A sensor converts a physical stimulus to an electric signal. For instance, an accelerometer can be made by placing a mass on a piezoelectric transducer. The mass helps to convert acceleration to a force through $F = ma$, where F is the force, m is the mass, and a is the acceleration. The force on the piezoelectric transducer generates a Voltage V proportional to the force. Such a series of linear relationships help to map the measurand a to an electrical measurement V in a linear sensor model. Sensors are not always linear. A thermocouple for instance, would use a material property of changing resistance in response to temperature. However, the change of resistance may not be linearly related to the temperature.

In contrast to rigid body sensors, soft sensors offer numerous advantages in specific applications. A soft sensor employs a flexible mechanical system to transform the measured signal into varying states such as vibrations or deformations within the soft mechanism, which are subsequently transmitted to a transducer for the generation of an electrical output.

For instance, consider a scenario where a permanent magnet is positioned alongside three Hall-effect sensors within a soft robotic fingertip (Fig. 1.4) [5]. This configuration enables the construction of a 3×3 covariance matrix based on the 3 channels of hall effect sensor readings for texture estimation. The elements within this matrix not only capture essential information about the texture but also provide insights into the physical properties of the fingertip and its texture-exploration behavior. Given the same texture and fingertip, the elements in the covariance matrix can change due to applied force or the speed of rubbing against the texture because they affect the vibration of magnets relative to the Hall-effect sensor. Since soft sensors allow us to use nonlinear soft tissue dynamics in the sensing function, diverging from rigid body sensors, such soft sensors offer the unique capability to explore in the space of mechanical and behavioral variables to enhance sensing resolution. This property can also be viewed as a sensor model or a likelihood function that can be tuned such as in a controllable stiffness finger [11].

Another example of a flexible wearable conductive Thermoplastic Polyurethane Material (TPU) sensor is shown in Fig. 1.5a. The sensor is produced using fused deposition modeling (FDM) printing technology. The resistance of the soft sensor changes when it is stretched or squeezed. The shape of the sensor is a serpentine curve. This structure can increase the length of the conductor per unit area and is more elastic, with more obvious changes in resistance. In this example, wearing it on the knuckle can be used to detect the curvature of the knuckle. Figure 1.5b shows the curve relationship between the resistance value of the sensor and the bending angle of the index finger. Figure 1.5c shows the fabrication process of the sensor. The printer first prints the sensor's substrate using normal TPU material (normal

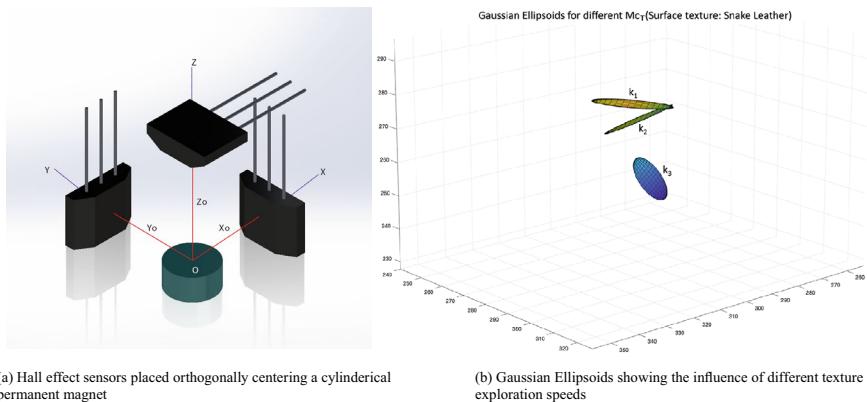


Fig. 1.4 **a** Hall effect sensors placed orthogonally centering a cylindrical permanent magnet.
b Gaussian ellipsoids showing the influence of different texture exploration speeds

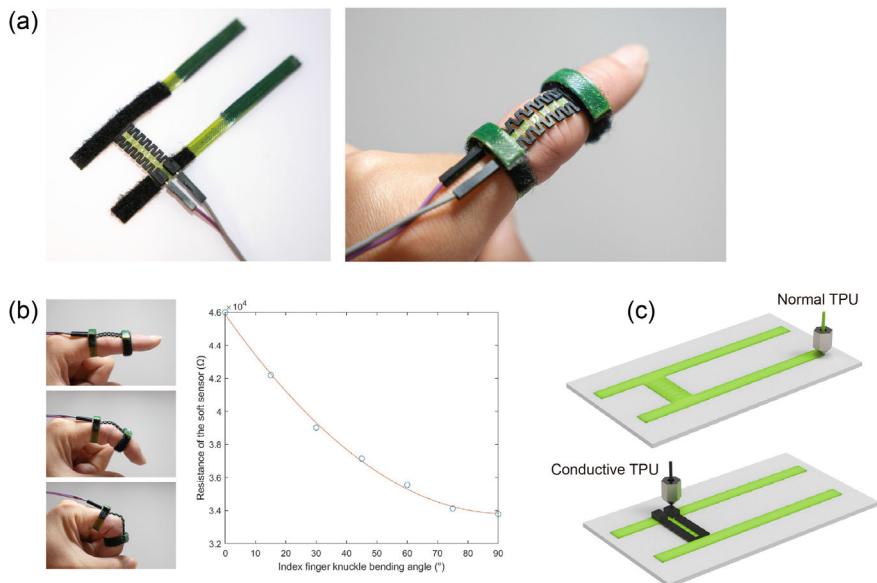


Fig. 1.5 A example is a soft sensor made with FDM printing technology using flexible conductive TPU. This wearable sensor can be used to monitor finger bending

TPU is an insulator, which can avoid signal interference caused by contact between skin and conductive TPU). When the substrate printing is completed, the filament is replaced with conductive TPU to continue printing, so that the conductive TPU can be perfectly adhered to the sensor substrate.

1.4 Trends in Soft Robot Applications

The paper titled ‘‘Roadmap on soft robotics: multifunctionality, adaptability and growth without borders’’ [21] gives a good overview about present challenges and future opportunities for soft robots. In terms of market trends, reports show that the soft robotics market is projected to grow at around 35% compound annual growth rate (CAGR) over 2021–2026 horizon [2]. The main driver for market growth is the need to have robots safely interacting with humans in shared spaces. Since human behavior is variable and difficult to predict, future soft robots are required to have designs that are safe at a physical level. Therefore, concepts such as embodied intelligence, embedded computing, and robust controllers are going to play key roles in the delivery of useful soft robots.

1.4.1 *Healthcare and Wearable Robots*

The largest market share is predicted to come from healthcare and wearable robotics sectors. The inherent safety of soft robots is ideal for patient care, surgery, and rehabilitation. Companies such as Ekso Bionics and ReWalk have developed wearable soft robotic exosuits to assist and rehabilitate patients with spinal cord injuries. Soft robotics has been a viable option for minimally invasive surgery (MIS) too. The EU funded STIFF-FLOP project designed an Octopus inspired continuum robot for MIS. A UK based company called Cambridge Medical Robotics is developing a surgical robot for MIS, and several other designs are being tested across the globe.

1.4.2 *Agriculture*

Soft robots are well-suited for applications in agriculture that require delicate handling of produce and navigation through natural, unstructured environments. For instance, soft robots can pick fruits without causing damage, help with weeding, and monitor plant health. Companies such as Octinion and Agrobot are developing new solutions to reduce food waste and to address labour shortages in agriculture.

1.4.3 *Home Robotics*

Soft robots have significant potential in home based robot companions [10], rehabilitation [6], remote health monitoring [23, 27], and cleaning [19]. In addition to safety of interacting with people, pets, and fragile objects in a home setting soft robots can navigate cluttered and dynamic home environments far better than a rigid

robot. Their deformable bodies allow them to squeeze into tight spaces [22], access confined areas, and perform tasks that rigid robots find challenging.

1.5 A Brief Overview of the Rest of the Chapters

Chapter 2 aims to establish a foundational understanding of bioinspiration in the context of soft robotics, drawing richly from the principles observed in nature. Given that a majority of living organisms predominantly feature soft tissues, underscored by the significance of body compliance in embodied intelligence, the chapter explores these concepts extensively. It not only elucidates the theoretical underpinnings but also offers comprehensive methodological guidelines for the application of bioinspired design. The journey commences with a systematic approach to crafting bioinspired (soft) robots, commencing with the meticulous observation of biological principles and their subsequent translation into viable engineering solutions. Furthermore, the chapter serves as a practical guide, offering insights into the effective utilization of bioinspired principles within the dynamic field of soft robotics.

Chapter 3 endeavors to explore the gaps in our comprehension of the disparities between robots and biological entities when navigating movements in a natural environment. In particular, we delve into the overarching inquiry of how the fusion of passive dynamics and computational processes can address intricate challenges in interactions. The seamless and graceful physical interactions we experience daily seem to emanate from a sophisticated interplay between the dynamics and computational capabilities within the brain, the body, and the environment. Nevertheless, attempting to decipher this intricate interconnection solely through human participants proves to be a challenging task. Therefore, we propose soft robots as a paradigm to test biological hypotheses.

Chapter 4 presents dynamic and kinematic modeling of soft robotics, examining the evolving landscape through several methodologies. Continuous Curvature Models show the intricacies linked with the inherently continuous and deformable characteristics of soft robots. Lumped Parametric Models provide insights into techniques that discretize soft robots into simpler, interconnected elements. Hybrid Models amalgamate the strengths of continuous curvature and lumped parametric models. Finally, Learning-Based Models leverage machine learning and data-driven approaches to model the complex kinematics and dynamics of soft robots.

Chapter 5 Introduces a novel generic modeling approach for soft robots, our method revolves around the fundamental concept of mechanical compliance. This concept serves as a concise representation of the deformable robot's behavior. Specifically, we delve into the derivation of the kinematics of a soft manipulator arm based on this pivotal notion. This chapter elucidate the process of calculating the compliance through a thorough analysis of the robot's materials and mechanical structure.

Chapter 7 delves into the core principles of fabrication, control mechanisms, and the dynamics of human-robot interaction, specifically focusing on an innovative class of collaborative robotic manipulators known as malleable robots. These robots distinguish themselves through their adaptable architectures, featuring varying stiffness levels, which contribute to achieving heightened dexterity while utilizing arms with lower mobility. Unlike traditional collaborative robots, or cobots, that typically integrate six or more degrees of freedom (DOF) in a serial arm, malleable robots are designed to excel in constrained spaces and seamlessly adapt to a spectrum of tasks.

Chapter 6 focuses on the exploration of soft robotic hands, emphasizing their unique characteristics and functionalities. Soft hands, crafted from or incorporating pliable materials, showcase remarkable adaptability and flexibility in the realms of object grasping and manipulation. We begin by illustrating how soft hands effectively alleviate contact forces during the grasping process. Subsequently, we delve into the fundamental principles governing soft robotic hands, delineating various categories such as soft pad hands, suction hands, jamming hands, bending finger hands, expanding membrane hands, and remotely deformed hands. The mechanical intricacies of these soft hands are then expounded upon. To enhance our understanding, we introduce an analytical model for soft pad hands, alongside computational models for both bending finger hands and expanding membrane hands.

Chapter 8 addresses practical considerations in the deployment of soft robots, focusing on soft continuum hose robots for 3D printing applications in the construction industry. By imbuing these hoses with robotic capabilities, there is substantial potential for fabricating and intricately repairing complex structures beyond current capabilities. The chapter provides modelling and control guidelines for such robots.

Chapter 9 discusses origami robotics, where robots are conceived as 2D sheets and skillfully folded into their 3D configurations. The process of folding emerges as an elegant solution to assembly, enabling thin sheets to transform into a diverse array of structures characterized by intricate geometry, sophisticated kinematics, and nuanced mechanical responses. What sets this approach apart is the seamless integration of actuators, sensing mechanisms, and computation directly into the sheet material. This innovative approach allows for the fabrication, self-assembly, and deployment of complete robots in a singular, unified process.

References

1. Albu-Schaffer A, Fischer M, Schreiber G, Schoeppe F, Hirzinger G (2004) Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints? In: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566), vol 4. IEEE, pp 3295–3301
2. Banerjee A (2020) Soft robotics market-growth, trends, Covid-19 impact, and forecasts (2021–2026). Res Markets, pp 1–136
3. Hritwick B, Mohamed S, Hongliang R (2018) Hydrogel actuators and sensors for biomedical soft robots: brief overview with impending challenges. *Biomimetics* 3(3):15
4. Borvorantanajanya K, Treratamakulchai S, Franco E, Baena FRY (2023) Area-based total length estimation for position control in soft growing robots. Technical report, EasyChair
5. Chaturanga DS, Wang Z, Noh Y, Nanayakkara T, Hirai S (2016) Magnetic and mechanical modeling of a soft three-axis force sensor. *IEEE Sens J* 16(13):5298–5307
6. Matteo C, Cecilia L, Arianna M, Paolo D (2018) Biomedical applications of soft robotics. *Nat Rev Mater* 3(6):143–153
7. Enrico F (2022) Model-based eversion control of soft growing robots with pneumatic actuation. *IEEE Control Syst Lett* 6:2689–2694
8. Franco E, Garriga-Casanovas A, Tang J, Baena FR, Astolfi A (2022) Adaptive energy shaping control of a class of nonlinear soft continuum manipulators. *IEEE/ASME Trans Mechatronics* 27(1):280–291
9. Liang H, Nicolas H, de Simon L, Luca S, Perla M, Fumiya I, Thrishantha N (2020) An abdominal phantom with tunable stiffness nodules and force sensing capability for palpation training. *IEEE Trans Robot* 37(4):1051–1064
10. Hedayati H, Bhaduri S, Sumner T, Szafir D, Gross MD (2019) Hugbot: a soft robot designed to give human-like hugs. In: Proceedings of the 18th ACM international conference on interaction design and children, pp 556–561
11. Nicolas H, Perla M, Fumiya I, Thrishantha N (2018) A variable stiffness robotic probe for soft tissue palpation. *IEEE Robot Autom Lett* 3(2):1168–1175
12. Hogan N (1985) Impedance control: an approach to manipulation: Part ii—implementation
13. Jeong U, Kim K, Kim SH, Choi H, Youn BD, Cho KJ (2021) Reliability analysis of a tendon-driven actuation for soft robots. *Int J Robot Res* 40(1):494–511
14. Lalitharatne TD, Tan Y, He L, Leong F, Van Zalk N, de Lusignan S, Iida F, Nanayakkara T (2021) Morphface: a hybrid morphable face for a robopatient. *IEEE Robot Autom Lett* 6(2):643–650
15. Laschi C, Mazzolai B, Cianchetti M (2016) Soft robotics: technologies and systems pushing the boundaries of robot abilities. *Sci Robot* 1(1):eaah3690
16. Lee B, Modrić Ž (1975) Tao of jeet kune do. Ohara Publications Burbank
17. Leung F, Lam CP, Cheung L, Donald IP, Yam Y, Chiu P, Lau KC (2022) Design of a robotic traction device for endoscopic submucosal dissection, pp 29–30
18. Leung FF, Borvorantanajanya K, Zari E, Garriga-Casanovas A, Franco E, Baena FR, Chiu PW, Yam Y (2024) Development of a hybrid manipulator with force estimation capability for lower GI interventions. Manuscript submitted for publication
19. Xiaomin L, Maozheng S, Yuhui F, Yunwei Z, Changyong C (2022) Worm-inspired soft robots enable adaptable pipeline and tunnel inspection. *Adv Intell Syst* 4(1):2100128
20. Martinez RV, Fish CR, Chen X, Whitesides GM (2012) Elastomeric origami: programmable paper-elastomer composites as pneumatic actuators. *Adv Funct Mater* 22(7):1376–1384
21. Mazzolai B, Mondini A, Del Dottore E, Margheri L, Carpi F, Suzumori K, Cianchetti M, Speck T, Smoukov SK, Burget I et al (2022) Roadmap on soft robotics: multifunctionality, adaptability and growth without borders. *Multifunctional Mater* 5(3):032001
22. Mulvey BW, Lalitharatne TD, Nanayakkara T (2023) DeforMoBot: a bio-inspired deformable mobile robot for navigation among obstacles. *IEEE Robot Autom Lett* 8(6):3828–3835

23. Yaokun P, Xu X, Shoue C, Yuhui F, Xiaodong S, Yiming D, Zhong-Lin W, Changyong C (2022) Skin-inspired textile-based tactile sensors enable multifunctional sensing of wearables and soft robots. *Nano Energy* 96:107137
24. 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
25. Shiva A, Stilli A, Noh Y, Faragasso A, De Falco I, Gerboni G, Cianchetti M, Menciassi A, Althoefer K, Wurdemann HA (2016) Tendon-based stiffening for a pneumatically actuated soft manipulator. *IEEE Robot Autom Lett* 1(2):632–637
26. Shtarbanov A (2021) Flowio development platform—the pneumatic “raspberry pi” for soft robotics. In: Extended abstracts of the 2021 CHI conference on human factors in computing systems, pp 1–6
27. Xinyang T, Liang H, Jiangang C, Wei C, Thrishantha N (2020) A soft pressure sensor skin for hand and wrist orthoses. *IEEE Robot Autom Lett* 5(2):2192–2199
28. Trivedi D, Rahn CD, Kier WM, Walker ID (2008) Soft robotics: biological inspiration, state of the art, and future research. *Appl Bionics Biomech* 5(3):99–117
29. Wu Y, Yim JK, Liang J, Shao Z, Qi M, Zhong J, Luo Z, Yan X, Zhang M, Wang X et al (2019) Insect-scale fast moving and ultrarobust soft robot. *Sci Robot* 4(32):eaax1594
30. Youn J-H, Jeong SM, Hwang G, Kim H, Hyeon K, Park J, Kyung K-U (2020) Dielectric elastomer actuator for soft robotics applications and challenges. *Appl Sci* 10(2):640
31. Zhou X, Spiers AJ (2023) Instagrasp: an entirely 3d printed adaptive gripper with tpu soft elements and minimal assembly time. arXiv preprint [arXiv:2305.17029](https://arxiv.org/abs/2305.17029)

Chapter 2

Bioinspiration



Cecilia Laschi and Barbara Mazzolai

Abstract Bioinspiration involves applying principles observed in nature to design and create various human-made products, ranging from engineering and materials to architecture. A significant area influenced by bioinspiration is soft robotics, drawing extensively from natural principles. Most living beings possess primarily or entirely soft tissues, emphasizing the importance of body compliance for embodied intelligence. This chapter delves into these concepts, offering detailed methodological guidelines for bioinspired design. It initiates by presenting a structured approach to designing bioinspired (soft) robots, beginning with the observation of biological principles and their subsequent translation into engineering solutions. Additionally, it provides a practical guide on utilizing bioinspired principles in the field of soft robotics. The chapter includes a mini-catalogue of principles pertinent to robotics, offering engineers a valuable resource. The principles are described qualitatively and mathematical descriptions are given wherever available. The potential of bioinspiration is still largely unexplored and this chapter contributes to its practical application in soft robotics.

2.1 Learning from Nature: A Method for Bioinspired Soft Robotics

Bioinspiration is a broad term that encompasses the uptake of principles observed in nature for designing and building human manufacts. It applies to many field of human activities, from engineering to architecture, to material synthesis, and many more. Bioinspiration is used in robotics, too. In a sense, the old dream of humankind of building self-moving machines since ancient times, and autonomous robots today, is inspired by the observation of living beings and the ambition of reproducing

C. Laschi (✉)
National University of Singapore, Singapore, Singapore
e-mail: mpeclc@nus.edu.sg

B. Mazzolai
Italian Institute of Technology, Genoa, Italy
e-mail: barbara.mazzolai@iit.it

ourselves. Despite many examples exist of bioinspired products, with some success cases in our everyday life, one may argue whether a systematic method for taking inspiration from nature exists. A general answer cannot be but negative. However, we may try to give general guidelines with a few fundamental steps to take if embarking into a bioinspired-design journey in robotics.

First of all comes nature. But when and how? It comes when we have robotics problems that are solved in nature. For instance, when we need to provide our robots with abilities that are shown by living beings, e.g., animals or plants: jumping, burrowing, fetching objects on the fly, understanding an object function, and many more. How it comes is more difficult to outline. Inspiration comes in a variety of ways. However, in engineering, we may use tools for objective observation and measurement of the phenomenon we are inspired by. This first step is important and should ensure that we are not distracted by the superficial characteristics of our model, like the physical appearance. A star-shaped robot is not necessarily inspired to a starfish. Instead, we should understand and explain the working principle, in engineering terms, that we wish to implement. The deeper we go in this direction, the more likely the implementation is successful. In some cases, it is helpful to build simple demonstrators of the principles, with a trial-and-error approach. Today's prototyping technologies are available and usable enough to give an advantage with respect to complex theoretical analyses.

Once a principle is clearly understood and validated, its mathematical description becomes crucial for the transition into engineering. A mathematical model serves as a foundation for robot design. At this stage, our focus shifts to the specific requirements of our robots, without the need to revisit the biological model, as the principle is embedded in the mathematical model. Regardless of whether the robot's size, shape, or appearance differs from our initial model, they are tailored to meet our requirements. The robot remains bioinspired because it operates based on the same principle observed in nature.

More precisely, the method for bioinspired design can be divided into the following steps:

1. Define the problem: Identify the problem or challenge that needs to be addressed. This could be anything from improving performance to improving energy efficiency. The problem needs to be defined to the deeper level of detail possible. An example can be the design of a more efficient transportation system, like a train. The problem should be defined in terms of fluid dynamics.
2. Study: Search existing solutions and approaches to the problem. Study how nature solves similar problems and identify biological systems, processes, and strategies that could be used as inspiration. In the train example, once the problem is expressed in fluid dynamics terms, the behaviour of natural systems in a fluid can provide an inspiration. Let's take a specific fluid dynamic problem as an example: the noise associated to the transition between two media of different pressure, like the entering of a train into a tunnel. Fish-catching birds which dive in water silently become an excellent source of inspiration.

3. Analyze: Analyze the biological systems, processes, and strategies that have been identified. Extract the key features and principles that could be applied to the design problem. In the train example above, the shape of kingfishers' beak explains the silent dive with no splash.
4. Apply: Apply the insights gained from biological systems to the design problem. Implement the principles that enable the features of the biological system. A train front can be reshaped to similarly reduce the transition noise.
5. Evaluate: Evaluate the concepts and test them against the original problem. Refine and iterate the design until a viable solution is found.
6. Implement: Implement the final design and continue to monitor and improve it over time.

As a matter of fact, when the Shinkansen super-fast train started its service in 1964, a problem arose for the so-called tunnel boom, an explosion-like noise due to the pressure wave created in front of the train when entering the tunnel. The Japanese engineer Eiji Nakatsu redesigned the Shinkansen front in a streamlined and pointed shape, similar to the beak of a kingfisher bird, which dives into water at high speed without making a splash. The train shape minimizes noise and vibrations, just like the kingfisher bird that makes no noise when it dives into water. Moreover, it has been found that, with his bioinspired solutions, Eiji Nakatsu made Shinkansen trains quieter and also 10% faster, with 15% less electrical power. Therefore, the Shinkansen is an excellent example of how nature can inspire the design of advanced technologies, improving efficiency, speed, and sustainability.

The bioinspired design method is a cyclical process that involves continuous learning, observation, and adaptation. It encourages designers to take a holistic approach to problem-solving and to look to nature for inspiration in creating innovative and sustainable solutions.

Figure 2.1 summarizes the proposed method for bioinspiration based on the above-mentioned steps.

2.2 Basics of Movement: Muscles

In animals, muscles are at the basis of most movements. Building artificial muscles is one of the long-lasting challenges in robotics, and beyond. As human beings, we are familiar with skeletal muscles, but more forms of muscles exist in the animal world, as this section shows.

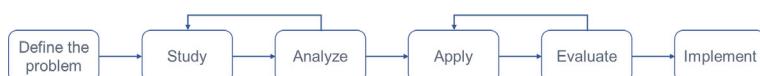


Fig. 2.1 A method for bioinspiration

2.2.1 *Skeletal Muscles*

Skeletal muscles are composed of fibers, activated by motoneurons. Neural activation generates a contraction, which shortens the muscle. In our main limbs, muscles are attached to bones by tendons. They are generally arranged in an antagonistic couple of muscles that move the same joint. Contraction of an agonist muscle moves the joint in a direction and contraction of the antagonistic muscle moves the same joint in the opposite direction. Co-contractions increase the overall stiffness.

While we are not interested in the detailed physiological mechanisms of muscle activation and contraction here, we are interested in muscle mechanics and how it can be modeled, for use in robotics. The classical muscle model proposed by Hill [1] is composed of a contractile unit (CE) with a parallel elastic component (PEC) and a series elastic component (SEC) (see Fig. 2.2, from [2]). The mathematical description of the model is:

$$(F + a)(v + b) = (F_{max} + a) \quad (2.1)$$

where

F is the tension (or load) in the muscle

v is the velocity of contraction

F_{max} is the maximum isometric tension (or load) generated in the muscle

a is the coefficient of shortening heat

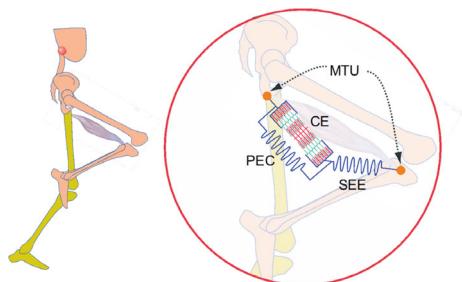
$b = a \cdot v_0 / F_0$

v_0 is the maximum velocity, when $F = 0$

2.2.2 *Hydraulic Skeletons*

Unlike skeletal muscles found in vertebrates, which are attached to bones and generate force through contraction, hydraulic skeletons rely on the movement of liquids within their body cavities to generate force and produce movement. The functioning of these systems depends on the fact that they are primarily composed of relatively

Fig. 2.2 Hill-type muscle model, reproduced with permission from [2]



uncompressible fluids and tissues, which ensures their constant volume. As a consequence, when the muscles contract and decrease in one dimension, there is a corresponding increase in another dimension. They are found in a wide range of animals or animals' structures, such as earthworms, the legs of spiders, or the larval stage of many invertebrates.

Hydrostatic skeletons consist of longitudinal and circular muscles that are arranged in layers around a fluid-filled cavity. When these muscles contract, they increase the pressure of the fluid in the cavity, which causes the body to bend or straighten. The direction of movement is controlled by the pattern of muscle activation and the relative strength of the longitudinal and circular muscles and the force is transmitted by internal pressure.

Hydrostatic skeletons have inspired the development of soft robots and other bioinspired technologies that use fluid-filled chambers and flexible materials to generate movement and force. They are also an important area of research in biomechanics and evolutionary biology, as they provide insights into the principles and mechanisms of movement in the majority of animals on earth, which relies on hydrostatic skeletons.

2.2.3 *Muscular Hydrostats*

A muscular hydrostat is a specialized type of hydrostatic skeleton found in some animals, such as the tongues of mammals (including humans), the trunks of elephants, and the tentacles and arms of some cephalopods (such as the octopus). These structures also rely on hydrostatic skeletal support, but they lack the fluid-filled cavities that characterize this skeletal type.

Muscular hydrostats consist of muscles that are arranged in complex patterns without any skeletal support, allowing them to produce highly flexible and precise movements. They are able to change their shape by contracting and relaxing different groups of muscles, rather than by bending or straightening a rigid skeleton.

In the case of the tongue, the muscular hydrostat is composed of many small intrinsic muscles that are able to move in multiple directions, allowing the tongue to be precisely controlled for functions such as swallowing, vocalization, and manipulating food.

The elephant trunk is also a muscular hydrostat, consisting of several layers of muscles that enable it to perform a wide range of movements, from delicate grasping to powerful lifting.

The muscular hydrostat structure of the octopus arms (see Fig. 2.3) is especially notable for its ability to generate strong, flexible movements that are capable of exerting a significant amount of force relative to the size of the animal, allowing the animal to manipulate objects and prey with great precision and strength. The octopus arm muscular hydrostat is composed of thousands of tiny muscular fibres that are arranged in a few major muscles: longitudinal (four main uniformly distributed muscles), transverse (all along the arm length) and oblique (wrapped around the arm

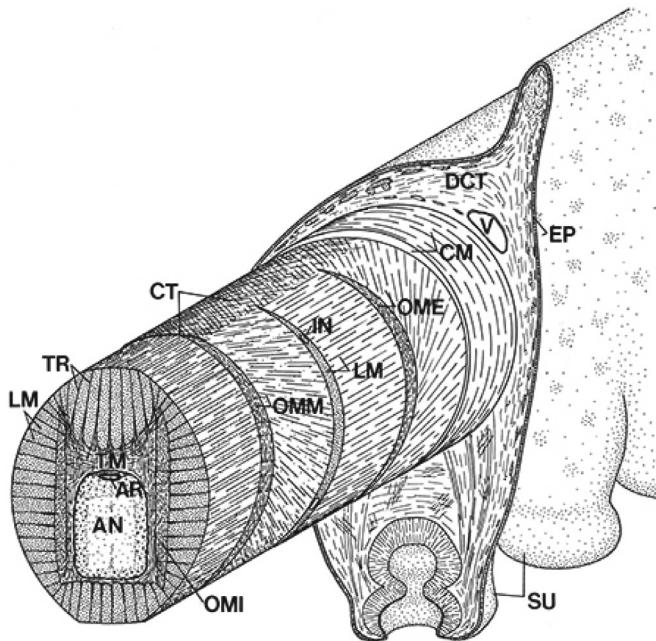


Fig. 2.3 Octopus muscular hydrostat, from [3]

both clockwise and counterclockwise). The octopus arm can elongate and shorten by transverse and longitudinal contractions, respectively, while the other muscles are relaxed. Selective longitudinal contractions generate bending, helped by local transverse contractions. Oblique contractions generate torsion. A mathematical relation between the radial strain ϵ_r and longitudinal strain ϵ_l is proposed by [3]:

$$\epsilon_r = (1 + \epsilon_l)^{-1/2} - 1 \quad (2.2)$$

The study of muscular hydrostats has contributed to the development of soft robotics and other bioinspired technologies that use flexible, muscle-like materials to generate movement and force in a more versatile and adaptable way.

2.3 Movement Without Muscles

Plants have often been overlooked as a source of inspiration for robotics and innovative engineering solutions. This may be attributed to their fundamentally different operational principles compared to animals, as well as the challenges involved in studying their movements and features. However, despite their sessile lifestyle, plants have evolved remarkable abilities to respond to various signals and effectively

adapt to changing environmental conditions. Despite lacking muscles, plants exhibit a wide range of non-muscular movements that occur over different timescales, ranging from hours or days to milliseconds [4, 5]. The mechanisms behind these diverse non-muscular movements have intrigued scientists since the pioneering work of Darwin [6–9].

Understanding these non-muscular movements in plants holds significant potential for advancements in applied sciences and engineering, particularly in the development of novel biomimetic actuation strategies and bioinspired devices [10]. Exploring and harnessing the principles behind plant movements can open doors to innovative solutions in various fields.

Among the various movements observed in plants, three main categories are here identified: nastic movements, tropisms, and passive movements.

2.3.1 Active Nastic Movements

Nastic movements refer to the movements induced by stimuli that lack a specific direction, resulting in a particular orientation of the plant (e.g., thigmonastism, seismonastism, hydronastism, photonastism). These plant movements can occur much more rapidly, especially when stored elastic energy is released. They are typically actuated by turgor changes or swelling/shrinking processes triggered by humidity fluctuations. Examples include the opening and closing of pine cones and the snap-trapping mechanism observed in *Dionaea muscipula* or Venus Flytrap. The direction of the plant's reaction is independent of the direction of the stimulus but is instead predetermined by the structure and form of the reacting organ. In the case of the Venus flytrap, when an insect touches at least two of the three tactile hairs inside its lobes within approximately 20 s, the plant rapidly closes its snap-trap to capture the animal. It only reopens the trap after digestion or an unsuccessful attempt.

More generally, nastic movements are typically induced by elastic changes in the size of specialized motor cells within plant tissues. These alterations are usually brought about by variations in osmotic pressure resulting from the influx or efflux of ions, which in turn triggers the movement of water into or out of the cells. In many plants, the shrinkage of these motor cells is responsible for the overall movement of the plant.

Plants are able to modulate their internal pressure in a selective way, by virtue of a finely regulated turgor-based strategy. Indeed, as a sort of “natural hardness”, turgor pressure ΔP generated by water flux is sustained by an osmotic pressure difference: $\Delta P = P_o - P_i$, where P_o and P_i respectively represent the osmotic pressures (also called osmotic potentials) outside and inside the cell [11].

The response of the plant is determined by the structure of the plant part and remains the same regardless of the direction from which the stimulus originated. In nastic movements, the asymmetry in the plant's response does not arise from a gradient in the stimulus. Instead, it is generated by the differential flow of ions, initiated by the stimulus, into and out of specialized cells [12].

2.3.2 Active Tropisms Movements

Tropisms in plants refer to the directional growth responses exhibited by plant organs in response to external stimuli. These stimuli can include factors such as light, gravity, touch, or chemicals. Tropisms can be positive or negative, depending on whether the plant organ grows towards or away from the stimulus.

For example, phototropism is the growth response of plants towards or away from a light source. When a plant is exposed to light from one direction, the cells on the shaded side of the shoot apical meristems elongate more rapidly than those on the illuminated side. This causes the shoot to bend towards the light source, allowing the plant to maximize its exposure to sunlight for photosynthesis. Positive phototropism occurs when plant organs, such as stems or leaves, bend or grow towards the light source, while negative phototropism refers to the opposite response, with plant organs bending or growing away from the light. Similarly, gravitropism is the growth response of plant organs in relation to gravity. Roots display positive gravitropism, growing towards the gravitational pull, while shoots exhibit negative gravitropism, growing against gravity, towards the light.

Plants use tropisms to adapt to their environment and maximize their chances of survival and reproduction. In fact, tropisms are crucial for plant growth and orientation in their environment, allowing them to optimize light absorption, nutrient uptake, and anchorage. These responses are regulated by plant hormones and the differential distribution of growth-promoting substances within the plant.

There are several formulas that have been developed to describe the growth in plant organs, such as roots. However, it is important to note that root growth is influenced by many variables, including the plant species, environmental conditions, and available nutrients, so the formulas can vary depending on the context.

One of the most common formulas for describing root growth is the Richards equation, which describes the kinetics of root growth as a function of time. This equation is often used in agronomy and plant biology to describe root growth under different environmental conditions. The Richards equation is expressed as:

$$L(t) = \frac{L_{max}}{1 + e^{-b(t-t_m)}} \quad (2.3)$$

where $L(t)$ represents the length of the root at time t , L_{max} is the maximum root length, b represents the rate of root growth, and t_m represents the time of maximum root growth.

The Chododny-Went theory suggests that both gravitropism and phototropism involve the redistribution of auxin in response to the external stimulus.

In gravitropism, i.e., the growth or movement of a plant in response to gravity, the Chododny-Went theory proposes that when a plant is placed horizontally, gravity causes the plant hormone auxin to accumulate on the lower side of the plant, which leads to increased cell elongation and curvature towards the ground. In other words, the Chododny-Went theory proposes that differential growth rates on opposite sides

of the plant stem or root cause curvature towards the gravity vector. According to this model, the gravity-induced redistribution of auxin leads to an asymmetrical growth response, which causes the curvature. The Cholodny-Went model for gravitropism can be represented by the following equations:

$$y = y_0 + \frac{1}{2}gt^2$$

where y is the displacement of the plant part from its initial position, y_0 is the initial displacement, g is the acceleration due to gravity, and t is the time elapsed.

$$\frac{d^2y}{dt^2} = kg(y - y_0)$$

where k is a constant that describes the sensitivity of the plant to gravity.

These equations describe the motion of the plant part under the influence of gravity and the plant's growth response to the gravity vector. The Cholodny-Went model for gravitropism has been used to explain the bending of roots and shoots in response to gravity, and has been supported by experimental evidence.

The Cholodny-Went theory similarly proposes that light causes an unequal distribution of auxin in phototropism, i.e., the growth or movement of a plant in response to light, which leads to differential growth and bending of the plant towards the light source.

2.3.3 *Passive Movements*

Plants have developed various mechanisms to disperse seeds from their parent plants and enhance germination, ensuring their survival. The study of seed dispersal mechanisms, along with their structural and functional characteristics, is a vibrant area of research encompassing ecology, plant diversity, climate change, and even material science and engineering.

Seed transport is considered “passive” as it does not involve active metabolism or the production of internal energy. Instead, the movement is facilitated by the inherent material and structural properties of the seed or fruit tissues [13–16]. These characteristics confer upon the structures that transport seeds a remarkable responsiveness to changes in environmental conditions, such as humidity and temperature. Moreover, these structures have evolved the ability to harness environmental factors, such as wind, water, and animals, as vectors for their mobility [17].

The mechanisms implemented by plants to transport seeds in the environment exemplify captivating and significant instances of morphological computation in

nature. They offer a diverse array of physical and mechanical features that are optimized for passive flying, landing, crawling, and drilling within various ecosystems. These features allow seeds to navigate and adapt to their surroundings, ensuring successful dispersal and survival.

Among the most ingenious in terms of spatial mobility are those structures that utilize their morphology and structural properties to be carried by the wind and disperse the seeds over long distances (e.g., “winged” and “parachute” seeds). Additionally, certain seeds can penetrate soil through a self-burial mechanism driven by the hygroscopic properties and morphological features of the structures that transport them (typically fruits). By studying the motion mechanisms of these structures, we can extract novel design principles that inspire the development of intelligent artificial systems.

Among several examples, the Geraniaceae family (e.g., in the Erodium or Pelargonium genera) employ an explosive dispersal strategy of seeds combined with hygroscopic motion to facilitate germination. This mechanism allows for autonomous movement on the terrain surface and penetration into soil fractures. Crawling and burying actions are made possible through the hygroscopic helical structure of the fruit, known as the awn, which responds to external humidity fluctuations by altering its configuration.

The metabolically inactive tissues within the awn are responsible for the passive movement of the seed-transport structures, characterized by a combination of bending and torsional deformation. This movement is enabled by the internal hierarchical structure of the awn, described as a composite material. Within this structure, the hygroscopic active element, crystalline cellulose microfibrils, is embedded in a hygroscopic passive soft matrix composed of structural proteins, polysaccharides, and aromatic compounds.

Cellulose possesses a hydrophilic nature, allowing water molecules to be stored between intermolecular hydroxyl groups through hydrogen bonding. Consequently, water adsorption leads to a volumetric transversal expansion of the cellulose tissues. This expansion results in the swelling of the wet tissues. Importantly, because water adsorption is a reversible process, the swollen tissue of the fruit can return to its initial size when it dries [18].

From the perspective of engineering design, the self-burial and flying abilities of the structures that transport seeds offer efficient three-dimensional movement in space, making them valuable sources of inspiration for innovative soft robots. These characteristics provide insights into the development of dynamic solutions for robotic systems. Figure 2.4 provides an overview of the investigation methodologies and tools utilized for the analysis of self-burial biological systems and the generation of data in this field.

Self-burying seed		Morphometric and structural characteristics parameters		Biomechanics	
<i>Erodium</i>	Awn	Methodology <ul style="list-style-type: none"> Morphometric analysis, sectioning, and weight measurements  <i>Pelargonium</i> 	Biological specification <ul style="list-style-type: none"> The driver for hygroscopic tissue expansion is determined by the arrangement of microfibrils forming the cell walls, composed of glycoproteins and cellulose, attached in multiple layers with different orientations (Jung et al., 2014) <i>Erodium cicutarium</i> <ul style="list-style-type: none"> Maximum awn diameter: 0.002 ± 0.001 m Awn height: 0.015 ± 0.005 m Number of turns: 9 ± 2 Awn spiral angle: $86 \pm 2^\circ$ Section width: 0.001 m Section height: 0.00025 m Mass: $5 \pm 1 \times 10^{-6}$ kg (Evangelista et al., 2011) Lignin composition and distribution (Abraham & Elbaum, 2013; Shin et al., 2018) Arrangement of active and inactive layer (Abraham & Elbaum, 2013; Shin et al., 2018) 	Methodology <ul style="list-style-type: none"> Coefficient of Hygroscopic Expansion (CHE) measurement through mass changes in controlled relative humidity changes (Dawson et al., 1997) Measurements of CHE with thermo-hygrostat (Ha et al., 2020) Measurements of the Tilt angles (Ψ) and cellulose microfibril angle (MAFH) with Small Angle X-ray Scattering (SAXS) 	Biological specification <ul style="list-style-type: none"> Pinecones CHE: 0.20 ± 0.04 for $\Delta RH = 1\%$ (Dawson et al., 1997) <i>Pelargonium appendiculatum</i> CHE: $0.15\text{--}0.2$ (Ha et al., 2020)
<i>Erodium</i>		Capsule <ul style="list-style-type: none"> Scanning electron microscopy (SEM) Spraying with water to characterize the number of wet-dry cycles necessary for self-burial of seeds Observations of seed establishment of unmodified and modified seeds in different substrates (e. g. small or large crevices sizes) Measurement of the depth of burial of unmodified and modified seeds 	The presence of hairs or tips affect the establishment of the seeds into soil substrates with different crevices sizes (Stamp, 1984)	 <ul style="list-style-type: none"> Material biomechanical properties of <i>Erodium cicutarium</i> are assumed equal to the properties of wood (Ashby and Jones, 1996; Evangelista et al., 2011) Drag force tests with load cell and dedicated setup 	 <ul style="list-style-type: none"> Young's modulus = 9×10^9 Pa Poisson's ratio = 0.33 Shear modulus = 3.4×10^9 Pa (Ashby and Jones, 1996; Evangelista et al., 2011) F_{drag} in beads = $2.5\text{--}3$ mN (Jung et al., 2017)
<i>Pelargonium</i>					<i>Pelargonium peltatum</i> <ul style="list-style-type: none"> $\Psi = 5^\circ, >3^\circ$ and 19° MAFH = 40, 16 and 70° <i>Erodium gruinum</i> <ul style="list-style-type: none"> $\Psi = 3^\circ, >3^\circ$ and 20° MAFH = 30, 10 and 80° <p>The three value are respectively for outermost sublayer, median sublayer and inner layer.</p> <p>(Abraham and Elbaum, 2013)</p> <i>Pelargonium appendiculatum</i> <ul style="list-style-type: none"> $R = 0.90 \pm 0.43$ mm $p = 1.85 \pm 0.24$ mm <p>For RH 50%</p> <p>Both values increase with the increase of humidity (Ha et al., 2020)</p>

Fig. 2.4 The table highlights the biological features, measurements, and characterization methods relevant to self-burying seeds, along with their associated specifications for soft robotic solutions [19]

2.4 Principles of Terrestrial Locomotion for Soft Robots

Terrestrial locomotion is achieved by animals in a variety of ways, some of which involves the use of two or more legs, while some others just use body movements and friction in the terrain. We see in the section fundamental modes for biped and multi-legged locomotion and for crawling-based locomotion strategies.

2.4.1 Biped Locomotion

Biped locomotion is based on a cyclic, alternate foot placement and push, which implements walking, running, and more locomotion patterns. We will focus on walking, which provides the basic models for biped locomotion. The cyclic walking pattern is composed of 2 main phases, i.e., swing phase and stance phase. If we observe one of the two legs, the swing phase is when the leg is not in contact with the ground and swing forward to reach the next contact placement. The stance phase is when the foot is in contact with the ground and the other leg swings forward. The stance phase starts with a heel-contact and ends with a toe-off, which in turn is the start of the swing phase, ending with heal-contact (see Fig. 2.5 and [20] for details).

The leg movement can be modeled as an inverted pendulum, provided that we have compliant components in our leg. The SLIP (spring-loaded inverted pendulum) model is in fact a classical model describing biped locomotion. Figure 2.6 [21] shows that the leg is modeled as a spring, with stiffness k and length l_0 , and the mass m is concentrated in a point at the top of the leg. The angle θ between the leg and the ground at the stance start is named angle of attack. After contact, the stance phase brings the inverted pendulum to the start of the swing phase, with an opposite angle of attack.

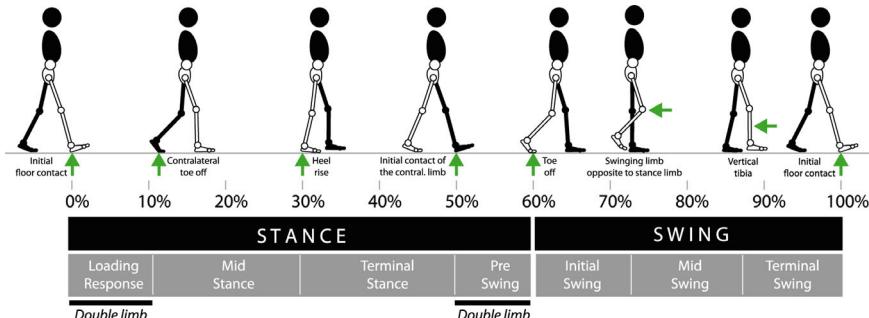


Fig. 2.5 Phases of biped walking, from [20]

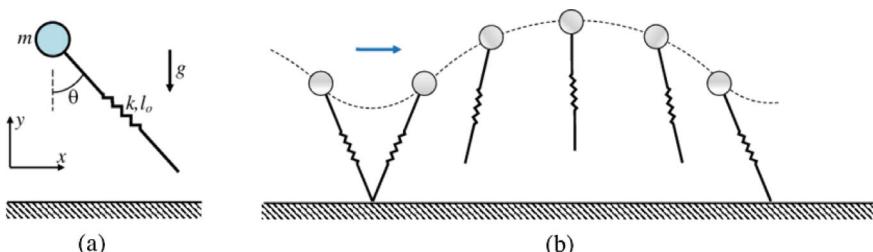


Fig. 2.6 SLIP model for biped walking, from [21]. **a** Model of the leg. **b** Sequence of one leg positions in a full cycle of stance and swing phases

In robots with rigid legs, biped locomotion is often implemented with a different strategy and model, by generating dynamically stable walking. The basic underlying concept is the zero-moment point (ZMP), i.e., the point where the resultant of forces intersects with the ground. It is the point on the ground surface where the net angular momentum is equal to zero. If it exists, it is constrained to lie within the support polygon, which is the convex hull of all points in contact with the ground.

In ZMP-based walking, one or both feet of the robot are always in contact with the ground. The robot can keep balance by making sure that the support polygon contains the ZMP, by controlling the position of the Centre of Mass (COM). In static walking, the projection of COM always stays inside the support polygon. If the robot completely stops moving at any moment during walking, it does not fall down because, for the robot at rest, the projection of COM onto the ground surface is equal to the ZMP. In dynamic walking, the projection of COM is not equal to ZMP and can fall outside of the support polygon during some period of motion.

2.4.2 *Multi-legged Locomotion*

Nature provides uncountable models for multi-legged locomotion, from quadruped mammals to six-leg insects, up to terrestrial and marine animals with 8 or more legs. Despite advantageous in terms of stability, multi-legged locomotion requires coordination of more complex patterns of leg motions, with respect to biped locomotion. The discovery of the role of a Central Pattern Generator (CPG) in the spinal cord in the '70s [22] provides a model for describing the generation of most rhythmic movements. Discovered in the lamprey spinal cord, CPGs have been demonstrated to be common to vertebrates and to explain locomotion, including biped walking, among other rhythmic movements.

CPGs are neuronal networks producing coordinated patterns of rhythmic activity. They are organized as couples of elements with at least one per limb. Sensory feedback is not needed for generating the rhythmic activity, but it plays a role in defining the rhythmic patterns and coordinate multiple limbs. Higher-level brain structures trigger and module the movement patterns, based on the perception of external conditions.

The basic CPG mechanism can be modelled with couples of oscillators, with descending connections that activate them sequentially and with lateral connections that activate alternate limb motion. Figure 2.7 shows the CPG model for a salamander locomotion, which combines body oscillations and limb movements. CPG are increasingly used in multi-legged locomotion in robotics. See [23] for an exhaustive review.

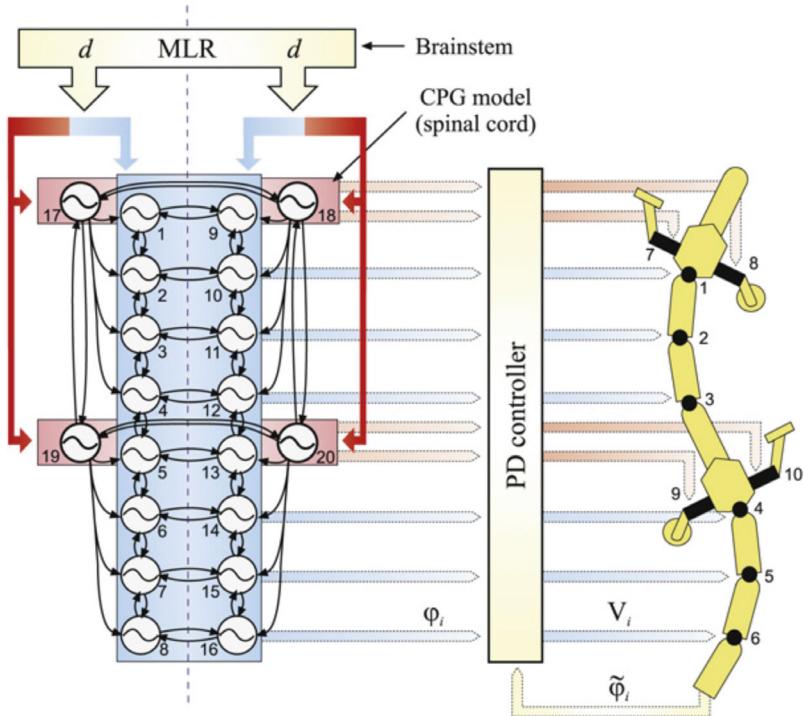


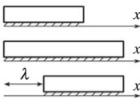
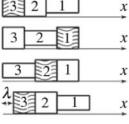
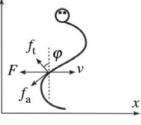
Fig. 2.7 A scheme of CPG, reproduced with permission from [23]. The model is composed of 20 oscillators which receive a drive signal from the brain representing the descending stimulation. The outputs of the CPG are desired joint angle positions that, in a robot, can be reached with a PID controller

2.4.3 Crawling

Crawling requires a soft, deformable body. Crawling animals are then interesting models for soft robots. We may identify three main crawling principles [24]: two-anchor, peristaltic, serpentine.

In two-anchor crawling (see caterpillars as examples), forward motion is obtained by body elongations and shortenings, with a friction coefficient which is lower in the forward direction than in the backward one. This is obtained, for instance, with backward-oriented bristles. The body is elongated to reach forward and the front part adheres to the ground without slipping backward when the body is shortened and the hind part is brought forward. Overall, this sequential elongation and shortening generates a forward motion (see Fig. 2.8, top).

In peristaltic crawling (typical in worms, as well as some human body organs, like the digestive system), forward motion is generated by a wave of radial contraction which travels backward along the body (see Fig. 2.8, middle).

gait type	figure	key locomotion law	description
crawling two-anchor		$\mu_{-x} > \mu_x$	A small element of the body with mass δm exerts a force of δmg on the ground. When the whole body elongates by λ , the friction force in the forward direction is $\delta mg\mu_x$, and should be smaller than that in the backward direction, $\delta mg\mu_{-x}$ (where μ_x and μ_{-x} are, respectively, kinetic and static friction coefficients).
peristaltic		$\mu_{-x}mg(1-n) > \mu_xmgn$ $n < \frac{\mu_{-x}}{\mu_{-x} + \mu_x}$	By considering a unitary body, the part of the body which is not elongated $[1 - n]$ should provide an anchor for the portion of the body n which is elongating: this means that the friction force in the backward direction should be smaller than that in the forward direction. From this relationship, we can estimate the portion of the body which can be elongated during each cycle.
serpentine		$\mu_t mg \cos \varphi \geq \mu_a mg \sin \varphi$ $\tan \varphi < \frac{\mu_t}{\mu_a}$	The body is subjected to an oscillatory motion, such that a small element of mass δm is moving in direction v and is subject to a friction force of F . If we split F into its tangential f_t and axial f_a components (with respect to the body), forward locomotion is possible when tangential μ_t and axial μ_a friction coefficient are properly designed with respect to body motion, φ .

Exploitation of deformable components for:

Adaptation to the environment; capability to work in harsh conditions; resilience to damages; reduced number of actuators.

Fig. 2.8 Bioinspired crawling models for soft robots, reproduced from [24]

In serpentine locomotion (see snakes as an example), forward motion is generated by a wave of deformation and proper friction coefficients. If the soft body is deformed as a perfect sinusoidal shape, it moves forward only if the forward component of transverse frictional force is greater than the backward component of axial frictional force, which yields a relationship between the body curvature and the friction coefficients (see Fig. 2.8, bottom).

2.5 Burrowing and Underground Locomotion

Classically adopted solutions for soil drilling and exploration still rely on cumbersome, energy-demanding, and intrusive technologies.

Current challenges include reducing the stresses exercised on the system, the energy expenditure, increasing the drilling system stabilization, providing steerability and branching to increase exploration of ample areas, and ensuring safety for the system, the operator, and the environment (e.g., reduction of vibrations, fuel leakage, soil removal).

Modeling soil penetration is highly challenging. Several complex processes occur during soil interactions, such as the separation of soil layers, the appearance of cracks, and the flow of soil particles. Since processes involving large displacements

and deformations cannot be modeled properly with finite element analysis, DEM numerical models were often preferred for modeling soil operations.

In this context, a bioinspired approach aims to identify strategies of movement and exploration in living organisms that are relevant for improving the performance of artificial systems. DEM-based models can be used to analyze bioinspired burrowing strategies to extend the findings to engineering applications.

Three-dimensional (3-D) DEM-based numerical model can also represent an interesting approach to simulate the penetration of a digger into the soil. In [25], a root-inspired digger penetrates a cohesionless granular medium by growing axially from the tip and building its structure, whereas its upper structure remains fixed with the surrounding soil. In particular, this study investigates the penetration force the intruder must apply to explore the soil successfully. Force requirements are analyzed as a function of the system size (diameter, D_{root}) and soil granularity (median particle size, D_{50}). Moreover, the estimated penetration pressure the root-like intruder needs to move into the soil is compared with the one needed by a system penetrating while being pushed from the top. The developed model can provide important guidelines on the intruder design specifications such as dimension, geometry, and actuation system requirements given soil specifications.

Among burrowing organisms, worm-like animals also provide interesting solutions for soil penetration. The basic musculature arrangement of these animals is characterized by circular and longitudinal muscle fibers. In annelids, such as earthworms, the coelom is divided into segments (metameres) by muscular septa. These septa prevent the movement of hydrostatic fluid from one segment to another during normal locomotion. Earthworms use alternating contractions of antagonistic muscles within these constant volume chambers to propel themselves below and above the soil surface, generating retrograde peristaltic waves.

When moving below the surface, the earthworm undergoes cycles of alternate subterminal radial expansion and longitudinal elongation, facilitating burrowing by extending the anteriorly placed discoidal crack. Additionally, radially expanded regions help the organism to anchor itself within the surrounding medium. Earthworm movement has been likened to plant root propagation, and the equations governing this penetration-expansion model have been found applicable to earthworm subsurface burrowing. While crawling above the surface, the earthworm's ventral side features hair-like protrusions that generate an anisotropic frictional force, preventing backward slipping and resulting in forward locomotion (Fig. 2.9).

Mechanical models have been developed to quantify common mechanical processes involved in soil penetration by earthworms and growing plant roots, including the energetic requirements for soil plastic displacement [26]. The study presents a basic mechanical model considering cavity expansion into plastic wet soil, involving wedging by root tips or earthworms through cone-like penetration. This is followed by cavity expansion due to pressurized earthworm hydroskeleton or root radial growth. Mechanical stresses and resulting soil strains determine the energy required for bioturbation under different soil hydro-mechanical conditions, considering a realistic range of root/earthworm geometries. Modeling results suggest that higher soil water

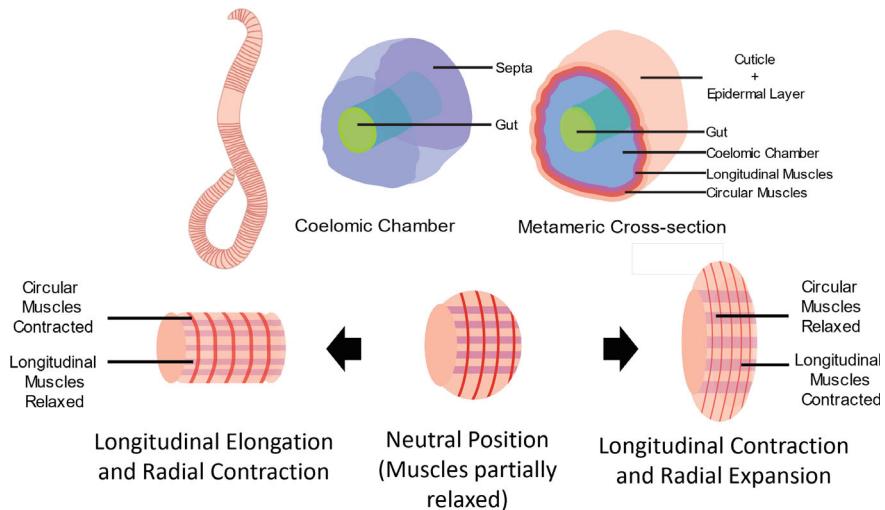


Fig. 2.9 Anatomy of earthworm segment cross-section showing constant volume coelomic chambers and setae and the working principles of the antagonistic muscles

content and reduced clay content reduce the strain energy required for soil penetration. The critical earthworm or root pressure increases with the diameter of the root or earthworm, but results are insensitive to the cone apex (the shape of the tip).

The invested mechanical energy per unit length increases with increasing earthworm and plant root diameters, whereas mechanical energy per unit of displaced soil volume decreases with larger diameters. This study provides a quantitative framework for estimating the energy requirements for soil penetration work done by earthworms and plant roots, and delineates intrinsic and external mechanical limits for bioturbation processes (Fig. 2.10).

The estimated energy requirements for earthworm biopore networks are linked to the consumption of soil organic matter, suggesting that earthworm populations are likely to consume a significant fraction of ecosystem net primary production to sustain their subterranean activities [26].

2.6 The Sense of Touch

Tactile perception is a fundamental sensory components in all living organisms. In humans, the sense of touch is conveyed by a distributed set of receptors, differently from other senses that rely on sensory organs. The skin can be considered as our sensory organ for touch. The resolution of receptor varies greatly in our skin, from below millimeter to a few centimeters, depending on the body part. The fingertip shows the highest resolution and it is estimated to contain around 2000 receptors. Four

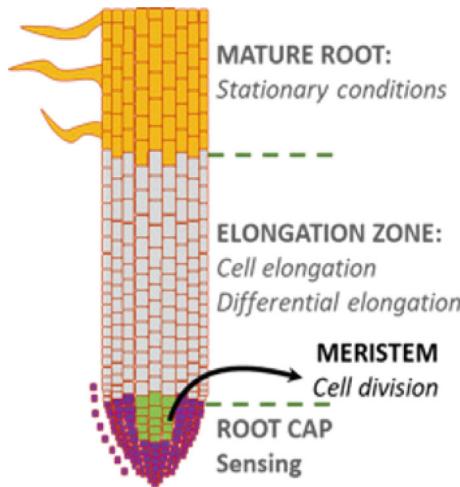


Fig. 2.10 Plant root structures and their functions. Four main regions can be identified, each with different roles. Root cap cells exude mucilage and slough off, which decreases friction during penetration by lubricating the surrounding soil. New cells are created by mitosis in the meristematic region. These cells are then elongated by osmotic pressure and move to the elongation zone. Cell division and cell elongation provide the force to penetrate into the soil; asymmetries with respect to the root axis results in bending. The mature root consists of elongated cells that are stationary and provide a strong anchor to the soil, thus supporting tip penetration

main types of human tactile receptors, or mechanoreceptors, have been identified and classified according to two important concepts: adaptation and receptive field [27].

Mechanoreceptors are specialized neurons where a mechanical stimulus deforms the membrane and open the channels that allow Na^+ and K^+ ions to flow, triggering the spiking mechanism. Due to the membrane elasticity, channels tend to close back after some time. Then, a constant stimulus is no more perceived. The speed of such adaptation classifies human mechanoreceptors as FA (Fast Adapting) or SA (Slowly Adapting).

The receptive field of a mechanoreceptor is the skin area it responds to. Depending on the depth on a mechanoreceptor in the skin tissues, the area is larger (deep mechanoreceptor) or smaller (superficial mechanoreceptor). The size of a receptive field set the resolution, too. Based on their receptive field, mechanoreceptors are classified are type I or type II.

Crossing the two classification axes, we have four types of mechanoreceptors, as follows:

- FA-I: Meissner corpuscles. They represent 43% of hand tactile units and are located in the extroflections corresponding to fingerprints in the dermis, with main axis normal to the skin surface. Their dimension is $80 \times 30 \mu\text{m}$. They respond to stimuli of 10.2 mN/mm^2 .

- SA-I: Merkel discs. They represent 25% of hand tactile units and are located in the extroflections corresponding to fingerprints in the dermis, arranged parallel to the skin surface. Their diameter is 10–15 μm . They respond to stimuli of 22.8 mN/mm^2 .
- FA-II: Pacini corpuscles. They are the deepest and the largest of our skin receptors. They represent 13% of hand tactile units and are located in the ipodermis. Their length is between 1 and 4 mm and their diameter is between 0.5 and 1 mm. Their threshold of activation is 9.5 mN/mm^2 . They have a high frequency response, which allows vibration detection.
- SA-II: Ruffini corpuscles. They are the less abundant, accounting for 19% of hand tactile units, in the dermis. Their activation threshold is 131.6 mN/mm^2 .

Remarkably, the combination of fingertip receptor responses encode the variety of tactile stimuli that we perceive and enable manipulation tasks, by detecting critical control points. In a simple grasp and lift task, FA receptors detect contact/release with the object and lift-off/landing of the object on the supporting plane, while SA detect constant contact [28]. The combination of fingertip receptors detect object shape (e.g., surface curvature), force direction, and mechanical properties (e.g., friction, mass distribution) [29]. We can then consider the fingertip as a tactile organ. Figure 2.11 summarizes the functions of the single receptors as well as the role of the fingertip as a tactile organ.

The description and numbers above gives a good guideline and outline how challenging is to provide robots with the sense of touch. Despite matching the richness, high resolution and sensitiveness of human touch is still an open challenge in engineering, soft robotics technologies offer viable solutions for building deformable skins with embedded sensing units. Working principles range from piezoresistive and resistive to optical, capacitive and magnetic. Smart materials also provide continuous sensing skins. See [30] for a review.

TACTILE AFFERENT	CLASS	DIMENSION	RECEPTIVE FIELD (diameter)	RESPONSE THRESHOLD* (pressure)	FREQUENCY	FUNCTION	ROLE IN GRASP CONTROL (FINGERTIP)
MEISSNER	FA-I	80x30 μm	9.4 mm^2 (3.4 mm)	0.58 mN (10.2 mN/mm^2)	8-64 Hz	THIN SHARP EDGES, VELOCITY CHANGES AND PRESSURE CHANGES	CONTACT, LOCAL SHAPES, FORCES ON THE FINGERTIP
MERKEL	SA-I	10-15 μm (diameter)	11 mm^2 (3.7 mm)	1.3 mN (22.8 mN/mm^2)	2-32 Hz	THIN SHARP EDGES, PRESSURE INTENSITY	CONTACT, FORCES ON THE FINGERTIP
RUFFINI	SA-II		58.9 mm^2 (8.7 mm)	7.5 mN (131.6 mN/mm^2)	< 8 Hz	THICK SMOOTH EDGES, PRESSURE INTENSITY	SKIN LATERAL STRETCH, FORCE ON THE FINGERTIP
PACINI	FA-II	1-4 mm x 0.5-1 mm	101.3 mm^2 (11.4 mm)	0.54 mN (9.5 mN/mm^2)	64-400 Hz	THICK SMOOTH EDGES, VIBRATIONS	MECHANICAL TRANSIENTS, VIBRATIONS
FINGERTIP						<ul style="list-style-type: none"> • FORCE ON THE FINGERTIP: INTENSITY AND DIRECTION • CONTACT LOCAL SHAPE: CURVATURE OF OBJECT SURFACE • TACTILE CONTROL EVENTS: SLIPPAGE, CONTACT START/END, START/END OF OBJECT CONTACT WITH THE ENVIRONMENT 	

Fig. 2.11 Human mechanoreceptors have different functions, depending on their class, and their combination enable the detection of contact characteristics and manipulation control events. 90% of type SA-I and FA-I receptors respond to a stimulus of 5 mN, corresponding to 87 mN/mm^2 , applied with a Von Frey hair of diameter 0.27 mm

2.7 Summary

Nature offers valuable insights for a variety of functions relevant to robotics, particularly in the context of movement and locomotion. Often, the application of these principles necessitates a soft and deformable body. Soft robots, in turn, can attain bioinspired capabilities that traditional robotics approaches cannot replicate.

This chapter presents a general methodology for identifying relevant principles within natural systems and translating them into models suitable for implementation in soft robots. Furthermore, the chapter introduces several foundational models that can readily be employed in soft robot design. These include fundamental models for movement, both with and without muscles, incorporating strategies inspired by plants. Additionally, models for legged locomotion, crawling, and burrowing are discussed.

It's worth noting that the potential of bioinspired principles in the realm of soft robotics remains largely unexplored. Many principles are yet to be discovered, with most of the known ones serving as sources of inspiration rather than being described through mathematical models. This chapter aims to contribute to their practical application in the field of soft robotics.

References

1. Hill AV (1938) The heat of shortening and the dynamic constants of muscle. *Proc Roy Soc Lond. Ser B Biol Sci* 126(843):136–195
2. Hooren B, Bosch F (2017) Is there really an eccentric action of the hamstrings during the swing phase of high speed running? Part I: a critical review of the literature. *J Sports Sci* 35:2313–2321
3. Kier WM (2016) The musculature of coleoid cephalopod arms and tentacles. *Front Cell Dev Biol* 4
4. Martone PT, Boller M, Burgert I, Dumais J, Edwards J, Mach K, Rowe N, Rueggeberg M, Seidel R, Speck T (2010) Mechanics without muscle: biomechanical inspiration from the plant world. *Integr Comp Biol* 50(5):888–907
5. Forterre Y (2013) Slow, fast and furious: understanding the physics of plant movements. *J Expe Bot* 64(15):4745–4760
6. Darwin C, Darwin FE (1888) The' power of movement in plants—1880
7. Hart JW (1990) Plant tropisms: and other growth movements. Springer Science & Business Media
8. Dumais J, Forterre Y (2012) “Vegetable dynamicks”: the role of water in plant movements. *Annu Rev Fluid Mech* 44:453–478
9. Mazzolai B, Beccai L, Mattoli V (2014) Plants as model in biomimetics and biorobotics: new perspectives. *Front Bioeng Biotechnol* 2:2
10. Burgert I, Fratzl P (2009) Actuation systems in plants as prototypes for bioinspired devices. *Philos Trans Roy Soc A Math Phys Eng Sci* 367(1893):1541–1557
11. Sinibaldi E, Argiolas A, Puleo GL, Mazzolai B (2014) Another lesson from plants: the forward osmosis-based actuator. *PloS One* 9(7):e102461
12. Speck T, Cheng T, Klimm F, Menges A, Poppinga S, Speck O, Tahouni Y, Tauber F, Thielen M (2023) Plants as inspiration for material-based sensing and actuation in soft robots and machines. *MRS Bull* 1–16

13. Abraham Y, Dong Y, Aharoni A, Elbaum R (2018) Mapping of cell wall aromatic moieties and their effect on hygroscopic movement in the awns of stork's bill. *Cellulose* 25:3827–3841
14. Abraham Y, Elbaum R (2013) Hygroscopic movements in Geraniaceae: the structural variations that are responsible for coiling or bending. *New Phytol* 199(2):584–594
15. Fratzl P, Barth FG (2009) Biomaterial systems for mechanosensing and actuation. *Nature* 462(7272):442–448
16. Mazzolai B, Mariani S, Ronzan M, Cecchini L, Fiorello I, Cikalleshi K, Margheri L (2021) Morphological computation in plant seeds for a new generation of self-burial and flying soft robots. *Front Robot AI* 8
17. Huss JC, Spaeker O, Gierlinger N, Merritt DJ, Miller BP, Neinhuis C, Fratzl P, Eder M (2018) Temperature-induced self-sealing capability of banksia follicles. *J Roy Soc Interface* 15(143):20180190
18. Cecchini L, Mariani S, Ronzan M, Mondini A, Pugno NM, Mazzolai B (2023) 4d printing of humidity-driven seed inspired soft robots. *Adv Sci* 10(9):2205146
19. Mazzolai B, Mariani S, Ronzan M, Cecchini L, Fiorello I, Cikalleshi K, Margheri L (2021) Morphological computation in plant seeds for a new generation of self-burial and flying soft robots. *Front Robot AI* 8:797556
20. Torricelli D, Gonzalez J, Weckx M, Jiménez-Fabián R, Vanderborght B, Sartori M, Dosen S, Farina D, Lefeber D, Pons JL (2016) Human-like compliant locomotion: state of the art of robotic implementations. *Bioinspir Biomim* 11(5):051002
21. Or Y, Moravia M (2016) Analysis of foot slippage effects on an actuated spring-mass model of dynamic legged locomotion. *Int J Adv Robot Syst* 13(1):04
22. Grillner S (1975) Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiol Rev* 55(2):247–304 PMID: 1144530
23. Auke Jan Ijspeert (2008) Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw* 21(4):642–653
24. Calisti M, Picardi G, Laschi C (2017) Fundamentals of soft robot locomotion. *J Roy Soc Interface* 14(130):20170101
25. Pirrone SRM, Dottore ED, Sibile L, Mazzolai B (2023) A methodology to investigate the design requirements of plant root-inspired robots for soil exploration. *IEEE Robot Autom Lett* 8(6):3438–3445
26. Ruiz S, Or D, Schymanski SJ (2015) Soil penetration by earthworms and plant roots—mechanical energetics of bioturbation of compacted soils. *PLoS One* 10(6):e0128914
27. Johansson RS, Vallbo AB (1979) Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin. *J Physiol* 286(1):283–300
28. Randall Flanagan J, Bowman MC, Johansson RS (2006) Control strategies in object manipulation tasks. *Curr Opin Neurobiol* 16(6):650–659. Motor Syst/Neurobiol Behav
29. Jenmalm P, Birznieks I, Goodwin AW, Johansson RS (2003) Influence of object shape on responses of human tactile afferents under conditions characteristic of manipulation. *Eur J Neurosci* 18(1):164–176
30. Roberts P, Zadan M, Majidi C (2021) Soft tactile sensing skins for robotics. *Curr Robot Rep* 2:07

Chapter 3

Soft Robots as a Platform to Understand Embodied Intelligence



Thrishantha Nanayakkara, Barry Mulvey, Shehara Perera, Yukun Ge,
Zhenhua Yu, and Parvathi Sunilkumar

Abstract This chapter addresses some of the possible missing pieces to understand the gulf between robots and biological beings in negotiating movements in a natural world. More specifically, we discuss the overarching question of understanding how passive dynamics and computing can come together to solve complex interaction problems. The elegance of physical interactions we have in daily life appears to come from a fine interplay among the dynamics and computing in the brain, the body, and the environment. However, it is difficult to map this out using human participants alone. Can we use robots as a proxy to living beings to understand the nuances of such interactions since we can control and measure details of information flow? We will group examples under three phenomena we believe are important to untangle the puzzle: Ghost circuits, kinematic tuning, and behavioral lensing on which we will elaborate further in this chapter.

3.1 Introduction

Nature is full of fascinating survival secrets. Mountain goats can conquer steep cliffs, trout can navigate turbulent waters, frogs can jump 10–20 times their body length, and owls can surprise their prey by approaching in silence. Though we generally give

T. Nanayakkara (✉) · B. Mulvey · S. Perera · Y. Ge · Z. Yu · P. Sunilkumar
Imperial College London, South Kensington, London, UK
e-mail: t.nanayakkara@imperial.ac.uk

B. Mulvey
e-mail: b.mulvey21@imperial.ac.uk

S. Perera
e-mail: u.perera@imperial.ac.uk

Y. Ge
e-mail: yukun.ge20@imperial.ac.uk

Z. Yu
e-mail: z.yu18@imperial.ac.uk

P. Sunilkumar
e-mail: p.sunilkumar22@imperial.ac.uk

full credit of the intelligence of living beings to their Central Nervous System (CNS), the physical body seems to hold some secrets of survival in natural environments.

In fact, the brain appears to know the functionality of the shape and dynamics of the body with which it works. For example, a mountain biker would instinctively adjust their position by bending their knees on rough terrain. Someone would pull their chair to a comfortable distance from the table before beginning to write or type. A physician would regulate the shape and stiffness of their hands and fingers during physical examination of a patient's abdomen. Consider how we would estimate the weight of an object: we would instinctively lift it and gently move it up and down to help us make our prediction.

These observations suggest that the way we feel the world and the stability of interactions depend on the kinematic and dynamic parameters of the body and how the CNS interacts with them.

The broad area of embodied intelligence views that the mechanisms in the physical body contribute to simplify the sensorimotor computation needed to survive [1, 2]. These ideas are based on the notion of embodiment and embeddedness of physically grounded agents exhibiting intelligent behavior [3]. The term *embodied* means that they have a physical body with certain kinematic and dynamic properties that determine how they interact with the environment. For instance, a mountain goat has a weight and size. It also has hoofs tuned to suit the steep cliffs. This broadly explains its embodiment. The term *embedded* means that the agent has to be an integral part of the environment and has to meet conditions imposed by the environment to survive. One such condition in the case of a mountain goat on a cliff is to be able to compute counter-slip action before slipping exceeds a limit (i.e. a deadline on meaningful reaction). Other conditions can range from constraints imposed by visibility, temperature, humidity, noise, and scarcity of food in the environment. The term *physically grounded* means that an embodied agent is in a network of other embodied units influencing each other. Since all physically grounded agents in a system of systems have to meet conditions imposed by their environment, they are under evolutionary pressure to tune embedded problem-solving mechanisms to be as efficient as possible.

With this background, we can broadly explain the term *embodied intelligence* to refer to the arbitration among sensors, neural circuits, and other bodily formations such as the musculo-skeletal system to meet conditions imposed by the environment. An important argument in this discussion is that the physical body should allow real-time tuning of its kinematics and dynamics to accomplish efficient embodied computation. One facet of embodied intelligence involves the fundamental capacity for adaptive structural adjustments to facilitate sensor-independent responses. For instance, passive dynamic walkers demonstrate ability to self-balance and walk without any sensors, numerical algorithms, or motors. Our recent work on taking a robotics approach to understand the ability of mountain goats to climb cliffs without slipping revealed that passive dynamics in the hoof—given the right combination of

stiffness in its joints—can improve slip resistance compared to a spherical foot [4]. These examples show that adaptations spanning across neural and physical structures are important for real-time action to maximise rewards.

3.2 Entropy and Information Theory

3.2.1 Shannon Entropy

Shannon entropy of a random variable x is defined as

$$I(x) = - \sum_{i=1}^n p(x_i) \log |p(x_i)| \quad (3.1)$$

where x_i is the i th value assumed by x , and n is the total number of states x can occupy.

For instance, let x be the volume of rainwater collected per hour in a jar kept outside. This can range from 0 to a highest recorded volume V_{max} . One can then break the range into n number of ranges of volumes with increments given by $\frac{V_{max}}{n}$. Therefore the i th level of volume would be $i \frac{V_{max}}{n}$. Then

$$p(x_i) = p\left((i-1)\frac{V_{max}}{n} \leq x < i\frac{V_{max}}{n}\right) \quad (3.2)$$

In this case, on a dry day the first volume level will have a probability of 1, and the other 0. Then Eq. 3.1 gives 0 for all volume levels adding up to 0. This will also be true for a very rainy day, because it will give a probability of 1 to the highest bin and 0 to the others. Therefore, Shannon entropy is a measure of the value of obtaining new samples. On a very dry day or a very rainy day, once you consistently get 0 entropy for new samples, there is no point in collecting more data. In other words, we can say that the system is predictable when the Shannon entropy is zero. However, when it has a value, there is uncertainty in the random variable.

3.2.2 Transfer Entropy or Information Gain

Transfer entropy quantifies the reduction of uncertainty of a random variable due to some action taken in a parameter or another variable that can influence the behavior of the said random variable given by

$$G(x) = - \sum_{i=1}^n p(x_i|\zeta_1) \log \left| \frac{p(x_i|\zeta_1)}{p(x_i|\zeta_0)} \right| \quad (3.3)$$

where ζ is a vector of control parameters. Imagine x is the vibration readings taken from an Inertial Measurement Unit (IMU) on an outdoor robot. The level of vibrations can depend on both the terrain conditions and the speed of the robot. Since the speed can be controlled but the terrain conditions cannot, in this case ζ can be the speed of the robot. Then if the uncertainty of the vibration levels drops at ζ_1 relative to those at ζ_0 , there is an entropy reduction by moving from ζ_0 to ζ_1 . The uncertainty reduction is also viewed as an information gain because the robot can make faster decisions when it feels the world with certainty.

3.2.3 Mutual Information

Mutual information quantifies the reduction of uncertainty in one random variable when there is knowledge about another random variable given by

$$H(x, y) = I(x) - I(x|y) \quad (3.4)$$

where x and y are random variables. In other words, two random variables are independent if mutual information is equal to the Shannon entropy of any one of the random variables.

Mutual information can also be written in the format of the transfer entropy given by

$$H(x, y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left| \frac{p(x, y)}{p(x)p(y)} \right| \quad (3.5)$$

where $p(x, y)$ is the joint probability distribution of x and y . The physical meaning is that mutual information can be viewed as the transfer entropy of having a joint distribution of x and y compared to x and y being two independent random variables.

Proof:

$$\begin{aligned} H(x, y) &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left| \frac{p(x, y)}{p(x)p(y)} \right| \\ &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left| \frac{p(y)p(x|y)}{p(x)p(y)} \right| \\ &= \sum_{x \in X} \sum_{y \in Y} p(x, y) [\log |p(x|y)| - \log |p(x)|] \end{aligned}$$

$$\begin{aligned}
&= \sum_{x \in X} \sum_{y \in Y} p(x)p(y|x) [\log |p(x|y)| - \log |p(x)|] \\
&= \sum_{x \in X} \sum_{y \in Y} p(x)p(y|x) [\log |p(x|y)| - \log |p(x)|] \\
&= \sum_{x \in X} \sum_{y \in Y} p(x)p(x|y)p(y) [\log |p(x|y)| - \log |p(x)|] \\
&= \sum_{x \in X} \sum_{y \in Y} -p(x)p(y)I(x|y) + p(x|y)p(y)I(x)
\end{aligned} \tag{3.6}$$

Since $\sum_{x \in X} \sum_{y \in Y} p(x)p(y) = 1$ and $\sum_{x \in X} \sum_{y \in Y} p(x|y)p(y) = 1$,

$$\begin{aligned}
\sum_{x \in X} \sum_{y \in Y} -p(x)p(y)I(x|y) + p(x|y)p(y)I(x) &= I(x) - I(x|y) \\
H(x, y) &= I(x) - I(x|y)
\end{aligned} \tag{3.7}$$

3.3 Definitions of Essential Concepts

3.3.1 Kinematics

In Greek, “kinein” is to move and “kinema” is motion. In the 19th century the latter was anglicised to refer to motion of objects without referring to the forces and moments that cause them to move. For instance, if we open a door mounted on hinges, the outer edge of the door always moves along the periphery of a circle with a center at the hinge. The relationship between the angular speed of the door and the speed of the outer edge depends on the width of the door (i.e. radius of the circle). Such relationships of motion are called kinematic relationships. Forward kinematics involves computing the Cartesian movements when the angular movements are known. In the case of a door for instance, forward kinematics involve computing the position of the edge of the door in some Cartesian coordinate system when the rotation angle of the door around its hinge is known. Inverse kinematics involve knowing the rotation angle when the position of the door edge is known.

We use such kinematic relations to plan how to move objects. For a door, this is simple and easy to comprehend. But kinematics of most biological bodies and robots are not trivial when there are many axes to move different parts of the object. For instance, when we want to reach to grasp an object, it involves solving inverse kinematics to command joints in our hands to move in particular ways to perform a reaching task. How biological brains solve this complex problem for many body parts at the same time is not well known. The computational problem becomes very complex when the number of joints involved increase. For instance, if we want to make a robot arm with 3 joints to move to a point in 3D Cartesian space, inverse

kinematics involve calculating how the 3 joints should be moved to reach the 3D point. However, a robot arm with more joints, say 7 joints, will involve a more complex calculation because there are multiple ways 7 joints can be moved to reach the same point in 3D space.

3.3.2 Force Fields

A force field as shown in Fig. 3.1 has a force vector at any given point in the space in which it operates. For instance, the gravitational field has a force at any given point within its influence. If you dip a stick in a river, you experience a drag force due to the flowing water, and this force changes across the cross section of the river. A vortex of water has another force field of drag forces that try to bring an object towards the center of the swirl. Some evidence in neuroscience show that the spinal cord has pre-programmed force fields in the joint space of limbs. When one site of the spinal cord is excited, a limb would move as if it is attracted to a point in 3D space irrespective of the initial posture. Experiments have also found that activating multiple sites can render movements that can be explained by summing up the individual force fields. Force fields are very important to understand how behavior emerges due to an interaction among systems.

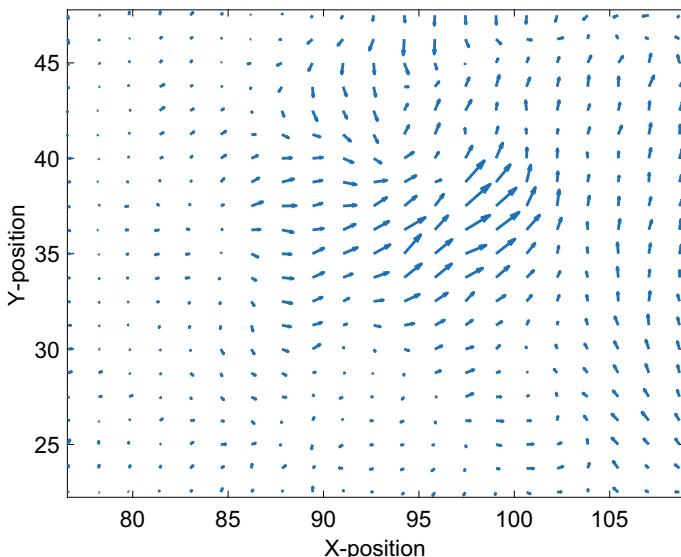


Fig. 3.1 A force field

3.3.3 *Dynamics*

The Greek word “dunamis” refers to power. It became “dynamics” in English in the 19th century to refer to movements and force relationships. If we consider an example of opening a door, even without friction, one has to apply a force to move the door. The required force depends on the mass of the door, the point at which force is applied, and the direction of the force. If we know the kinematics of the door, we will know that any point on the door moves on a circle with the center at the hinge axis of the door. We can use this kinematic relationship to minimize the force required to open the door by applying the force perpendicular to the door (along the tangent of the circle of movement). What matters in all other forces is the projection on the direction perpendicular to the door. Therefore, kinematics set out trajectories along which forces create movements. Therefore, forces required to render movements depend not only on the mass of an object, but also on the kinematics involved. When an object is linked to an environment with movable objects, we cannot separate the object and the environment to plan the required forces. The object and the environment can establish various kinematic relations in complex ways. Some kinematic constraints are visible while others are not. For instance, in the case of the door, when somebody’s hand pushes the handle of the door, a new kinematic system is formed by the door and the hand of the person. Now the motor programs in the brain must treat this new coupled kinematic system to solve the door closing problem. However, the kinematic system is visible. It is not so visible in the case of a fish swimming in water, because the links created by hydraulic vortexes are not apparent.

3.4 Broad Computational Objectives of Embodied Intelligence

Embodied problem solving involves finding ways to tune different parts of the body resulting in useful kinematic and dynamic relations to make perception more accurate and action more timely, efficient, and stable. The CNS of living beings constantly try to tune the body to accomplish this goal. In the following sections, we expand the discussion to propose several key computational goals for embodied intelligence.

3.4.1 *Realtime/Embedded Computing*

Realtime computing, also known as embedded computing in physical robots, refers to a computing framework organized to process input information to produce “good enough” action to meet deadlines in each context. Therefore, computations that have short deadlines will require a different method than those with longer deadlines. For instance, if your finger touches a flame, the pain signal would travel through the

peripheral nervous system and take a shortcut through the spinal cord to generate motor action to take the hand away from the flame as fast as possible. If the computing of motor commands were to happen in the brain, the signals would take longer to travel and motor commands would be too late to meet the deadline on taking the finger away from the flame before it causes damage. The computation via the spinal cord is not as optimum as it would have been if the brain were to consider factors such as energy and avoiding obstacles during the hand movement. However, the sub-optimum spinal reflex meets the deadlines of taking the finger away from the flame. It is important to keep in mind that embedded computing is a subset of mechanisms in the broad notion of embeddedness of a physically grounded agent. For instance, if an animal slips on a steep slope, even spinal reflexes may be too late to act. Therefore, some other passive mechanisms in the body such as hoofs can kick into action even to perform a mundane slip resistance function before neural based commands arrive.

For instance, mountain goats (Fig. 3.2) who live on cliffs will need mechanisms to generate slip resistance action before a slip develops beyond a limit. It is important to note that optimal action is useless if it comes after damage is already done. Embedded computing imposes pressure to have mechanisms which generate good enough action to meet deadlines. The steep cliff environment of a mountain goat imposes strict deadlines to finish any computing/thinking to take counter-slip action. Since deadlines on a given cliff environment have a distribution due to varying cliff angles and surface properties, the bodily mechanisms also should have a hierarchy of solutions to generate good enough actions depending on the situation. Therefore, the embedded nature of computing in an embodied agent imposes evolutionary pressure to tune multiple mechanisms to generate dynamic reactions to states. Therefore, the evolutionary pressure is on the entire interplay among CNS, sensing, actuation, and the kinematics and dynamics of the body and the environment.



Fig. 3.2 Mountain goats on steep slopes in Spain. Courtesy, Akila Anuradha Ekanayake

The action generation process in a mountain goat can be a combination of motor control action involving the CNS and passive dynamics in the musculo-skeletal system in the hoofs and legs. In the case of figure skaters, we know that they brake on ice by turning the skating blades inwards while sliding the feet apart. This involves motor action of tuning the kinematics of the legs that result in favorable passive interaction dynamics to brake on ice within the expected time. Another example is hearing. Since hearing should happen when the auditory source is present, the passive dynamics in the tapered membrane in the cochlea provides fast frequency separation of sound signals so that the brain can process frequency domain signals. When pressure waves cause the tapered membrane to vibrate, different frequency components get separated into known locations along the tube. Cochlea hair then pick these frequency components to feed the brain. Mapping the pressure signal from its time domain to the frequency domain is a computation purely implemented using the physics of a tapered membrane vibration. This also means that when the viscosity of cochlea liquid surrounding the cochlear membrane changes, the computing function also changes.

3.4.2 *Entropy Reduction*

Shannon Entropy quantifies the amount of new information in sampling (please see Sect. 3.2.1 for more details). An agent interacting with a natural environment is faced with the challenge of estimating states of interaction such as force, vibrations, noise levels, etc. Then, unlike a rigid robot without many options, an agent with a soft embodiment can take various actions to influence the behavior of random variables to reduce their perceived uncertainty. This is known as transfer entropy reduction or information gain (see Sect. 3.2.2).

It is important to note that most biological sensors required for motor control are located in the muscles themselves. The spindle sensors that give us the perception of joint angles and their speeds are located among the muscle fibres. The tendon organs that give us the perception of forces are located at the interface between a muscle and a bone. Since such vital sensors in a biological agent are in a soft body that can be tuned through muscle activity (mediated by neural networks), sensors and the activity of motor neural networks cannot be easily separated. In other words, the body, the CNS, and sensors are entangled such that the CNS can tune the body to influence the functions of sensors. For instance, when a cyclist reaches a bumpy terrain, they come off the seat to ride with bent knees. The dynamics at the knees provide a mechanical attractor that reduces the uncertainty of the world that the brain must deal with. This cycling posture filters disturbances from the bumpy ground allowing the brain to focus on the riding task more efficiently. Otherwise, it would be inundated with sensor feedback about the shape of the ground. Here, the brain takes a smart approach by generating motor commands to tune the knee joint so that it simplifies the riding motor control problem.

3.4.3 Dynamic Tuning of State Space

An agent can benefit by tuning the physical kinematics and dynamics to change the size of the state space involved in control to suit the situation. For instance, in the case of vision, the foveated images passed from the retina help the neural circuits to dedicate limited computing resources to process relevant visual information for a particular context. In an octopus, unrolling the hyper-redundant tentacle along a straight line to catch prey uses passive dynamics to simplify the state space by fixing the zenith and azimuth in a spherical coordinate system.

For continuum bodies like octopus tentacles, the problem of interacting with uncertain hydraulic environments becomes very complex if we frame the control problem in the wrong way. As shown in Fig. 3.3, it is known that the octopus unwinds its tentacle along a straight line in a throwing action to grasp prey located at a point in 3D space [5]. If it misses the target, it does not try to correct the extended tentacle. Instead, it pulls the tentacle back and throws it again. This implies that natural systems use low dimensional kinematics to solve complex inverse kinematic problems. It should be also kept in mind that the octopus can exhibit such smart behavior in water. Therefore, the buoyancy of the tentacle in water and hydraulic vortices during unwinding to shoot at a target are part of the behavior we observe.



Fig. 3.3 The octopus tentacle rolls out along a straight line to reach a target. Courtesy, Cecilia Laschi

3.5 Basic Mechanisms of Embodied Intelligence (EI)

In this section, I will introduce three important phenomena to explain embodied intelligence. The idea of “ghost circuits” describes how coupled dynamics between an agent and the environment can exhibit a higher level of dynamic problem-solving capacity than the agent taken in isolation. The idea of “kinematic tuning” explains how an agent would tune its shape and mechanical impedance to suit motion and force control happening in parallel. The idea of “behavioral lensing” describes how an agent would take action to elicit favourable interaction dynamics to sharpen perception during interaction with physical objects.

3.5.1 *Ghost Circuits in Compliant System Interactions*

Species and their environment evolve together by one influencing the other. It is difficult to understand one entity in isolation because their status is essentially a summary of a long history of a dynamic interplay among all entities in a given eco-system. States such as velocity, acceleration, internal forces, stress, and strain levels in each soft entity appear and disappear in dynamic interactions. The transient and steady state levels of these states guide the evolution of an embodied agent in a process that tries to tune the embodied and embedded process of action and perception to survive. Therefore, parts of a mechanical network organised to solve a given dynamical problem can be distributed among several entities in each environment. A living being can have only part of such a complex mechanical network. The idea of ghost circuits comes from this phenomenon where a meaningful dynamic behavior can emerge and then dissolve in a collection of embodied entities depending on transient conditions.

This book introduces the notion of “ghost circuits” as a phenomenon centred around a system of systems coming together to establish meaningful dynamic relations to render useful forces and movements.

Let us take an example of a network of tapered springs shown in Fig. 3.4. Each spring can have different stiffness and length (kinematic parameters). When an external perturbation is given to one point in the network, the whole network starts to oscillate with varying patterns of movements across the network as seen from the optical flow vectors. This system can be used to approximate a complex nonlinear dynamical system where the oscillation of a single point is linked to other elements in the network via multiple paths. Then, primitives of the nonlinear system can be tapped at various locations on the network to be linearly combined to approximate the overall nonlinear dynamical system. It is important to note that a change in one element such as the spring circled in the network can influence the entire network behavior depending on how it is coupled with other elements.

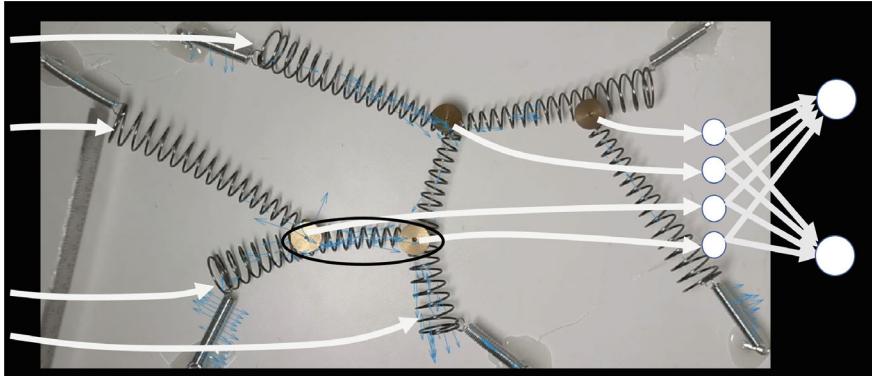


Fig. 3.4 A network of springs interacting together to accomplish a computational goal

3.5.1.1 Example of a Spider and the Web

For instance, in the case of a spider and the web shown in Fig. 3.5, the spider (as one system) can establish a bigger system together with the web to improve the efficiency of accomplishing a goal such as locating its prey on the web. The spider and the web work together to solve the problem of locating the prey through kinematic pathways that can be tuned. Then the desired dynamics emerge when the right conditions exist in the kinematic relationships. In the spider's case, it can be a reduction in uncertainty of the location of the prey felt by the spider's haptic sensors.

3.5.1.2 Example of Turbulent Water and Efficient Swimming of Fish

Another example is the observation of a dead trout “swimming” against turbulent water, as shown in Fig. 3.6. We can try to understand the mechanisms and control pathways involved by fitting the convenient model where a brain sends motor commands to the muscles to create wavy movements to swim. However, the observation that a dead trout body can move against a turbulent water stream challenges this conventional model [6].

Experiments under controlled conditions would reveal that turbulent water flowing against the body creates passive wavy movements in the fish body at the right frequency to create useful vortexes that push it against the stream. It then opens the possibility that the brain might be involved in minimalist interference such as tuning the stiffness of the body to produce useful vortexes in turbulent flow of water with minimum energy used for purposeful swimming. The dynamic coupling between the body and the stream of turbulent water can be viewed as a “ghost circuit” between the organism and the environment. Part of this mechanical circuit exists in the environment. In the case of the trout, this part is in the flowing water, while the other part is in the body. These separate parts come together in a unified dynamic system



Fig. 3.5 A spider in a web. Courtesy, Parvathi Sunilkumar

to render useful behavior when certain conditions are satisfied, such as stiffness of the body and flow of the water. The circuit then dissolves when the conditions are not satisfied.

Force-generating mechanisms do not exist in either the fish body or the turbulent water in isolation. If turbulent water had this capacity, a rigid stick or a soft ribbon would also swim forward. The dead trout could swim forward because it is natural for the trout's body and the turbulent water to establish kinematic relations that organise interaction forces in a meaningful way. Therefore, we can infer that a live trout could tune the stiffness at different parts of the body to control how the interaction between the body and turbulent water create useful vortexes to render desired swimming dynamics.

Figure 3.6 also shows how a school of fish may make a larger ghost circuit where vortexes created around one fish are efficiently used by other fish through carefully positioning themselves along vortex pathways with correct phase lag.

It is difficult to observe this phenomenon of ghost circuits using conventional approaches such as anatomical analysis of the fish body alone. The phenomenon is

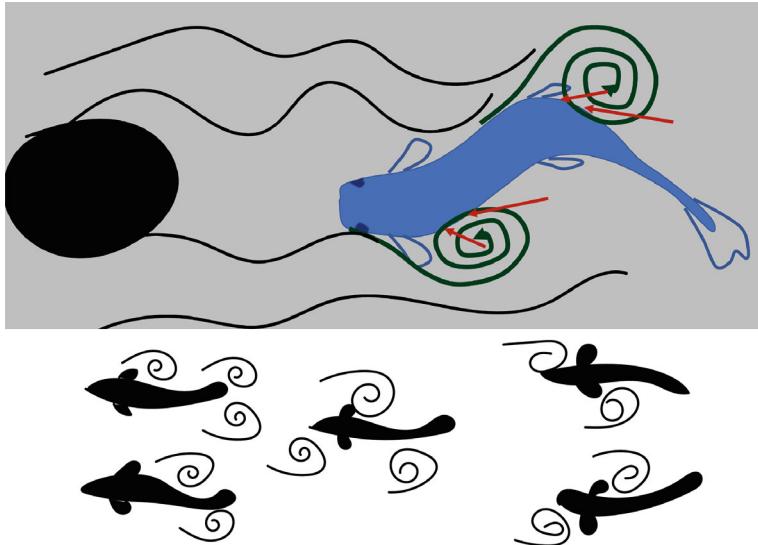


Fig. 3.6 Interaction between turbulent water and a fish

far more elusive because the body can be tuned using motor commands to stiffen different parts of itself to tune the ghost circuit with the environment. For this reason, we must take a new approach of viewing the entirety of the body, CNS, and the environment as parts that can interact with each other to produce useful behavior.

In the case of the dead trout, forces exerted on the water by its tail can reflect back on another part of the body through hydraulic channels. These hydraulic channels function like links in a rigid system such as previous example involving the human hand and the door. Therefore, these unseen kinematic chains determine how dynamics develop. Dynamics in turn determine how kinematic chains develop. Such inter-dependence often leads to illusive circuits along which forces and torques create movements.

The beauty of such illusive circuits is that they can be tuned by changing the behavior of a single object in the system of objects involved. If we take the spider and the web, it is hard to separate the spider, the web, and the branches of a tree in which the web exists when it comes to understanding the dynamics of movement of objects concerned. Therefore, the spider can tune the tension of the web in different patterns to simplify the problem of locating an insect stuck in the web.

Therefore, ghost circuits are kinematic meshes that can be tuned to produce different dynamics in the entirety of a system of systems. Consider a system of nonlinear dynamic systems given by

$$\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{\xi}_i, \boldsymbol{u}_i, t), \quad i = 1, 2, 3, \dots, N \quad (3.8)$$

where t is time, i is the index of the dynamic system, \mathbf{x}_i is its state variable vector, $\boldsymbol{\xi}_i$ is the vector of system parameters, and \mathbf{u}_i is the control input vector.

According to the contraction theory [7], the dynamic system $f_i(\cdot)$ is stable if

$$\forall \mathbf{x}_i, \forall t > 0, \quad \frac{1}{2} \left(\frac{\partial f_i}{\partial \mathbf{x}_i} + \frac{\partial f_i^T}{\partial \mathbf{x}_i} \right) < 0 \quad (3.9)$$

This means, a sub-system $f_i(\cdot)$ can tune its stability by searching for suitable $\boldsymbol{\xi}_i$ for bounded inputs. The sub-system $f_i(\cdot)$ is useful if the state vector \mathbf{x}_i moves within a cone $\psi_i(t)$ aiming at some desired direction at time t . A ghost circuit is a system of coupled sub-systems where the input of each sub-system can be a function of states of other systems given by

$$\mathbf{u}_i = g_i(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_N) \quad (3.10)$$

where $g_i(\cdot)$ is a nonlinear function that maps states to a control input.

In a linear case,

$$\mathbf{u}_i = W [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_i^T, \mathbf{x}_{i-1}^T, \mathbf{x}_{i+1}^T, \dots, \mathbf{x}_N^T]^T \quad (3.11)$$

where W is a matrix of coefficients to linearly combine state vectors of each sub-system.

It is important to note that the function $g_i(\cdot)$ plays a critical role for stable information transfer between two sub-systems of dynamic systems. For instance, a musculoskeletal system is a mechanical dynamic system. The spindle sensors in the muscles would map muscle contraction and its speed to electrical signals that would be fed to the neural networks in the central nervous system (CNS) for motor control. The interface between muscle contraction and electrical signals is a function $g_i(\cdot)$ whose shape determines how the CNS reads muscle activity. In biological systems, this shape of $g_i(\cdot)$ can be tuned using muscle contraction itself because sensors such as spindles are part of the muscles.

The ghost circuit as a whole is stable if each sub-system is stable, or if the system of systems operates in a contraction region [7] in the grand state-space $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_{i-1}^T, \mathbf{x}_{i+1}^T, \dots, \mathbf{x}_N^T]^T$. The ghost circuit is useful if $\forall i \in \mathfrak{N}, \forall t > 0, \mathbf{x}_i(t) \in \psi_i(t)$.

There is evolutionary pressure on each sub-system of such a ghost circuit to be stable and useful. It can do so not only by tuning its system parameters $\boldsymbol{\xi}_i$ but also by minimising the dependency of its inputs on states of other sub-systems. This puts evolutionary pressure on W to be as sparse as possible, i.e. the dynamics of a sub-system disturb a minimum number of other sub-systems. For instance, in the case of the dead trout swimming against a stream, the passive dynamics of the fish can be given by $\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \boldsymbol{\xi}_1, \mathbf{u}_1, t)$, where \mathbf{x}_1 is the direction of movement in 3D space, $\boldsymbol{\xi}_1$ defines the kinematic and dynamic parameters such as the body shape, stiffness, and mass distribution. $\dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2, \boldsymbol{\xi}_2, \mathbf{u}_2, t)$ is the system describing the

hydrodynamics in the neighborhood of the fish, where \mathbf{x}_2 is the speed of water flow in 3D space, $\boldsymbol{\xi}_1$ defines the hydrodynamic parameters such as pressure, density, and viscosity of the water. Given a suitable $g_1(\cdot)$ and $g_2(\cdot)$, the two systems can form a coupled nonlinear dynamic system such that $\forall t > 0$, $\mathbf{x}_1(t) \in \psi_1(t)$, where $\psi_1(t)$ points upstream.

All living species show adaptations to suit environmental conditions. Since the environment also holds potential to change, living beings keep diversity within the same species while making gradual changes in physical parameters. Since we know that physical parameters such as the shape of organs (e.g. shape of the cochlea, digits in a hoof), and shape of sliding surfaces of joints (e.g. the knee joint) determine kinematics and dynamics of interaction, we can view gradual physical adaptations as a process of tuning to improve the efficiency of ghost circuits for real-time dynamic problem solving.

3.5.2 Kinematic Tuning

The biological motor system comprises a complex network of neural pathways within the CNS that orchestrates the translation of sensory inputs into neural instructions, subsequently directing muscles to execute movements. In humans, this intricate system encompasses pivotal components, including the spinal cord, cerebellum, pre-motor cortex, and primary motor cortex. Nevertheless, it is important to note that the organisation of these neural structures may differ among different species. The directives issued by the motor system to induce muscle contractions are referred to as motor commands.

Motor commands in biological systems operate at two levels: 1) motor commands that can tune the body (MC_T), such as setting the right stiffness and limb configurations for a given task, and 2) motor commands that can directly control movements and interaction forces (MC_M).

For instance, a cyclist would continuously tune the impedance of their bent knee joints (MC_T type) while riding (MC_M type) on bumpy terrain. Another example is soft tissue palpation where participants would change the stiffness of their fingers (MC_T type) while regulating their speed and palpation force (MC_M type). The efficiency of MC_M type commands depends on the how well MC_T type motor commands are regulated.

A general nonlinear dynamical system given by Eq. 3.8 can be re-written in terms of MC_T and MC_M type commands given by

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i, \boldsymbol{\xi}_i(\mathbf{u}_{MC_T}), \mathbf{u}_{MC_M}, t), \quad i = 1, 2, 3, \dots, N \quad (3.12)$$

where x , \mathbf{u}_{MC_T} , \mathbf{u}_{MC_M} , and t are the state vector, MC_T -type control commands, MC_M -type control commands, and time, respectively.

3.5.2.1 Insights from Rigid Body Dynamics

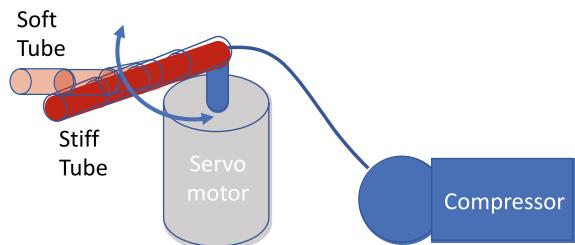
Rigid body dynamics are given by $\tau = H(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + J^T f_{ext}$, where τ is the joint torque vector, θ is the joint angle vector, f_{ext} is the external force vector applied at the end effector, J is the Jacobian matrix, $H(\theta) = \sum_{j=1}^N (I_j + m_j J_j^T J_j)$, I_j is the moment of inertia matrix for the j-th joint, m_j is the mass of the j-th link, J_j is the Jacobian matrix at the j-th joint, $G(\theta)$ is the torque vector due to gravity and other potential energy sources such as springs, $V(\theta, \dot{\theta}) = H(\theta)\dot{\theta} + D(\theta)\dot{\theta} - d(\theta^T H(\theta)\theta)/d\theta$, and $D(\theta)$ is the angle-dependent damping matrix. It is important to notice that the elements in H , V , G , J , and D are functions of the joint angle vector θ . MC_T type control commands denoted by u_{MC_T} can be used to tune the joint stiffness influencing $G(\theta)$ in addition to controlling the posture vector θ . The remaining terms in the torque vector are MC_M -type action denoted by u_{MC_M} to control interaction forces and joint speed. Since $\dot{\theta} = H(\theta)^{-1} [\tau - V(\theta, \dot{\theta}) - G(\theta) - J^T f_{ext}]$, by defining $x = [\theta, \dot{\theta}]^T$, we can write the dynamic equation in the form of an ordinary differential equation given by $\dot{\theta} = f(x, \xi(u_{MC_T}), u_{MC_M}, t)$, where x , u_{MC_T} , u_{MC_M} , and t are the state vector, MC_T -type control commands, MC_M -type control commands, and time, respectively.

3.5.2.2 Insights from Soft Body Systems

Imagine there is a hollow rubber tube with one closed end as shown in Fig. 3.7. The tube can be stiffened by pumping air into it through the open end. Otherwise, it can be bent easily. When one end of the tube is attached to the shaft of a servo motor, the tube can swing back and forth. However, the path of the tube's free end depends on the tube's stiffness. If it is very stiff, the path will be circular. When the tube is soft, the path depends on the acceleration and speed profile of the motor shaft. Though the differences of motion for a tube can be easy to imagine, it becomes more complex when several controllable stiffness members (like beams) are connected to make a more complex mechanism.

Therefore, even if the resting state kinematics (i.e. no motion) of a mechanism is the same, the dynamics and kinematics during motion depend on the stiffness of the members of mechanisms. Previously, we discussed that kinematics determine

Fig. 3.7 Controllable stiffness tube waving



how forces move objects. Now we see that controllable stiffness mechanisms have a more complex map from forces to movements conditioned by the stiffness of its members. Through careful observation, we notice that stiffness decides how external forces and inertial forces cause structural changes. Structural changes such as bending, compression, or stretching cause changes in kinematics resulting in changes in the way forces cause movements (please see Sect. 3.3 for details). When there is no movement or when movements are quasi-static (i.e. very low acceleration), only the open loop effect of forces causing structural changes is present. But when there are notable accelerations in members of a mechanism, the inertial forces themselves can cause structural changes augmenting the map from external forces to subsequent movements. Therefore, the stiffness of a mechanism plays a mediating role to establish feedback loops between forces and movements.

3.5.2.3 Kinematic Tuning to Regulate Ghost Circuits

We mentioned in the discussion surrounding ghost circuits that kinematic parameters such as stiffness and dimensions of agents in a network of systems play an important role in network dynamics. Therefore, it is important to note that kinematic tuning (using MC_T) is an important process to influence the emergent dynamics in the network. This phenomenon rests in the core of ghost circuits.

In the school of fish shown in Fig. 3.6, the careful tuning in the geometric relationships among members of the school of fish can also be seen as kinematic tuning in a broad sense. This fine-tuning of geometric relationships can be viewed as a form of kinematic adjustment in a broader context. These adjustments extend beyond just regulating individual fish's body stiffness; they also encompass the careful positioning of each fish in relation to its peers. This collective effort aims to optimize the utilization of passing vortices, ultimately giving rise to the emergence of an efficient ghost circuit that is shared by the entire school of fish.

3.5.2.4 Kinematic Tuning to Improve Perception

First let us review the physical meaning of eigenvalues and eigenvectors.

Figure 3.8 shows the graphical view of multiplying a vector v_0 by a (2×2) matrix called A to obtain v_1 . Essentially two things happen in this process. The vector v_1 has scaled and rotated with respect to v_0 . This scaling and rotating can also be seen as only scaling of the projected components of v_0 on to the eigenvectors of A labeled e_1 and e_2 by the corresponding eigenvalues. In other words $v_1^{e_1} = \lambda_1 v_0^{e_1}$ and $v_1^{e_2} = \lambda_2 v_0^{e_2}$, where λ_1 and λ_2 are eigenvalues corresponding to eigenvectors e_1 and e_2 , and $v_0^{e_1}$ and $v_0^{e_2}$ are projections of v_0 onto e_1 and e_2 respectively.

Our recent findings on soft tissue palpation show that in tasks such as estimating the depth of a hard nodule in a soft tissue accompany two phenomena of motor action in participants. One is a process of muscle co-contraction with statistical patterns unique to the depth of the nodule which is also verified by independent

$$v_1 = A_{(2 \times 2)} v_0$$

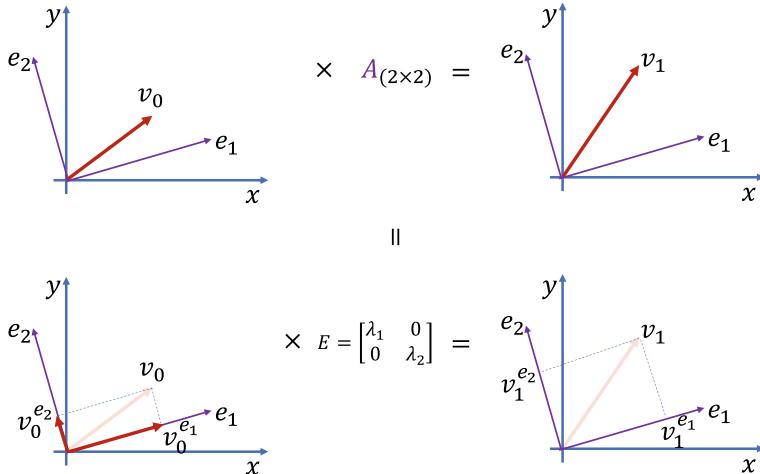


Fig. 3.8 The meaning of eigenvalues and eigenvectors in matrix multiplication

robotic experiments. Our experiments using human participants showed that the antagonistic muscles in the upper arm experienced co-contraction variations when they palpated a soft tissue to classify their perception of the depth of a hard nodule in to one among four previously learnt categories as shown in Fig. 3.9.

When we plotted the level of muscle co-contraction as a state transition probability matrix (STPM) as shown in Fig. 3.9c, we noticed that the second largest eigenvalue of the STPM increased with the depth of the nodule [8] as shown in Table 3.5.2.4.

Nodule depth [mm]		Eigenvalues				n
2	0.0006	0.2945	0.3936	0.5984	1	5.83
4	0.1393	0.2772	0.4069	0.6628	1	7.28
8	0.1958	0.3468	0.4270	0.6992	1	8.37

Now let us try to understand this using some mathematical formulations. We know that all eigenvalues in an STPM are less than or equal to 1. Mathematically, this means that the projection of the initial muscle co-contraction level on each eigenvector will decay at the rate of the corresponding eigenvalue (please see Sect. 3.3 for an explanation of basics) as shown in Fig. 3.10.

Those with small eigenvalues will decay faster than the others. The projection with eigenvalue 1 will not decay. The projection with the smallest eigenvalue will decay fastest. Since the projection on the eigenvector with the second largest eigenvalue will be the slowest to decay, the convergence time to the absorbing state (the projection with eigenvalue 1) will be largely determined by the second largest eigenvalue. We use this property to estimate the time it would take for the haptic exploration to

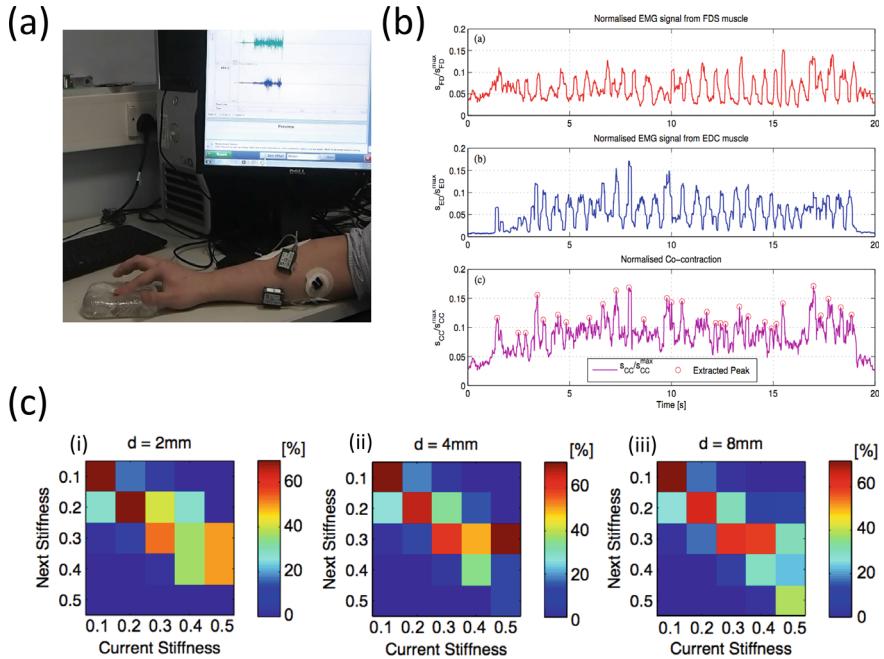


Fig. 3.9 Muscle co-contraction during palpation to estimate the depth of a nodule in a soft tissue. **a** The experimental condition with two electromyography (EMG) sensors placed on antagonistic muscles on the lower arm, **b** raw data of muscle EMG and co-contraction pattern, and **c** the state transition probability matrices for nodules buried at different depths

converge, or in other words, the time to make a decision. The size of the projection on the i -th eigenvector after n -iterations is $s_n = s_0 \lambda_i^n$. Then we can find the time it takes for the ratio of the projection to reach a threshold of $\frac{s_n}{s_0} \leq \epsilon$ after n -iterations using $n \log |\lambda_i| = \log |\epsilon|$. Then for the second largest eigenvalue, $n = \log |\epsilon| / \log |\lambda_2|$. This means, for a given threshold of $\epsilon < 1$, a stiffness transition probability matrix with the second largest eigenvalue close to 1 will take a long time to reach a decision.

This prediction matched our experimental observations, where we observed stiffness variation having larger λ_2 for nodules buried deeper in the soft tissue. Therefore, internal tuning of the joint stiffness using MC_T type commands is an important part of haptic information gain.

3.5.2.5 Kinematic Tuning to Exploit Passive Dynamics

Work on rodents and cephalopods show that the biological control architectures are hierarchical rather than centralized [9] where local control loops have autonomy to take certain reflex actions that do not require model-based long loop control. This architecture allows physical dynamics of the body to solve certain computational

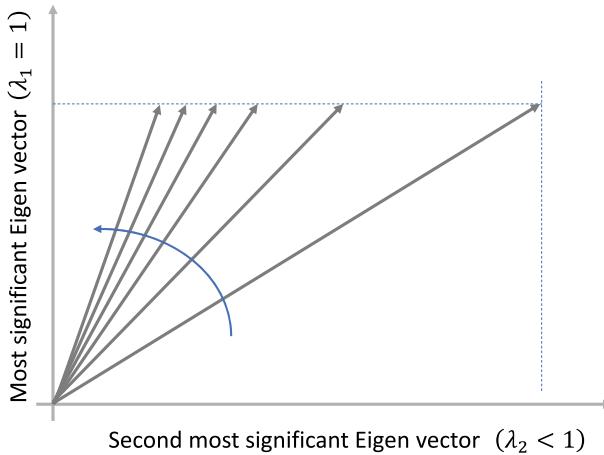


Fig. 3.10 How a vector in the state space rotates towards the absorbing state across iterations

problems autonomously, and the complex sensor-based controllers can then focus on low entropy signals for model-based control for overall coordination.

We notice in biological counterparts that they have a remarkable ability to use passive dynamics of the body to control movements whenever possible. The aforementioned reaching movements in the octopus is one example. This also means that fixed sampling intervals are not needed for control. In the case of the octopus, it checks the accuracy of its throw only after the tentacle reaches its prey.

3.5.2.6 Attempt for Kinematic Tuning Leading to Adaptation Across Physical and Numerical Systems

Adaptations spanning neural and physical structures are important for real-time action to maximise rewards. Take for example the case of the dead trout swimming against a stream of turbulent water. First the kinetic energy of the turbulent water exerts forces that cause the soft body to bend passively. Different stiffness levels along the body will render different levels of deformations along the body. Such deformations change the kinematic relations between external forces and bending patterns. If the bending patterns help to augment the force fields in water in terms of vortices helpful to push the body against the water stream, the fish body can passively establish a “ghost circuit” for force fields that develop to render meaningful work.

The emergence of bodily formations encompassing the body and the environment is closely linked to the idea of ghost circuits because these bodily formations emerge as a real-time attempt to tune the body and environment to solve the problem of reducing the uncertainty of task relevant states and are hence in a state of flux. These mechanical circuits emerge during interaction and dissolve thereafter. They have a ghostly nature of appearance and dissolution. This makes it very difficult to spot

them in anatomical analysis of a living being separated from its environment. Of particular interest, the possibility that variation of meta parameters of these ghost circuits and the resulting rewards may hold clues as to how they would adapt and evolve in a given environment.

3.5.3 Behavioral Lensing to Sharpen Perception

When asked to estimate the weight of an object, one would bob it up and down gently before coming to a conclusion. We regulate speed and force applied at fingers during soft tissue examination.

If we refer back to Fig. 3.4, imagine that one of those springs is a finger and that the rest of the spring network is the soft tissue being examined. The spring representing the finger can attempt to find the best way of oscillating the rest of the network, i.e., to elicit states within the finger spring to best identify the nonlinear dynamic parameters of the rest of the network. We introduce the term behavioral lensing to describe this process of finding the best perturbation pattern to identify a nonlinear system being physically examined. It leads to an enhanced perception of task relevant states with lower uncertainty leading to higher rewards.

3.6 A Framework to Design Robots for Embodied Intelligence

In the above discussion, we mentioned two types of action which an agent can take to interact meaningfully with an environment. There is one class of control commands denoted \mathbf{u}_{MC_T} to tune the body and the other class denoted \mathbf{u}_{MC_M} to control movements and interaction forces. Since MC_T type commands try to tune the shape, stiffness, or damping of various degrees of freedom in the robot, their statistical summary given other information (such as concurrent MC_M type commands given states) and resulting rewards from value functions can be used to derive useful guides to re-design the robot. Therefore, the process can start with an initial design of a robot that can improve its embodied intelligence through an iterative process.

Let $\pi_1(\mathbf{u}_{MC_T} | s)$ be the policy that maps a vector of states s to a vector of MC_T type control commands, and let $\pi_2(\mathbf{u}_{MC_M} | s)$ be that for MC_M type control commands. Let $V(\mathbf{u}_{MC_T}, \mathbf{u}_{MC_M}, s)$ be a value function that estimates the total discounted sum of rewards that the robot can earn by following π_1 and π_2 given a state s at time t . The interplay among these vectors of information over a sufficient length of time can provide useful information to iterate the hardware design of a robot as shown in Fig. 3.11 which is based off the design of a deformable mobile robot presented in [10].

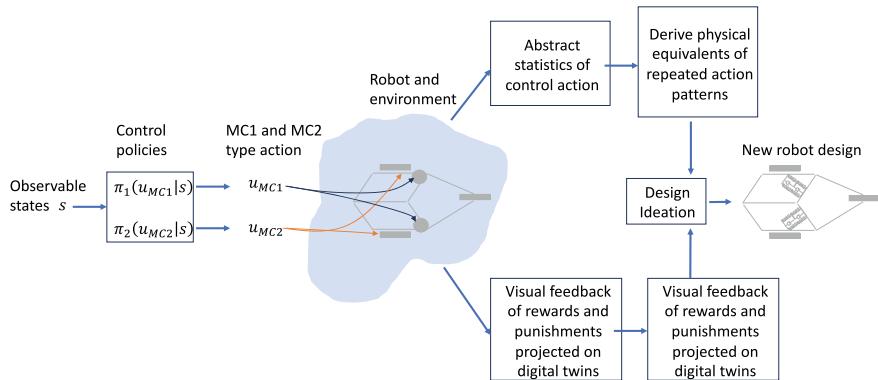


Fig. 3.11 A general framework for design ideation to improve physical hardware designs of robots based on their interaction with the environment

Here we propose two broad approaches. Both approaches require an initial robot design based on basic information about the working environment. The actual deployment would then help to test the initial hypotheses of the design process based on measurements. However, when dealing with personal robots deployed in private spaces, a significant limitation arises: restrictions on data transfer to the robot designers. This restriction hinders the reliability of the iterative design process.

To overcome the limitations of data transfer in private environments, a more generalizable approach is warranted. One possible solution is to utilize statistical analyses of control actions. By examining patterns and trends in the robot's actions within different environments, designers can extract valuable information and derive insights that inform the hardware design of the robot. This statistical approach enables the clustering of various user environments, enabling the creation of robot designs that cater to distinct classes of environments.

Another promising avenue is the use of digital twins for the personal robot. A digital twin is a virtual replica of the physical robot that mimics its behavior and interaction with the environment. By leveraging digital twins, designers can simulate the robot's actions and observe the resulting rewards and punishments from the working environment. This virtual testing environment provides a wealth of data without relying on the transfer of sensitive private information. It also allows for the derivation of abstract design guidelines, empowering designers to innovate based on grounded facts and observations.

The robot designers could take one or both streams of information as inputs for design ideation to iterate the hardware design of the robot to suit different classes of environments. This approach, for instance, will provide information to cluster different user environments. The other advantage is to be able to derive abstract design guidelines without needing to know details of sensitive private data. Moreover, this approach stimulates the creativity of the design teams based on grounded facts.

3.6.1 Action Space Statistics Driven Approach

After a robot interacts with a given environment over a sufficient length of time, useful insights can be found in regions of state and action space where $V(\mathbf{u}_{MC_T}, \mathbf{u}_{MC_M}, s) \geq \beta$, where $\beta > 0$ is a value threshold. Different methods can be developed to understand the hardware equivalents of actions for high reward. For instance, the statistics of \mathbf{u}_{MC_T} for such high reward regions would show the distribution of stiffness variation of a particular joint. This distribution can then be approximated by a number of primitive distributions that can be implemented in an assembly of variable stiffness hardware components as shown in Fig. 3.12.

3.6.2 Digital Twin Driven Approach

Design involves creativity based on interpretation of evidence [11]. Often the challenge is to represent evidence from multiple sources so that the human brain can identify useful patterns on which to base innovative ideas. Work completed on visualisation of dependencies among keywords has shown potential for cross-pollination across different scientific domains [12]. Recent findings show that heuristic prototypes—such as bio-inspiration and examples in other domains that contain useful clues—significantly improve capacity to innovate solutions to new problems [13]. These works highlight the open challenge to present facts to stimulate innovative ideas. Representing a large number of variables in a graphical format is challenging. Taking a step beyond static interactive graphs, we hypothesise that videos showing concurrent changes in a digital twin of the robot’s morphological feature will allow human participants to extract key patterns for design innovation.

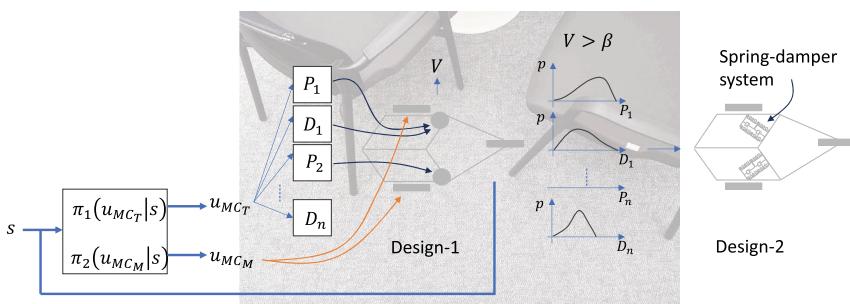


Fig. 3.12 A design iteration framework to delegate control responsibility to hardware. The \mathbf{u}_{MC_T} commands to tune the proportional (P) and derivative (D) gains of certain joints can provide useful statistical distributions of their required impedance in a given environment. This information can then be used to design spring-damper networks to provide the same functionality the motors provide with variable gains

For instance, a user may display gradients of $V(\mathbf{u}_{MC_T}, \mathbf{u}_{MC_M}, s) \geq \beta$ on \mathbf{u}_{MC_T} and s overlaid on the digital twin of the robot. In the case of the robot shown in Fig. 3.13, color shading at the joints can represent their stiffness under \mathbf{u}_{MC_T} commands. A graphical format can be used such as a circle with radius proportional to the gradients of V against \mathbf{u}_{MC_T} . Since the robot designer can see the speed of the robot due to MC_M type commands and the response of joints due to changes in MC_T type commands, the graphical sequences will help to build memory primitives that will, over time, help the designer to build new hypotheses of design improvements to be tested in an iterative manner.

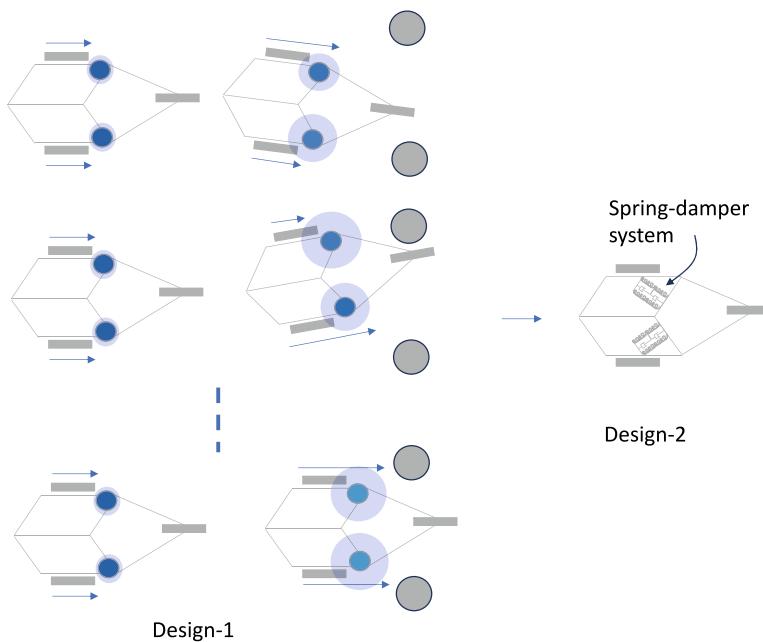


Fig. 3.13 An alternative design iteration framework to delegate control responsibility to hardware. The \mathbf{u}_{MC_T} commands to tune the mechanical impedance of certain joints are visualised using colors (with dark colors being very stiff and light colors being very soft). The gradient of $V(\mathbf{u}_{MC_T}, \mathbf{u}_{MC_M}, s) \geq \beta$ against \mathbf{u}_{MC_T} is proportional to the radius of a transparent blue circle projected onto each joint. The memory consolidation process of such graphical information can then lead to design spring-damper networks to provide the same functionality given by the motors with variable gains

3.7 A Review of Examples of Using Soft Robots to Test Hypotheses on EI

3.7.1 Bio-Inspired Deformable Mobile Robot

Many animals are adept at conforming their natural body shapes to negotiate geometric constraints in their environments. Over thousands of years, they have evolved to be able to tune their body parameters to adapt to their surroundings. Animals such as cats, rats, cockroaches, and octopuses can squeeze through apertures smaller than their natural body shapes. This ability to fit through gaps in cluttered settings is a key adaptation to simplify their environmental interactions [14]. As shown in Fig. 3.14, cats can readjust their body dimensions by adapting their flexible collarbones, shoulders, and spines which assist with flexion and torsion [15, 16].

Embodied artificial intelligence proposes that robot bodies and brains are co-developed, similar to the evolution of animals [17]. Taking this approach could enable soft robots to represent bio-inspired artificial intelligence that would not be possible with rigid robotics [18, 19]. Properties such as adaptivity, robustness, versatility, and agility—which are desirable characteristics of biological organisms—can strongly benefit the design of autonomous robots [2]. Soft robots can achieve tasks—such as squeezing, stretching, climbing, growing, and morphing—which would not be possible with rigid robotic approaches [20].

Shape-changing robots have emerged due to technology developments such as deformable soft robotics. Origami robots, which use folding techniques, offer possibilities for achieving different robot morphologies [21]. The ability of robots to change shape affords opportunities for functionality enhancement and expansion through adaptation. Grand challenges to achieve this vision include enabling shape sensing, automating shape-changing, and integrating functional materials into systems [22]. Addressing these challenges can enable robots to adapt their morphologies to their environments by taking actions such as growing, evolving, self-healing, developing, and biodegrading [23, 24]. Morphological computation and adaptation offer several advantages for robots operating in different environments. Traversal, rather than circumnavigation, of obstacles is more efficient (i.e. less travel time and lower energy expended) and could even be necessary for mission success. Robots



Fig. 3.14 The ability of animals—such as cats—to morph their shape in order to traverse spaces smaller than their natural body states is the bio-inspiration for the design of DeforMoBot, a deformable mobile robot

which can rearrange their natural shape, and thus change their centre of mass, obtain intrinsic robustness and increased ability to overcome obstacles. It can also assist them in finding compensatory behaviors in response to damage or injury, at which animals are adept [25].

DeforMoBot is a bio-inspired deformable mobile robot which uses embodied intelligence to traverse obstacles in unstructured environments [10]. The design of the robot body detailing sensors and components is shown in Fig. 3.15. The robot has a hexagonal shape composed of acrylic sheets with connected hinges acting as the joints of the body. Three additional links within the outer shell form two parallelograms and a kite within the hexagon. The “spine” of the robot is characterised by a linear guide rail (attached to the middle acrylic sheet), while spring-loaded 3D-printed “whiskers” are placed at its front tip. Umbrella-inspired mechanisms are used at the whiskers and at the front corners of the main robot body to ensure symmetrical movement of both during shape adjustment. The shape of the robot body attempts to mimic the transversal contraction of the cat by being able to change between a regular hexagonal shape to an elongated rhombus shape.

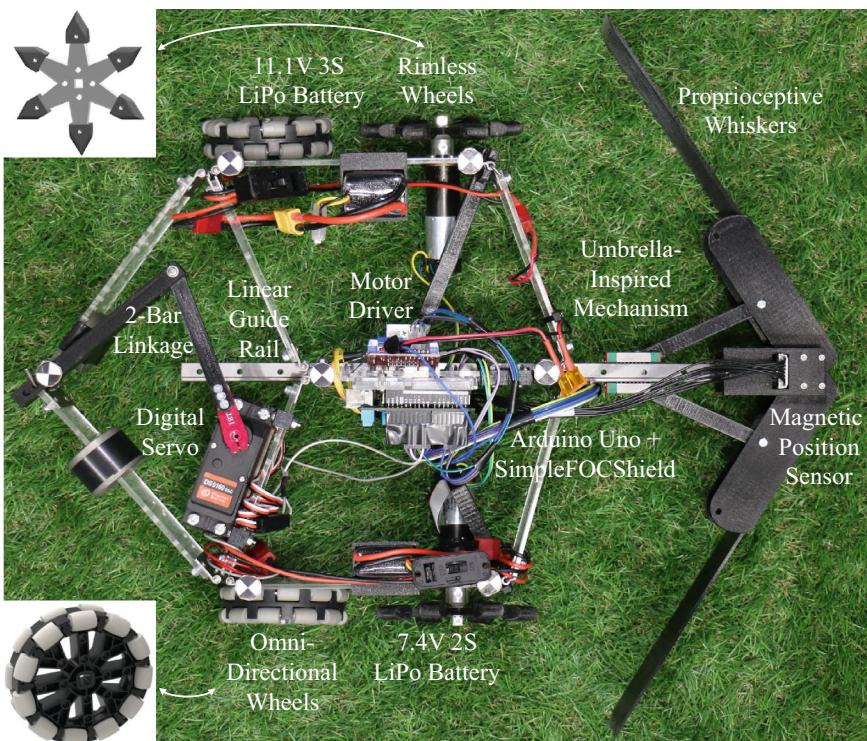


Fig. 3.15 Design of the bio-inspired deformable mobile robot DeforMoBot, detailing sensors and components

The whiskers can rotate around a fixed axis where the orientation of a neodymium magnet is measured in real time by a magnetic encoder. As shown inset of Fig. 3.15, motor-driven 3D-printed rimless wheels at the front of the robot provide traction and grip on unstructured terrain, while passive omni-directional wheels at the back assist with the robot's shape-changing ability. Depending on the given task or goal, other wheels can also be employed, such as the PaTS-Wheel, a passively-transformable single-part wheel that can render hooks when presented with obstacles or uneven terrain [23]. The wheels and digital servo—controlling the robot's body shape via a 2-bar linkage—are powered by Lithium Polymer (LiPo) batteries.

The robot aims to traverse obstacles in its path while keeping a wide body shape for stability. Given these objectives, we employ a Position to Shape Mapper (PSM) modelling the relationships between the change in the servo angle $\Delta\Phi$ and the change in the whisker angle $\Delta\Omega$ (in blue, left y-axis) and between the change in the servo angle $\Delta\Phi$ and the 2-bar linkage angle θ as quadratic polynomials, for example $\Delta\Phi = a\Delta\Omega^2 + b\Delta\Omega + c$ where a , b , and c are tuned coefficients to couple the whisker and desired robot body shape. We propose an algorithm which the deformable mobile robot can employ to efficiently progress through obstacles, taking account of both the real-time whisker angle deformation and the current shape of the robot body. The setup is controllable, meaning that different algorithms can be deployed and tested.

To demonstrate and test the robot's shape-changing ability, we conduct various experiments to observe how it interacts with obstacles in its path. The robot obtains the whisker angle measurements from the magnetic encoder and applies the PSM to change shape, adhering to the algorithm previously detailed. The servo angle is reactive in real-time to data measurements of the whisker angle apart from a delay when the robot is widening its shape so that this happens only when the body has mostly or fully passed through the obstacles. Visual markers are placed at different points on the robot to precisely follow its movements and changing body shape through the scene so that we can directly compare the transient responses of the robot's body shape and the progression of its path as it traverses different obstacles.



Fig. 3.16 DeforMoBot, a bio-inspired deformable mobile robot, changes its shape depending on its surroundings to squeeze through gaps and traverse obstacles in its path

As shown in Fig. 3.16, the robot is tested navigating a track of immovable stone blocks, the width of which varies from wide to narrow. The robot continuously updates its width to efficiently traverse the obstacle course. We previously conducted experiments to observe how the robot interacts with obstacles with different parameters. In [10], we used boxes, cushions, clothes, and stone blocks as the obstacles since this covers a range of different physical properties including mass, shape, texture, rigidity/compliance, and hardness/softness. We tested the robot across various approach angles including 90° (directly facing the obstacles), 60°, and 30°. Based on the compliance of objects, the robot can push them away (in cases of lighter obstacles), narrow its body to squeeze through the gap (in cases of heavier obstacles), or use a combination of these actions.

After numerous design iterations to our robot, we learnt several lessons from the process which might prove useful to others interested in robot design and construction.

First, the design of a robust yet agile robot body is highly desired and will result in better performance. The ability and control of the shape changing nature of our robot was difficult to achieve as we wanted a data-efficient yet strong solution. We attempted interaction with many different joints and linkages on the robot, using different types of servos and gimbal motors—as well as trying varying pulley and linkage systems—before reaching our solution. Significant design iterations such as using a strong servo with a 2-bar linkage and employing an umbrella-inspired symmetrical design led to more straightforward modelling and improved performance.

Secondly, it is important to calculate an accurate model from the proprioceptive feedback to the body. Overcompensation by the body could be inefficient, while undercompensation could lead to undesired problems such as mission failure. In accordance with this, measurements of the body and whiskers should be considered to achieve the desired goal; originally, we employed short whiskers and noticed that the wheels sometimes became stuck on the obstacles. Making the whiskers longer not only improved the robot’s proprioceptive reach, but it also enabled the whiskers to protect the wheels and body when navigating obstacles, like how a bird might use its wings to cover its body in flight or navigating small apertures.

Finally, the interaction between the robot and its terrain should be considered. When the robot was navigating in structured indoor environments, four omnidirectional wheels were used to ensure smooth navigation and to assist with shape change. However, when testing the robot in more unstructured settings such as grasslands, the wheel motors proved too weak for the robot to drive well, and the omnidirectional wheels could get clogged with debris and become stuck. To solve these problems, we used higher-torque motors and used a combination of motor-driven rimless wheels at the front (to provide good grip and traction) and passive omnidirectional wheels at the back (to assist with the shape-changing ability). We are also interested in exploring the benefits of additional sensing modalities for predictive and adaptive navigation in unstructured environments [24].

Our experimental results highlight the importance of kinematic tuning, ghost circuits, and behavioral lensing.

Kinematic tuning is clear since the proprioceptive whiskers and shape adjustment algorithm are tuned to the physical capabilities, coupling perception and embodiment through MC_T commands to change the robot's shape. When the robot can push away lighter obstacles, it only slightly adjusts its body shape and maintains its stability. For more moderately-weighted obstructions which the robot cannot fully move, it still tries to persevere and adjusts its body shape accordingly. When the robot cannot move or push the obstacles at all, it completely relies on adjusting its configuration until it has successfully traversed the obstacles at which time it can resume its natural body shape. Since the robot tries to navigate between obstacles, it can be upheld by the objects themselves. When narrowing its shape, the robot becomes longer which could affect its turning capabilities in constrained surroundings. However, in doing so it reduces its turning circle and can better jostle through challenging apertures. The long whiskers help the robot to turn and protect the robot's body.

In addition, DeforMoBot provides examples of ghost circuits. When the robot body meets obstacles, the system is coupled with the obstacle's mechanical dynamics. This allows the robot to take advantage of mechanical support from the obstacle to stabilise itself in addition to perceiving the obstacle dynamics.

DeforMoBot could also use behavioral lensing by regulating speed and force on the obstacle—using MC_M commands—in order to better feel the obstacle properties.

These types of improved capabilities contribute to robots having better perception and proficiency especially in unstructured and challenging environments where they can navigate more efficiently and effectively. The marginal gains of increasing the number and accuracy of sensing modalities and degrees of freedom should be explored to improve predictive and adaptive navigation of robots.

3.7.2 Bio-Inspired Angle-Dependent Damping to Stabilise Knee Movement During Locomotion

As discussed previously, when travelling over rough terrain, a cyclist will lift themselves off their seat and ride with bent knees. Tuning the angle-dependent impedance of their knee joints allows them to filter out such disturbances. Understanding the computational role of the compliant knee joint in simplifying complex dynamic interactions of legged robots with their surroundings can lead to significant advancements in hybrid active-passive based control approaches for designing efficient walking robots. According to the findings of an in-vivo study of the human knee joint, muscle contraction significantly increases the stiffness and damping properties of the knee joint, while limb inertia remains the same [26]. Because of this, it can be deduced that the internal impedance of the knee joints varies depending on the angle.

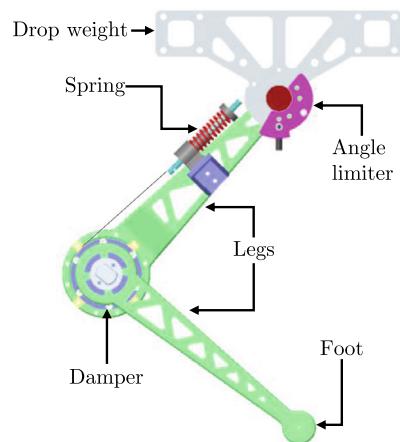
It is possible to provide an effective morphological solution to filter perturbations by installing an angle-dependent damper that has been carefully designed and constructed. We present a novel design for a passive knee joint that incorporates angle-dependent mechanical damping into its structure. The occurrence of metastable

variations in states can be reduced with the help of this design, which makes use of internal viscous damping within the knee joint of a compliant leg structure. The purpose of this design is to improve the knee joint's stability and reliability by reducing the peak ground reaction force as well as the variability with which it is associated.

Figure 3.17 shows a simplified robotic leg including the damper, spring, leg segments, and drop weight. The damper is fitted to the joint of the leg along with a parallel spring which resembles the human knee joint. Drop tests were carried out to provide empirical evidence of the usefulness of variable damping. During the drop test, the leg fitted with the damper and spring is allowed to freefall from a predetermined height. When the leg hits the drop platform it compresses, and the spring damper system absorbs energy from the impact. Both events happen simultaneously. The process of absorbing energy causes a decrease in the velocity of the system, which ultimately results in a state of equilibrium with a constant deflection.

As shown in the Fig. 3.18, the test setup with angle-dependent damper shows a low deflection and settles faster compared to the constant dampers. Initially, during the drop, the damping coefficient is low. Which allows the leg to compress and absorb the energy. During the compression, the damping coefficient increases, and leg compression reduces results in smaller variation. The MC_T type command corresponds to the adjustment of the amount of damping to settle the movement faster, while the MC_M type command corresponds to the dropping of the leg from a height which is similar to the locomotion of a legged animal.

Fig. 3.17 CAD Model:
simplified robotic leg
including the damper, spring,
leg segments and drop
weight



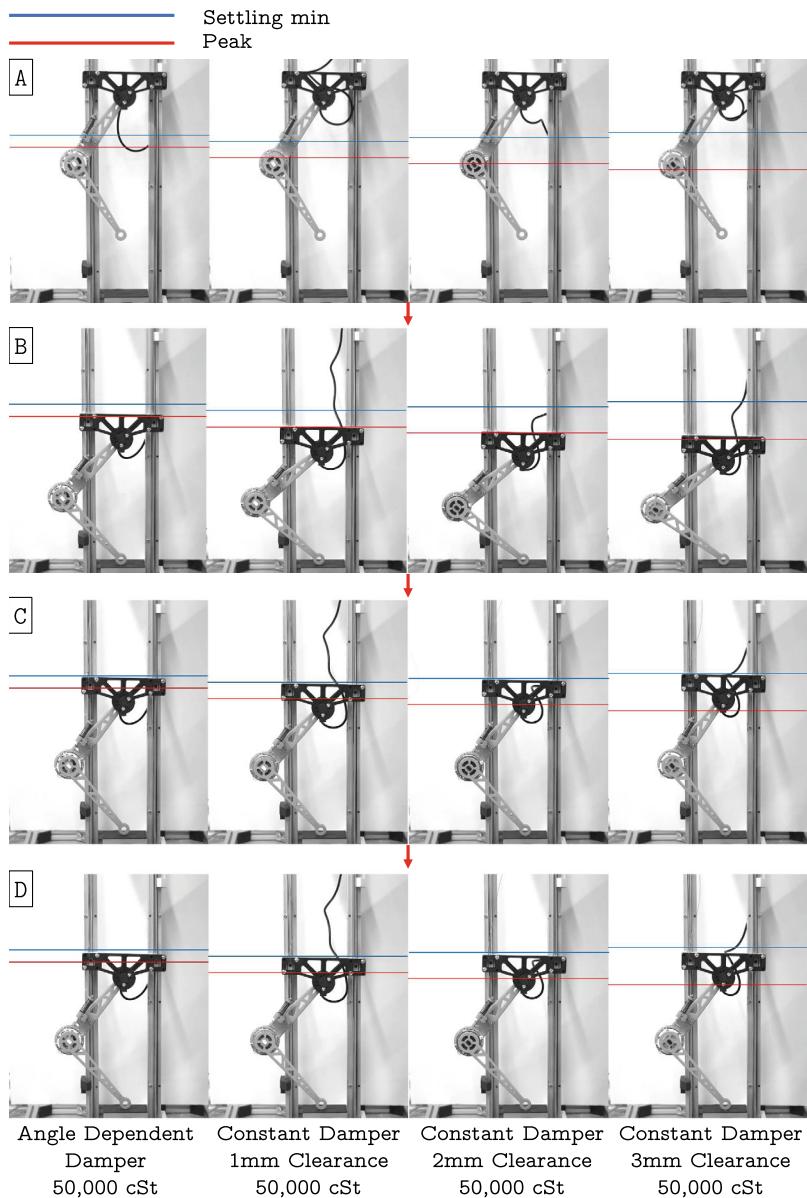


Fig. 3.18 Snapshots of a drop test for three constant dampers and an angle-dependent variable damper (A) $t = 0$, (B) $t = \text{maximum compression of leg}$, (C) $t = \text{settling minimum of the leg}$, (D) $t = \infty$ (settled position). The angle-dependent variable damper settles with a smaller variation

3.7.3 *Morphological Adaptation as an Optimum Responsibility Sharing Process*

Short-term adaptation and long-term evolution occurs across the entire plasticity range in the body starting from neural network topologies to sliding surface profiles at joints. Structures with high plasticity tend to take responsibility for real-time modeling such as tracking the dynamic model of an object being manipulated. Those structures with lower plasticity tend to take the responsibility of modeling baseline dynamics shared across many experiences over generations of a species. Then the question arises as to whether real-time attempts to model the dynamics of experiences provide information for long term adaptations.

It is difficult to test this question in real biological species due to the timescales involved. In a limited controlled experiment, we asked the question as to whether the history of feedback control action on a given degree of freedom of a physical dynamic system could carry information for physical adaptations that would reduce the burden of the original closed loop controller.

We took an inverted pendulum to represent the torso of a biped walker and used a closed-loop regulator-type controller to generate forces MC_M on the base that slides on a rail [27]. A backdrivable servo motor at the pivot joint of the pendulum was first set to a passive mode of monitoring the action of the closed loop controller. Other than this observation in the state space, the motor at the pivot joint was not informed of any other details of the implementation of the control action. Then the backdrivable motor found a pattern of state trajectories in the phase portrait (joint angle and its velocity space) that showed a spiral shape of convergence towards the origin whenever a disturbance was given to the pendulum.

Data collected across repeated disturbances of different magnitudes could help this motor to uncover the spiral pattern common to all kinds of disturbances. Then, a comparison with a bank of physical phenomena that could be implemented by the backdrivable motor through its impedance parameters revealed that a basic pattern match can be made by growing a visco-elastic property at the pivot joint MC_T . Then a simple search for the best visco-elastic parameters that minimises the overall cost of balancing the pendulum led to a settlement where a part of the responsibility of balancing the inverted pendulum was taken by the visco-elastic property implemented by the backdrivable motor at the pivot joint, and the rest of the responsibility was taken by the closed loop controller. This is reminiscent of how the closed loop model-based controllers in the CNS and the ligaments in the hip joint would co-evolve in a biped walker without a central arbitration.

The long term attempts of the model based controllers in the CNS can function as an environmental feedback for the visco-elastic structures at the hip joint to evolve with the common aim of reducing the overall effort taken for upright biped walking. If we extrapolate this phenomenon, the long-term action of the visco-elastic ligaments at the hip joint can provide consistent patterns of stresses on bone structures. This can be environmental feedback for the bones to grow or deplete material leading to

shape and strength adaptations in low plastic structures such as bones in the long run.

Therefore, a chain of local information relayed from one structure to another can lead to an overall adaptation to share responsibility of control across different neural and mechanical structures. It is important to notice here that each local sub-system tries to improve its own contribution using locally available resources. The closed loop controllers may tune feedback gains in the controller and the observer (synaptic plasticity in the CNS of a biological counterpart). The motor at the pivot joint contributes by tuning its visco-elastic properties implemented using proportional and derivative gains of the servo controller in this case. If it were to be a local 3D printer, then it would use material available locally to build a visco-elastic structure (ligaments and antagonistic muscles in a biological counterpart). The bones on the other hand could grow more material and change the lattice structures within the bone to efficiently distribute the stresses it experiences. Therefore, long term adaptation or evolution takes place along the morphological gradients determined by the local resources available at each sub-system of a species.

3.7.4 Origami Endoscopy Capsule

The peristalsis of the digestive tract pushes the chyme forward in the direction of peristalsis. An endoscopy capsule is a tiny camera that is mainly used to examine the small intestine. Due to its limited size and power supply, there is no drive system. It can only be propelled forward passively by intestinal peristalsis, which prevents doctors from controlling it to find lesions. However, in our previous research, we discovered a special origami structure that can be pushed in the opposite direction by intestinal peristalsis [28]. Using this structure, we can control the direction and speed of the capsule's movement with limited energy.

The origami capsule concept is shown in Fig. 3.19. The origami structure is attached to the back of the endoscopy capsule and is manufactured of elastic material

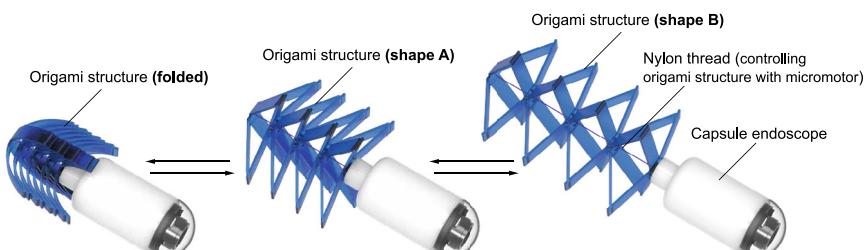


Fig. 3.19 The micromotor in the endoscopy capsule can pull the Nylon thread which runs through the origami structure to control the shape of the origami structure

(with shore hardness around 95 A). The structure can be produced using photosensitive resin printing technology and its shape can be adjusted by a micromotor pulling the nylon thread that runs through the structure. Under different shapes, the origami structure will show various dynamic properties in the same intestinal peristalsis wave. When the origami structure is folded, the size of the whole capsule can be small. The ice or digestible shell covers the capsule to make it easier for a patient to swallow. In the stomach, the capsule will be released from the shell and pushed into the small intestine. In the small intestine, the origami structure can be extended to shape A in Fig. 3.19 so that the capsule will be pushed in the same direction as the peristalsis. When the capsule finds the lesion and needs to go back to collect more data, the origami structure will be compressed to shape A, which can retrograde under the peristalsis. The retrograde and prograde motion are both driven by peristalsis, and only a small amount of energy is needed to change the origami shape.

Although the origami structure is elastic and soft, we can simplify it as a stiffness linkage to easily explain the dynamic principle. The Fig. 3.20 shows the simplified geometry of the origami structure (shape A) and illustrates the mechanism of the retrograde motion. We designed the shape by taking inspiration from ears of wheat. If a stalk of wheat slips into your sleeve and you try to shake it out by waving your arm, you will find that it gets deeper and deeper. This is because there are many wheat awns on the ear which point in one direction, which restricts it to move in only that direction. The shape A consists of four concave quadrilaterals strung together (Fig. 3.20a). The length is L , and the width is H , where H is similar to the intestine diameter. Analysing a single quadrilateral (Fig. 3.20b), note that the side BC of the quadrilateral extends outwards slightly, which is CD . It imitates the wheat awn by restricting the origami structure's direction of movement. The line passing through AB is the axis of symmetry of the structure. The angle between CD and this line is α . In shape A, α is greater than 90° , so the line CD faces to the right, and the origami structure is restricted from moving left.

We then require it to move. We took inspiration from the fact that some worms move by stretching and contracting their longitudinal length. A concave quadrilateral has the property of a negative Poisson ratio structure that if the horizontal length is stretched or compressed, the vertical length will also be stretched or compressed [29]. In Fig. 3.20b, the distance from vertex C to the structure symmetry axis is h , and the spread between vertex A and B is l . Figure 3.20c shows the peristalsis compresses the geometry of the concave quadrilateral. The distance from vertex C to the structure symmetry axis is contracted to h' , resulting in the distance between vertices A and B compressing to l' (meanwhile, the length of the whole structure L also becomes shorter). Then, after the peristalsis wave passes through the origami structure, l' will return to l because of the elasticity of the concave quadrilateral.

Figure 3.20d illustrates the process of retrograde motion. We assume the peristalsis wave propagates from left to right. As the wave passes through the concave quadrilaterals sequentially from left to right, the position of the leftmost quadrilateral will be stationary. However, the others are shifted a small distance to the left (because L becomes shorter). After the peristalsis disappears, the elasticity stretches the origami structure back to its original shape. Thus, all the quadrilaterals will move

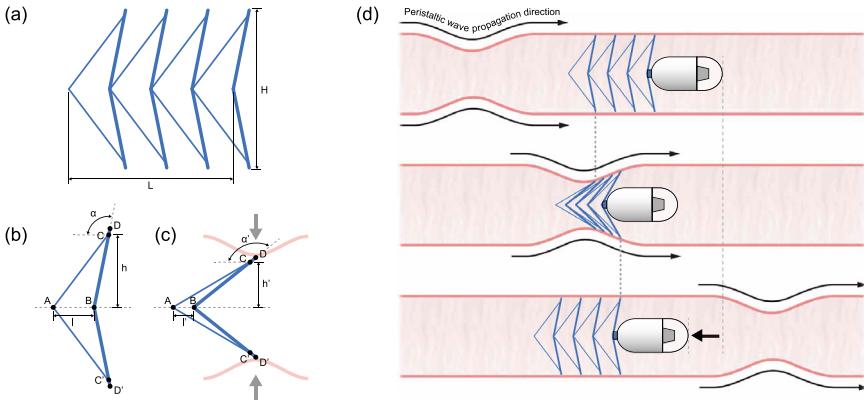


Fig. 3.20 The mechanism of the origami structure is pushed backwards by intestinal peristalsis

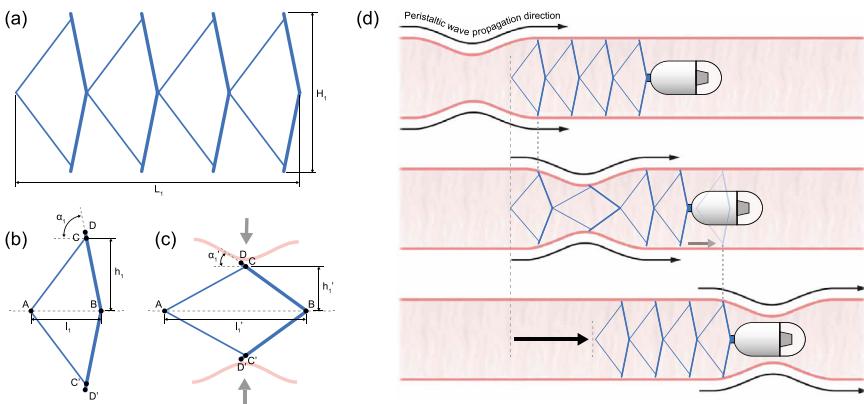


Fig. 3.21 The mechanism of the origami structure is pushed forwards by intestinal peristalsis

left slightly except the rightmost one. When this process finishes, the endoscopy capsule has been pushed along by peristalsis.

The Fig. 3.21 shows the simplified geometry of the origami structure (shape *B*) and illustrates the mechanism of the prograde motion. When the nylon thread does not exert any tension on the origami structure, it returns to its original shape (shape *B*, shown in Fig. 3.21a). It has length L_1 and width H_1 , where H_1 is also similar to the intestine diameter. These four quadrilaterals become convex, the geometry of which is shown in Fig. 3.21b, while the compressed geometry is shown in Fig. 3.21c. Compared with the concave quadrilateral in shape *A*, the convex quadrilateral differs in two ways. The first is the angle α_1 between side CD and the axis of symmetry, which is less than 90° , so side CD faces towards the left, and the structure is limited to movement to the right. The second is that the convex quadrilateral is a positive Poisson

ratio structure. Hence, when the distance h_1 between vertex C and the symmetry axis contracts due to peristalsis, the distance l_1 between A and B stretches.

Figure 3.21d shows how peristalsis propels the capsule with a shape B origami structure forward. As the peristalsis travels from left to right and squeezes the origami structure, the leftmost quadrilateral does not move, but the others move slightly to the right. After the peristaltic wave disappears, the rightmost one is stationary, and the others move a small distance to the right.

It is worth noting that there are also retrograde peristaltic waves in the intestine [30]. However, neither shapes A nor B can be pushed by retrograde waves because the vertices, like the wheat awn, restrict the direction of movement. Therefore, this structure's benefit is that it makes the endoscopy capsule always move in the expected direction without being affected by irregular peristalsis.

The moving speed of the capsule (whether shape A or B) can be controlled by the rate at which the origami structure has been compressed. Let's take shape A as an example. If the nylon thread is slightly tauter, the width H becomes shorter, causing the same peristalsis to compress the quadrilaterals by less. The difference between h' and h becomes smaller, resulting in a smaller change in l and L . Each peristalsis only pushes the capsule a shorter distance than before, which is how we control the velocity of the capsule.

As shown in Fig. 3.22, in order to observe and simplify our experiments, we tested the origami structure in a two-dimensional intestinal phantom. We used two silicone layers (Smooth-on Ecoflex 00-30) to simulate the intestine walls. Two rollers passed

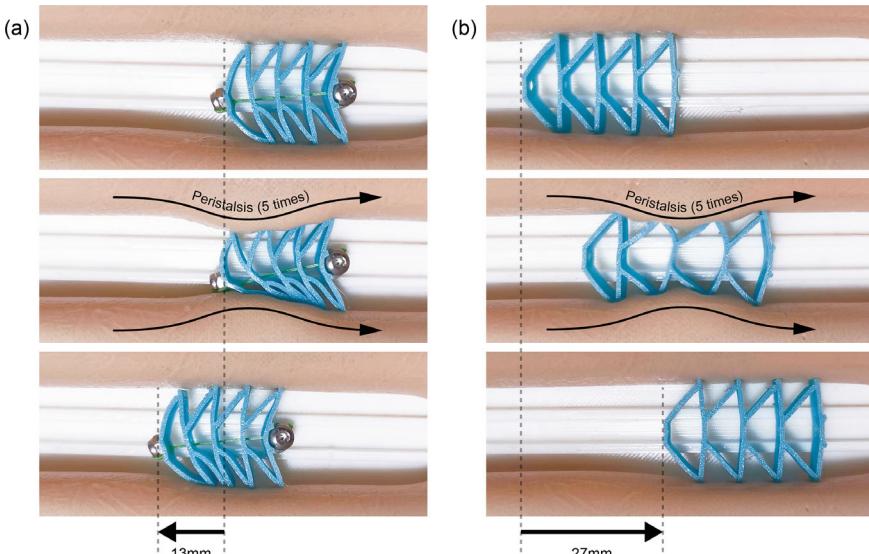


Fig. 3.22 After being squeezed by intestinal peristalsis in the intestinal phantom, the origami structure in shape A moves to the left, and the origami structure in shape B moves to the right

from left to right on the intestinal wall to simulate intestinal peristalsis. The origami structure is made of 95A TPU material through FDM printing technology. In Fig. 3.22a, the origami structure is constrained into shape A with a nylon thread. In Fig. 3.22b, the origami structure remains in its maximum expanded state, which is shape B. When shape A and shape B are squeezed by intestinal peristalsis five times, they are moved 13 mm to the left and 27 mm to the right, respectively.

This endoscopy capsule is a special case that only relies on MC_T type commands, that is, it can switch the direction it is pushed by intestinal peristalsis simply by reconstructing the origami structure. For the movement of robots with very limited power, such as endoscopy capsules, rationally abandoning MC_M type commands and using the external environment to passively move the robot can save energy and increase operating time.

3.7.5 *Tapered Whisker Reservoir Computing System for Mobile Terrain Identification*

Embodied artificial intelligence proposes the integration of sensory and motor capabilities into robotic systems to achieve human and animal-like perception of the world and interaction with it. In the natural world, the whisker, vestibular system, and cochlea serve as three examples of compliant mechanical systems found in biological organisms, utilizing the non-linear vibrational dynamics of their own bodies to address real-time perception challenges. Taking the cochlea as an example, the brain's ability to process signals across different frequency domains arises from the fact that when sound waves induce vibrations within the cochlea, the shape of the cochlear membrane separates different frequency components to known locations along the tube, enabling the brain to efficiently process frequency-domain signals. Although the tapered tube of the cochlea passively provides frequency separation of sound signals, it effectively and rapidly maps vibrational signals from the temporal domain to the spectral domain. Inspired by computations that purely exploit the physical principles of tapered membrane vibrations, as well as other similar inspirations, researchers have proposed an abstract machine learning algorithm called reservoir computing, which relies on mechanical systems for signal processing and classification.

The diagram of reservoir computing is depicted in Fig. 3.23b, at the centre of the physical reservoir computing system is a high-dimensional nonlinear complex physical system (called the *reservoir*), which takes an external time-varying and time-series input $\mu(t)$ and maps it nonlinearly into its high-dimensional state space. In general, if the mechanical system serving as a reservoir is sufficiently complex and exhibits abundant nonlinearity, complex input signal classification $y(t)$ can be achieved by using simple logistic regression to find the weights W_t of its linear outputs. Therefore, the soft body of the robot itself could work as the reservoir computer to achieve cost-efficient morphological computation.

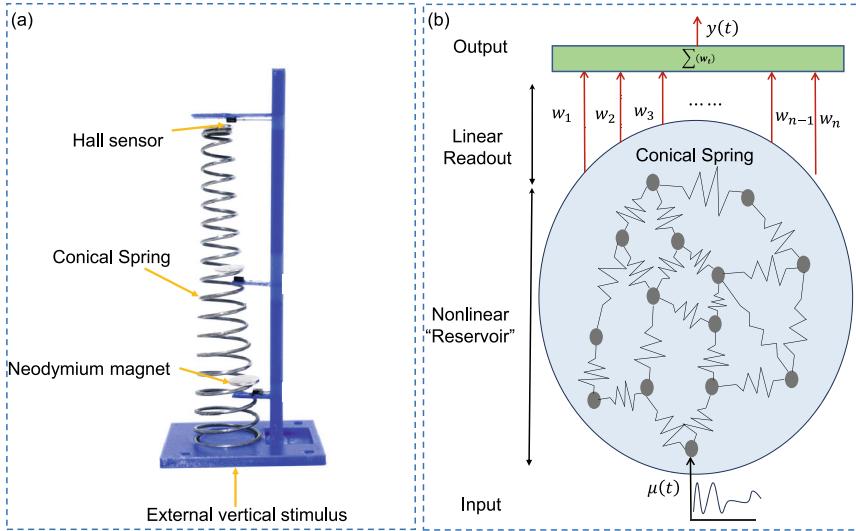


Fig. 3.23 **a** The tapered whisker sensor designed for the mobile robot terrain identification and navigation. **b** Schematic of the reservoir computing system, showing the internal nonlinear reservoir and the linear readout function. In this research, the nonlinear tapered spring works as a reservoir, and the tapered spring induces vibration components of different frequencies along the axial positions of the whisker when there is a time-series stimulus from the base in the vertical direction of the whisker sensor. A whisker-based reservoir computing system can accommodate an arbitrary number of outputs, wherein only the matrix W_t requires training via linear logistic regression. However, in our specific experiment focusing on terrain recognition, we have achieved sufficient accuracy by utilizing only three outputs

Taking inspiration from the human cochlea and animal whisker, we present the use of tapered whisker-based reservoir computing (TWRC) system mounted on a mobile robot for high-accuracy and cost-efficient terrain classification and roughness estimation of unknown terrain [31]. In this research, we leverage the concept of vibration frequency separation along the axis of a vertically mounted tapered spring on a mobile robot to implement a reservoir computer system for terrain classification, based on our previous research [32]. The proposed approach employs a nonlinear whiskered sensor, as shown in Fig. 3.23a, for identifying different types of terrain. The physical reservoir consists of a high-carbon conical nonlinear spring with specific dimensions: a free length of 100 mm, a wire diameter of 1 mm, a bottom diameter of 20 mm, and a tip diameter of 10 mm. Within the conical spring beam, three neodymium permanent magnets are embedded at the bottom, middle, and top sections. To measure the terrain-induced oscillations, three SS49E linear Hall sensors are positioned orthogonally beneath the neodymium permanent magnets using a cantilever beam. The cantilever, composed of sturdy acrylic pillars, ensures that the relative positions of the three Hall sensors remain relatively constant while the robot is in motion. The whisker sensor's output reading is limited by the Hall sensor, which operates within a range of 1–4 V for magnetic fields ranging from—1000

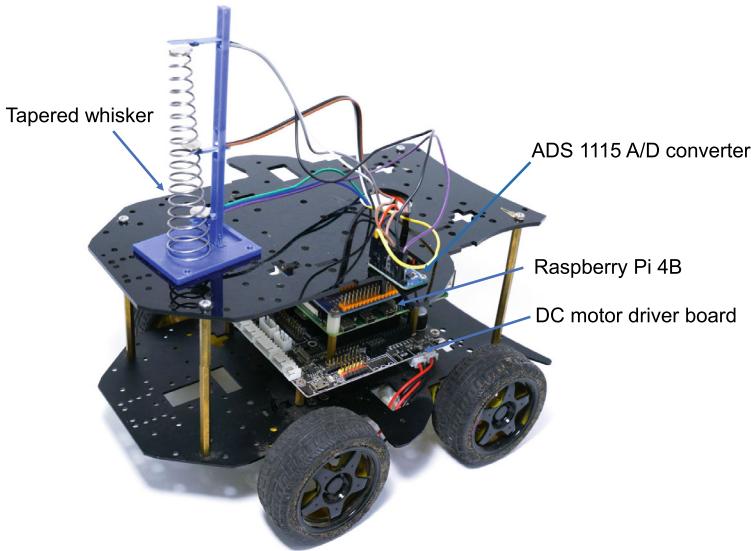


Fig. 3.24 A differential-drive mobile robot with a bio-inspired tapered whisker sensor vertically installed on the front to achieve reservoir computing-based terrain classification and texture-guided navigation

Gauss to 1000 Gauss. The SS49E Hall sensor exhibits a sensitivity of 1.4 mV/Gauss, enabling the detection of extremely faint magnetic field vibrations and facilitating the discrimination of closely similar terrains or surfaces.

To ensure early recognition of new terrain information and mitigate potential risks, the whisker sensor is strategically installed in front of the wheeled robot. This placement takes advantage of the fact that the robot's front wheels are the first to encounter external terrain excitation, as depicted in Fig. 3.24. During the robot's steady-state motion across various terrains, the conical whiskered spring shaft experiences vertical vibrations induced by the external environment. As a result, the tapered spring undergoes nonlinear displacement perpendicular to its axis. This deformation of the shaft also imparts vibrations to the embedded magnet. Consequently, the Hall effect sensor detects continuous low-frequency electrical voltage signals arising from the nonlinear changes in magnetic flux. As shown in Fig. 3.24, the output signals from the three Hall sensors are routed to a Raspberry Pi 4B through an ADS1115 analogue-to-digital converter. This setup facilitates the transfer of sensor data for further processing and analysis. The ADS1115 A/D converter operates within a frequency range of 8–860 Hz. This broad frequency range ensures that the robot captures an adequate amount of information even within short time windows. To achieve synchronized and real-time data collection, robot motion and control, reservoir computing, terrain identification, and surface roughness prediction in unfamiliar environments, the Raspberry Pi 4B

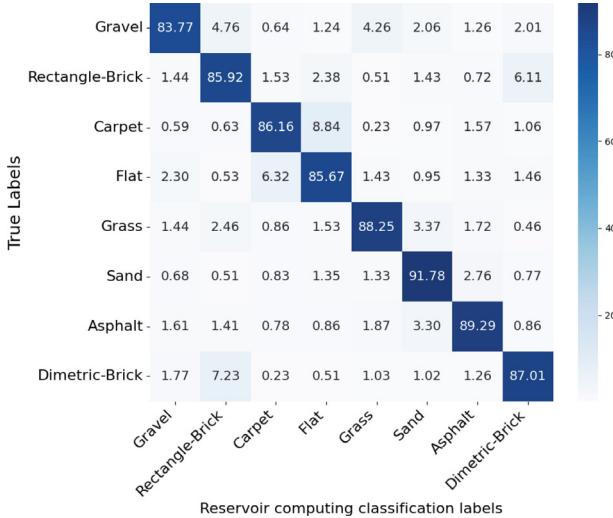


Fig. 3.25 The average classification outcomes of the TWRC system were evaluated over 25 random trials, with the robot moving at a velocity of 0.2 m/s and utilizing a time window duration of 1.5 s

serves as the central hub. Due to its efficient processing capabilities, the Raspberry Pi 4B can handle these tasks effectively while requiring minimal computing power. By coordinating all these operations on the Raspberry Pi 4B, the system achieves seamless integration and real-time terrain identification and decision-making capabilities, enabling the robot to cost-efficiently classify different terrains [31], predict surface roughness accurately [33], and auto-navigate unknown terrains only relying on the tapered whisker reservoir computing system [34].

The classification confusion matrix presented in Fig. 3.25 illustrates the performance of the TWRC system across 25 randomized experiments, yielding an average classification rate of 87.2% within a time window duration of 1.5 s, with the AD converter's sampling frequency of 100 Hz. These findings demonstrate the TWRC system's effective capabilities in identifying and classifying even similar terrain surfaces. Although reservoir computing typically necessitates a high-dimensional and complex nonlinear physical entity to serve as the reservoir computer, along with a large number of outputs, we have demonstrated in this study that our specific terrain recognition requirements can be met by relying solely on a limited set of three outputs, as shown in Fig. 3.23a.

Kinematic tuning is evident in these experiments, as observed from the results in Fig. 3.26, where the mobile robot can enhance its perception accuracy of terrain classification by adjusting its speed using the MC_M command. However, as mentioned in the previous subsection, the efficiency of MC_M -type commands depends on how well MC_T -type motor commands are regulated.

Through these experiments, we validate the importance of robot kinematic tuning and behavioral lenses for robots to have better perception and proficiency, by coupling

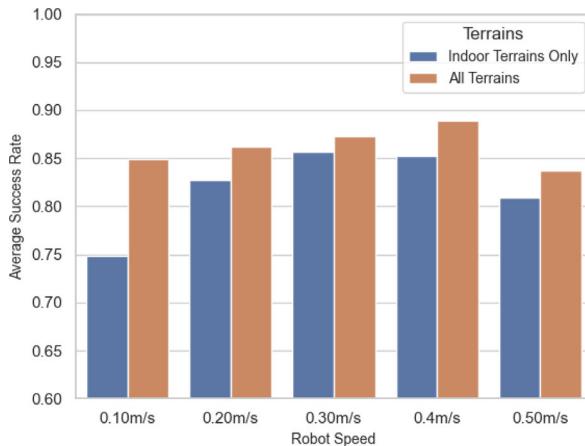


Fig. 3.26 Reservoir computing average prediction accuracy of eight terrains when the whiskered mobile robot receives different MC_M commands to change its speeds at 0.1, 0.2, 0.3, 0.4, and 0.5 m/s

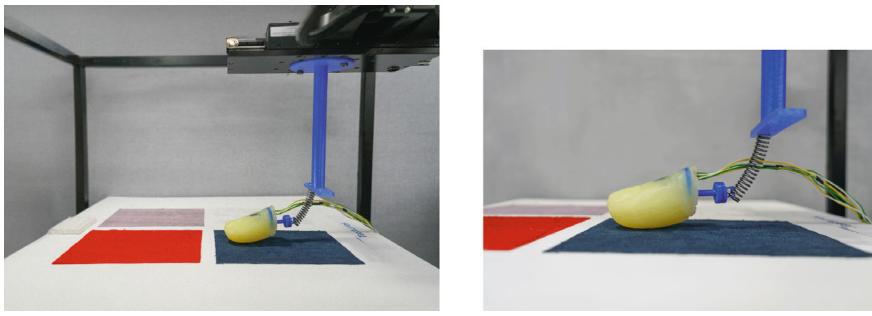


Fig. 3.27 Experimental setup with the fingertip model affixed to the XY Table with the help of a spring attachment to generate the motion of a finger stroke

perception and embodiment through MC_T and MC_M commands. This study explores the application of physical whisker-based reservoir computing in achieving rapid and cost-effective environment perception and navigation in mobile robots. It aims to bridge the gap between real-time high-performance requirements and low-power onboard hardware systems for robots (Fig. 3.27).

3.7.6 Texture Perception Using Finger Haptics

The ability to recognize materials and textures is an important skill for robots. To create systems with this capability, it is necessary to develop tactile sensors that are

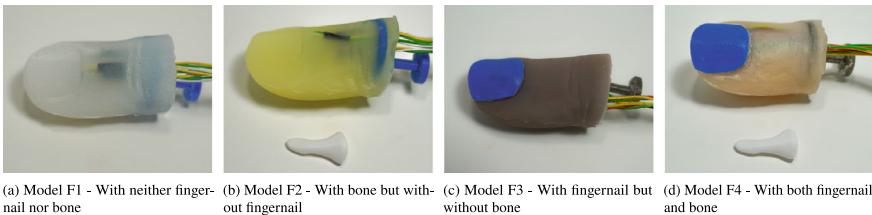


Fig. 3.28 Fingertip models used for the experiment

sufficiently sensitive and have high spatial resolution. The biological tactile system is extremely complex, with many sensors tightly packed into a square centimetre of skin [35]. These sensors can sense a wide range of stimuli, including temperature, pressure, vibrations, and pain. Humans mostly use their fingers to explore their environment and obtain tactile information, even though their bodies are covered in touch sensors. A few artificial tactile devices, such as tactile fingers, have been developed in the past for use in robot applications. These fingertips were designed specifically for manipulating objects and classifying textures [36–39]. They typically had a range of sensing components as well as tactile sensors covered in a soft overlay. These comprised accelerometers, micro electromechanical (MEMS) force sensors, polyvinylidene fluoride films, and strain gauges. These artificial tactile systems attempted to mimic some of the functions of human touch in robotic applications by merging these various sensing technologies. The functioning of humanoid robots working in unstructured surroundings depends heavily on tactile sensing. The surface characteristics of the items these robots deal with must be sensed, evaluated, and understood. For several types of robot systems, such as service robots, medical robots, and exploration robots, the capacity to recognise surface textures is extremely crucial. To efficiently carry out their intended jobs, these robots' tactile systems must be able to recognise and discriminate between various textures.

We present different models representing different morphological features of a fingertip to study tactile sensing for classification of different textures. We used four different kinds of models. As shown in Fig. 3.28, each fingertip model represents a distinct combination of anatomical features like the distal phalange bone and the fingernail:

- Model 1 (F1): Designed with neither the fingernail nor the distal phalange bone.
- Model 2 (F2): Features the distal phalange bone but lacks the fingernail.
- Model 3 (F3): Includes the fingernail but excludes the distal phalange bone.
- Model 4 (F4): Complete with both the distal phalange bone and the fingernail.

We present these fingertip models with three one-dimensional hall effect sensors placed in the X, Y and Z directions respectively. The tactile data is obtained from each of these sensors for comparison. The covariation between these tactile data streams is found out and the corresponding Frobenius distance is calculated. The construction of each fingertip model utilizes a soft compound, specifically Ecoflex-0030 silicone.

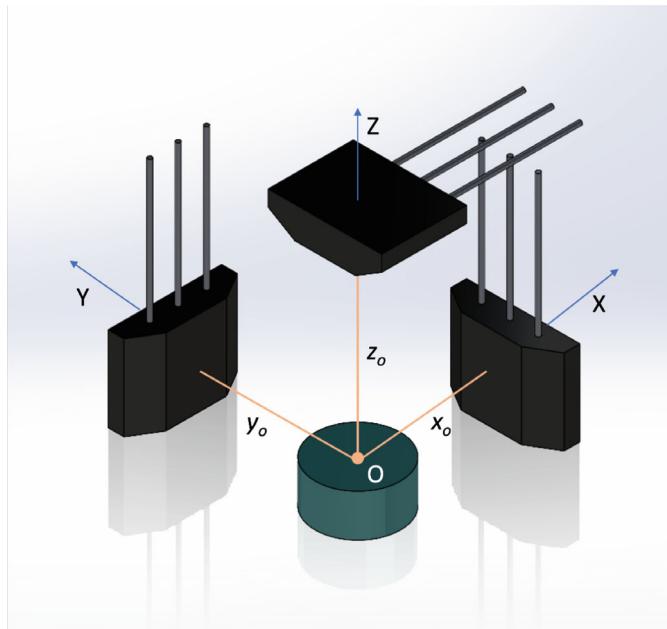


Fig. 3.29 Hall effect sensors placed orthogonally centering a cylindrical permanent magnet

A specially designed mould for the fingertip is developed through SolidWorks software and produced using 3D printing technology. The silicone mixture is poured into this mould and then placed in a vacuum chamber to allow trapped air bubbles to be released. Once this process is completed and all bubbles have been expelled, the silicone-filled mould is transferred to a heater for curing at a temperature of 50° Celsius. Meanwhile, the components for the distal phalange bone and the fingernail are also designed using SolidWorks software. These components are then materialised through 3D printing, with PLA being used for the bone and PETG for the fingernail. The three hall effect sensors are placed perpendicular to each other, centered on a neodymium cylindrical magnet with a diameter of 3 mm. They are placed inside each of the fingertip model just before the silicone mixture is poured into the mould. To develop the soft sensor configuration as shown in Fig. 3.29, a cuboid mould of required dimensions is 3D printed. Within the confines of this geometrically structured mould, a neodymium magnet is systematically placed in the centre of the base of the mould. Then the sensors are placed as shown in the Fig. 3.29. The dimensions of this mold are meticulously determined to ensure that when hall sensors are oriented along the x , y , and z axes, they are at a spatial separation of x_o , y_o and z_o in the corresponding axes. Only when a force is applied will the magnet gets displaced and the variation in magnetic field affects the sensor readings. Subsequent to the precise positioning of the sensors, a silicone compound is introduced, serving to encapsulate and subsequently solidify the integrated configuration of sensors and magnet.



Fig. 3.30 Different textures used for the experiment

The fingertip model (one at a time) is attached to a spring holder set up, which in turn is attached to an XY Table (AeroTech ANT130-160-XY-25DU-XY-CMS UPPER) as shown in Fig. 3.27. There are vertical and horizontal linear stages in the XY table allowing the holder with fingertip model to traverse in the desired direction. The linear stages are connected to a computer and thus the data retrieval and the movement of the linear stages are synchronous to each other. The experimental data is collected and fed into the Matlab software for comparison.

For data collection, 6 different textures (as shown in Fig. 3.30) are used, which include visible textures like bubble wrap, denim, felt, and invisible textures like cardboard and snake leather. These textures were fixed on a base and the fingertip model was slid against each of the textures with the help of the XY table. A single palpation was considered as the total motion of the fingertip at a palpation velocity of 50 mm/s for a stroke length of 15 cm. Correspondingly the tactile data is collected. In this experiment, 20 repetitions were carried out for each fingertip model against each texture. From the collected tactile data, the covariance matrix of each fingertip model on each texture was evaluated. For feature generation, the Frobenius distance between the covariance matrices of each fingertip model were calculated. The Frobenius distances of the covariance matrices for each of the models F1, F2, F3 and F4 are shown in the Tables 3.1, 3.2, 3.3 and 3.4 respectively (Fig. 3.31).

It is evident from the heatmaps that the maximum Frobenius distance (0.05) is obtained with the model F4, that is the one with both the distal phalange bone and the fingernail. More the distance implies that the model can distinguish between

Table 3.1 Frobenius distances between covariance matrices of Model F1

Textures	Bubble wrap	Cardboard	Denim	Felt	Snake leather
Bubble wrap	0	0.0178	0.0125	0.0126	0.0102
Cardboard	0.0178	0	0.0298	0.0284	0.0149
Denim	0.0125	0.0298	0	0.0057	0.0200
Felt	0.0126	0.0284	0.0057	0	0.0175
Snake leather	0.0102	0.0149	0.0200	0.0175	0

the textures. Hence model F4 with the maximum Frobenius distance gives better classification of the texture. The distal phalange bone in the model gives a stable contact aiding in texture discrimination. Inside each finger model, there are 3 hall effect sensors placed orthogonal to each other with a cylindrical magnet. While the finger slides over a surface, these sensors come in contact of the magnetic field which is enhanced by the distal phalange bone. The fingernail acts as a lens enhancing the haptic perception. This haptic lensing affects the distribution of forces and vibrations, thereby improving the perceived haptic properties. In this case, the spring (as shown in Fig. 3.27b) acts as the stiffness of the finger during sliding (MC_T type). Regulating the sliding speed and palpation force is an MC_M type intervention. Figure 3.32 shows the Gaussian ellipsoids on showing the influence of MC_T . Three different springs of stiffness k_1 , k_2 and k_3 , in the ratio 1:11:63 were considered respectively.

Acknowledgements The work in this chapter was supported by UK Engineering and Physical Sciences Research Council (EPSRC) RoboPatient project under Grant EP/T00603X/1, EPSRC MOTION Grant EP/N03211X/2, and EU Horizon 2020 research and innovation programme under grant agreements 101016970 (Natural Intelligence for Robotic Moni-toring of Habitats).

Table 3.2 Frobenius distances between covariance matrices of Model F2

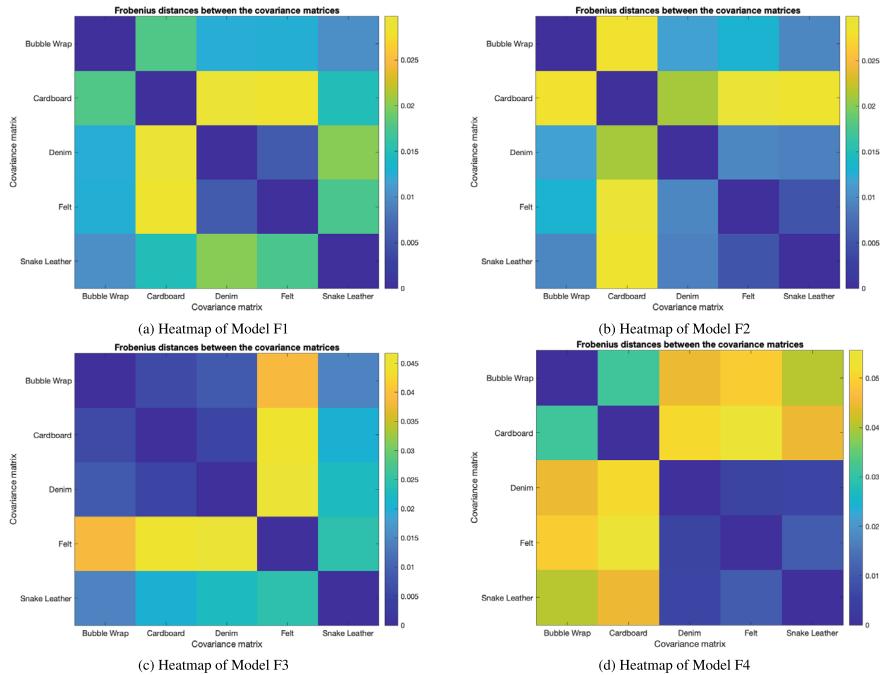
Textures	Bubble wrap	Cardboard	Denim	Felt	Snake leather
Bubble wrap	0	0.0282	0.0117	0.0132	0.0095
Cardboard	0.0282	0	0.0210	0.0298	0.0284
Denim	0.0117	0.0210	0	0.0095	0.0089
Felt	0.0132	0.0298	0.0095	0	0.0050
Snake leather	0.0095	0.0284	0.0089	0.0050	0

Table 3.3 Frobenius distances between covariance matrices of Model F3

Textures	Bubble wrap	Cardboard	Denim	Felt	Snake leather
Bubble wrap	0	0.0060	0.0087	0.0385	0.0144
Cardboard	0.0060	0	0.0052	0.0444	0.0202
Denim	0.0087	0.0052	0	0.0467	0.0225
Felt	0.0385	0.0444	0.0467	0	0.0245
Snake leather	0.0144	0.0202	0.0225	0.0245	0

Table 3.4 Frobenius distances between covariance matrices of Model F4

Textures	Bubble wrap	Cardboard	Denim	Felt	Snake leather
Bubble wrap	0	0.0313	0.0447	0.0495	0.0405
Cardboard	0.0313	0	0.0509	0.0556	0.0449
Denim	0.0447	0.0509	0	0.0060	0.0062
Felt	0.0495	0.0556	0.0060	0	0.0108
Snake leather	0.0405	0.0449	0.0062	0.0108	0

**Fig. 3.31** Heatmaps representing the Frobenius distances between the covariance matrices

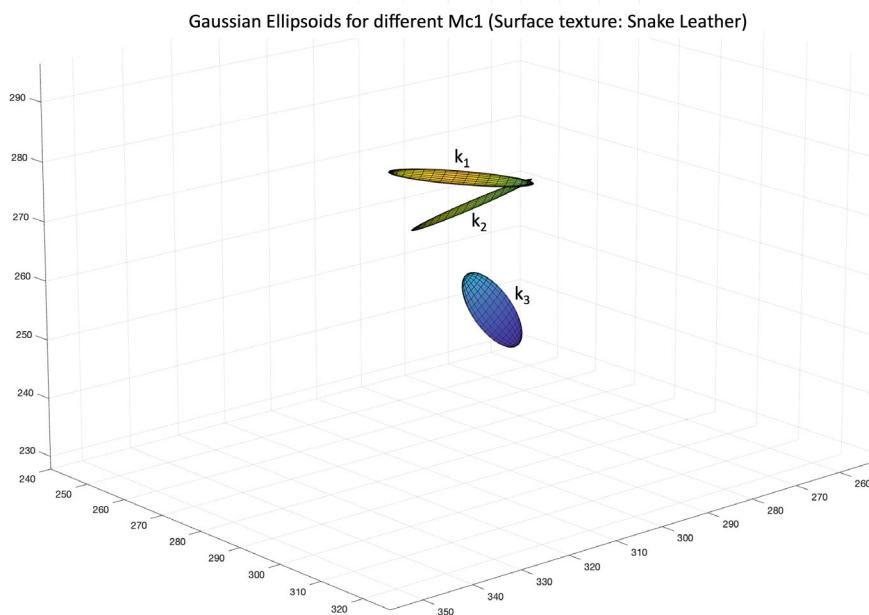


Fig. 3.32 Gaussian Ellipsoids showing the influence of MC_T

References

1. Cangelosi A, Bongard J, Fischer MH, Nolfi S (2015) Embodied intelligence. In: Springer handbook of computational intelligence, pp 697–714
2. Pfeifer R, Lungarella M, Iida F (2007) Self-organization, embodiment, and biologically inspired robotics. *Science* 318(5853):1088–1093
3. Arkin RC (1998) Behavior-based robotics. MIT press
4. Abad S-A, Herzig N, Sadati SMH, Nanayakkara T (2019) Significance of the compliance of the joints on the dynamic slip resistance of a bioinspired hoof. *IEEE Trans Robot* 35(6):1450–1463
5. Sumbre G, Fiorito G, Flash T, Hochner B (2006) Octopuses use a human-like strategy to control precise point-to-point arm movements. *Curr Biol* 16(8):767–772
6. Beal DN, Hover FS, Triantafyllou MS, Liao JC, Lauder GV (2006) Passive propulsion in vortex wakes. *J Fluid Mech* 549:385–402
7. Lohmiller W, Slotine JJE (1998) On contraction analysis for non-linear systems. *Automatica* 34(6):683–696
8. Sornkarn N, Nanayakkara T (2016) Can a soft robotic probe use stiffness control like a human finger to improve efficacy of haptic perception? *IEEE Trans Haptics* 10(2):183–195
9. Hochner B (2012) An embodied view of octopus neurobiology. *Curr Biol* 22(20):R887–R892
10. Mulvey BW, Lalitharatne TD, Nanayakkara T (2023) DeforMoBot: a bio-inspired deformable mobile robot for navigation among obstacles. *IEEE Robot Autom Lett* 8(6):3828–3835
11. Hicks BJ, Culley SJ, Allen RD, Mullineux G (2002) A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. *Int J Inf Manag* 22(4):263–280

12. Chen L, Wang P, Dong H, Shi F, Han J, Guo Y, Childs PRN, Xiao J, Wu C (2019) An artificial intelligence based data-driven approach for design ideation. *J Vis Commun Image Represent* 61:10–22
13. Luo J, Li W, Qiu J, Wei D, Liu Y, Zhang Q (2013) Neural basis of scientific innovation induced by heuristic prototype. *PLoS One* 8(1):e49231
14. Dickinson MH, Farley CT, Full RJ, Koehl M, Kram R, Lehman S (2000) How animals move: an integrative view. *Science* 288(5463):100–106
15. Richmond FJ, Bakker DA (1982) Anatomical organization and sensory receptor content of soft tissues surrounding upper cervical vertebrae in the cat. *J Neurophysiol* 48(1):49–61
16. Andrew Russell R (1992) Using tactile whiskers to measure surface contours. In: Proceedings 1992 IEEE international conference on robotics and automation. IEEE Computer Society, pp 1295–1296
17. Pfeifer R, Bongard J (2006) How the body shapes the way we think: a new view of intelligence. MIT Press
18. Kim S, Laschi C, Trimmer B (2013) Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol* 31(5):287–294
19. Rus D, Tolley MT (2015) Design, fabrication and control of soft robots. *Nature* 521(7553):467–475
20. Laschi C, Mazzolai B, Cianchetti M (2016) Soft robotics: technologies and systems pushing the boundaries of robot abilities. *Sci Robot* 1(1):eaah3690
21. Rus D, Tolley MT (2018) Design, fabrication and control of origami robots. *Nat Rev Mater* 3(6):101–112
22. Shah D, Yang B, Kriegman S, Levin M, Bongard J, Kramer-Bottiglio R (2021) Shape changing robots: bioinspiration, simulation, and physical realization. *Adv Mater* 33(19):2002882
23. Godden T, Mulvey BW, Redgrave E, Nanayakkara T (2024) PaTS-wheel: a passively-transformable single-part wheel for mobile robot navigation on unstructured terrain. *IEEE Robot Autom Lett* 9(6):5512–5519
24. Mulvey BW, Nanayakkara T (2024) HAVEN: haptic and visual environment navigation by a shape-changing mobile robot with multimodal perception. *Sci Rep*
25. Cully A, Clune J, Tarapore D, Mouret J-B (2015) Robots that can adapt like animals. *Nature* 521(7553):503–507
26. Zhang L-Q, Nuber G, Butler J, Bowen M, Rymer WZ (1997) In vivo human knee joint dynamic properties as functions of muscle contraction and joint position. *J Biomech* 31(1):71–76
27. Akhond S, Herzig N, Wegiriya H, Nanayakkara T (2019) A method to guide local physical adaptations in a robot based on phase portraits. *IEEE Access*, 7:78830–78841
28. Ge Y, Lalitharatne TD, Nanayakkara T (2022) Origami inspired design for capsule endoscope to retrograde using intestinal peristalsis. *IEEE Robot Autom Lett* 7(2):5429–5435
29. Schenk M, Guest SD et al (2011) Origami folding: a structural engineering approach. *Origami* 5:291–304
30. Avvari RK (2020) Biomechanics of the small intestinal contractions. In: Xingshun Q, Sam K (eds) Digestive system, chapter 4. IntechOpen, Rijeka
31. Yu Z, Perera S, Hauser H, Childs PRN, Nanayakkara T (2022) A tapered whisker-based physical reservoir computing system for mobile robot terrain identification in unstructured environments. *IEEE Robot Autom Lett* 7(2):3608–3615
32. Yu Z, Sadati SM, Wegiriya H, Childs P, Nanayakkara T (2021) A method to use nonlinear dynamics in a whisker sensor for terrain identification by mobile robots. In: IEEE international conference on intelligent robots and systems (IROS)
33. Yu Z, Hadi Sadati SM, Hauser H, Childs PRN, Nanayakkara T (2022) A semi-supervised reservoir computing system based on tapered whisker for mobile robot terrain identification and roughness estimation. *IEEE Robot Autom Lett* 7(2):5655–5662
34. Yu Z, Hadi Sadati SM, Perera S, Hauser H, Childs PRN, Nanayakkara T (2023) Tapered whisker reservoir computing for real-time terrain identification-based navigation. *Sci Rep* 13(1):5213
35. Johansson RS, Randall Flanagan J (2009) Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nat Rev Neurosci* 10(5):345–359

36. Nicholas W, Santos VJ, Johansson RS, Loeb GE (2008). Biomimetic tactile sensor array. *Adv Robot* 22(8):829–849
37. Jamali N, Sammut C (2011) Majority voting: material classification by tactile sensing using surface texture. *IEEE Trans Robot* 27(3):508–521
38. Oddo CM, Controzzi M, Beccai L, Cipriani C, Carrozza MC (2011) Roughness encoding for discrimination of surfaces in artificial active-touch. *IEEE Trans Robot* 27(3):522–533
39. Chathuranga DS, Hirai S et al (2013) Investigation of a biomimetic fingertip's ability to discriminate fabrics based on surface textures. In: 2013 IEEE/ASME international conference on advanced intelligent mechatronics. IEEE, pp 1667–1674

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, Louisiana State University, Baton Rouge, LA 70803, USA

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

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, compared 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.

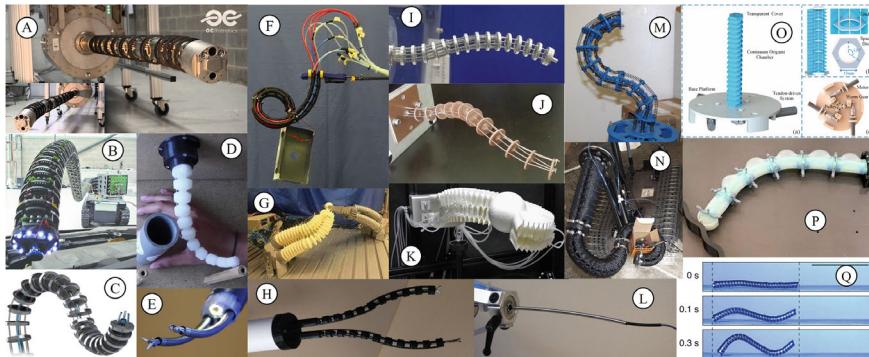


Fig. 4.1 Various Types of Continuum and Soft Robots. **a** OC Robotics—Series II, X125 snake-arm robot at UCL [3], **b** continuum manipulator [4], **c** tendon-driven continuum robot [15], **d** Tendon-actuated soft robot, **e** joystick-controlled concentric robot [29], **f** variable length multi-section 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]

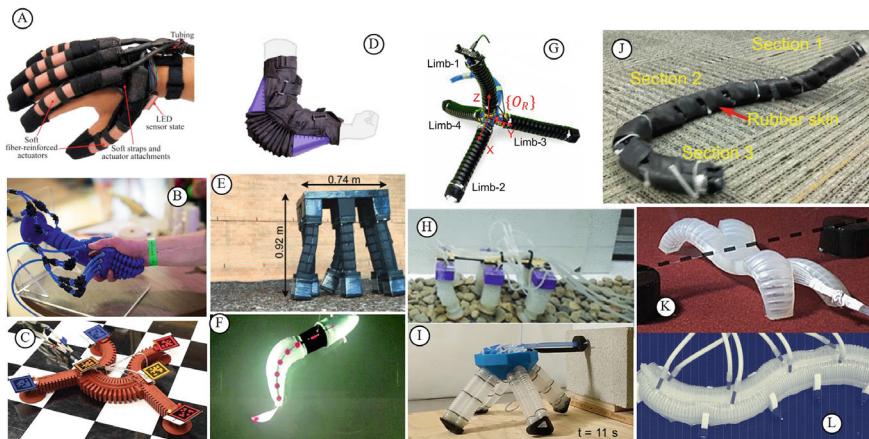


Fig. 4.2 **a** Soft robotic glove [28], **b** 3D-printed soft robotic hand [32], **c** gecko-inspired soft robot [33], **d** soft elbow exosuit [38], **e** Meter-scale soft hexapod robot [18], **f** Soft biomimetic fish robot [35], **g** Soft tetrahedral robot [27], **h** Sorx: soft pneumatic hexapedal 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 pneumatically 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

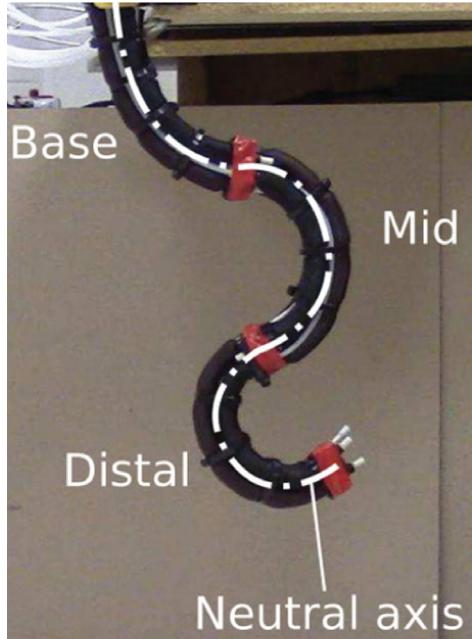
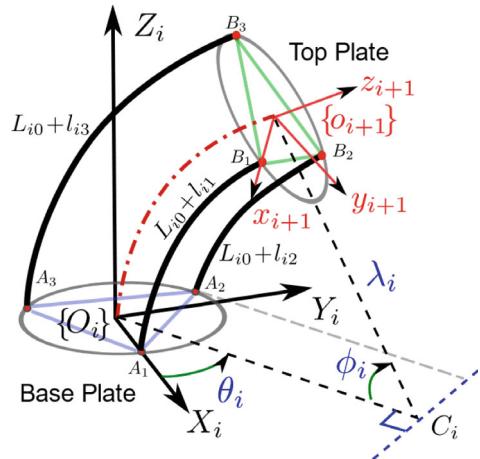


Fig. 4.4 Joint-space of a continuum/soft robotic section



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 $\mathbf{q} = [l_1(t), l_2(t), l_3(t)]^T \in \mathbb{R}^3$.

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_i , angle subtended by the bending arc $\phi \in [0, 2\pi]$, and angle of the bending plane with respect to the $+X$ axis, $\theta \in [-\pi, \pi]$.

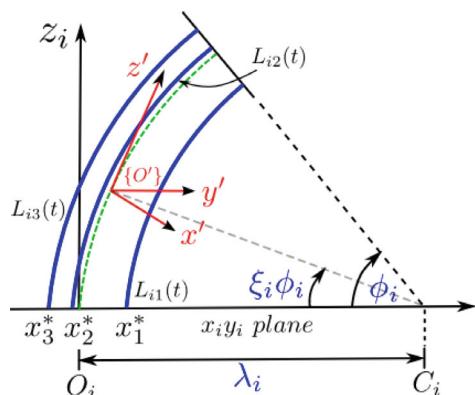
4.2.1.1 Deriving Curve Parameters in Joint Space Variables

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

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

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)



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

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

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

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

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

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

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

Similarly, rearranging Eq. (4.2) gives

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

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

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

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

$$\lambda_i (\mathbf{q}_i) = \frac{(3L_{i0} + l_{i1} + l_{i2} + l_{i3}) r_i}{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}} \quad (4.9)$$

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

$$\phi_i(\mathbf{q}_i) = \frac{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{3r_i} \quad (4.10)$$

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

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

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

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

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

$$\phi_i(\mathbf{q}_i) = \frac{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{3r_i} \quad (4.13)$$

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

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

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

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

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, $\mathbf{q} = [\lambda, \phi, \theta]^T \in \mathbb{R}^3$ is the curve parameter vector, $\mathbf{R} \in SO(3)$, $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ are the rotational and translational matrices of the robot.

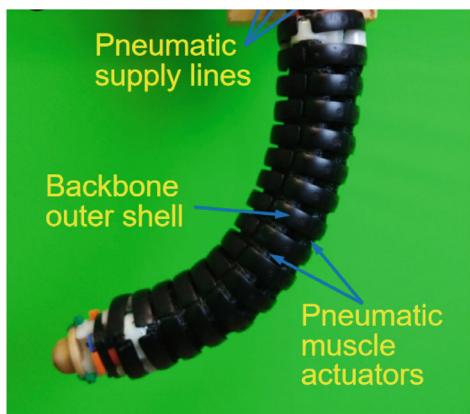
4.2.1.3 Constant-Length System Modeling

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).

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

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., $l_{j1} = -(l_{j2} + l_{j3})$, 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



$$\phi_j = \frac{2}{3r} \sqrt{\sum_{i=1}^3 (l_{ji}^2 - l_{ji}l_{j\text{mod}(i,3)} + 1)} \quad (4.17)$$

$$\theta_j = \arctan \left\{ \sqrt{3} (l_{j3} - l_{j2}), l_{j2} + l_{j3} - 2l_{j1} \right\}$$

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).

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

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 $\{O\}$, $\mathbf{T}^i : (\mathbf{q}^i, \xi_i) \mapsto \mathbb{SE}^3$, is given by

$$\mathbf{T}^i = \prod_{k=1}^i \mathbf{T}_k = \begin{bmatrix} \mathbf{R}^i & \mathbf{p}^i \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.19)$$

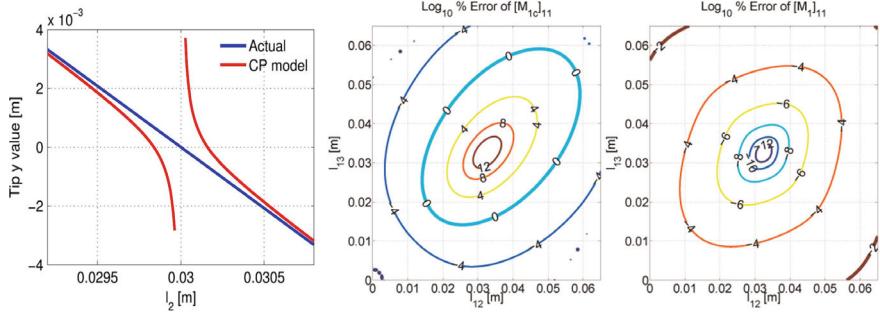
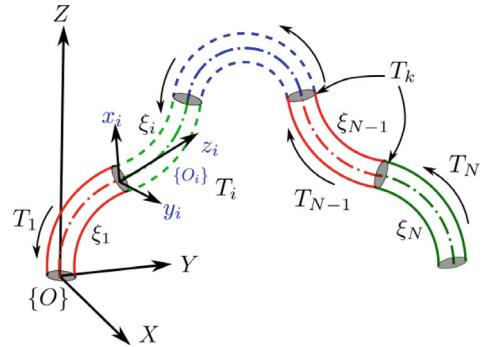


Fig. 4.7 (LEFT) Errors of $[T]_{24}$, given by Eq. (4.18), within singular neighborhood against l_2 when $l_2 \rightarrow 0.03$ where $l_1 = l_3 = 0.03$. The error spans within a singularity neighborhood thus eliminating the possibility of conditional HTM's to counter the numerical instabilities. Comparison 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 $\mathbf{R}^i(\mathbf{q}^i, \xi_i) \in \mathbb{R}^{3 \times 3}$ and $\mathbf{p}_i(\mathbf{q}^i, \xi_i) \in \mathbb{R}^3$ define the position and orientation of $\{O'_i\}$ 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

$$\begin{aligned} \mathbf{R}^i &= \mathbf{R}^{i-1} \mathbf{R}_i \\ \mathbf{p}^i &= \mathbf{p}^{i-1} + \mathbf{R}^{i-1} \mathbf{p}_i \end{aligned} \quad (4.20)$$

where $\mathbf{R}^{i-1}(\mathbf{q}^{i-1}) \in \mathbb{R}^{3 \times 3}$ and $\mathbf{p}_i(\mathbf{q}^{i-1}) \in \mathbb{R}^3$ 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 $\{O'_i\}$, $\boldsymbol{\omega}_i(\mathbf{q}^i, \dot{\mathbf{q}}^i) \in \mathbb{R}^3$ can be defined as

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

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

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

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

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, $\mathbf{J}_i^\omega(\mathbf{q}^i, \xi_i) \in \mathbb{R}^{3 \times 3n}$ and $\mathbf{J}_i^v(\mathbf{q}^i, \xi_i) \in \mathbb{R}^{3 \times 3n}$ respectively are derived. Here also, we use the property $\boldsymbol{\omega}_i = \boldsymbol{\Omega}_i^\vee$ to define $\mathbf{J}_i^\Omega(\mathbf{q}^i, \xi_i) \in \mathbb{R}^{3 \times 9n}$, as

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

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

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

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

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

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

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

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

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

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

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

$$D_1 = s_1 + \varepsilon t_1 \quad (4.27)$$

$$D_2 = s_2 + \varepsilon t_2 \quad (4.28)$$

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

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

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

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

where \odot is the dual quaternion multiplication, \circ 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.

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

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

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

$$\mathbf{t} = \frac{1}{2} \langle 0, \mathbf{p} \rangle \circ \mathbf{s} \quad (4.34)$$

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

$$\mathbf{R} = \begin{bmatrix} 1 - 2(s_2^2 + s_3^2) & 2(s_1s_2 - s_0s_3) & 2(s_0s_2 + s_1s_3) \\ 2(s_1s_2 + s_0s_3) & 1 - 2(s_1^2 + s_3^2) & 2(s_2s_3 - s_0s_1) \\ 2(s_1s_3 - s_0s_2) & 2(s_0s_1 + s_2s_3) & 1 - 2(s_1^2 + s_2^2) \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} 2t_x \\ 2t_y \\ 2t_z \end{bmatrix} \quad (4.35)$$

Coordinate transformation in Dual Quaternion Systems

When it comes to coordinate transformations, the traditional approach involves multiplying transformation matrices, denoted as $T_1 \cdot T_2 \cdot T_3 \dots \cdot T_n$. In the realm of dual quaternions, these transformations are represented as $Q_1 \odot Q_2 \odot Q_3 \dots \odot Q_n$, 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 quaternion 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 λ , ϕ , and θ 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 \mathbf{q} that optimize a cost function while satisfying constraints:

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

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

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

where $f(\mathbf{q})$ is the cost function to be minimized, $\mathbf{g}(\mathbf{q})$ represents inequality constraints (e.g., actuator range constraints), and $\mathbf{h}(\mathbf{q})$ 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 \mathbf{q}^* that minimize the cost function while satisfying the constraints

$$\mathbf{q}^* = \operatorname{argmin}_{\mathbf{q}} f(\mathbf{q}) \quad (4.39)$$

The solution \mathbf{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 $\dot{\mathbf{q}}$ and the end-effector velocities $\dot{\mathbf{x}}$ can be expressed using the Jacobian as follows:

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

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

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

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

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

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 ($\xi : 0 \rightarrow 1$).

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

where $\delta M = [\delta m I_{3 \times 3} \ 0_{3 \times 3}; 0_{3 \times 3} \ L_{\delta m r^2}]$ 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

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

where m_i is the mass of the robot, $\mathbf{g} = [0, 0, g]^T$ is the gravitational acceleration vector, and \mathbf{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 $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{P}(\mathbf{q})$. By applying the Lagrangian, the generalized equation of motion (EoM) can be expressed as

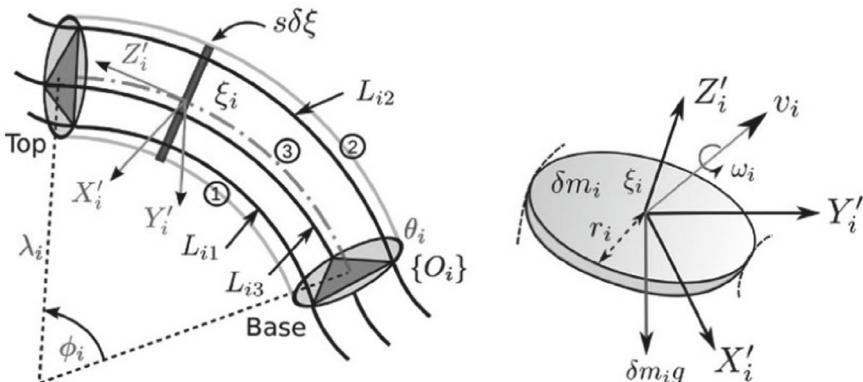


Fig. 4.9 **a** Schematic illustration of the infinitesimally thin slice at ξ_i on any i th continuum section along with curve parameters $\{\lambda_i, \phi_i, \theta_i\}$ (listed in Appendix B.1), actuator variables $\{O_i\}$, and $\{O'_i\}$. **b** Velocities and forces acting on the thin slice with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

$$\frac{d}{dt} \frac{\partial}{\partial \dot{\mathbf{q}}} \mathcal{L} - \frac{\partial}{\partial \mathbf{q}} \mathcal{L} = \mathbf{F}_e \quad (4.44)$$

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

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

where $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ is the generalized inertia matrix, $\mathbf{C} \in \mathbb{R}^{3 \times 3}$ 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 \mathbf{q} .

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

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

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

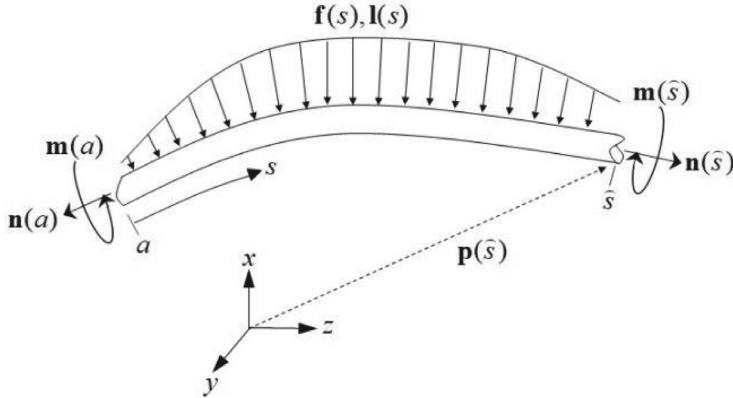


Fig. 4.10 An arbitrary section of the rod from a to b , subject to distributed forces $f(s)$ and moments $l(s)$ at a given time t [16]

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

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

where \mathbf{N} is Force vector, \mathbf{M} is Moment vector, \mathbf{t} is Tangent vector along the rod's centerline, \mathbf{n} is Normal vector, \mathbf{m} is Director vector, κ is Curvature vector, κ' 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

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

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

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

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

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

Bézier 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.

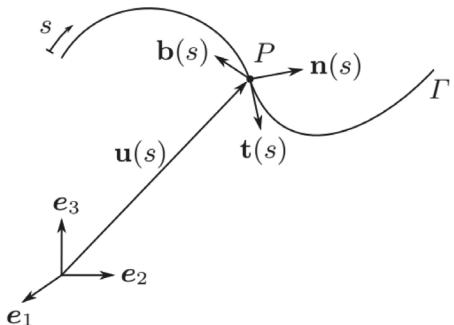
Having defined the control points for each Bézier segment, the subsequent step is to evaluate the Bézier curve at various parameter values. This evaluation process generates a point cloud that effectively represents the shape of the soft robotic arm. This approach allows for the precise modeling of the 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, \dots, P_n and a knot vector $[t_0, t_1, \dots, t_{n+k+1}]$ is given by:

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

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, \dots, 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 $[t_0, t_1, \dots, t_{n+k+1}]$.

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 (\mathbf{T}), the normal vector (\mathbf{N}), and the binormal vector (\mathbf{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).

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

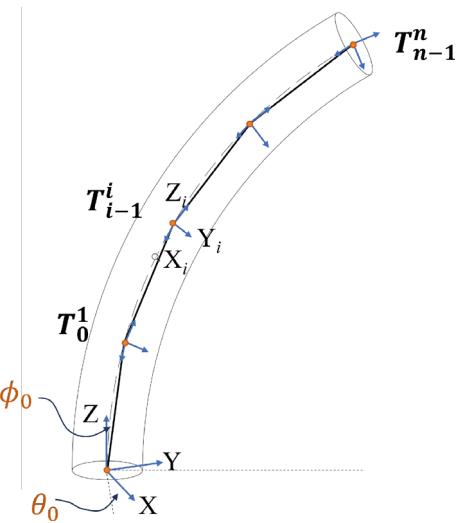
4.3 Lumped Parametric Models

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.

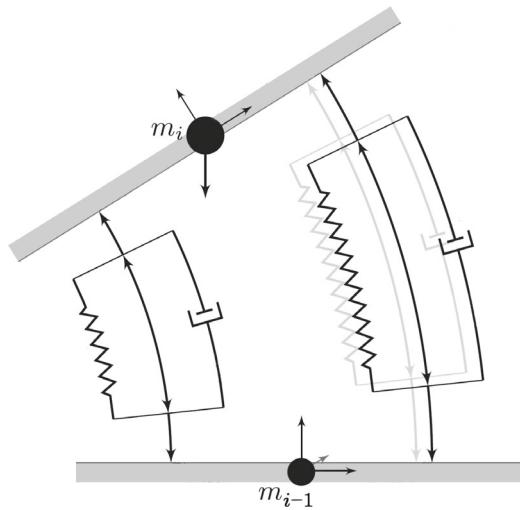


Fig. 4.13 Lumped-mass approximation with spring and dampers for material properties applied to a planar soft robotic arm. The segments are approximated as point masses, and in this case, there is no rotational energy considered. If a plate were used instead, rotational energy would be accounted for,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

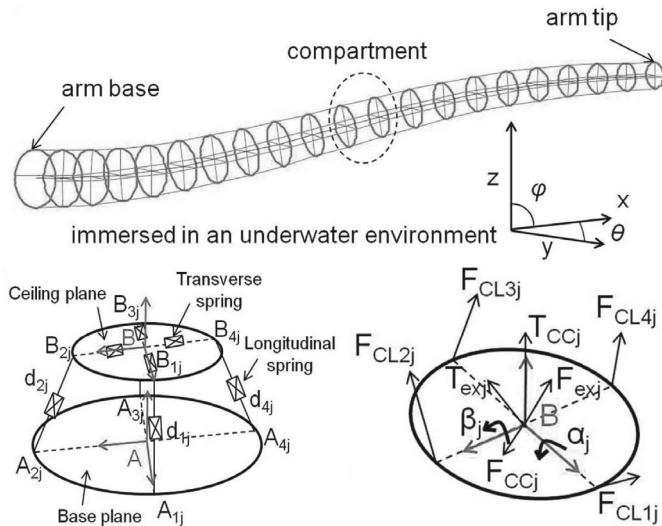


Fig. 4.14 Discretized representation of an octopus arm actuated parallel units actuated by two pairs of linear actuators. The arm consists of multiple discretized segments and this modular structure allows for flexible and coordinated motion, resembling the versatility of an octopus arm in real-world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

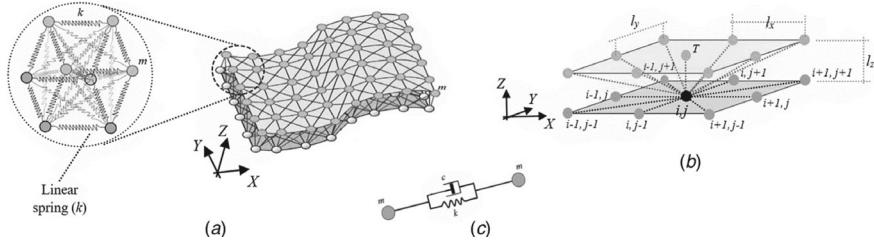
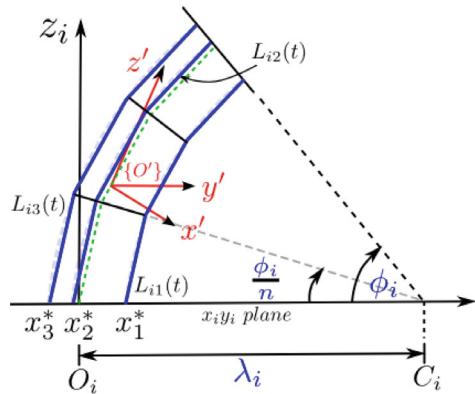


Fig. 4.15 **a** Illustration of the LDCS lumped-mass model with an enlarged view of one corner. **b** Depiction of the mass arrangement, showing the connections to the central masses i and j in the bottom layer of the two-layer model. **c** Detailed view of the spring-damper links connecting adjacent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

$$\mathbf{T}_0^n = \mathbf{T}_0^1 \cdot \mathbf{T}_1^2 \cdot \mathbf{T}_2^3 \cdot \dots \cdot \mathbf{T}_{i-1}^i \cdot \dots \cdot \mathbf{T}_{n-1}^n \quad (4.63)$$

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).

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

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

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

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

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

$$\sum F_i = m \cdot \ddot{x}_i \quad (4.66)$$

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

$$\sum \tau_i = I_i \cdot \ddot{\theta}_i \quad (4.67)$$

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

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \quad (4.68)$$

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

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

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

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

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:

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

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.

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

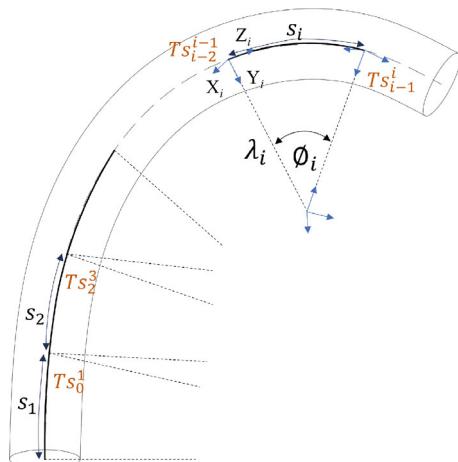
where λ_i is the radius of the constant curvature arc s_i . The angle $\phi_i = \frac{s_i}{\lambda_i}$ is the arc angle, which is considered to be the configuration variable of the kinematic. $\mathbf{T}_y(\cdot)$ is the homogeneous translation matrix along the y -axis. The $\mathbf{R}_x(\cdot)$ is the 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

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

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 $\{\overline{O}_i\}$, and introduce a homogeneous transformation matrix (HTM), denoted as $\bar{\mathbf{T}}_i : (\mathbf{q}_i) \mapsto \mathbb{SE}^3$, with respect to $\{O_i\}$, defined as

$$\bar{\mathbf{T}}_i = \int \mathbf{T}_i = \begin{bmatrix} \bar{\mathbf{R}}_i & \bar{\mathbf{p}}_i \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.74)$$

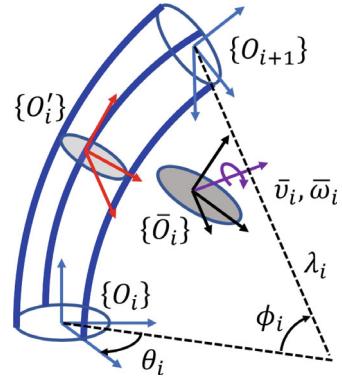
where $\bar{\mathbf{R}}_i = \int \mathbf{R}_i(\mathbf{q}_i) \in \mathbb{R}^{3 \times 3}$ is the resultant rotation matrix and $\bar{\mathbf{p}}_i = \int \mathbf{p}_i(\mathbf{q}_i) \in \mathbb{R}^3$ is the position vector [?]. Note that the CoG is a function of \mathbf{q}_i and therefore varies as the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

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

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

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

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



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

$$\begin{aligned}\bar{\mathbf{R}}^i &= \mathbf{R}^{i-1} \bar{\mathbf{R}}_i \\ \bar{\mathbf{p}}^i &= \mathbf{p}^{i-1} + \mathbf{R}^{i-1} \bar{\mathbf{p}}_i\end{aligned}\quad (4.76)$$

where \mathbf{R}^{i-1} and \mathbf{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 $\{\bar{\mathbf{O}}_i\}$), $\bar{\boldsymbol{\omega}}_i (\dot{\mathbf{q}}^i, \ddot{\mathbf{q}}^i) \in \mathbb{R}^3$ and $\bar{\mathbf{v}}_i (\dot{\mathbf{q}}^i, \ddot{\mathbf{q}}^i) \in \mathbb{R}^3$, can be derived as

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

where \mathbf{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 $\bar{\boldsymbol{\omega}}_i = \bar{\boldsymbol{\Omega}}_i^\vee$ to compute $\bar{\boldsymbol{\Omega}}_i (\dot{\mathbf{q}}^i, \ddot{\mathbf{q}}^i) \in \mathbb{R}^{3 \times 3}$.

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 $\bar{\mathbf{J}}_i^\Omega (\dot{\mathbf{q}}^i) \in \mathbb{R}^{3 \times 9n}$, its Hessian $\bar{\mathbf{H}}_i^\Omega (\dot{\mathbf{q}}^i) \in \mathbb{R}^{9n \times 9n}$, the linear body velocity Jacobian, $\bar{\mathbf{J}}_i^v (\dot{\mathbf{q}}^i) \in \mathbb{R}^{3 \times 3n}$, and its Hessian $\bar{\mathbf{H}}_i^v (\dot{\mathbf{q}}^i) \in \mathbb{R}^{9n \times 3n}$ are provided respectively by Eqs. (4.78), (4.79), (4.80), and (4.81) as

$$\bar{\mathbf{J}}_i^\Omega = \bar{\mathbf{R}}_i^T \left[\mathbf{J}_{i-1}^\Omega \bar{\mathbf{R}}_i \mid \bar{\mathbf{R}}_{i,q_i^T} \right] \quad (4.78)$$

$$\bar{\mathbf{H}}_i^\Omega = \left[\begin{array}{c|c} \bar{\mathbf{R}}_i^T \mathbf{H}_{i-1}^\Omega \bar{\mathbf{R}}_i & \mathbf{0} \\ \hline \bar{\mathbf{R}}_{i,q_i}^T \mathbf{J}_{i-1}^\Omega \bar{\mathbf{R}}_i \cdots \mid \bar{\mathbf{R}}_{i,q_i}^T \bar{\mathbf{R}}_{i,q_i^T} \cdots \\ + \bar{\mathbf{R}}_i^T \mathbf{J}_{i-1}^\Omega \bar{\mathbf{R}}_{i,q_i} \mid + \bar{\mathbf{R}}_i^T \bar{\mathbf{R}}_{i,q_i^T, q_i} \end{array} \right] \quad (4.79)$$

$$\bar{\mathbf{J}}_i^v = \bar{\mathbf{R}}_i^T [\mathbf{J}_{i-1}^v + \mathbf{J}_{i-1}^\Omega \bar{\mathbf{p}}_i | \bar{\mathbf{p}}_{i,q_i^T}] \quad (4.80)$$

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

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, $\mathcal{K}_i^\omega : (\mathbf{q}^i, \dot{\mathbf{q}}^i) \mapsto \mathbb{R}$, is given by

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

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

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

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

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

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

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

$$\begin{aligned}\bar{\mathcal{K}}_i^v &= \frac{1}{2} \bar{\mathbf{v}}_i^T \mathcal{M}_i^v \bar{\mathbf{v}}_i = \frac{1}{2} m_i (\mathbf{v}_{i-1}^T \mathbf{v}_{i-1} + 2\mathbf{v}_{i-1}^T \boldsymbol{\Omega}_{i-1} \bar{\mathbf{p}}_i \cdots \\ &\quad + 2\mathbf{v}_{i-1}^T \dot{\bar{\mathbf{p}}}_i + \bar{\mathbf{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \boldsymbol{\Omega}_{i-1} \bar{\mathbf{p}}_i + 2\bar{\mathbf{p}}_i^T \boldsymbol{\Omega}_{i-1}^T \dot{\bar{\mathbf{p}}}_i + \dot{\bar{\mathbf{p}}}_i^T \dot{\bar{\mathbf{p}}}_i)\end{aligned}\quad (4.85)$$

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

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

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

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

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^v 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 $(\alpha_l, \alpha_r, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \boldsymbol{\Omega}_i - 1) \in \mathbb{R}^{11}$. 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}^{ω} , we separately compute the terms $\mathbb{T}2(\int \mathbf{R}_i^T \boldsymbol{\Omega}_i - 1^T \boldsymbol{\Omega}_i - 1 \mathbf{R}_i)$, $\mathbb{T}2(\int \dot{\mathbf{R}}_i^T \boldsymbol{\Omega}_i - 1 \dot{\mathbf{R}}_i)$, and $\mathbb{T}2(\int \ddot{\mathbf{R}}_i^T \dot{\mathbf{R}}_i)$.

Similarly, for $\bar{\mathcal{K}}^{\omega}$, we compute the terms $\mathbb{T}2(\bar{\mathbf{R}}_i^T \boldsymbol{\Omega}_i - 1^T \boldsymbol{\Omega}_i - 1 \bar{\mathbf{R}}_i)$, $2\mathbb{T}2(\dot{\bar{\mathbf{R}}}_i^T \boldsymbol{\Omega}_i - 1 \dot{\bar{\mathbf{R}}}_i)$, and $\mathbb{T}2(\ddot{\bar{\mathbf{R}}}_i^T \dot{\bar{\mathbf{R}}}_i)$ separately. The energy difference, $\mathcal{K}^{\omega} - \bar{\mathcal{K}}^{\omega}$, 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 $\frac{m_i}{2}$.

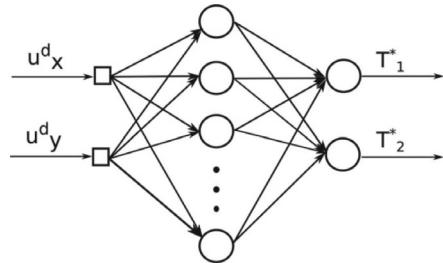
4.4.2.4 Potential Energy of Continuum Sections

As reported in [9], a continuum arm is subjected to gravitational and elastic potential energies. Elastic potential energy, given by $\mathcal{P}^e = \frac{1}{2}\mathbf{q}^T \mathbf{K}_e \mathbf{q}$, only depends on \mathbf{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 $\mathcal{P}_i^g = \int m_i \mathbf{g}^T \mathbf{p}^i$ and $\bar{\mathcal{P}}_i^g = m_i \mathbf{g}^T \bar{\mathbf{p}}^i$ respectively. Note that, \mathcal{P}_i^g does not contain products of integrable terms. Therefore, \mathcal{P}_i^g can be simplified to $\mathcal{P}_i^g = m_i \mathbf{g}^T (\int \mathbf{p}^i)$ and from the definition in Eq. (4.75), then becomes $\mathcal{P}_i^g = m_i \mathbf{g}^T (\bar{\mathbf{p}}^i) = \bar{\mathcal{P}}_i^g$. 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

$$y = g_M(A_M, \dots, g_2(A_2, g_1(A_1, x)) \dots) \quad (4.88)$$

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:

$$y = f(x) + \epsilon \quad (4.89)$$

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:

$$f(x) = \beta_0 + \beta_1 x \quad (4.90)$$

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

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n \quad (4.91)$$

where $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ 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

$$f(x) = \sum_i \alpha_i K(x, x_i) + b \quad (4.92)$$

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:

$$y = f(x) + \epsilon \quad (4.93)$$

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:

$$y_* = f(x_*) + \epsilon_* \quad (4.94)$$

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.

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

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 This 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 multissection continuum arms. *Int J Robot Re* 35(6):695–722

10. Godage IS, Walker ID (2015) Dual quaternion based modal kinematics for multisession continuum arms. In: 2015 IEEE international conference on robotics and automation (ICRA), pp pp 1416–1422. IEEE
11. Godage IS, Webster RJ, Walker ID (2019) Center-of-gravity-based approach for modeling dynamics of multisession 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. (2023) 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, Katzschnmann 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 b閦ier 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-b閦ier 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

Chapter 5

Mechanical Compliance: From Soft Robot Modeling Theory to FEM Computation



Christian Duriez

Abstract This chapter proposes a generic modeling method for soft robots. This method is based on the notion of mechanical compliance, which provides a compact description of the behavior of a deformable robot. In particular, the kinematics of a soft manipulator arm can be derived from this notion. The chapter then shows how this compliance can be calculated from an analysis of the robot's materials and mechanical structure. Starting from continuum mechanics, we describe how the equations can be integrated using the finite element method (FEM). This method is known to be computationally time-consuming, so the chapter briefly introduces the notion of model reduction by projection. Then, using a Lagrangian formulation, actuation, end-effectors, sensors and contacts are introduced which are essential for robot modeling and not often found in FEM formulations. In particular, models for fluidic, tendon-based and articulation-based actuators are detailed. Finally, this chapter describes how to obtain inverse models on soft robots and how to use them to invert kinematics, measure external forces without any force sensor or take contacts into account in the robot model-based control.

5.1 Introduction

This chapter summarizes the work carried out by the DEFROST team these last years on the mechanical modeling of soft robots using the finite element method, computed in real time. The work of the team is constantly implemented in the framework **SOFA** which is open-source, allowing the community to have an implementation of what is described in this chapter. The way in which this chapter has been written is original in relation to the team's previous publications. The emphasis is put on purpose on mechanical compliance, which, is very convenient and generic for soft robot modeling theory.

Soft robots have solid structures and create their motion by deformation. The behavior of the robot depends on the type of material used to build its body. From

C. Duriez (✉)

Inria Research Director, Defrost Team, Inria, University of Lille, CNRS, Centrale Lille, UMR 9189 CRISyAL, 59000 Lille, France

e-mail: christian.duriez@inria.fr

a mechanical point of view, we can also speak of deformable robots, just as we speak of deformable solids in mechanics. The kinematics at the scale of the soft robot as a whole (the relationship between the actuator motions and the robot body motions) depends on mechanics and in particular on the balance between internal forces, external forces, and actuation-related forces.

This is one of the main difference between rigid and deformable robots. At this macroscopic scale, the kinematic relations of the robot are therefore no longer only geometrical as for rigid robots, but derive from the mechanics of the structure, and in particular from its structural mechanical compliance. In this chapter, we will see how the mechanical compliance (the opposite of stiffness), gives the movement created on a structure at given pointsets, called effector or sensor spaces, by the action of forces exerted on it by actuators (actuator space) or contacts (contact space). This compliance, derived from the robot's deformable statics, enables to write the forward kinematics and inverse kinematics of these robots.

The computation of this compliance is obtained from solid body deformation laws of continuum mechanics. Continuum mechanics focus on deformations at a small scale: that of all particles that compose the robot's deformable body. It builds kinematic relations between particle motion and deformation tensors allow to derive physical models, based on the constitutive laws of material. As it includes the material properties, it is an excellent basis for accurate modeling of soft robots. We will very briefly describe the type of equations we obtain for 3D elastic solid bodies and we will try to draw some links with the equations used in robotics. Unfortunately, there is no general analytical solution for these PDE, except in very idealized (and unrealistic) cases.

In numerical mechanics, a variety of numerical methods have been developed to find convergent approximate solutions, and one of the most famous one is the Finite Element Method. Numerical methods are commonly used in engineering to integrate these equations and model deformable structures. The finite element family is one of these numerical methods and is probably the most popular for this purpose. In civil engineering for example, the finite element method is used to assess structure strength and stability. But in these types of applications, there is no special concern of having interactive simulations, running at high frame rate. One of the main challenge in modeling soft robots is to make numerical methods compatible with real-time simulation which is necessary for control. We will also briefly discuss this point, in particular by presenting model order reduction.

Once numerical methods explained for obtaining deformable robot models, we'll see how to model actuators as constraints, using Lagrange multipliers. We'll then extend the constraint-based approach to other spaces of interest in robotics: effectors, sensors and contacts in particular, and we'll see why and how to project mechanical compliance into these different spaces. This will enable us to derive direct and inverse models of deformable soft robots, including in contact situations.

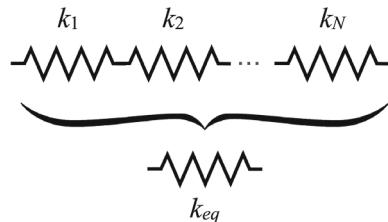
We emphasize that this chapter is still relatively short, and that the reader should not hesitate to refer to other books, particularly on continuum mechanics and FEM. For more detailed information on the modeling of the soft robots presented in this chapter, please refer to the publications and theses of the DEFROST team.

5.2 Modeling Soft Robot Through Mechanical Projected Compliance

In mechanics, compliance is defined as the inverse of stiffness. This concept is best known in mechanics for finding the equivalent spring for springs placed in series. In the 1D case of springs in series shown in the Fig. 5.1, we find the equivalent spring by applying the formula:

$$1/k_{eq} = 1/k_1 + 1/k_2 + \cdots + 1/k_N$$

Fig. 5.1 Use mechanical compliance to find the value of a spring equivalent k_{eq} to springs connected in series



In this series alignment of springs, the interacting forces are the same at each link. But potentially, this force, applied to each end of the springs, creates a particular deformation, depending on the stiffness of the spring. So, in the broadest sense, compliance describes the movement created by a force. On a deformable structure such as a soft robot, compliance is obtained by taking into account the entire structure, integrating the stiffness of the materials in the domain, taking into account boundary conditions and inverting the expression for the tangent stiffness of the entire structure. The expression of compliance is therefore not easy to obtain on soft robots in general. It is often non-linear and depends not only on the overall configuration of the robot, referred to here as \mathbf{q} , but also on the law of material behavior and boundary conditions.

Soft robots are often made of elastic materials (silicone, plastic, rubber, etc.) to which actuating forces are applied, as well as external forces (gravity, contact, etc.). The aim of modeling is to describe the movements created by these forces. In soft robotics, actuation forces are generally linked to an effort parameter (pressure in a cavity, tension in a tendon, torque exerted by a motor, etc.). This effort has a motion dual (cavity volume, tendon length, motor rotation, etc.). Mechanical compliance provides a very compact description of the coupling between these actuating force parameters and the movements they create on the structure at certain specific points.

We'll be talking about projected compliance, since we'll be looking at the value of this compliance in the actuator/effectuator/sensor subspaces, but also using subspaces linked to contact when the robot comes into contact with its environment.

The robot diagram below schematically describes the notion of projected compliance in the actuator/effectuator spaces and the link with the kinematics on a deformable robot.

In the spirit of virtual work, projected compliance allows us to describe a relationship between notions of variation in displacement $\Delta\delta$ created by increments in force on actuators $\Delta\lambda_a$ (see Fig. 5.2). This relationship is linearized around the current configuration for small force increments. When observing motion from effector points, we use $\Delta\delta_e = \mathbf{W}_{ea}(\mathbf{q})\Delta\lambda_a$. Thanks to compliance projected into the effector/actuator space, we have here a compact expression for the variation in motion on the effectors created by an increment of forces on the actuators. Similarly, we can use projected compliance to describe the motion, in actuator space, created by this increment of actuator forces (forces are often coupled by the mechanics): $\Delta\delta_a = \mathbf{W}_{aa}(\mathbf{q})\Delta\lambda_a$. Here, compliance is projected on both sides (force/movement) into actuator space.

So, if we have the expression of $\mathbf{W}_{aa}(\mathbf{q})$ and $\mathbf{W}_{ea}(\mathbf{q})$ for the current robot configuration \mathbf{q} , we can describe the **kinematic model of soft manipulators**, by the form:

$$\Delta\delta_e = \mathbf{W}_{ea}(\mathbf{q})\mathbf{W}_{aa}^{-1}(\mathbf{q})\Delta\delta_a = \mathbf{J}_{SR}(\mathbf{q})\Delta\delta_a \quad (5.1)$$

This expression gives the variation in movement created on the end-effector, by a variation in movement on the actuator. Thanks to projected compliance, we have a general expression for the Jacobian for soft robots $\mathbf{J}_{SR}(\mathbf{q}) = \mathbf{W}_{ea}(\mathbf{q})\mathbf{W}_{aa}^{-1}(\mathbf{q})$. This

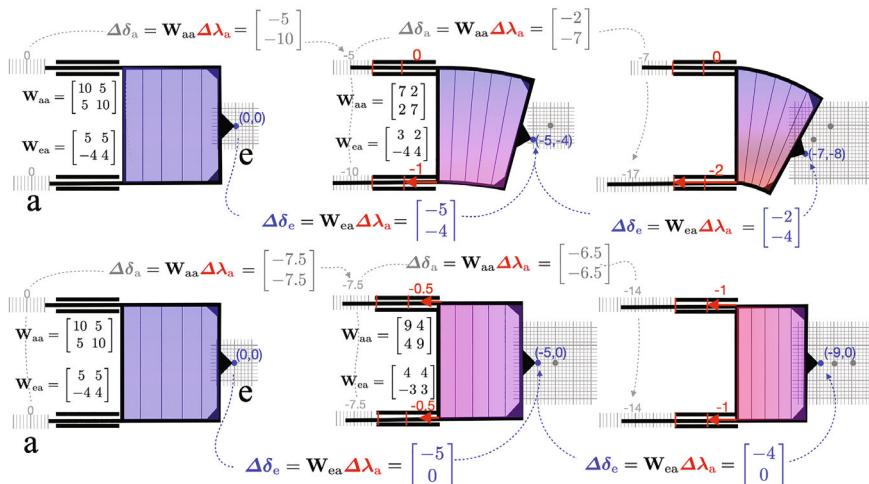


Fig. 5.2 Kinematics through the mechanical compliance of soft robots: To describe the kinematics of a soft robot, the mechanical compliance of the manipulator structure is projected into actuator space with \mathbf{W}_{aa} and into effector/actuator space with \mathbf{W}_{ea} . Thanks to these two operators, we can obtain the displacements of the actuators ($\Delta\delta_a$) and of the effector ($\Delta\delta_e$) when increments of forces are applied to the actuators ($\Delta\lambda_a$). To illustrate, the figure shows a dummy soft robot, actuated by two linear pistons. The robot's behavior is decomposed in three steps with two different sets of incremental actuator forces on the top and bottom lines. We show that the value of compliances is not constant due to non-linear deformations. In this example, the kinematics of the soft robot can be calculated using the Eq. 5.1

Jacobian has similar characteristics of the Jacobians used on rigid manipulators, in particular the loss of rank when there is a singularity.

In the case of the dummy robot presented in Fig. 5.2, the value of the jacobian at the starting position is:

$$\mathbf{J}_{SR}(\mathbf{q}) = \mathbf{W}_{ea}(\mathbf{q})\mathbf{W}_{aa}^{-1}(\mathbf{q}) = \begin{bmatrix} 5 & 5 \\ -4 & 4 \end{bmatrix} \begin{bmatrix} 10 & 5 \\ 5 & 10 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{5}{15} & \frac{5}{15} \\ \frac{-12}{15} & \frac{12}{15} \end{bmatrix}$$

If we apply the first actuator displacement increments (see the passage from column 1 to column 2 in the Fig. 5.2), we obtain the corresponding displacement of the effector:

$$\underbrace{\begin{bmatrix} \frac{5}{15} & \frac{5}{15} \\ \frac{-12}{15} & \frac{12}{15} \end{bmatrix}}_{\mathbf{J}_{SR}} \underbrace{\begin{bmatrix} -5 \\ -10 \end{bmatrix}}_{\Delta\delta_a} = \underbrace{\begin{bmatrix} -5 \\ -4 \end{bmatrix}}_{\Delta\delta_e} \quad \text{and} \quad \underbrace{\begin{bmatrix} \frac{5}{15} & \frac{5}{15} \\ \frac{-12}{15} & \frac{12}{15} \end{bmatrix}}_{\mathbf{J}_{SR}} \underbrace{\begin{bmatrix} -7.5 \\ -7.5 \end{bmatrix}}_{\Delta\delta_a} = \underbrace{\begin{bmatrix} -5 \\ 0 \end{bmatrix}}_{\Delta\delta_e}$$

Then, when the robot deforms, the Jacobian matrices (and the matrices of projected compliance) are changing due to deformation non-linearities (see matrices in column 2 in the Fig. 5.2).

Note that the formula requires the matrix \mathbf{W}_{aa} to be invertible. This matrix represents the mechanical coupling between actuators due to the deformable structure. In practice, we often try to decouple the actuators as much as possible in the design so this matrix is usually invertible. Given the high number of degrees of freedom on the deformable structure, it would be surprising, from a design point of view, to couple by actuation the same degrees of freedom in the same directions.

The compliance allow to describe the kinematics of soft deformable robots. We'll see in Sect. 3 how this compliance is obtained for different types of actuators, and all the uses we can make of it, from robot kinematics, inverse kinematics, force estimation, motor/sensor coupling... But before going into detail, let's return to the original problem: how can we model and compute the projected compliance $\mathbf{W}_{ij}(\mathbf{q})$?

The remainder of the chapter will explain how this projected compliance can be obtained using FEM models, in particular to be able to use material behavior laws, and take account of the strong non-linearities in these models.

5.3 Notions of Continuum Mechanics

The aim of this section on continuum mechanics and numerical methods is to give a few basic notions. For more detailed information, we recommend the use of reference books such as [8, 9].

5.3.1 Continuum Mechanics of Elastic Solids

First of all, as its name suggests, continuum mechanics considers solids to be filled by a continuum of material points. It's therefore a relatively macroscopic vision of mechanics, at a scale where the medium can be considered continuous (we're not at the scale of atoms, which make up matter in a discrete way).

5.3.1.1 Motion Tracking

First of all, as in rigid robotics, the starting point is the parametrization of the motion. In continuum mechanics, motion is traditionally described using two different views, the so-called Lagrangian view and the so-called Eulerian view. Considering that most soft robots use elastic materials, the notion of a reference domain makes sense. This reference domain corresponds to the reached position by the elastic structure when no force is applied to it.

In the Lagrangian view, all particles defined on the domain are tracked from their position in the reference configuration to the current, deformed configuration. In this approach, many calculations are performed in the reference configuration (this allows a certain number of pre-calculations to be performed if this configuration remains unchanged).

In practice, we can define an application \mathbb{A} which tracks the motion. For any point $\mathbf{p} = [x, y, z]$ defined on the domain \mathbb{A} gives its corresponding unique point $\mathbf{p}' = [x', y', z'] = \mathbb{A}(\mathbf{p})$ in the deformed domain (see Fig. 5.3).

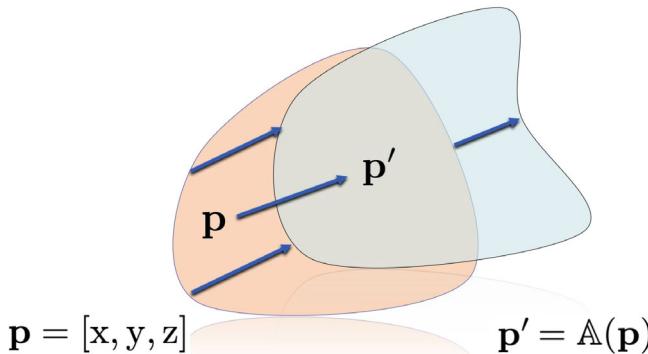


Fig. 5.3 Lagrangian view: for any point \mathbf{p} in the reference domain, the application $\mathbb{A}(\mathbf{p})$ provides \mathbf{p}' the unique corresponding point in the deformed domain

5.3.1.2 Strain Tensors

Once this “motion tracking” has been set up for any point in the domain, the deformation is measured using the spatial derivation of this \mathbb{A} function.

In this way, we can look at the variation in motion of infinitely close points (see Fig. 5.4) in the deformed shape, compared with the initial shape.

$$d\mathbf{p}' = \frac{\partial \mathbb{A}}{\partial \mathbf{p}} d\mathbf{p} = \mathbf{F} d\mathbf{p} \quad (5.2)$$

In this equation $\mathbf{F} = \frac{\partial \mathbb{A}}{\partial \mathbf{p}}$ is the deformation gradient, which is arguably the simplest tensor to geometrically describe a differential of particle motion, and therefore potentially a deformation. Thus, if $\mathbf{F} = \mathbf{Id}$ (\mathbf{Id} being the identity), at any point in the domain, then the body undergoes no deformation, but is potentially animated by a uniform rigid translation.

This tensor is said to be “translation invariant”. To have a zero value when in pure translation, we can write this tensor with the gradient of displacements. We then define $\mathbb{A} = \mathbf{Id} + \mathbb{U}$, \mathbb{U} being the displacement field of any point \mathbf{p} of the domain to its corresponding deformed \mathbf{p}' . In this case, the deformation gradient can be re-written as

$$\mathbf{F} = \mathbf{Id} + \mathbf{grad}(\mathbb{U}) \quad (5.3)$$

However, the value of \mathbf{F} can be influenced by movements other than deformation. For example, a rigid rotation will generate a value of $\mathbf{F} \neq \mathbf{Id}$. So we’ll be looking to construct tensors that are independent of this rigid rotation. To do this, we can perform a polar decomposition of the tensor $\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}$ where \mathbf{U} and \mathbf{V} are symmetric tensors and \mathbf{R} is an anti-symmetric and proper orthogonal tensor assimilated to a rotation matrix. $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$. This is illustrated in 2D on Fig. 5.5.

Comment: Note that to compute a 3D rotation from a polar decomposition of \mathbf{F} , the tensor must be defined along the 3 directions. For slender domains like curve or surfaces, used for rods or shells, the rotation cannot be computed that way as there are some directions of non deformation (the section in rods, the thickness in shells). This is why, in the literature, the **Cosserat theory** was developed with the use of a

Fig. 5.4 Illustration of the deformation gradient

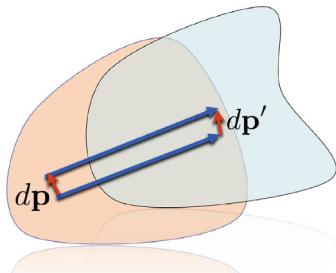
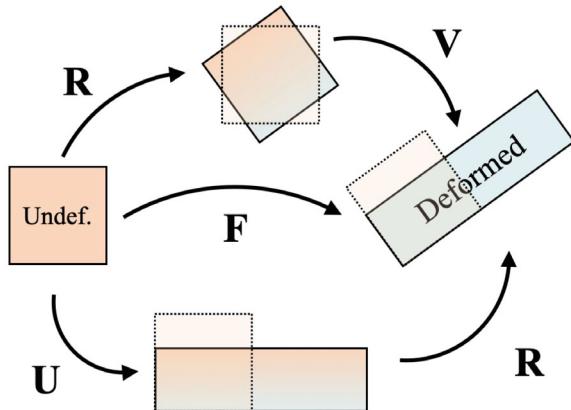


Fig. 5.5 Right and Left polar decomposition of the deformation gradient



definition of domains full of infinitely small rigid bodies tracked both in translation and rotation.

We can define two strain tensors: the right Cauchy-Green tensor:

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} = (\mathbf{R}\mathbf{U})^T (\mathbf{R}\mathbf{U}) = \mathbf{U}^T \mathbf{R}^T \mathbf{R} \mathbf{U} = \mathbf{U}^2 \quad (5.4)$$

and the left Cauchy-Green tensor:

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T = (\mathbf{V}\mathbf{R})(\mathbf{V}\mathbf{R})^T = \mathbf{V}\mathbf{R}\mathbf{R}^T \mathbf{V}^T = \mathbf{V}^2 \quad (5.5)$$

In practice, in the context of elasticity or hyperelasticity, the right Cauchy-Green tensor is more commonly used, as strain is analyzed in the undeformed domain.

This tensor is used to write another widely-used strain tensor, the Green-Lagrange tensor $\mathbf{E} = \mathbf{C} - \mathbf{Id}$, which can also be written as the square norm of the difference between $d\mathbf{p}$ and $d\mathbf{p}'$:

$$\frac{1}{2}(\partial\mathbf{p}'^2 - \partial\mathbf{p}^2) = \frac{1}{2}(\partial\mathbf{p}^T \mathbf{F}^T \mathbf{F} \partial\mathbf{p} - \partial\mathbf{p}^2) = \partial\mathbf{p}^T \underbrace{\frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{Id})}_{\mathbf{E}} \partial\mathbf{p} \quad (5.6)$$

Following this definition $\mathbf{E} = \frac{1}{2}(\mathbf{U} - \mathbf{Id})$. When we write the deformation gradient $\mathbf{F} = \mathbf{Id} + \mathbf{grad}(\mathbb{U})$, like in Eq. (5.3), we obtain an usual formulation of the Green-Lagrange tensor:

$$\mathbf{E} = \frac{1}{2}(\mathbf{grad}(\mathbb{U}) + \mathbf{grad}(\mathbb{U})^T + \mathbf{grad}(\mathbb{U})^T \mathbf{grad}(\mathbb{U})) \quad (5.7)$$

When the displacement field \mathbb{U} is small, one can use the linear part of this deformation tensor:

$$\boldsymbol{\epsilon} = \frac{1}{2}(\mathbf{grad}(\mathbb{U}) + \mathbf{grad}(\mathbb{U})^T) \quad (5.8)$$

Comment: In a corotational formulation, one can use the rotation \mathbf{R} computed for the polar decomposition of \mathbf{F} to build a rigid transformation around each point (translation and rotation). Then, the displacement field \mathbb{U} of the neighbourhood of this point can be computed in a local frame. In general, this allows to use the small displacement tensor $\boldsymbol{\epsilon}$ for quite large rotations but locally small strains. But note that the corotational formulation can also be used for large strains using \mathbf{E} .

5.3.1.3 Stress Tensors

So far, all manipulations have been purely geometric. Strain tensors provide an idea of the geometric deformation of the solid. But depending on the material's behavior, this deformation will create stresses in the domain, i.e. internal forces, which we will also describe with tensors.

The Cauchy tensor $\boldsymbol{\sigma}(\mathbf{p}')$ describes the state of stress at any point on the object and in all directions within the deformed shape. Its components are homogeneous with pressure (in Pa), since they correspond to a force exerted on a unit area around a point ($N\ m^{-2}$). This tensor is symmetrical with $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$. The diagonal terms of the stress tensor correspond to tension/compression and the non-diagonal terms to shear.

At the boundary of the domain, these internal stresses balance out with the pressure forces $\vec{\mathbf{f}}$ exerted at the surface as illustrated in Fig. 5.6

$$\vec{\mathbf{f}} = \boldsymbol{\sigma} \cdot \vec{\mathbf{n}} \quad (5.9)$$

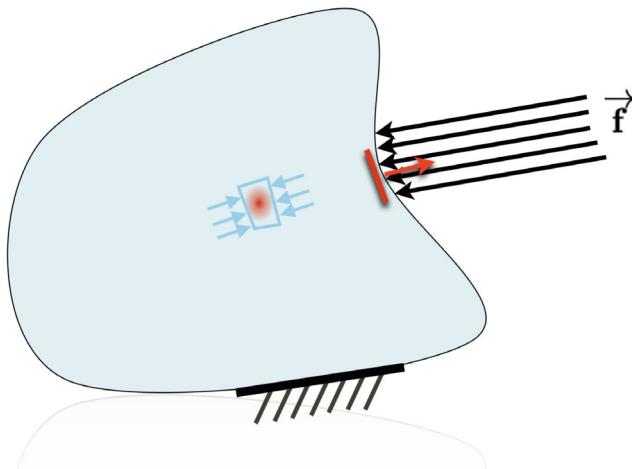


Fig. 5.6 Stress corresponds to the internal force exerted on unit areas around the points of the domain. It has to be balanced with the forces at the boundary

Comment: The Cauchy tensor σ is therefore defined in the deformed form, on which the forces that creates the deformation are exerted. But as stated before we have defined the strain in the reference configuration. So, two additional stress tensors can be used. The First Piola Kirchhoff boldsymbol σ_{PK1} for which the strain is defined in the reference domain and the stress in the deformed configuration. And the Second Piola Kirchhoff stress tensor σ_{PK2} , for which the stress is defined in the reference configuration.

$$\sigma_{PK2} = J\mathbf{F}^{-1}\sigma\mathbf{F}^{-T} = \mathbf{F}^{-1}\sigma_{PK1} \quad (5.10)$$

5.3.1.4 Constitutive Laws

The stress being defined, the constitutive law will be used to characterize the mechanical behavior of the material. This law provides a relationship between strain and stress: given its properties the material will generate internal stress when deforming. In the context of this chapter, we will not detail all the possible constitutive laws but just provide a quick overview.

- **Hooke's law** Adapted to linear elastic deformations, it is probably the most simple constitutive law: it is considered that the stress is linearly proportional to the strain. As an elastic law, when there is no effort applied to the deformable body, there is no deformation. Note that this linear behavior is quite frequently observed at relatively low strain levels, and sometimes even with large displacements, i.e. with a non-linear strain tensor to take into account the rigid rotations. Many models (such as the Cosserat beam models widely used in the continuous robot literature) use an exact calculation of the geometric part of the strain, but employ a linear behavior law. In a fully 3D formulation (without the assumption of rods or shells), a linear relationship between the stress and strain tensors can be written using a 4th-order tensor. In practice, with Voigt's notations (stress and strain tensors in vectors of size 6, taking advantage of symmetries) we can write this relationship in matrix form, with a 6×6 matrix \mathbf{C} .

$$\sigma = \mathbf{C}\mathbf{E} \quad (5.11)$$

If the material is homogeneous and isotropic, two parameters are used by the behavior law: Young's modulus gives the material's level of stiffness and the Poisson ratio gives its level of compressibility. It should be noted that a totally incompressible material cannot be modeled with such a law: this would mean setting the Poisson ratio at 0.5, but the law diverges at such a level, and numerically this leads to a "locking" of deformations. To model complete incompressibility, numerical techniques are used (addition of a pressure unknown in the FEM formulation, use of Lagrange multipliers, etc.), which creates additional difficulties [1].

- **Anisotropy:** Some work demonstrated the interest of using anisotropic material to influence the kinematics of soft robots [2]. We can model anisotropic constitutive

law by enriching Hooke's law with directions in which the stiffnesses are not the same. In this case, the behavior law is enriched with the use of several Young's Modules and a definition of the direction of the anisotropies [3].

- **Hyperelasticity.** Some materials, such as rubbers, silicones and other elastomers [4, 5], widely used in soft robotics, allow very high levels of deformation, while maintaining an elastic behavior (no residual deformations when the loading cancels). But this behavior becomes non-linear and is called hyperelastic. In practice, to calculate stress, an energy density function W is used. This function will depend on the strain, and the behavior law will be derived from this relationship:

$$\sigma = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} \quad (5.12)$$

Of course, we need to define the function W . Here, the behavior is no longer necessarily linear. Several models exist: St Venant Kirchhoff (linear model), Arruda-Boyce, Mooney-Rivling, Ogden etc. The formulations of these models are often based on the invariants of the right-hand Cauchy Green tensor. These models sometimes have many parameters that are difficult to find experimentally. However, there is a database for materials often used in soft robotics, which provides parameter fittings for the most common behavior laws [6].

- **Other behaviors** Other behaviors can be modeled, such as visco-elasticity [7]. Plasticity and fatigue can also be used but usually they require to keep the loading history.

5.3.1.5 Variational Formulation

The behavior of the internal forces has been modeled as depending on the deformation of the medium and the stress generated by the behavior law. Continuum mechanics proposes to describe the dynamics of each point of the medium using the equation:

$$\rho \ddot{\mathbf{p}} = \operatorname{div}(\boldsymbol{\sigma}) + f_{ext} \quad (5.13)$$

where ρ represents the mass density, $\ddot{\mathbf{p}}$ the acceleration of the particles, $\operatorname{div}(\boldsymbol{\sigma})$ provides the internal force acting on each particle which are balanced with the external forces f_{ext} (gravity or boundary forces from the environment). If some parts of the soft robot are fixed, $d\mathbf{p} = 0$ can be imposed as a boundary condition of the deformable domain.

With this equation, the problem is written in the **strong form** for each point (and there is an infinite number) of the domain. In most of the 3D deformation cases, there is no analytical solution to that equation and that would provide the behavior of each point of the domain. For simplified domains, like beams or shells more analytical solution exist but there are still often very specific. Numerical methods are therefore used to solve the problem. In the case of deformable solid mechanics, methods based a weak form of the problem (which include finite elements) are particularly effective.

Here again, numerous works provide details on the mathematical way of obtaining the weak formulation and the foundations of the associated numerical methods. We can quickly draw a parallel of this principle with tools used in rigid robotics: Assuming that the strain-stress relationship is non-linear, we can write the variation of the density of the strain-energy over the entire volume v of the structure using Eq. 5.12 to obtain the weak form:

$$\partial\mathcal{W} = \int_v \partial W \, dV = \int_v \boldsymbol{\sigma}^T \partial \mathbf{E} \, dV \quad (5.14)$$

1. We find the variation of energy by summing the work of each point in the domain. We draw a first parallel between this equation and the **virtual work** used in rigid robot models: we could suppose that there is a joint at each point of the deformable robot. In the space of each articulation, the motion (\approx the displacement) would be represented by the strain and the effort (\approx the force) would be represented by the stress. Here we use Voigt notation to write the strain and stress (in particular the dot product $\boldsymbol{\sigma}^T \partial \mathbf{E}$). *Note that many approaches using Cosserat's theory explicitly use strain-stress space as the joint space of their model. Once the motion has been solved in this space, it can then be traced back to the motions using a method of integration along the curve, which would be far more complex to do in a surface or volume domain.*
2. By combining this integral form with Eq. 5.13, we can also obtain volume terms (notably kinetic energy) and equilibrium with surface terms (virtual work of surface forces, see in Eq. 5.9). So the second parallel we can make with classical modeling in robotics, is that this energetic approach brings to **Lagrangian mechanics**, (also widely used for articulated rigid structures) and that will be used in Sect. 5.6 to integrate actuators, sensors, effectors and also contacts as constraints.

5.4 Computing Soft Robot Configuration and Compliance from Finite Element Modeling

As in the previous section, the aim is not to describe the finite element method of deformable structures in detail. The reader is encouraged to read some books to improve the knowledge on such methods [8, 9].

The aim is to understand how this method can be used to model a soft robot, in particular to find its configuration as a function of force fields (actuators, internal forces, external forces, etc.) and to calculate the projected compliance of soft robots, which, as we have introduced in the first section, is a particularly useful tool for describing the motion behavior of soft robots.

To understand the usefulness of finite element methods for elastic solids, we need to go back to continuum mechanics and realize that the equations presented in the

previous section have no analytical solution, except for simplified cases (simplified geometries, constitutive laws and loading).

5.4.1 Discussion About Dimensionalities

In terms of continuum mechanics, we can distinguish between the space in which the points move (usually 2D or 3D, but in robotics we're using 3D) and the size of the objects: a very slender object (or robot) with a length that is very large compared to its thickness in the other two dimensions can use simplified rod theory.

String, beams, rods: The simplest model will be the stretched soft string, which only deforms along its direction of length. Then, there are several beam theories, which makes the simplifying assumption that the cross-section of the long object is not deformable. Euler-Bernoulli, Kirchhoff and Timoshenko theories fall into this category. Cosserat rod theory obtains an exact calculation of the strain. For all, it is usually supposed that the material is homogeneous, isotropic and obeying Hooke's law. From a robotics point of view, the model used for "continuum robots" generally fall into this category.

Membranes, plates, shells: An object (or robot) with a large surface area relative to its thickness can also be analysed under simplified theory. Membranes, plates and shells are often analyzed by assumption that the thickness of the structure remains constant under loading. In the membrane case strains are small and are limited to in-plane components. The thickness of the structure is much smaller compared to its other dimensions. Plate models include bending moments but transverse shear effects are neglected. In shell theory the transverse shear is included. The thickness of the shell is small compared to its other dimensions, but not necessarily as small as in membranes or plates. Again for all these models, the material is often assumed to be homogeneous, isotropic, and obeying Hooke's law.

Volume deformations: Finally, if the object (or robot) is not significantly smaller in size than the others, then we use the general theory outlined above. As there is no assumption about the strain (the kinematics of the deformation), any constitutive law can be used in this context.

5.4.2 Elements: Geometry, Topology, Interpolation and Nodes

These models, derived from continuum theory, can be computed numerically using finite element modeling, but the types of elements used will differ. Elements are defined by their topology in the sense that they will connect some nodes. As these nodes are defined in space, it also defines a geometry. Finally, we will assume that

any point of the elements has its kinematics linked to the kinematics of the nodes through the interpolation.

Figure 5.7 provides some examples of elements and geometrical role and support interpolation. To integrate beam theory (Bernoulli, Kirchhoff, Cosserat, ...), segments or curves are used. Interpolation functions are based on the curvilinear abscissa, and often have a polynomial form. Nodes may have more degree of freedom than just their translation. Some rotations can also be included in the formulation. For instance we can have frame nodes with 3 translations and 3 rotations on beam elements. For surface elements, this is the same approach: In plate or shell elements, some degrees of freedom in rotations are included in the formulation of the interpolation over the element. For volume element, usually, the degrees of freedom of nodes are only in translation but depending on the polynomial interpolation, the number of nodes can vary.

In all cases, these degrees of freedom of the elements are noted in a vector \mathbf{x} in the following.

Interpolation functions are used to describe the behavior of the position (or displacement) and velocity field of all the points inside an element as a function of the positions (or displacements) and velocities of the nodes. These functions are often polynomials but many other approaches exist. By meshing the domain, i.e. the shape of the deformable object (in this case, the robot), we obtain a position and velocity field that is continuous over the entire domain, but whose motion is parameterized by the motion of the nodes. The mesh topology used to perform the FEM calculation will give the connections between the nodes, which are linked by elements. Some methods (called meshless methods) allow to compute these connections without a mesh, by neighborhood. In such case, the support of the interpolation is not defined

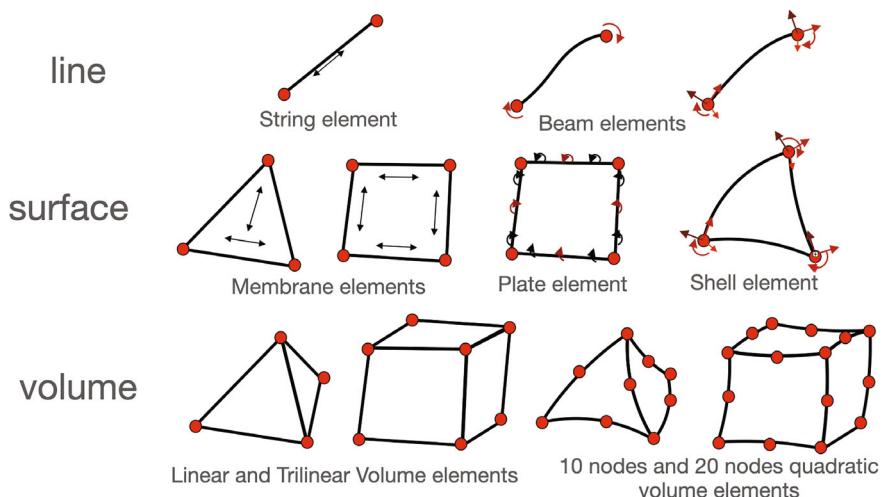


Fig. 5.7 Examples of elements used in FEM

by an element shape but by the function itself (this is not detailed here, see [10] for more information).

5.4.3 Internal Forces Computation and Assembling

At the level of each element, we can therefore integrate the laws of mechanics using a variational formulation, integrating the interpolation defined on the element. This formulation enables the computation of the variation of the deformation energy as a function of the movement of the nodes.

The finite element interpolation is defined on the mesh which discretizes the 3D domain \mathcal{D} on which we compute the deformations of the robot (in general the domain \mathcal{D} corresponds to its shape). Let us consider that the mesh connects a set of \mathcal{N} nodes. For any particle \mathbf{p} in the domain \mathcal{D} , the motion is interpolated using the node motion.

$$\mathbf{p}(x, y, z) = \sum_{i \in \mathcal{N}} N^i(x, y, z) \mathbf{x}^i$$

where $N^i(x, y, z)$ are piecewise polynomials with compact support. These polynomials are associated to nodes and their support is usually defined on one element. The spacial derivatives can now be computed using the interpolation, for instance:

$$\frac{\partial \mathbf{p}}{\partial x} = \sum \frac{\partial N^i(x, y, z)}{\partial x} \mathbf{x}^i ; \quad \frac{\partial \mathbf{p}}{\partial y} = \sum \frac{\partial N^i(x, y, z)}{\partial y} \mathbf{x}^i ; \quad \frac{\partial \mathbf{p}}{\partial z} = \sum \frac{\partial N^i(x, y, z)}{\partial z} \mathbf{x}^i$$

With a full Lagrangian approach, interpolations and derivatives are performed in the form at rest, enabling precalculations to be made. Thanks to the fact that the interpolation functions have a compact support, we can calculate the value of the strain and stress tensors element by element. Internal strain energy from Eq. (5.15) can be approximated by summing over each elements e :

$$\partial \mathcal{W}(\mathbf{x}) = \int_D \boldsymbol{\sigma}^T \partial \mathbf{E} dV \approx \sum \int_e \boldsymbol{\sigma}^T(\mathbf{x}) \partial \mathbf{E}(\mathbf{x}) dV \quad (5.15)$$

The internal forces $\mathbf{F}(\mathbf{x}) = \frac{\partial \mathcal{W}(\mathbf{x})}{\partial \mathbf{x}}$ are obtained by the derivative of this deformation potential energy. Again, we can make a parallel with the principle of virtual work, as the work created by the nodes' movements need to be equal to the variation of the strain energy: $\partial \mathcal{W}(\mathbf{x}) = \mathbf{F}(\mathbf{x})^T \partial \mathbf{x}$.

By deriving the expression a second time, the tangent stiffness can be found at the level of each element. But the stiffnesses and internal forces calculated for each element add up at the nodes. We therefore assemble these internal forces and matrices according to the topology and numbering of the nodes. The result is the internal forces and the internal stiffness of the structure. This structural stiffness serves as the basis for calculating compliance (inverse of stiffness).

5.5 Equations of Motion

Based on the FEM (or equivalent) formulation described above, the material behavior law is now integrated across all mesh elements. In this way, at the scale of a deformable robot, the internal forces created by the deformation of the robot structure can be recovered. The “structural” aspect is important, as the mechanical behavior of a soft robot depends on both the material used and the design of its hole structure. For a numerical model of an elastic or hyperelastic robot, the internal forces depend mainly on the position of the nodes, and sometimes on the velocity (viscoelastic case).

Solving the equations of motion on a FEM model of the soft robot involves calculating the positions and velocities of the mesh nodes. Depending on the soft robot and the application, quasi-static or dynamic approaches can be used. Dynamic effects on soft robots, such as vibrations, are often unwanted and avoided. In practice, in particular for medical applications, actuations of soft robots are usually done at low velocities. In such cases, the inertia forces vanish. However if the robot has no fixed point or no rigid base, the dynamic approach is mandatory (for locomotion for instance).

5.5.1 Quasi-Static Motion

The conditions to use a quasi-static approach are twofold: (1) some nodes of the FEM model of the robot are fixed or attached to a rigid part (2) the velocities or the mass/stiffness ratio are sufficiently low so that inertial terms vanish. For instance, at micro scale the mass decreases much faster than stiffness. For micro soft manipulators, even at high speeds, mass can be neglected [11].

The configuration of the robot at a given time is then obtained by solving the static equilibrium between the internal forces computed by the FEM model $\mathbf{F}(\mathbf{x})$ (where \mathbf{x} is the position vector of the FE nodes) and the external loads \mathbf{F}_{ext} , including the efforts exerted by the actuators. In some case, we can add the forces created by the gravity \mathbf{Mg} . Note that in FEM, using *mass lumping* on volume elements can lead to constant mass matrices that do not depends on the position of the nodes. It is not the case with beam FEM elements for instance.

$$\mathbf{F}(\mathbf{x}) + \mathbf{Mg} + \mathbf{F}_{ext} = \mathbf{0} \quad (5.16)$$

In the general case, the internal forces are a non-linear function. At each step i of the simulation, we compute a linearization of $\mathbf{F}(\mathbf{x})$ by using a Taylor series expansion, leading to this first order approximation:

$$\mathbf{F}(\mathbf{x}_i) \approx \mathbf{F}(\mathbf{x}_{i-1}) + \frac{\partial \mathbf{F}(\mathbf{x}_{i-1})}{\partial \mathbf{x}} d\mathbf{x} = \mathbf{F}(\mathbf{x}_{i-1}) + \mathbf{K}(\mathbf{x}_{i-1}) d\mathbf{x} \quad (5.17)$$

where $d\mathbf{x} = \mathbf{x}_i - \mathbf{x}_{i-1}$ and \mathbf{K} is the tangent stiffness matrix mentioned in Sect. 5.4.3 that depends on the current position of the FE nodes. This matrix is obtained by assembling the contributions of each element of the FEM model.

Combined with Eq. 5.16, it provides a linear matrix system $\mathbf{Ax} = \mathbf{b}$ to solve at each time stem:

$$\underbrace{-\mathbf{K}(\mathbf{x}_{i-1})}_{\mathbf{A}} \underbrace{d\mathbf{x}}_{\mathbf{x}} = \underbrace{\mathbf{F}(\mathbf{x}_{i-1}) + \mathbf{Mg} + \mathbf{F}_{ext}}_{\mathbf{b}} \quad (5.18)$$

At each step, the matrix $\mathbf{K}(\mathbf{x}_{i-1})$ is updated with the position of the nodes of the previous step. By solving this linear equation, a new value of $d\mathbf{x}$ can be obtained and the solution converges when $\mathbf{x}_i = \mathbf{x}_{i-1}$. This can be considered as a Newton-Raphson iterative method, in particular if \mathbf{Mg} and \mathbf{F}_{ext} are constant and non dependant on \mathbf{x} . In the next section (Sect. 5.6), we'll look at how to integrate actuator forces, which are not constant given forces but depend on the structure's internal forces.

Note that the matrix \mathbf{A} is usually highly sparse and positive. However it is rank deficient: the rank of the matrix corresponds rank = $s - 6$ with s being the size of the matrix. The 6 redundant equations corresponds to the rigid degrees of freedom. To inverse the matrix, we need to remove (at least) these 6 degrees of freedom. In practice we often fix all the nodes that are attached to a static support. One possible numerical technique is to remove the lines and column corresponding to these nodes from the equilibrium equation, considering that $d\mathbf{x} = 0$ for these nodes. An other technique (in particular when the rigid support is moving) is to use Lagrange multipliers, introduced in Sect. 5.6.

5.5.2 Dynamic Motion

To model the dynamics of a deformable robot using FEM, we rely on the second law of Newton:

$$\mathbf{M}(\mathbf{x})\mathbf{a} = \mathbf{F}(\mathbf{x}, \mathbf{v}) + \mathbf{M}(\mathbf{x})\mathbf{g} + \mathbf{F}_{ext} \quad (5.19)$$

The mechanical states is given by \mathbf{x} the position and \mathbf{v} the velocities of the FEM mesh nodes. With the general term $\mathbf{F}(\mathbf{x}, \mathbf{v})$, the computation of the internal forces can also include viscosity effects. The vector \mathbf{a} is the acceleration of these nodes. The mass matrix $\mathbf{M}(\mathbf{x})$ is always positive definite. The mass may be a function of the node position, but when using mass lumping, the matrix is constant and diagonal. In the following we will use simply \mathbf{M} .

The deformable dynamics evolves in time and need to be integrated. For that we use a time numerical scheme. The choice of this scheme has a big influence on the results:

- **With explicit schemes**, at each time step, a new value of the acceleration \mathbf{a}^+ can be obtained by inverting the mass matrix. The current position \mathbf{x}^- and velocity \mathbf{v}^- of

the nodes are computed using values of the previous steps. After the computation of the new acceleration \mathbf{a}^+ , the new velocities \mathbf{v}^+ and positions \mathbf{x}^+ can be obtained by applying the explicit time integration scheme (using a time step h).

$$\begin{cases} \mathbf{a}^+ = \mathbf{M}^{-1}(\mathbf{F}(\mathbf{x}^-, \mathbf{v}^-) + \mathbf{Mg} + \mathbf{F}_{ext}) \\ \mathbf{v}^+ = f(\mathbf{a}^+, \mathbf{v}^-, h) \\ \mathbf{x}^+ = f(\mathbf{a}^+, \mathbf{v}^-, \mathbf{x}^-, h) \end{cases} \quad (5.20)$$

With explicit scheme, the computation of the non linear internal forces are quite easy to obtain. And combined with mass lumping (with a diagonal matrix \mathbf{M}) the computation of a new acceleration is straightforward and can be easily computed in parallel. However, this method is **conditionally stable**: h should be sufficiently small, in particular when the size of the elements is small (detail mesh) and mass/stiffness ratio is low.

- With **implicit schemes**, the model is **unconditionally stable**, however the system is more difficult to solve because the current position \mathbf{x}^+ and velocity \mathbf{v}^+ of the nodes used in the dynamic system depends on the current acceleration \mathbf{a}^+ . A non linear system has to be solved which takes much more time, but the time step h can be chosen freely.

$$\begin{cases} \mathbf{Ma}^+ = \mathbf{F}(\mathbf{x}^+, \mathbf{v}^+) + \mathbf{Mg} + \mathbf{F}_{ext} \\ \mathbf{v}^+ = f(\mathbf{a}^+, \mathbf{v}^-, h) \\ \mathbf{x}^+ = f(\mathbf{a}^+, \mathbf{v}^+, \mathbf{x}^-, h) \end{cases} \quad (5.21)$$

Both strategies have their advantages and drawbacks. In the following, we have developed the method for implicit scheme, in particular to introduce more easily the non-smooth mechanics of contacts, as proposed in [12]. Indeed, when a contact occurs on a FEM node, the velocity of this node instantaneously changes and the acceleration is not defined. We will then use a low order time stepping scheme (backward Euler) which replaces the acceleration by a change of velocity $d\mathbf{v}$, and forces by impulses (forces multiplied by the time step):

$$\begin{cases} \mathbf{Md}\mathbf{v} = h\mathbf{F}(\mathbf{x}_i, \mathbf{v}_i) + h\mathbf{Mg} + h\mathbf{F}_{ext} \\ d\mathbf{v} = \mathbf{v}_i - \mathbf{v}_{i-1} \\ \mathbf{x}_i = \mathbf{x}_{i-1} + h\mathbf{v}_i \end{cases} \quad (5.22)$$

We can used the Taylor series expansion, leading to the following first order approximation:

$$\mathbf{F}(\mathbf{x}_i, \mathbf{v}_i) \approx \mathbf{F}(\mathbf{x}_{i-1}, \mathbf{v}_{i-1}) + \underbrace{\frac{\partial \mathbf{F}}{\partial \mathbf{x}}} \mathbf{d}\mathbf{x} + \underbrace{\frac{\partial \mathbf{F}}{\partial \mathbf{v}}} \mathbf{d}\mathbf{v} \quad (5.23)$$

where \mathbf{D} is the damping matrix. By combining with the time stepping (Eq. 5.22), we obtain the following linear system $\mathbf{Ax} = \mathbf{b}$ to solve at each time step:

$$\underbrace{(\mathbf{M} - h\mathbf{D} - h^2\mathbf{K})}_{\mathbf{A}} \underbrace{\frac{d\mathbf{v}}{x}}_{\mathbf{x}} = \underbrace{h\mathbf{F}(\mathbf{x}_{i-1}, \mathbf{v}_{i-1}) + h^2\mathbf{K}\mathbf{v}_{i-1} + h\mathbf{M}\mathbf{g} + h\mathbf{F}_{ext}}_{\mathbf{b}} \quad (5.24)$$

Note that here, the matrix \mathbf{A} is always positive definite, thanks to the presence of the mass matrix. The system can be solved even if the robot is not attached to any rigid or static parts.

In both cases (quasi-statics and dynamics), we have introduced the equations that leads to the solving of a sparse linear matrix system $\mathbf{Ax} = \mathbf{b}$ with matrix \mathbf{A} and vector \mathbf{b} changing at each step. In the static case, the matrix \mathbf{A} represents the tangent stiffness of the deformable object and in the dynamic case, the matrix \mathbf{A} can be considered as the tangent of the impedance integrated by the numerical scheme.

5.6 Lagrangian Mechanics Formulation

As seen in paragraph Sect. 5.3.1.5, in a weak form (finite element method or other numerical method), the calculation of internal forces due to deformations is derived from an energy potential. This is one of the foundations of Lagrangian mechanics. The advantage of an energetic approach such as Lagrangian mechanics is that different robot motion parameterization spaces can be used, while guaranteeing energetic equivalence. Moreover, in the previous section, to build the linear system solved at each time step, we have considered that the external forces do not depend on the position of the nodes which is not always true, in particular for computing actuator forces or contact forces. We will therefore introduce **Lagrange multipliers** to model these forces. The equation of motion to be solved at each time step will become:

$$\begin{cases} \mathbf{Ax} = \mathbf{b} + \mathbf{H}^T \boldsymbol{\lambda} & \text{in the static case} \\ \mathbf{Ax} = \mathbf{b} + h\mathbf{H}^T \boldsymbol{\lambda} & \text{in the dynamic case} \end{cases} \quad (5.25)$$

In the specific case of modeling a deformable robot, we will then distinguish 3 parameterization spaces:

1. **FEM DOFs, node motion or equivalent:** To be able to solve continuum mechanics using a numerical method, the number of degrees of freedom must be sufficiently high for convergence. In FEM, the objective is that by changing the mesh size, the solution found by the numerical method is essentially the same. To achieve this goal, and in particular when we want to achieve convergence on the strain field (which is a derivative of the node displacement field), we will often reduce the size of the elements and considerably increase the size of the model, which can take us far away from the computation times required for use in robotics.
2. **Reduced DOFs, Reduced order model:** The number of degrees of freedom generated by numerical methods is much higher than the usual number of degrees of freedom used in robotics. But the objective is not the same as above. In

robotics, the aim is to parameterize all the configurations the robot can take during its use and interaction with its environment, whereas with numerical methods, the aim is convergence. To use the model in robotics, we can therefore project the FEM model into a very reduced subspace of degrees of freedom, by reparameterizing the motion with a reduced number of DOFs. However, care must be taken not to alter the accuracy of the FEM model.

3. **Sub-spaces: actuators, effectors, sensors and external constraints:** The actuation space is very important in robotics: it determines the motions that can be controlled on the robot. Robots are often largely under-actuated: the actuation space is a subspace of all parameterized deforming movements. In the general case of a deformable robot, the actuation space is coupled: an effort on an actuator leads to a movement or a change in effort on the other actuators. In addition, the robot's movement will be controlled at certain points called "end-effectors", often located at the end of the robot, but sometimes at other points to control the configuration. Here too, we're talking about a subspace of possible movements. Finally, sensors can be placed on the robot to define a subspace of movements that can be measured and external interactions, like contacts, will be defined as constraints on a subpart of the DOFs.

In both cases, the dimensions of the parameterization spaces (2) and (3) are much smaller than (1). But it's important to note that (2) will give a parameter space describing all possible robot movements. We can choose to reduce these possible movements, but the space (2) must always remain larger than the subspaces defined in (3), otherwise hyperstatisms could occur. Moreover, the mapping between the different spaces is not made in the same way. The mapping between the degrees of freedom of the FEM (1) and the reduced DOFs (2) is made by stiffness projection, whereas the mapping between the degrees of freedom of the FEM (1) (or the reduced DOFs(2)) to the subspaces (3) is made by compliance projection. We will describe the two projections in the following subsections

5.6.1 DOF Reduction and Stiffness Projection

Lagrangian mechanics modeling stipulates that whatever coordinate system is chosen, energy must be conserved. Deformation energy is potential energy. The principle of virtual work can be applied to modify the coordinate system, without affecting the energy. So if we introduce new dofs \mathbf{q} with a mapping function \mathbb{M} such as

$$\mathbf{x} = \mathbb{M}(\mathbf{q}) \text{ and } d\mathbf{x} = \underbrace{\frac{\partial \mathbb{M}}{\partial \mathbf{q}}}_{\mathbf{J}} d\mathbf{q} \quad (5.26)$$

The work of the forces \mathbf{F} must remain the same with the new coordinate space defined by \mathbf{q} . Let τ be the forces in this space. If the work w is the same in the two coordinates system for the same displacement increment, we have:

$$w = d\mathbf{q}^T \boldsymbol{\tau} = d\mathbf{x}^T \mathbf{F} \Leftrightarrow d\mathbf{q}^T \boldsymbol{\tau} = d\mathbf{q}^T \mathbf{J}^T \mathbf{F} \Leftrightarrow \boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \quad (5.27)$$

As in rigid robotics, we use the Jacobian \mathbf{J} and the transposed Jacobian \mathbf{J}^T to project the mechanics into a reduced motion parameter space. Two use cases are now explained to show the interest of such a projection.

- **stiffening:** In the case of hybrid “soft-rigid” modeling, such a projection can be used. When the stiffness of the material is sufficiently high in certain regions, or when the boundary conditions impose rigid movements on certain parts of the mesh, we will consider a subpart of the nodes do not undergo any deformation (see Fig. 5.8). This part of the mesh is then considered to have a rigid motion ($\mathbf{x}_r, \mathbf{v}_r$) and can therefore be parameterized with a transformation (translation and rotation) (\mathbf{q}_1) with only 6 degrees of freedom. If many nodes are *stiffened*, the number of degrees of freedom can be significantly reduced. Of course several independent rigid regions can be modeled. In that case the dimension of (\mathbf{q}_1) is 6 multiplied by the number of independent rigid regions. The motion of the other part of the nodes ($\mathbf{x}_{\bar{r}}, \mathbf{v}_{\bar{r}}$) is still considered unknown. We will therefore define an identity mapping (often including a renumbering) with the space of degrees of freedom $\mathbb{I}(\mathbf{q}_2)$. For this part of the mapping, the number of degrees of freedom does not change.

$$\begin{cases} \mathbf{x}_r = \mathbb{T}(\mathbf{q}_1) & \mathbf{v}_r = \mathbf{J}_{\mathbb{T}} \dot{\mathbf{q}}_1 \\ \mathbf{x}_{\bar{r}} = \mathbb{I}(\mathbf{q}_2) & \mathbf{v}_{\bar{r}} = \mathbf{I}_{\bar{r}} \dot{\mathbf{q}}_2 \end{cases} \quad (5.28)$$

When applied on Eq. 5.25 in the static case, we obtain the following system to be solved.

$$\begin{bmatrix} \mathbf{J}_{\mathbb{T}}^T \mathbf{A}_{rr} \mathbf{J}_{\mathbb{T}} & \mathbf{J}_{\mathbb{T}}^T \mathbf{A}_{r\bar{r}} \mathbf{I}_{\bar{r}} \\ \mathbf{I}_{\bar{r}}^T \mathbf{A}_{\bar{r}r} \mathbf{J}_{\mathbb{T}} & \mathbf{I}_{\bar{r}}^T \mathbf{A}_{\bar{r}\bar{r}} \mathbf{I}_{\bar{r}} \end{bmatrix} \begin{bmatrix} d\mathbf{q}_1 \\ d\mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathbb{T}}^T \\ \mathbf{I}_{\bar{r}}^T \end{bmatrix} (\mathbf{b} + \mathbf{H}^T \boldsymbol{\lambda}) \quad (5.29)$$

The system to be solved is therefore smaller in size, but the matrix projection calculations can add computational cost, so computation time is not necessarily faster than for the full system. On the other hand, this modeling approach is very useful

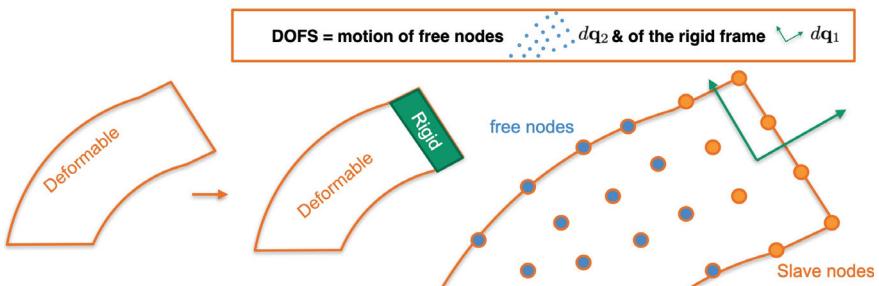


Fig. 5.8 Modeling the stiffening of a part of a deformable model through reduction of DOFs

for roboticists, as we find the rigid transformations $\mathbb{T}(\mathbf{q}_1)$ and associated Jacobians $\mathbf{J}_{\mathbb{T}}$ which are widely used in rigid robotics. On the other hand, FEM models with strong heterogeneities (with very steep and very soft regions) often lead to problems of poor conditioning. The approach outlined here helps to overcome them.

In dynamics, note that for high-speed movements, the $\mathbf{J}_{\mathbb{T}}(\mathbf{q}_1)$ matrix is non-linear and some derivative terms of the form $\dot{\mathbf{J}}_{\mathbb{T}}\dot{\mathbf{q}}_1$ will appear when calculating acceleration. Same, for very strong external forces: it could be useful sometimes to add the apparent stiffness (created by the non-linearity of the mapping) $\frac{\partial \mathbf{J}_{\mathbb{T}}(\mathbf{q}_1)\mathbf{F}_{ext}}{\partial \mathbf{q}_1}$ in the matrix A. For the moment, these terms have not yet been explored to any deep extent. For the sake of brevity, we won't elaborate on these points here.

- **Model order reduction:** To overcome the difficulty of quickly calculating the full FEM model, one possible approach is to reduce the size of the model by projecting the equations from the FEM into a reduced space. This technique also allows to obtain a dynamic reduced model. In engineering tools, a well known projection technique is the modal analysis. After calculating the structure in FEM, the main deformation modes are calculated and the model is projected in this space. We can thus define the size of the model according to the frequency range of vibration we are interested in (generally, the FEM model generates many numerical vibrations of very high frequency, which can be assimilated to numerical noise). The modal analysis is often limited to linear deformations but can be extended to non-linear [13].

Other techniques, such as POD, allow to compute the principal modes of deformations according to a data set pre-computed from an SVD decomposition (See Fig. 5.9). This technique first requires a calculation phase on the complete model. But then, the model motion can be projected through Φ in a reduced space α :

$$\Phi^T \mathbf{M} \Phi \ddot{\alpha} = \Phi^T \mathbf{F}(\Phi \alpha) + \Phi^T \mathbf{M} \mathbf{g} + \Phi^T \mathbf{F}_{ext} + \Phi^T \mathbf{H}^T \lambda$$

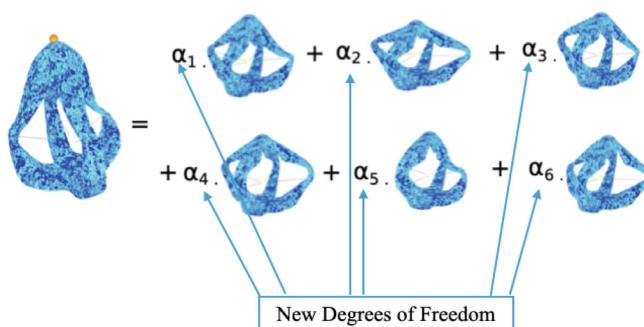


Fig. 5.9 Reduction of the degrees of freedom: In this figure, we suppose that the motion of the soft robot can be captured by a linear combination of 6 modes. The intensity of these modes are the new degrees of freedom

The approach is compatible with non-linear model and the approach allows to dramatically reduce the number of DOFs of the models (typically a space of 50 modes can be sufficient for a complex robot) [14]. By further reducing the state space of the system (by tolerating larger modeling errors) state observers can be introduced and perform dynamic control [15].

After integration of the dynamics using Eq. 5.24 or using the quasi-static assumptions, the reduced system to be solved at each step is:

$$\begin{cases} \Phi^T \mathbf{A} \Phi d\alpha = \Phi^T \mathbf{b}(\Phi\alpha) + \mathbf{H}^T \lambda & \text{in the static case} \\ \Phi^T \mathbf{A} \Phi d\alpha = \Phi^T \mathbf{b}(\Phi\alpha) + h \mathbf{H}^T \lambda & \text{in the dynamic case} \end{cases} \quad (5.30)$$

5.6.2 Compliance Projection

The compliance projection is based on condensation. This technique reduces the equations of equilibrium and motion on the loaded degrees of freedom. The unloaded degrees of freedom then become slaves. In general, on soft robots, actuation is not defined in a single node. Therefore, one can condense by adding Lagrangian constraints to the system. Let's suppose a robot made with one cavity and we load the structure of the robot by changing the volume $\delta_v(\mathbf{x})$ of this cavity (for instance with pressurized air or water). This volume depends on the position of the nodes of the mesh that are on the surface of the cavity. We can add the constraint $\delta_v(\mathbf{x}) = v_{\text{input}}$ to load the structure and deform the robot.

By solving this non-linear constraint system, a pressure will be applied on these nodes, so only a sub-part of the nodes of the robot model will be loaded. In practice, to solve the system, the model is linearized around the current position \mathbf{x} and a Lagrange multiplier λ is introduced to replace the external loads \mathbf{R} and compute the pressure:

$$\begin{cases} \mathbf{F}(\mathbf{x} + \Delta\mathbf{x}) + \mathbf{Mg} + \mathbf{H}^T \lambda \approx \mathbf{F}(\mathbf{x}) + \mathbf{K}(\mathbf{x})d\mathbf{x} + \mathbf{Mg} + \mathbf{H}^T \lambda = \mathbf{0} \\ \delta_v(\mathbf{x} + \Delta\mathbf{x}) \approx \delta_v(\mathbf{x}) + \mathbf{H}d\mathbf{x} = v_{\text{input}} \end{cases}$$

with $\mathbf{K}(\mathbf{x}) = \frac{d\mathbf{F}}{d\mathbf{x}}$ and $\mathbf{H} = \frac{d\delta_v}{d\mathbf{x}}$. At equilibrium, we can obtain the instantaneous relationship between the change of pressure $\lambda \rightarrow \lambda + \Delta\lambda$ and the change of volume as a single equation by condensation of the mechanical system:

$$\Delta\delta_v = \delta_v(\mathbf{x} + d\mathbf{x}) - \delta_v(\mathbf{x}) = \underbrace{\mathbf{H}\mathbf{K}(\mathbf{x})^{-1}\mathbf{H}^T}_{\mathbf{W}_{aa}} \Delta\lambda \quad (5.31)$$

With \mathbf{W}_{aa} a projected compliance in the actuator space. With several actuators, this matrix provides the mechanical coupling between actuators. By playing with

the definition of functions δ (we can define δ_a for the actuators, δ_e for the end-effectors and δ_s for the sensors), we can obtain the coupling between actuators and end-effectors (\mathbf{W}_{ea}) or between sensors and end-effectors (\mathbf{W}_{se}). Kinematic models of the robot can be derived from this:

$$\Delta\delta_e = \underbrace{\mathbf{W}_{ea} \mathbf{W}_{aa}^{-1}}_{\mathbf{J}} \Delta\delta_a$$

where \mathbf{J} is the jacobian of the soft robot.

For inverse kinematics, one can, by the optimisation $\min(\delta_e)_{\lambda_a}$ compute the inverse model by minimizing the distance between the end-effector position and an objective position, by actuating the structure through λ_a (see Sect. 5.8.2). The same strategy can be used to measure a force at the end-effector position λ_e , by minimizing the distance between the sensor values in the model and in real-life $\min(\delta_s)_{\lambda_e}$ (see Sect. 5.8.3).

Note that the problem being non-linear, the condensation requires the computation and the factorization of $\mathbf{K}(\mathbf{x})$ which is a very large matrix, in order to obtain \mathbf{W} . More details will be provided in the next section.

But at this point, we'd like to remind you of the different spaces in which the equations of deformable motion are written, and more specifically here, the equations of deformable robots. This is illustrated on Fig. 5.10 First, there's the behavior of materials that are subjected to deformation. To characterize motion and effort in materials, we have used Green-Lagrange strain \mathbf{E} and stress σ . Then we have interpolated and integrated the deformations over the shape of the robot thanks to Finite Element Method. The motion is discretized at the level of the nodes, thanks to their position \mathbf{x} and the internal forces $\mathbf{F}(\mathbf{x})$. After that we have projected the mechanics in a subspace and we have defined reduced dofs \mathbf{q} and efforts τ . Finally, we did the projection in the actuator and effector spaces with δ and λ . In all these changes of space, there's a guarantee of energy equivalence, with the principle of virtual work.

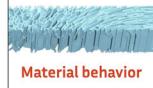
Space	Constitutive law	Finite Element Model	DOF Reduction	Compliance Projection
Illustration	 Material behavior	 FEM nodes	 Free nodes Joints	 Effector Actuators
Motion / Effort	\mathbf{E} / σ	\mathbf{x} / \mathbf{F}	\mathbf{q} / τ	δ / λ

Fig. 5.10 The different spaces in which the deformable mechanics of soft robots have been described

5.7 Actuator Models

Having described how actuator space is introduced in the form of a Lagrange multiplier, we'll now be able to describe in more detail the meaning of this multiplier for three different types of actuation.

5.7.1 Pressure Based Actuation

The pressure inside the cavity λ_a is distributed as a force on the nodes placed at the cavity surface thanks to the matrix \mathbf{H}_a^T . When the pressure is imposed, the value of the Lagrange multiplier is known ($\lambda_a = p_a$). This is often the case with pneumatic actuators. But when using hydraulic actuation, the volume in the cavity is imposed and the pressure λ_a is unknown. The Lagrange multiplier is solved by adding a new equation which forces the local linearization of the volume δ_a computed on the FEM mesh to follow the imposed volume v_a . The Fig. 5.11 presents the systems of equations to be solved in both cases of imposed pressure or imposed volume. Note that in the case of liquid, if the cavity is big, the weight of the liquid could also influence the behavior [16].

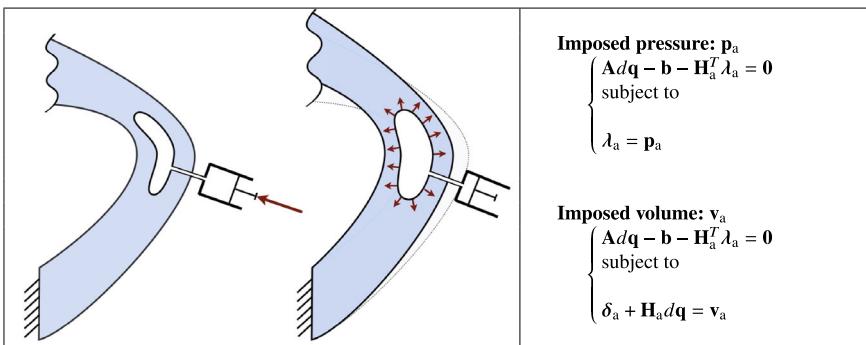


Fig. 5.11 Pressure based actuator

5.7.2 Tendon Based Actuation

In the tendon case (see Fig. 5.12), we assume that the cable is infinitely stiff and has no internal bending and torsion forces. In such case, we can use a geometrical representation of the cable (described below). In [17], a deformable rod model of the

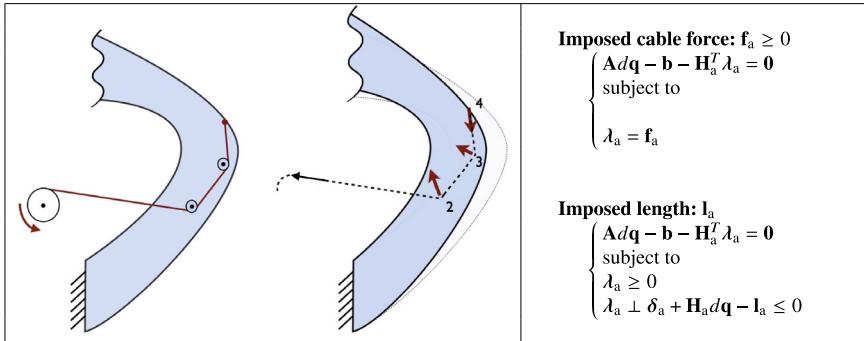


Fig. 5.12 Tendon (Cable) actuator

tendon is coupled with a FEM model of the robot, but this approach is not detailed here.

In the case of a geometric model of the cable, the Lagrange multiplier λ_a is the tensile force inside the cable. This force is distributed over the structure according to the cable path. Each time the cable path is inflected (deviating from the straight line), tangential forces appear, proportional to the tensile force. We can therefore find a matrix \mathbf{H}_a which distributes the force over the nodes of the mesh (nodes 2, 3, 4 on the figure). Note that if this cable path does not correspond to a series of segments and nodes in the FEM mesh, then the matrix \mathbf{H}_a also contains interpolation values of the points defining the cable with respect to the mesh nodes.

When the force \mathbf{f}_a is known and imposed, it must be positive (one can only pull on a cable, not push) and in that case the value of the Lagrange multiplier λ_a is known. When the cable is controlled in position, we can impose the length of the cable \mathbf{l}_a ... only when the cable is taut. We thus have to solve the problem with two inequalities: the force is positive and the maximum length of the cable is the imposed length (the length of the cable is linearized given the current configuration of the robot). Between this two inequalities, there is a complementarity: the force is strictly positive ($\lambda_a > 0$) only when the cable is taut ($\delta_a + \mathbf{H}_a d\mathbf{q} - \mathbf{l}_a = 0$) and $\lambda_a = 0$ when the cable is loose ($\delta_a + \mathbf{H}_a d\mathbf{q} - \mathbf{l}_a < 0$).

5.7.3 Direct Actuation with Stiffness Projection

In Sect. 5.6.1, we've seen how to model the absolute stiffening (no more deformation) of a part of the robot. This stiffening is often required to actuate the deformation with motors directly connected to these rigid parts. The most common case is probably that illustrated in Fig. 5.13, where a motor drives the rotation of part of the deformable model's nodes. It's also common to have the same thing happen, but with a translating motor. We can also find this in more specific cases, such as (rigid) magnets placed in

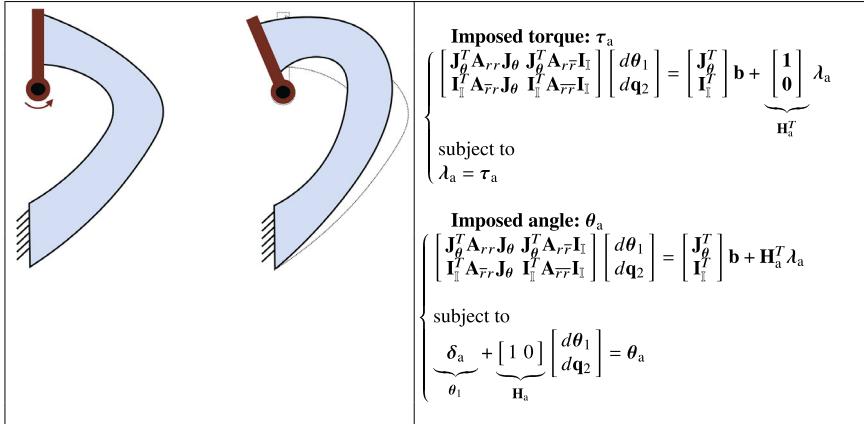


Fig. 5.13 Direct actuation on rigid part

a deformable robot driven by a magnetic field. In all these cases, there is a stiffening combined with an actuation placed on these rigid zones.

As described in Eq. 5.29, we will reduce the degrees of freedom of the deformable model by separating the nodes placed on the rigid part r and the other nodes \bar{r} . In the case of Fig. 5.6.1, the degree of freedom of the rigid part is directly given by the change of angle and denoted $d\theta_1$ (subscript 1 is used to keep the similarity with Eq. 5.29). The motion of all these nodes placed on the rigid boundary can be parameterized by this angle. The other nodes keep their degrees of freedom, which are set in $d\mathbf{q}_2$.

It can be seen that when torque is applied to the motors, $\lambda_a = \tau_a$, the torque is applied at the rotating rigid degree of freedom. Thus, the expression of the matrix \mathbf{H}_a^T gives the identity on the first line (which concerns the expression of the forces of the rigid part) and 0 on the second (which concerns the forces on the nodes remaining free).

Like the previous actuators, the motion can be imposed. In the example, a rotation angle θ_a can be imposed by the motor (if controlled in position). In such case, the value of the torque λ_a is unknown and depends on the overall equilibrium of the robot structure. A new equation is thus defined to solve the Lagrange multiplier, which corresponds to impose this angle. In such case, the constrain equation is very intuitive: δ_a corresponds to the value of the angle θ_1 at the beginning of the step and the variation $d\theta_1$ should be found so that $\theta_1 + d\theta_1 = \theta_a$.

5.8 Compliance Based Models of Soft Robots

In the previous section, the models of 3 types of actuator used in soft robotics are presented. In each case, we have used of a Lagrange multiplier λ_a to obtain the actuator force, a matrix \mathbf{H}_a which distributes the actuator forces over the deformable system, and its transpose, the matrix \mathbf{H}_a^T which projects the motion in the dof space into the actuator space. In the previous chapter, we saw that in such a case, we could use the compliance projection of the deformable robot model (see Eq. 5.31). These two aspects will be combined here to describe the kinematics of the soft robot and, above all, to calculate its inverse model very efficiently, in real time using the projected compliance.

This section shows the end result of the entire modeling process in the form of projected compliance, introduced at the beginning of the chapter. As we said at the beginning of the chapter (and even in the title!), we believe this operator of projected compliance is key to the modeling of soft robots. The following paragraphs are therefore essential to understand how to derive a truly generic model of soft robot.

5.8.1 Effector Model and Projected Compliance

We call *effectors*, some degrees of freedom of the soft robot, that we wish to control. As said previously, soft-robot are usually underactuated so, we can only control a subpart of the degrees of freedom. As for actuators, a matrix \mathbf{H}_e will be defined for the effectors. This matrix allows to obtain the motion of the effector $\Delta\delta_e$ given the motion of the degrees of freedom of the FEM model.

$$\Delta\delta_e = \mathbf{H}_e d\mathbf{q} \quad (5.32)$$

In practice, we often define some nodes of the mesh as the effectors so $\mathbf{H}_e = [0 \dots 0 \ \mathbf{I} \ 0 \dots 0]$ where \mathbf{I} is the identity matrix of the dimension of the selected node (3 for a position node, 6 for a position and orientation node, like for beam and shell elements see subsection 5.4.2). Sometimes it is interesting to define several effector points and/or to control particular directions (x, y but not z for example). The principle remains the same, it only changes the size of $\Delta\delta_e$ and the identity matrix \mathbf{I} . A Lagrange multiplier λ_e and the matrix \mathbf{H}_e^T can be also defined to apply a load on the effectors. If no load is applied on the effector point(s), then $\lambda_e = 0$.

In some cases, such as locomotion for instance, the effector can be defined as the center of mass (or the barycenter of the nodes). In these cases, $\Delta\delta_e$ provides the instantaneous displacement of the center of mass of the robot. The jacobian matrix is the same for all nodes i and is given by:

$$\text{Center of mass : } \mathbf{H}_e^i = \frac{1}{w} \quad \text{Barycenter : } \mathbf{H}_e^i = \frac{1}{\#\text{nodes}}$$

with $\#nodes$ the total number of nodes and w the mass of the object.

Thanks to the compliance projection presented in Eq. 5.31, we can obtain the mechanical coupling in the effector and actuator spaces. Starting from a given configuration $\mathbf{A}\mathbf{d}\mathbf{q} - \mathbf{b} - \mathbf{H}_a^T \boldsymbol{\lambda}_a = \mathbf{0}$ and introducing $\Delta\boldsymbol{\lambda}_a$ a small change of actuation force, the motion created in the effector and actuator spaces can be computed by

$$\Delta\boldsymbol{\delta}_e = \underbrace{\mathbf{H}_e \mathbf{A}^{-1} \mathbf{H}_a^T}_{\mathbf{W}_{ea}} \Delta\boldsymbol{\lambda}_a \text{ and } \Delta\boldsymbol{\delta}_a = \underbrace{\mathbf{H}_a \mathbf{A}^{-1} \mathbf{H}_a^T}_{\mathbf{W}_{aa}} \Delta\boldsymbol{\lambda}_a$$

with \mathbf{W}_{ea} and \mathbf{W}_{aa} the smallest possible operators that provides respectively the coupling by the computed compliance of the robot between actuators and effectors and between actuators.

5.8.2 Kinematics and Inverse Model by Optimization

As explained in Fig. 5.2, thanks to \mathbf{W}_{ea} and \mathbf{W}_{aa} , we can compute the local kinematics of the soft robot, i.e., the jacobian between motion in actuator space and motion in effector space.

$$\mathbf{J}_{sr} = \mathbf{W}_{ea} \mathbf{W}_{aa}^{-1} \text{ and } \Delta\boldsymbol{\delta}_e = \mathbf{J}_{sr} \Delta\boldsymbol{\delta}_a$$

It has to be emphasized that the values of $\mathbf{W}_{ea}(\mathbf{q})$ and $\mathbf{W}_{aa}(\mathbf{q})$ are not constant in general and depends on the configuration \mathbf{q} of the robot, as the matrices $\mathbf{A}(\mathbf{q})$ and $\mathbf{H}(\mathbf{q})$ depends on \mathbf{q} . The computation of matrices \mathbf{W} can be challenging to compute in real-time, in particular if the size of the matrix \mathbf{A} is big (i.e. a dense FEM mesh). However, the matrix \mathbf{A} is highly sparse and always positive definite. Some very efficient solvers based on the factorization of matrix \mathbf{A} are available in packages for linear algebra like Eigen [18], Csparse [19] and others. Moreover, the computation of this matrix \mathbf{W} in real-time has been optimized in different cases, like for slender structure (continuum robots) [20] or for corotational case [21] with an approximation. Recent techniques, based on GPUs, have been developed to compute in few ms a very good approximation of the matrix in dense contact cases (with a matrix \mathbf{W} involving more than 200 constraints on a meshes with more than 5000 nodes !) [22].

Inverse model of the soft robot can be computed by pseudo-inverse of the \mathbf{J}_{sr} but quadratic optimisation is a better option to introduce some bounds and deal with the redundancy of the robots. In practice, let's provide a desired position \mathbf{p}_e to the effector. We suppose that the previous step of the simulation provides a position \mathbf{q}_0 of the deformable model and $\boldsymbol{\delta}_e(\mathbf{q}_0)$ provides the current position of the effector. We thus would like to modify the actuation of the robot (i.e. find $\Delta\boldsymbol{\lambda}_a$) so that the effector moves $\Delta\boldsymbol{\delta}_e$ to minimize the distance $(\boldsymbol{\delta}_e(\mathbf{q}_0) + \Delta\boldsymbol{\delta}_e - \mathbf{p}_e)^2$. Introducing $\boldsymbol{\delta}_{e0} = \boldsymbol{\delta}_e(\mathbf{q}_0) - \mathbf{p}_e$ the violation of the desired motion at the beginning of the control step, we can write the following optimisation problem to find the actuation:

$$\left\{ \begin{array}{l} \min_{\Delta\lambda_a} (\delta_{e0} + \Delta\delta_e)^2 \text{ (with } \Delta\delta_e = \mathbf{W}_{ea}\Delta\lambda_a) \\ \text{subject to} \\ \delta_{\min} \leq \delta_{a0} + \Delta\delta_a \leq \delta_{\max} \text{ (with } \Delta\delta_a = \mathbf{W}_{aa}\Delta\lambda_a) \\ \text{(if cable)} \lambda_a + \Delta\lambda_a \geq 0 \end{array} \right.$$

where δ_{\min} and δ_{\max} are the actuating stops which define the limits of the actuators movements. An other constraint on the maximum speed of the actuator $-\Delta_{\max} \leq \Delta\delta_a \leq \Delta_{\max}$ can also be defined. This optimization problem falls into the category of QP problems for which numerous solvers are also available.

The matrix of the QP is $\mathbf{Q} = \mathbf{W}_{ea}^T \mathbf{W}_{ea}$ which is symmetric positive by construction, however it is not always fully definite (convex problem). Indeed, when the number of actuators is greater than the size of the effector space, multiple configuration can lead to a minimum. In practice, there is redundancy in the actuation and several solution are possible. Some QP algorithms are able to find one solution among all possible, however, the solution may oscillate between possible solutions.

To have a stable behavior, we add to the objective function a minimization of the mechanical work of the actuator force $w_a = \Delta\delta_a^T(\lambda_a + \Delta\lambda_a) = \Delta\delta_a^T\lambda_a + \Delta\lambda_a^T \mathbf{W}_{aa}^T \Delta\lambda_a$. Matrix \mathbf{W}_{aa} being symmetric definite and positive, $\epsilon \mathbf{W}_{aa}$ can be added in the QP matrix $\mathbf{Q} = \mathbf{W}_{ea}^T \mathbf{W}_{ea} + \epsilon \mathbf{W}_{aa}$ with a sufficiently small coefficient ϵ to regularize the problem while keeping a precise solution.

5.8.3 Sensor Model and External Forces Measurement

In the same way as we defined the actuators and effectors on soft robots, we can extend the formulation to the extrinsic or intrinsic sensors that can be placed on the robot. Let's imagine for instance, a camera system with marker points measuring the position of points on the robot (extrinsic) and a sensor measuring curvature on a robot (intrinsic). In the first case the measure of the position sensor is a selection of node position on the robot mesh $\delta_s = \mathbf{H}_s \mathbf{q}$ with $\mathbf{H}_s = [0 \dots 0 \mathbf{I} 0 \dots 0]$.

In the second case, the output of the sensor is a continuous non-linear function of the node position $\delta_s(\mathbf{q}) = Curv(\mathbf{q})$ which provides the local curvature measured by the sensor. On a given position a first order approximation of this function can be computed using $\delta_s(\mathbf{q} + d\mathbf{q}) \approx \delta_s(\mathbf{q}) + \mathbf{H}_s d\mathbf{q}$. This formulation is then very generic. In [23] a flex sensor is used and δ_s is found by fitting a function on data collected from the simulation and sensors during a defined motor trajectory phase.

To estimate the shape of the robot, we suppose that we have a vector of measured quantities δ_s^m . When some external forces are applied at on effectors, we can introduce a optimization problem to estimate these external forces λ_e :

$$\left\{ \begin{array}{l} \min_{\Delta\lambda_e} (\delta_s(\mathbf{q}) + \mathbf{H}_s d\mathbf{q} - \delta_s^m)^2 \\ \text{with} \\ \Delta\delta_s = \mathbf{H}_s d\mathbf{q} = \mathbf{W}_{se}\Delta\lambda_e \end{array} \right.$$

If more sensors are used, it can also estimate the errors on the actuator forces based on the model:

$$\left\{ \begin{array}{l} \min_{\Delta\lambda_e, \Delta\lambda_a} (\delta_s(\mathbf{q}) + \mathbf{H}_s d\mathbf{q} - \delta_s^m)^2 \\ \text{with} \\ \Delta\delta_s = \mathbf{H}_s d\mathbf{q} = \mathbf{W}_{se} \Delta\lambda_e + \mathbf{W}_{sa} \Delta\lambda_a \end{array} \right.$$

For this inverse problem to be solved under the good conditions (i.e. with a unique solution), the number of independent measurements on the sensors must be greater than the number of unknowns on the effectors and actuators. The rank (and the conditioning) of matrices \mathbf{W}_{se} and \mathbf{W}_{sa} are also key to obtain a full rank matrix in the Quadratic Programming optimisation that are used to solved this optimisation. For more details on the sensors see [24] or [25]. Such approach can be used in closed look control strategies [26].

5.9 Contact Interaction with the Environment

Most of today's rigid manipulators are designed to have the greatest possible rigidity between their base and end-effector. It is this rigidity (high stiffness) that gives them good positioning accuracy and repeatability in space. One of the consequences of this design is to limit interaction and contact with the environment. Numerous algorithms have been developed for anti-collision and obstacle avoidance, to prevent any interaction with the environment outside the end-effector. On current collaborative robots, if a contact or collision is detected on the robot's body, the control law provides for the robot to stop its movement, notably for safety reasons.

On humanoid robots, contact are considered for trajectory optimisation, but they are often limited to robot's feet and hands, all other mechanical contacts with the environment are usually avoided. Compliant commands implemented on some collaborative robots can be used to define a target force applied to a contact on the robot's end-effector, sometimes in certain directions. But here again, these commands are limited to contacts on the end effector of the robot.

The specificity of soft robots is their high mechanical compliance, enabling them to come into contact with their direct environment over part of their body. This ability to come into contact also changes the way model are conceived. Contact modeling is an integral part of modeling this type of robot. In this section, we discuss the general principles of modeling these contacts, which remains an active research topic.

5.9.1 Contact Model

The first step is to determine the points of contact between the robot model and the model of its environment. Collision detection may appear simple from a geometric point of view, but it is complex from a computational point of view. Indeed, the natural complexity is quadratic: it would consist in testing the set of geometric primitives (often triangles) of the robot model with the set of primitives of the environment model. This quadratic complexity can rapidly become the bottleneck for the computation of the contacting models. This is why many works have proposed algorithmic solutions to compute efficiently the collision detection between deformable objects [27].

Once contact has been detected, one needs a behavioral law to describe the physics of contact. In the context of continuum mechanics, we use Signorini's law to describe the normal part of the contact using a complementarity constraint. This law provides a macroscopic view of contact, consistent with continuum mechanics and multi-contact cases [28]. In contact zones, at each point, a signed distance $\delta_c \geq 0$ is defined between the two solids in contact, which is normal to the contact. This distance must remain positive or zero to prevent interpenetration of the solids (which would be not physical). Furthermore, Signorini's law defines the force (or stress) $\lambda_c \geq 0$ along the same normal direction to the contact surface. This force is also positive or zero to mean that contact can repel objects but cannot attract them to each other (excluding normal adhesion or bonding between objects). Finally, there is complementarity between two cases: (1) contact where the distance is zero at the point of contact and the repulsive force is non-zero $\delta_c = 0$ and $\lambda_c > 0$ (2) non-contact where the distance is strictly positive at the point of contact and there can be no repulsive force $\delta_c > 0$ and $\lambda_c = 0$ (See Fig. 5.14). To deal with the multi-contact case, the distances and forces at contact are gathered in vector form with δ_c and λ_c respectively. Complementarity means that the vector product between these two vectors is zero, so they are perpendicular. Signorini's law is written:

$$0 \leq \delta_c \perp \lambda_c \geq 0 \quad (5.33)$$

In 3D, at each point of contact, the two directions tangent to the surface form a plane in which the instantaneous sliding motion at contact is defined. It is therefore in this plane that we can define a law of friction, such as Coulomb friction. Friction is also defined by the complementary constraint between two states: (1) adhesion (or dry friction) which corresponds to the total suppression of tangent sliding and a force strictly included in the Coulomb cone and (2) sliding (or dynamic friction) which corresponds to the case where the relative motion is non-zero in the tangent plane and opposes (in direction) the force that is strictly on the surface of the Coulomb cone. However, so as not to unnecessarily complicate the remaining of the chapter, we'll consider the frictionless case. To add Coulomb friction, please refer to [29, 30].

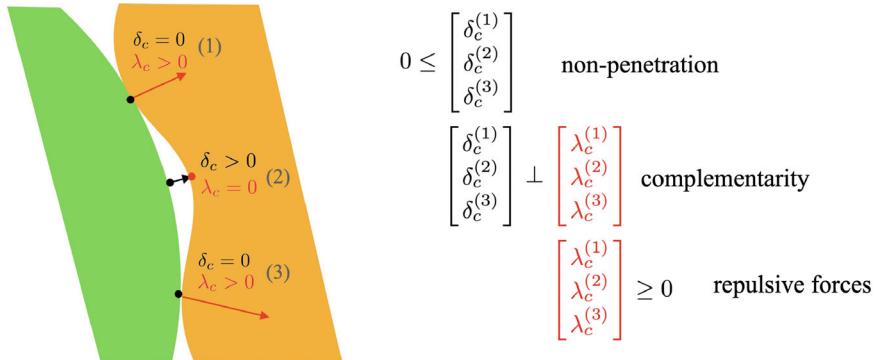


Fig. 5.14 Signorini's unilateral law for multicontact case: the three principles of this law are (i) to prevent penetration at contact points, (ii) to have a complementarity between distance and contact force and (iii) to impose repulsive forces. The drawn forces are applied to the yellow object but opposite forces are applied to the green solid

5.9.2 Direct Solving Process Using Compliance Projection

In Sect. 5.7, to introduce actuation in the deformable mechanical models of robots, we have used Lagrange multipliers and a constraint based formulation. Contact will provide additional constraints and complementarity conditions. Moreover, the model of the environment will provide additional degrees of freedom, leading to the following constraint problem:

$$\text{solve } \left\{ \begin{array}{l} \left[\begin{array}{cc} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_e \end{array} \right] \left[\begin{array}{c} d\mathbf{q} \\ d\mathbf{q}_e \end{array} \right] - \left[\begin{array}{c} \mathbf{b} \\ \mathbf{b}_e \end{array} \right] - \left[\begin{array}{c} \mathbf{H}_a^T \lambda_a + \bar{\mathbf{H}}_c^T \lambda_c \\ \bar{\mathbf{H}}_c^T \lambda_c \end{array} \right] = \mathbf{0} \\ \text{with} \\ \delta_a + \mathbf{H}_a d\mathbf{q} = \mathbf{u} \\ 0 \leq \delta_c + \mathbf{H}_c d\mathbf{q} + \bar{\mathbf{H}}_c d\mathbf{q}_e \perp \lambda_c \geq 0 \end{array} \right.$$

With \mathbf{A}_e , $d\mathbf{q}_e$ and \mathbf{b}_e represents the matrix, the degrees of freedom and the forces vector of the deformable model representing the environment. By the action-reaction principle, contact forces are applied to the nodes of the robot deformable models through \mathbf{H}_c^T matrix and to the nodes of the environment model through the $\bar{\mathbf{H}}_c^T$ matrix. The distance at each contact point is also influence by the motions of both the robot and its environment leading to a final distance at the end of the step being: $\delta_c + \mathbf{H}_c d\mathbf{q} + \bar{\mathbf{H}}_c d\mathbf{q}_e$.

The robot and environment models are only coupled by a few contact equations. In general, the number of degrees of freedom of the model (size of $d\mathbf{q}$ and $d\mathbf{q}_e$) is much greater than the number of contacts. It is therefore interesting to solve the

problem indirectly by the following steps and the use of the projection of compliance through condensation, as seen in Eq. 5.31.

$$\left\{ \begin{array}{l} \text{Step 1 : solve the linear system :} \\ \left[\begin{array}{c|c} \mathbf{A} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{A}_e \end{array} \right] \left[\begin{array}{c} d\mathbf{q}^0 \\ d\mathbf{q}_e^0 \end{array} \right] - \left[\begin{array}{c} \mathbf{b} \\ \mathbf{b}_e \end{array} \right] = \mathbf{0} \\ \\ \text{Step 2 : solve the Mixed Complementarity Problem (find } \boldsymbol{\lambda}_a \text{ and } \boldsymbol{\lambda}_c \text{) :} \\ \left\{ \begin{array}{l} \delta_a + \underbrace{\mathbf{H}_a \mathbf{A}^{-1} \mathbf{H}_a^T}_{\mathbf{W}_{aa}} \boldsymbol{\lambda}_a + \underbrace{\mathbf{H}_a \mathbf{A}^{-1} \mathbf{H}_c^T}_{\mathbf{W}_{ac}} \boldsymbol{\lambda}_c = \mathbf{u} \\ 0 \leq \delta_c + \underbrace{\mathbf{H}_c \mathbf{A}^{-1} \mathbf{H}_a^T}_{\mathbf{W}_{ca}} \boldsymbol{\lambda}_a + \underbrace{(\mathbf{H}_c \mathbf{A}^{-1} \mathbf{H}_c^T + \bar{\mathbf{H}}_c \mathbf{A}_e^{-1} \bar{\mathbf{H}}_c^T)}_{\mathbf{W}_{cc}} \boldsymbol{\lambda}_c \perp \boldsymbol{\lambda}_c \geq 0 \end{array} \right. \\ \\ \text{Step 3 : find the final motion :} \\ \left\{ \begin{array}{l} d\mathbf{q} = d\mathbf{q}^0 + \mathbf{A}^{-1} (\mathbf{H}_a^T \boldsymbol{\lambda}_a + \mathbf{H}_c^T \boldsymbol{\lambda}_c) \\ d\mathbf{q}_e = d\mathbf{q}_e^0 + \mathbf{A}_e^{-1} \bar{\mathbf{H}}_c^T \boldsymbol{\lambda}_c \end{array} \right. \end{array} \right. \quad (5.34)$$

To solve the linear system (step 1) and find the final motion (step 3), a factorization of matrices \mathbf{A} and \mathbf{A}_e or iterative solvers can be employed. This factorization can be reused to obtain matrices \mathbf{W}_{ij} . For step 2, a convex optimisation solver is used to find the solution to the Mixed Complementarity Problem (MCP).

5.9.3 Inverse Modeling with Contact

In Sect. 5.8.2 we saw that inverse modeling could be found on soft robots thanks to a QP (Quadratic Programming) optimization method based on condensed compliance models. In the previous section, we showed that this condensed compliance can also be found in step 2, at the contact level, to write the MCP (Mixed Complementarity Problem) that allows us to know the contact (and actuation) forces in a direct problem. We show here that we can combine these two optimization problems to find the inverse model of soft robots in contact with their environment. To do this the 3 steps described in Eq. (5.34) are repeated, but the MCP in step 2 is replaced a QPCC (Quadratic Problem with Complementarity Conditions) to follow a trajectory defined in a at the effector $\delta_e(t)$. At step 2, the goal is then to minimize $\delta_e(t)$ by using actuation and contacts.



Fig. 5.15 The goal position is illustrated by a green circle in both figures. On the left, the obstacle is not modeled in the inverse model. The obstacle prevents the real robot from reaching the goal. On the right, however, contact with the obstacle is taken into account in the inverse model, enabling the real robot to find a solution to the inverse problem of reaching the goal position while sliding over the obstacle

$$\left\{ \begin{array}{l} \text{Step 2 : solve the Quadratic Problem with Complementarity Constraints (QPCC)} \\ \min_{\Delta\lambda_a, \Delta\lambda_c} (\delta_{e0} + \Delta\delta_e)^2 \text{ (with } \Delta\delta_e = W_{ea}\Delta\lambda_a + W_{ec}\Delta\lambda_c) \\ \text{with the actuator constraints like} \\ \delta_{\min} \leq \delta_{a0} + \Delta\delta_a \leq \delta_{\max} \text{ (with } \Delta\delta_a = W_{aa}\Delta\lambda_a + W_{ac}\Delta\lambda_c) \\ \text{and with to contact constraints} \\ 0 \leq \delta_{c0} + \Delta\delta_c \perp \lambda_c + \Delta\lambda_c \geq 0 \text{ (with } \Delta\delta_c = W_{ca}\Delta\lambda_a + W_{cc}\Delta\lambda_c) \end{array} \right.$$

One result of using this optimization can be seen in the Fig. 5.15 from [31]. It shows a soft robot, driven by 8 cables and a translational movement of its base. It is shown that when the obstacle is taken into account, the inverse model finds (in an open loop) the actuation that enables it to slide along the obstacle in order to reach the goal (see the [movie](#)). Once again, optimization is based essentially on projected compliance in the actuator, effector and contact spaces.

5.10 Conclusion

The aim of this chapter is to demonstrate the modeling of soft robots through the calculation of mechanical compliance. We have discussed the definition of mechanical compliance, its numerical calculation depending on the material properties and using the finite element method. We've seen how it can be used to model soft robots behavior. We hope this chapter will help democratize this approach, which is not fully focused solely on the FEM aspect, but rather on numerical methods for condensed models. In particular, we feel it's important to emphasize that mechanical compliance can be calculated using numerical methods other than FEM.

We believe that our approach to the kinematics of soft robots, by passing this compliance projected into the robot's useful spaces (actuator spaces, effector spaces, sensor spaces, contact spaces, etc.), is today the most generic. In particular, the

properties of the (possibly heterogeneous) materials used to manufacture the robot are taken into account when calculating the robot's kinematics.

Only little discussion on the robots dynamics has been included in this chapter, as it was felt more important to focus on the mechanical compliance. However, the method is generally compatible with dynamics, especially if we combine with the model reduction method to make calculations fast enough.

Work has been carried out or is in progress to complement this approach, notably with closed-loop control (static and dynamic) [15, 23, 24, 26]. There is undoubtedly still a long way to go, particularly in the multi-physics aspects of sensors and actuators, but also in the acceleration of computation time to be able to use these approaches in on-line predictive control.

References

1. Frâncu M, Asgeirsson A, Erleben K, Rønnow MJL (2021) Locking-proof tetrahedra. ACM Trans Graph (TOG) 40(2):1–17
2. Vanneste F, Goury O, Duriez C (2021) Enabling the control of a new degree of freedom by using anisotropic material on a 6-DOF parallel soft robot. In: 2021 IEEE 4th international conference on soft robotics (RoboSoft). IEEE, pp 636–642
3. Vanneste Félix, Goury Olivier, Martinez Jonas, Lefebvre Sylvain, Delingette Hervé, Duriez Christian (2020) Anisotropic soft robots based on 3d printed meso-structured materials: design, modeling by homogenization and simulation. IEEE Robot Autom Lett 5(2):2380–2386
4. Mustaza SM, Elsayed Y, Lekakou C, Saaj C, Fras J (2019) Dynamic modeling of fiber-reinforced soft manipulator: a visco-hyperelastic material-based continuum mechanics approach. Soft Robot 6(3):305–317
5. Ferrentino P, Tabrizian SK, Brancart J, Assche GV, Vanderborght B, Terry S (2021) Fea-based inverse kinematic control: hyperelastic material characterization of self-healing soft robots. IEEE Robot Autom Mag 29(3):78–88
6. Marechal Luc, Balland Pascale, Lindenroth Lukas, Petrou Fotis, Kontovounios Christos, Bello Fernando (2021) Toward a common framework and database of materials for soft robotics. Soft Robot 8(3):284–297
7. Jian Qu, Khan Kamran, Meng Songhe, Muliana Anastasia (2023) Modeling nonlinear viscoelastic responses of flexible composites for soft robotics applications. Mech Adv Mater Struct 30(14):2793–2805
8. Przemieniecki JS (1985) Theory of matrix structural analysis. Courier Corporation
9. Zienkiewicz OC, Taylor RL (2005) The finite element method for solid and structural mechanics. Elsevier
10. Nguyen VP, Rabczuk T, Bordas S, Duflot M (2008) Meshless methods: a review and computer implementation aspects. Math Comput Simul 79(3):763–813
11. Leveziel M, Haouas W, Laurent GJ, Gauthier M, Dahmouche R (2022) Migrobot: a miniature parallel robot with integrated gripping for high-throughput micromanipulation. Sci Robot 7(69):eabn4292
12. Baraff D, Witkin A (1998) Large steps in cloth simulation. In: Computer graphics proceedings, annual conference series. Association for Computing Machinery SIGGRAPH, pp 43–54
13. Barbić J, James DL (2005) Real-time subspace integration for St. Venant-Kirchhoff deformable models. ACM Trans Graph (TOG) 24(3):982–990
14. Goury Olivier, Duriez Christian (2018) Fast, generic, and reliable control and simulation of soft robots using model order reduction. IEEE Trans Robot 34(6):1565–1576

15. Thieffry Maxime, Kruszewski Alexandre, Duriez Christian, Guerra Thierry-Marie (2018) Control design for soft robots based on reduced-order model. *IEEE Robot Autom Lett* 4(1):25–32
16. Rodríguez A, Coevoet E, Duriez C (2017) Real-time simulation of hydraulic components for interactive control of soft robots. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 4953–4958
17. Adagolodjo Yinoussa, Renda Federico, Duriez Christian (2021) Coupling numerical deformable models in global and reduced coordinates for the simulation of the direct and the inverse kinematics of soft robots. *IEEE Robot Autom Lett* 6(2):3910–3917
18. Guennebaud G, Jacob B, Avery P, Bachrach A, Barthelemy S et al (2010) Eigen v3
19. Davis T (2005) Csparse: a concise sparse matrix package
20. Duriez C, Cotin S, Lenoir J, Neumann P (2006) New approaches to catheter navigation for interventional radiology simulation. *Comput Aided Surg* 11(6):300–308
21. Saupin G, Duriez C, Cotin S (2008) Contact model for haptic medical simulations. In: International symposium on biomedical simulation. Springer, pp 157–165
22. Zeng Z, Courtecuisse H (2023) Efficient parallelization strategy for real-time FE simulations. [arXiv:2306.05893](https://arxiv.org/abs/2306.05893)
23. Koehler M, Bieze TM, Kruszewski A, Okamura AM, Duriez C (2023) Modeling and control of a 5-DOF parallel continuum haptic device. *IEEE Trans Robot*
24. Zhang Z, Dequidt J, Kruszewski A, Largilliere F, Duriez C (2016) Kinematic modeling and observer based control of soft robot using real-time finite element method. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 5509–5514
25. Navarro SE, Goury O, Zheng G, Bieze TM, Duriez C (2019) Modeling novel soft mechanosensors based on air-flow measurements. *IEEE Robot Autom Lett* 4(4):4338–4345
26. Bieze TM, Largilliere F, Kruszewski A, Zhang Z, Merzouki R, Duriez C (2018) Finite element method-based kinematics and closed-loop control of soft, continuum manipulators. *Soft Robot* 5(3):348–364
27. Teschner M, Kimmerle S, Heidelberger B, Zachmann G, Raghupathi L, Fuhrmann A, Cani M-P, Faure F, Magnenat-Thalmann N, Strasser W et al (2005) Collision detection for deformable objects. In: Computer graphics forum, vol 24. Wiley Online Library, pp 61–81
28. Kikuchi N, Oden JT (1988) Contact problems in elasticity: a study of variational inequalities and finite element methods. SIAM
29. Duriez C, Dubois F, Kheddar A, Andriot C (2005) Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Trans Vis Comput Graph* 12(1):36–47
30. Duriez C (2013) Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions. Université des Sciences et Technologie de Lille-Lille I
31. Coevoet Eulalie, Escande Adrien, Duriez Christian (2017) Optimization-based inverse model of soft robots with contact handling. *IEEE Robot Autom Lett* 2(3):1413–1419

Chapter 6

Soft Hands for Grasping and Manipulation



Shinichi Hirai

Abstract This chapter focuses on soft robotic hands. Soft hands composed of or involving soft materials exhibit adaptability and flexibility in object grasping and manipulation. First, we show how soft hands relax contact forces during grasping. Principles of soft robotic hands are then described to categorize the soft hands into soft pad hands, suction hands, jamming hands, bending finger hands, expanding membrane hands, and remotely deformed hands, with mechanics of soft hands. Analytical model for a soft pad hands and computational models for a bending finger hand and an expanding membrane hand are introduced. Soft robotic hands can grasp delicate and flexible food items, suggesting the soft hands are applicable to food manipulation. Thus, applications of soft robotic hands to food manipulation are described, focusing on tailored design of soft robotic hands for various food items. Soft sensors for soft robotic hands are then described, starting from their principles, including resistive, capacitive, and magnetic sensing, to entropy reduction in soft tactile sensing.

6.1 Introduction

Manipulation is an action to operate objects in the environment. Human hands perform various operations such as writing, peeling, and mating (Fig. 6.1), which are all manipulation. In writing, a human grasps and moves a pen to draw characters on a paper. In peeling, one hand grasps a fruit and another hand grasp a knife. Moving both hands cooperatively, peeling is performed. In mating, a pen and its cap are grasped and are aligned to put the cap on the pen. Manipulation is performed directly by human hands or through tools such as knives, folks, and screw drivers. Robotic manipulation is an attempt to perform manipulative operations by mechanical devices, which are often referred to as robotic hands or end-effectors.

Forces are applied to objects during manipulation. Human hands often apply normal and friction forces during manipulative operations. Adhesion forces can be also applied in human manipulation. For example, when humans pick up sesame seeds

S. Hirai (✉)

Wakakusa 12, Kusatsu, Shiga 5250045, Japan

e-mail: hirai@se.ritsumei.ac.jp

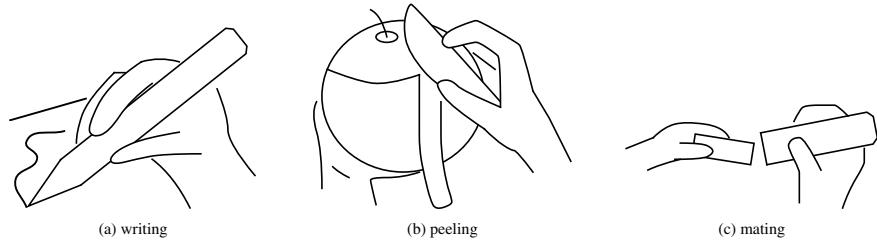


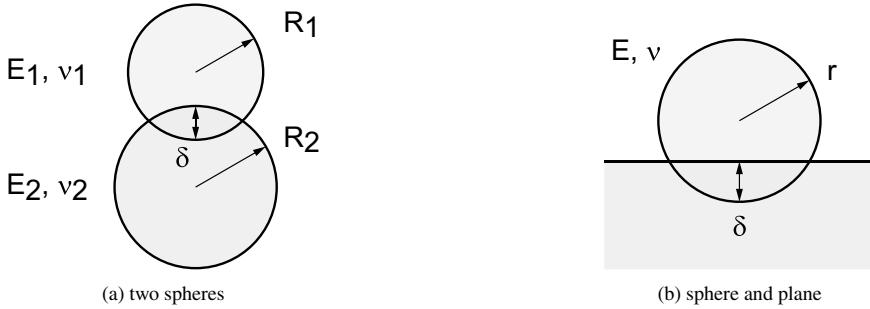
Fig. 6.1 Human manipulation

on a table by their wet fingers, wet fingers apply adhesion forces to the seeds so that the seeds stick to the fingers. Note that normal forces, friction forces, and adhesion forces require contact between human hands and objects. During robotic manipulation, robotic hands can apply not only contact forces but also non-contact forces such as suction forces, Bernoulli forces, electrostatic forces, and magnetic forces. However, non-contact manipulation tends to yield instability in object holding. Thus, robotic hands basically contact with objects to hold them stably. For example, suction cups contact with objects to hold the objects by suction forces. Consequently, contact between hands and objects is essential for manipulation.

Manipulation includes grasping and releasing. Initially, hands are not in contact with objects to be manipulated. Grasping is a process from the initial non-contact state to a contact state where hands are in contact with the objects. Releasing is a process from a contact state to the final non-contact state after manipulation. Regrasping is a process from one contact state to another. Namely, a manipulation process is a series of transitions among the non-contact state and contact states. When a hand grasps an object, contact between the two causes a collision force. When both the hand and the object are hard, a large collision force might damage the hand or the object. *Soft robotic hands* are then applied to robotic manipulation to relax collision forces. A soft hand and an object yield wider contact area, resulting small stress and less deformation of the grasped object. Additionally, soft hand deformation around the contact area improves adaptability against uncertainties of object geometry and surface condition. Therefore, soft hands can manipulate delicate objects or objects with large uncertainties. This chapter describes soft robotic hands for manipulation.

6.2 How Soft Hands Relax Contact Forces

Let us investigate how soft hands relax contact forces during grasping based on Hertzian contact model [1, 2]. Hertzian contact model formulates the contact between two spherical elastic bodies (Fig. 6.2a). Radius, Young's modulus, and Poisson's ratio of one spherical body are denoted by R_1 , E_1 , and ν_1 ; those of another body are R_2 , E_2 , and ν_2 . Two spherical bodies are pushed each other, resulting that the distance between two bodies is below $R_1 + R_2$. Let δ be the pushed distance from the initial

**Fig. 6.2** Hertzian contact

contact. Namely, the distance between the two centers is given by $R_1 + R_2 - \delta$. Reaction force applied to each body is then described as follows:

$$F = \frac{4}{3} \bar{E} \bar{R}^{\frac{1}{2}} \delta^{\frac{3}{2}}, \quad (6.1)$$

where

$$\frac{1}{\bar{E}} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}, \quad \frac{1}{\bar{R}} = \frac{1}{R_1} + \frac{1}{R_2}.$$

Assume that a spherical fingertip is in contact with a flat surface of a rigid body (Fig. 6.2b). Radius, Young's modulus, and Poisson's ratio of the fingertip are denoted by r , E , and ν . Letting $R_1 = r$, $E_1 = E$, $\nu_1 = \nu$, $R_2 = \infty$, and $E_2 = \infty$, we have

$$F = \frac{4}{3} \frac{E}{1 - \nu^2} r^{\frac{1}{2}} \delta^{\frac{3}{2}}. \quad (6.2)$$

The maximum contact force in grasping can be determined based on physical properties of target objects. For example, the maximum contact force should be small for delicate or fragile objects. Let F^* be the maximum force allowed in grasping. Then, the maximum value of pushed distance δ is given by

$$\delta^* = \left(\frac{3F^*}{4} \frac{1 - \nu^2}{E} r^{-\frac{1}{2}} \right)^{\frac{2}{3}}$$

which implies

$$\delta^* \sim E^{-\frac{2}{3}}$$

Compare two elastic fingertips; one is made of rubber while the another is made of steel. Let δ_{rubber}^* and δ_{steel}^* are the maximum values of δ for the two fingertips. Young's

modulus E_{rubber} of rubber is about or less than 100 MPa and Young's modulus E_{steel} of steel is over 100 GPa. Then, we have

$$\frac{E_{\text{rubber}}}{E_{\text{steel}}} \leq 10^{-3},$$

which directly yields,

$$\frac{\delta_{\text{rubber}}^*}{\delta_{\text{steel}}^*} \geq 100.$$

The maximum value δ^* determines allowable tolerance of geometry and error in positioning. For example, when $\delta_{\text{rubber}}^* = 1 \text{ mm}$, we find $\delta_{\text{steel}}^* \leq 10 \mu\text{m}$. This implies that a rubber fingertip accepts 1 mm tolerance or error at maximum but a steel fingertip cannot accept tolerance or error over $10 \mu\text{m}$. Tolerance or error below $10 \mu\text{m}$ is acceptable in electric or automobile industries but not acceptable in human daily life and in food or garment industries. This is the reason why soft hands are effective for manipulation in human daily life and industries with much uncertainty in geometry and positioning of target objects.

6.3 Soft Hand Classification

6.3.1 *Passive/Active Deformation in Soft Hands*

Soft hands are totally or partially made of soft materials to relax contact forces during grasping and manipulation. Soft materials in soft hands deform passively or actively. Focusing on the deformation, soft hands can be divided into two categories: *passive deformation hands* and *active deformation hands*. Soft materials in passive deformation hands deform according to external forces applied to the materials. Soft materials in passive deformation hands are attached to hard materials. Motion of the hard materials drives the soft materials, yielding their contact with target objects and their passive deformation. On contrary, soft materials in active deformation hands deform through internal forces, which can be controlled by hand users. Soft materials in active deformation hands are able to deform without the contact with target objects. Active deformation of the soft materials yields the contact with target objects for their grasping and manipulation.

Passive deformation hands include soft pad hands, suction hands, and jamming hands. Active deformation hands include bending finger hands, expanding membrane hands, and remotely deformed hands.

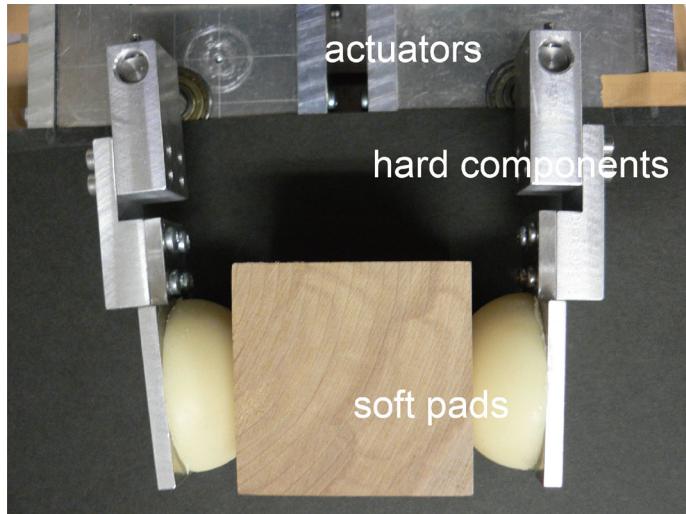


Fig. 6.3 Soft pad hand [8]

6.3.2 Soft Pad Hands

Soft pad hands [3–7] are categorized into passive deformation hands. A soft pad hand has multiple fingers. Each finger consists of a pad consisting of soft materials attached to a component made of hard materials (Fig. 6.3). Hard components are driven by actuators and soft pads move according to the motion of the hard components. Soft pads then deform passively when they are in contact with target objects to be grasped.

Since soft pads are in contact with a target object during its grasping, impact forces at the contact can be mitigated. Additionally, as contact area between a soft pad and the object increases, friction between a soft pad and the target object increases during grasping along with the deformation of the pad, which enables stable grasping and manipulation. Furthermore, creating grooves and unevenness on the surface of a pad increases the friction. Note that grasping forces of a soft pad hand originate from actuators. Actuator torques are directly transmitted to soft pads through hard components. Consequently, a soft pad hand can generate relatively large grasping forces.

6.3.3 Suction Hands

Suction hands [9–12] are categorized into passive deformation hands. A suction hand consists of one or multiple suction cups made of soft materials. When soft suction cups contact with a target object to be grasped, the pressure inside suction cups is

reduced to generate suction forces, allowing the target object to be grasped or lifted. Even though suction forces are non-contact forces, suction cups are in contact with a target object to avoid instability in non-contact grasping and manipulation.

Soft suction cups reduce collision forces when the cups contact the target object. Note that suction cups deform passively when they are in contact with the target objects. A suction hand can generate a large lifting force when the contact surface of a target object is smooth and flat. On the other hand, suction hands are not effective when the surface is uneven, which may cause a gap between cups and the surface, resulting in insufficient suction pressure inside the cups. Despite of such shortage, various suction hands are commercially available due to their simple structure and low cost.

6.3.4 Jamming Hands

Jamming hands [13–17] are categorized into passive deformation hands. Elasticity and viscosity of powders increase when powder density increases, as contact and friction forces among the powers increase. Jamming hands apply this phenomenon, which is referred to as jamming. A jamming hand comprises an elastic bag filled with powders (Fig. 6.4a). The bag is connected to a negative pressure source. In the natural state where powder density is small, friction among powders is small, resulting in an external force easily deforming the bag. During grasping, a jamming hand is pushed into a target object in its natural state so that the bag deforms according to the object surface. Negative pressure is then applied inside the bag so that the powders are pushed against one another. This increases the density of the powder inside the bag, making the bag stiff. The bag remains its deformed shape along the target object surface, enabling the object to be lifted (Fig. 6.4b).

Since the elastic bag of a jamming hand deforms according to its contact with a grasped object, jamming hands are categorized into passive deformation hands. On the other hand, elasticity of the bag can be controlled actively by the negative pressure applied to the bag. This is a unique feature of jamming hands.

6.3.5 Bending Finger Hands

Bending finger hands are categorized into active deformation hands. A bending finger hand (Fig. 6.5) consists of one or multiple bendable fingers made of elastomer. Bending fingers are pneumatically or electrostatically driven.

PneuNet actuators are often applied to pneumatically-driven bendable finger hands [18–23]. A PneuNet actuator is an elastomer tube with multiple chambers aligned along its one side. Applying pneumatic or fluidic pressure into the chambers causes their expansion and adjacent chambers push against each other. Subsequently, the chamber side elongates more than its opposite side, resulting in flexural deformation of the actuator. PneuNet-based bendable finger hands consist of two or multiple

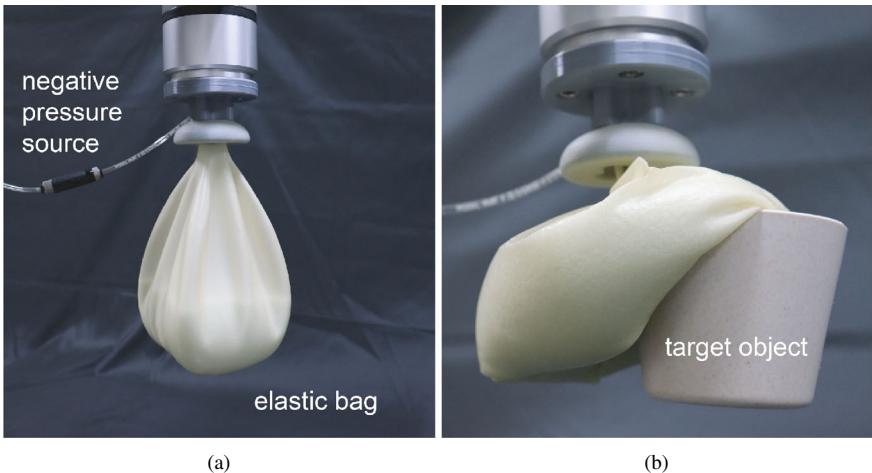
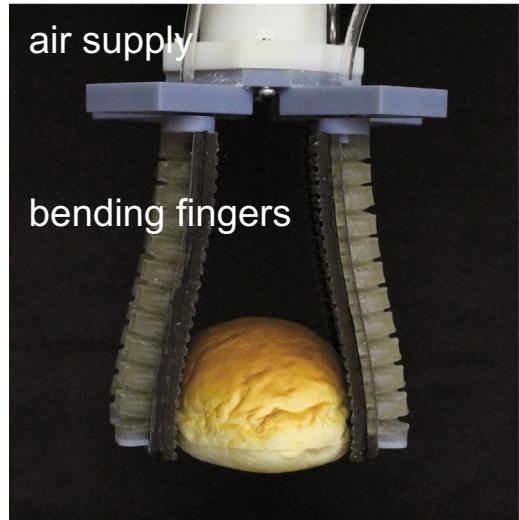


Fig. 6.4 Jamming hand

Fig. 6.5 Bending finger hand [24]



PneuNet actuators, which work as bendable fingers. A target object between bendable fingers or between a bendable finger and a hard component can be grasped when the fingers bend. Note that PneuNet-based bendable fingers deform actively with application of pneumatic or fluidic pressure.

We can fabricate PnetNet-based bending fingers by 3D printing or molding. Currently, elastic materials are available for several 3D printers. Parts of a PnetNet-based bending finger are printed and are glued to fabricate a bending finger. Using sacrifice material, a 3D printer can print a PnetNet-based bending finger, of which air chambers are filled with sacrifice material. Removing sacrifice material by heat or water,

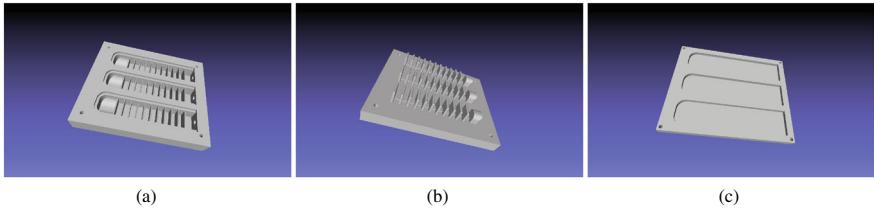


Fig. 6.6 Molds for bending fingers

we can fabricate a bending finger. Up to now, molding is widely applied to fabricate PnetNet-based bending fingers. Figure 6.6 shows molds for fabricating bending fingers. Liquid elastomers are poured into a bottom mold (Fig. 6.6a). Then, a chamber mold (Fig. 6.6b) is mated to the bottom mold to generate chambers. Curing liquid elastomers and detaching the bottom and chamber molds, we obtain chamber parts of bending fingers. Another mold (Fig. 6.6c) is for cover parts of bending fingers. Gluing two fabricated parts, we obtain PnetNet-based bending fingers.

Pneumatic bendable actuators involving multiple chambers are proposed [25, 26]. Applying pneumatic pressure to one or several chambers of a multi-chamber actuator causes its flexural deformation accordingly. A hand consisting of multi-chamber actuators can grasp a target object. Also, this hand can conduct in-hand manipulation of a grasped object by controlling pneumatic pressure applied to the chambers of the actuators. Bendable finger hands driven by balloon actuators are proposed [27, 28]. Balloons are attached on a surface of an elastomer finger. Expanding balloons causes flexural deformation of the finger. Thus, a hand consisting of balloon-based bendable fingers can grasp a target object by activating the balloons.

Dielectric elastomer actuators are applied to bendable finger hands [29, 30]. A dielectric elastomer actuator is composed of an elastomer sandwiched by two electrodes. This actuator expands when voltage is applied to the electrodes. Attaching a dielectric elastomer actuator on one side of a bendable film and activating the actuator cause flexural deformation of the file. Thus, a hand consisting of films with dielectric elastomer actuators can grasp a target object. Additionally, dielectric elastomer actuators yield electrostatic adhesion, which contributes to secure grasping.

6.3.6 *Expanding Membrane Hands*

Expanding membrane hands [31–34] are categorized into active deformation hands. An expanding membrane hand has one or several inflatable membranes made of elastomer attached to hard components (Fig. 6.7). Inflatable membranes expand when air pressure is applied to the membranes. A target object between inflatable membranes or between an inflatable membrane and a hard component can be grasped when the membranes expand. Inflatable membranes deform actively with application

Fig. 6.7 Expanding membrane hand



of air pressure. Expanding membrane hands grasp target objects through contact forces caused by pressure applied to inflatable membranes. The magnitude of contact forces depends on pressure and contacting area. Subsequently, increasing the pressure and the contact area increases the contact forces.

During grasping, inflatable membranes expand but hard components do not move. Additionally, since soft inflatable membranes and hard components can be thin, expanding membrane hands can be thin. As a result, an expanding membrane hand can grasp target objects in narrow space.

6.3.7 Remotely Deformed Hands

Remotely deformed hands are categorized into active deformatoin hands. A remotely deformed hand is composed of soft structures and actuators. Actuators are connected to soft structures. When the actuator applies a force or a displacement to the soft structure at the connected point, the structure deforms so that the part away from the connected point grasps a target object. Figure 6.8a shows an example of remotely deformed hands. This hand consists of a cap-shaped elastic structure and a pneumatic cylinder attached to the top of the structure. When a pneumatic cylinder pulls up the top, the bottom of the cup closes as shown in Fig. 6.8b, enabling grasp a taget object inside the cap.

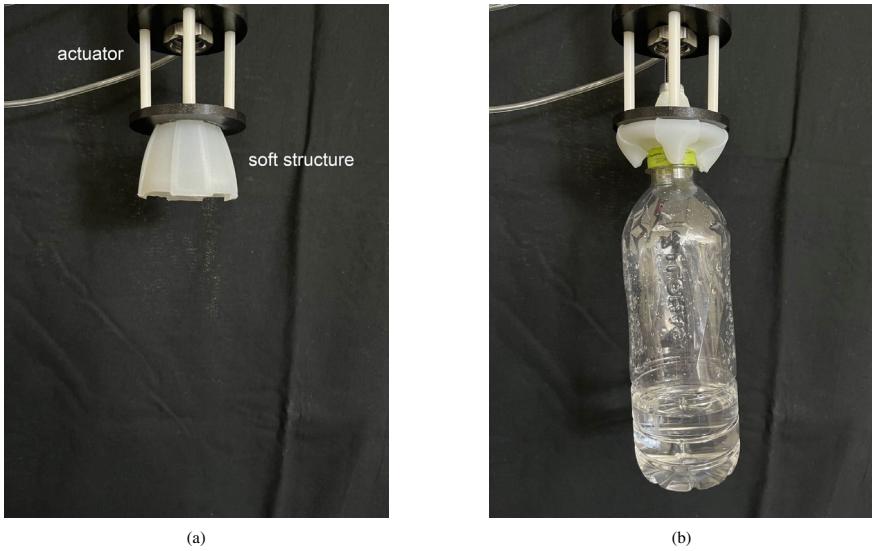


Fig. 6.8 Remotely deformed hand

6.4 Mechanics of Soft Hands

6.4.1 Mechanics of Soft Pad Hand

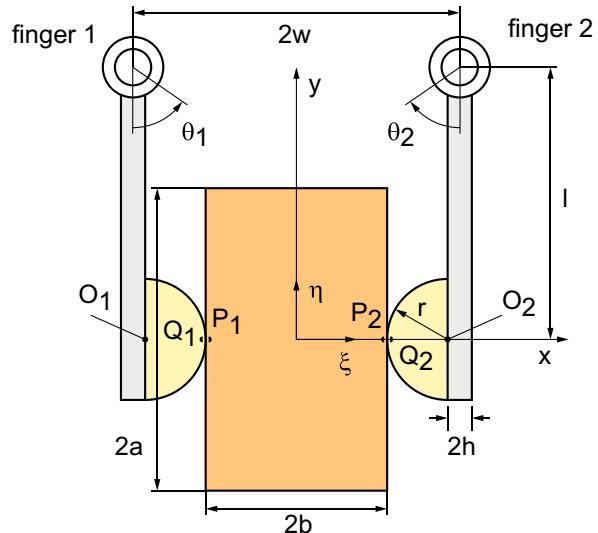
This section focuses on the mechanics of grasping performed by a soft pad hand with two fingers. A hemispherical soft pad is attached to one end of each finger while its other end is driven by a rotational actuator. The bottom surface of a hemispherical soft pad is fixed on the surface of each finger. Strain potential energy of a hemispherical soft pad is formulated as a closed-form function [8]. We apply the closed-form function to investigate the grasping performed by a soft pad hand. Soft pads consist of an elastic material with Young's modulus E . When a hemispherical soft pad is in contact with a flat surface of a rigid body, the strain potential energy of the pad is formulated as follows:

$$U(\theta, d_n, d_t) = \frac{\pi E d_n^3}{3 \cos^2 \theta} + \pi E (d_n^2 d_t \tan \theta + d_n d_t^2) \quad (6.3)$$

where d_n and d_t are the maximum normal and the tangential displacements, respectively, and θ is the relative angle between the fingertip and flat surface.

We formulate mechanics of a soft pad hand grasping a rectangle rigid body of its width $2a$ and height $2b$ (Fig. 6.9). A hemispherical soft pad of radius r is attached to a rigid finger of thickness $2h$, which is driven by a rotational actuator. Let l be the distance between the center of finger rotation and the line connecting the top and the center of a hemispherical pad. The hand consists of a pair of fingers. The distance

Fig. 6.9 Soft pad hand grasping rectangle rigid body



between two centers of finger rotation is given by $2w$. Let θ_1 and θ_2 be the rotational angles of the left and right fingers, respectively. The rotational angles increase when the fingers move inward and decrease when the fingers move outward. Let $O-xy$ be the spatial coordinate system while $C-\xi\eta$ be the object coordinate system. Position of the object is given by spatial coordinates of point C and its orientation is specified by the angle between x - and ξ -axes. Let $[x_{\text{obj}}, y_{\text{obj}}]^T$ be the positional vector of the object and θ_{obj} be its orientation angle. Assume that each pad is in one-point contact with a side of the rectangle at $\theta_1 = 0, \theta_2 = 0$, $[x_{\text{obj}}, y_{\text{obj}}]^T = [0, 0]^T$, and $\theta_{\text{obj}} = 0$, which directly yields $w = a + r + h$.

We analyze the contact between the left pad and the left side of the rectangle. Assume that the top point of the hemispherical soft pad is initially in contact with one point of the left side. Let P_1 be the top point and Q_1 be the point on the left side the top point is initially in contact with. The relative orientation angle between the left finger and the rectangle is equal to $\theta_{\text{obj}} - \theta_1$. We calculate the maximum normal displacement d_{n1} and the tangential displacement d_{t1} of the left pad. Let O_1 is the center of the hemispherical soft pad. Spatial position vector of point O_1 is described as:

$$\begin{bmatrix} O_{1x} \\ O_{1y} \end{bmatrix} = \begin{bmatrix} -w \\ l \end{bmatrix} + \begin{bmatrix} C_1 & -S_1 \\ S_1 & C_1 \end{bmatrix} \begin{bmatrix} h \\ -l \end{bmatrix}$$

where $C_1 = \cos \theta_1$ and $S_1 = \sin \theta_1$. The left side is described as $-\xi - a = 0$ in object coordinate system, which can be converted into $L(x, y) = -C_{\text{obj}}(x - x_{\text{obj}}) - S_{\text{obj}}(y - y_{\text{obj}}) - a = 0$ in spatial coordinate system, where $C_{\text{obj}} = \cos \theta_{\text{obj}}$ and $S_{\text{obj}} = \sin \theta_{\text{obj}}$. Function $L(x, y)$ represents a signed distance between the left side and $[x, y]^T$, which takes positive values in a half plane including point O_1 . Thus, distance

between the left side and point O_1 is given by $L(O_{1x}, O_{1y})$. Calculating $d_{n1} = r - L(O_{1x}, O_{1y})$, we have

$$d_{n1} = r - C_{\text{obj}}(x_{\text{obj}} + w) - S_{\text{obj}}(y_{\text{obj}} - l) + a + hC_{\text{obj}-1} - lS_{\text{obj}-1} \quad (6.4)$$

where $C_{\text{obj}-1} = \cos(\theta_{\text{obj}} - \theta_1)$ and $S_{\text{obj}-1} = \sin(\theta_{\text{obj}} - \theta_1)$. Let R_1 be an intersecting point of O_1P_1 and the left side of the rectangle. Note that tangential displacement d_{t1} is given by Q_1R_1 along the left side. Since $O_1R_1 = (r - d_{n1})/C_{\text{obj}-1}$, position vector of point R_1 is described as:

$$\begin{bmatrix} R_{1x} \\ R_{1y} \end{bmatrix} = \begin{bmatrix} O_{1x} \\ O_{1y} \end{bmatrix} + \frac{r - d_{n1}}{C_{\text{obj}-1}} \begin{bmatrix} C_1 \\ S_1 \end{bmatrix}$$

Noting that the unit tangential vector of the left side is given by $[S_{\text{obj}}, -C_{\text{obj}}]^T$, we have

$$d_{t1} = S_{\text{obj}}(Q_{1x} - R_{1x}) - C_{\text{obj}}(Q_{1y} - R_{1y}) \quad (6.5)$$

where

$$\begin{bmatrix} Q_{1x} \\ Q_{1y} \end{bmatrix} = \begin{bmatrix} C_{\text{obj}} & -S_{\text{obj}} \\ S_{\text{obj}} & C_{\text{obj}} \end{bmatrix} \begin{bmatrix} -a \\ 0 \end{bmatrix} + \begin{bmatrix} x_{\text{obj}} \\ y_{\text{obj}} \end{bmatrix} \quad (6.6)$$

Consequently, strain potential energy of the left pad is $U(\theta_{\text{obj}} - \theta_1, d_{n1}, d_{t1})$.

We analyze the contact between the right pad and the right side of the rectangle. Assume that the top point of the hemispherical soft pad is initially in contact with one point of the right side. Let P_2 be the top point and Q_2 be the point on the right side the top point is initially in contact with. The relative orientation angle between the right finger and the rectangle is equal to $\theta_{\text{obj}} + \theta_2$. We calculate the maximum normal displacement d_{n2} and the tangential displacement d_{t2} of the right pad. Let O_2 is the center of the hemispherical soft pad. Spatial position vector of point O_2 is described as:

$$\begin{bmatrix} O_{2x} \\ O_{2y} \end{bmatrix} = \begin{bmatrix} w \\ l \end{bmatrix} + \begin{bmatrix} C_2 & S_2 \\ -S_2 & C_2 \end{bmatrix} \begin{bmatrix} -h \\ -l \end{bmatrix}$$

where $C_2 = \cos \theta_2$ and $S_2 = \sin \theta_2$. The right side is described as $\xi - a = 0$ in object coordinate system, which can be converted into $R(x, y) = C_{\text{obj}}(x - x_{\text{obj}}) + S_{\text{obj}}(y - y_{\text{obj}}) - a = 0$ in spatial coordinate system. Function $R(x, y)$ represents a signed distance between the right side and $[x, y]^T$. Calculating $d_{n2} = r - R(O_{2x}, O_{2y})$, we have

$$d_{n2} = r + C_{\text{obj}}(x_{\text{obj}} - w) + S_{\text{obj}}(y_{\text{obj}} - l) + a + hC_{\text{obj}+2} + lS_{\text{obj}+2} \quad (6.7)$$

where $C_{\text{obj}+2} = \cos(\theta_{\text{obj}} + \theta_2)$ and $S_{\text{obj}+2} = \sin(\theta_{\text{obj}} + \theta_2)$. Let R_2 be an intersecting point of O_2P_2 and the right side of the rectangle. Note that tangential displacement

d_{t1} is given by Q_2R_2 along the right side. Since $O_2R_2 = (r - d_{n2})/C_{\text{obj}+2}$, position vector of point R_2 is described as:

$$\begin{bmatrix} R_{2x} \\ R_{2y} \end{bmatrix} = \begin{bmatrix} O_{2x} \\ O_{2y} \end{bmatrix} + \frac{r - d_{n2}}{C_{\text{obj}+2}} \begin{bmatrix} -C_2 \\ S_2 \end{bmatrix}$$

Noting that the unit tangential vector of the left side is given by $[S_{\text{obj}}, -C_{\text{obj}}]^T$, we have

$$d_{t2} = S_{\text{obj}}(Q_{2x} - R_{2x}) - C_{\text{obj}}(Q_{2y} - R_{2y}) \quad (6.8)$$

where

$$\begin{bmatrix} Q_{2x} \\ Q_{2y} \end{bmatrix} = \begin{bmatrix} C_{\text{obj}} & -S_{\text{obj}} \\ S_{\text{obj}} & C_{\text{obj}} \end{bmatrix} \begin{bmatrix} a \\ 0 \end{bmatrix} + \begin{bmatrix} x_{\text{obj}} \\ y_{\text{obj}} \end{bmatrix} \quad (6.9)$$

Consequently, strain potential energy of the right pad is $U(\theta_{\text{obj}} + \theta_2, d_{t2}, d_{l2})$.

Figure 6.10 shows a simulation of a soft pad hand grasping a rectangle rigid body. The soft pad hand can control the orientation of the grasped rectangle through rotational angles of both fingers. MATLAB programs are available in https://+++/Chapter_8_MATLAB_Programs.zip. Download and unzip the file, move to folder SoftPadHand, and run file `soft_pad_hand.m` for simulation.

6.4.2 Mechanics of Expanding Membrane Hand

This section focuses on the mechanics of grasping performed by a expanding membrane hand. We apply finite element analysis (FEA) to compute the deformation of elastic membranes. Figure 6.11 shows calculation of membrane deformation. An elastic membrane is attached to a rigid component (Fig. 6.11a). Membrane region is divided into 20×2 triangles with $n = 2 \times 21$ nodal points. Let P_i be the i -th nodal point and \mathbf{x}_i be its position at the natural shape. Nodal points move when the membrane deforms. Let \mathbf{u}_i be the displacement vector of nodal point P_i . Then, the deformation of the membrane is described by the following collective vector:

$$\mathbf{u}_N = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix} \quad (6.10)$$

which is referred to as *nodal displacement vector*. Let λ and μ be Lamé's constants of membrane material. Assume that λ and μ are constant over the membrane region.

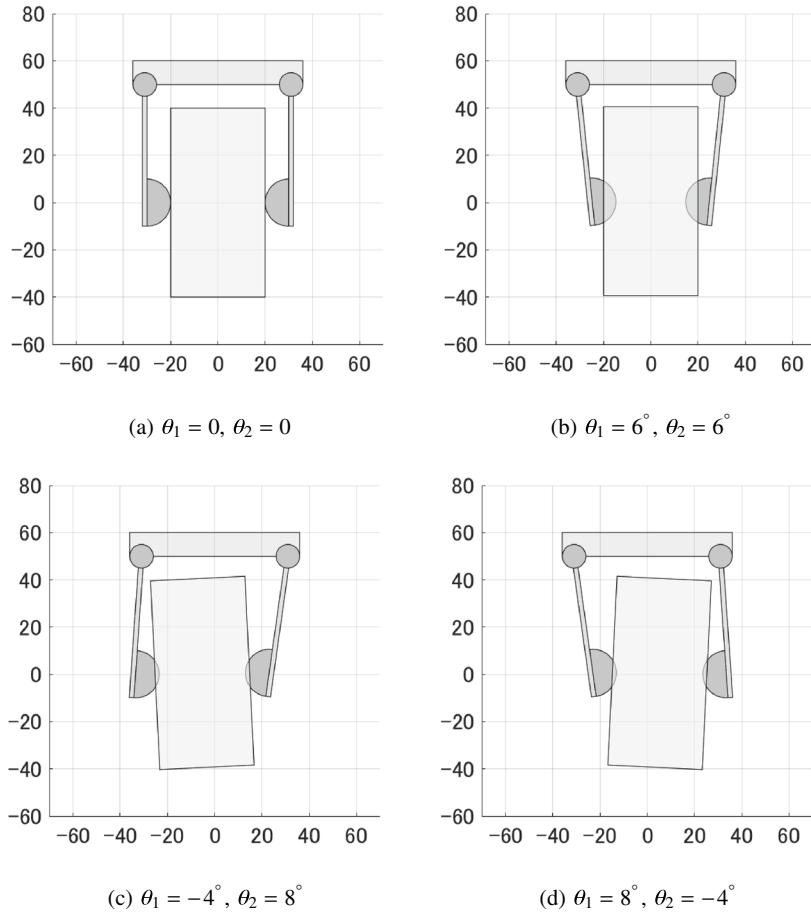


Fig. 6.10 Simulation of soft pad hand grasping rectangle rigid body

Strain potential energy stored in the membrane is then given by

$$U = \frac{1}{2} \mathbf{u}_N^T K \mathbf{u}_N \quad (6.11)$$

where

$$K = \lambda J_\lambda + \mu J_\mu \quad (6.12)$$

is referred to as *stiffness matrix*. Matrices J_λ and J_μ depend positional vectors of nodal points at the natural shape, that is, \mathbf{x}_1 through \mathbf{x}_n .

When air pressure p is applied into the chamber between the membrane and the rigid component, the membrane deforms (Fig. 6.11b). Let V be the volume of the membrane. Work done by air pressure is then described as

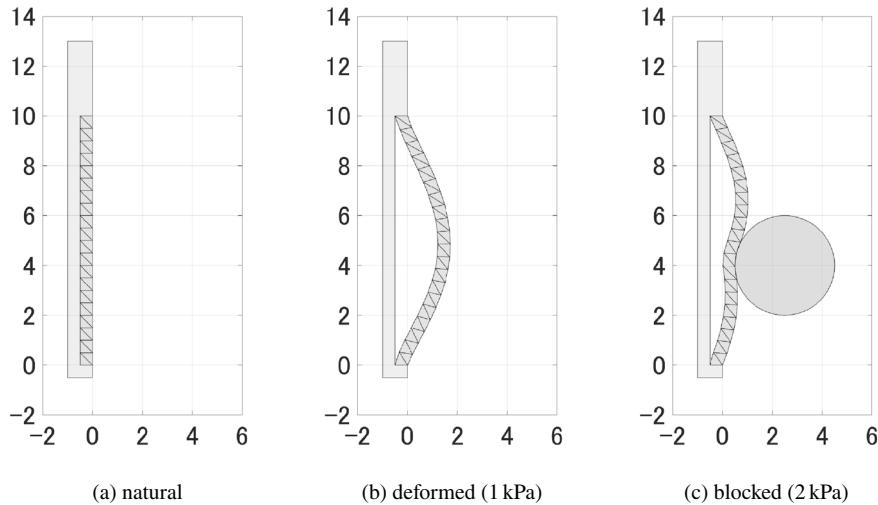


Fig. 6.11 Expansion of an elastic membrane

$$W = pV \quad (6.13)$$

Note that volume V of a function of nodal displacement vector \mathbf{u}_N . Top and bottom faces of the membrane are fixed to the rigid component, implying that displacement vectors vanish at nodal points on top and bottom faces. These geometric constraints are integrated into

$$A^T \mathbf{u}_N = \mathbf{0} \quad (6.14)$$

where matrix A specifies displacement vectors at nodal points on top and bottom faces.

According to variational principle in statics, internal energy $I = U - W$ reaches its minimum at the equilibrium under geometric constraints. Consequently, we can calculate membrane deformation by solving the following conditional minimization problem:

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{u}_N) = \frac{1}{2} \mathbf{u}_N^T K \mathbf{u}_N - p V(\mathbf{u}_N) \\ \text{subject to} \quad & A^T \mathbf{u}_N = \mathbf{0} \end{aligned} \quad (6.15)$$

This conditional minimization problem can be solved numerically, obtaining numerical solution of \mathbf{u}_N . Figure 6.11b shows a calculated deformation. Membrane of $10\text{ cm} \times 5\text{ mm} \times 1\text{ cm}$ is attached to a rigid component. The membrane is made of elastic material of Young's modulus 0.1 MPa and Poisson's ratio 0.48 , which are equivalent to $\lambda = 0.081\text{ MPa}$ and $\mu = 0.0034\text{ MPa}$. Pressure $p = 1\text{ kPa}$ is applied into the chamber. Membrane expansion can be computed well.

Assume that an obstacle is fixed in the environment. Let $g(\mathbf{x})$ be a function describing the obstacle. This function takes positive values inside the obstacle, negative values outside it, and zero along its boundary. Nodal points should be outside or on the boundary, we have the following inequalities:

$$g(\mathbf{x}_i + \mathbf{u}_i) \leq 0, \quad (i = 1, 2, \dots, n) \quad (6.16)$$

Then, we obtain the following conditional minimization problem:

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{u}_N) = \frac{1}{2} \mathbf{u}_N^T K \mathbf{u}_N - p V(\mathbf{u}_N) \\ \text{subject to} \quad & A^T \mathbf{u}_N = \mathbf{0} \\ & g(\mathbf{x}_i + \mathbf{u}_i) \leq 0, \quad (i = 1, 2, \dots, n) \end{aligned} \quad (6.17)$$

This conditional minimization problem can be solved numerically, obtaining numerical solution of \mathbf{u}_N . Figure 6.11c shows a calculated deformation. A circular obstacle of radius $r = 2$ cm is fixed at its center $\mathbf{c} = [2.5, 4.0]^T$ cm. The corresponding obstacle function is given by

$$g(\mathbf{x}) = r - \|\mathbf{x} - \mathbf{c}\|$$

Pressure $p = 2$ kPa is applied into the chamber. Membrane deformation can be computed well.

Figure 6.12 shows simulation results of the deformation of a membrane expanding hand. The hand is grasping a circular object of radius $r = 2$ cm fixed at its center \mathbf{c} . Figure 6.12a shows a hand consists of two membranes and a circular object. Pressure $p = 3$ kPa is applied into the two chambers. Figure 6.12b–d show the results with $\mathbf{c} = [0, 5]^T$ cm, $\mathbf{c} = [0, 4]^T$ cm, and $\mathbf{c} = [0, 6]^T$ cm, respectively. From the deformation of the membranes, we can calculate forces applied to the circular object by the membranes. Arrows in the figures indicate the applied forces. Summing up the forces that each membrane applies, we obtain grasping forces left and right fingers exert on the circular object. Let $\mathbf{f}_{\text{left}} = [f_{\text{left}}^x, f_{\text{left}}^y]^T$ and $\mathbf{f}_{\text{right}} = [f_{\text{right}}^x, f_{\text{right}}^y]^T$ be grasping forces exerted by left and right fingers. Table 6.1 summarizes the calculated grasping forces. When this expanding membrane hand picks up a circular object on a table, the hand applies grasping forces \mathbf{f}_{left} and $\mathbf{f}_{\text{right}}$ as well as friction forces. Friction forces are described as $\mu R(\pi/2)\mathbf{f}_{\text{left}}$ and $\mu R(-\pi/2)\mathbf{f}_{\text{right}}$, where μ denotes coefficient of friction and $R(\theta)$ represents rotation matrix of angle θ . Thus, total force applied to the circular object at its picking is formulated as follows:

$$(\mathbf{f}_{\text{left}} + \mu R(\pi/2)\mathbf{f}_{\text{left}}) + (\mathbf{f}_{\text{right}} + \mu R(-\pi/2)\mathbf{f}_{\text{right}})$$

Let m be the mass of the circular object and g be the acceleration of gravity. Vertical component of the total force should exceed or equal to mg so that the circular object can be picked up:

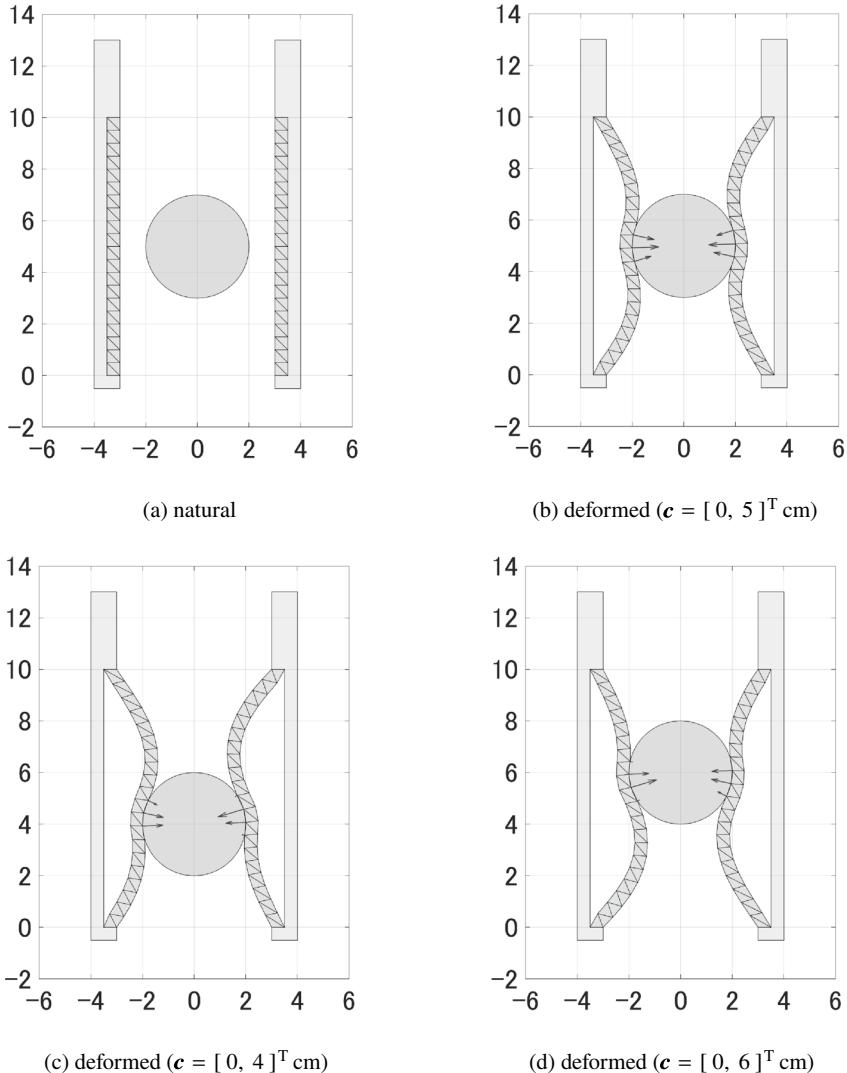


Fig. 6.12 Simulation results of expanding membrane hand

$$(f_{\text{left}}^y + \mu f_{\text{left}}^x) + (f_{\text{right}}^y - \mu f_{\text{right}}^x) \geq mg$$

yielding

$$\begin{aligned} \mu &\geq \mu_{\text{lower}} + \mu_{\text{inc}} m \\ \mu_{\text{lower}} &= \frac{-(f_{\text{left}}^y + f_{\text{right}}^y)}{f_{\text{left}}^x - f_{\text{right}}^x}, \quad \mu_{\text{inc}} = \frac{g}{f_{\text{left}}^x - f_{\text{right}}^x}, \end{aligned}$$

Table 6.1 Calculated grasping forces and friction parameters

c (cm)	$[0, 5]^T$	$[0, 4]^T$	$[0, 6]^T$
f_{left} (N)	$\begin{bmatrix} 1.2487 \\ 0.0348 \end{bmatrix}$	$\begin{bmatrix} 1.2166 \\ -0.2028 \end{bmatrix}$	$\begin{bmatrix} 1.2436 \\ 0.2227 \end{bmatrix}$
f_{right} (N)	$\begin{bmatrix} -1.2487 \\ -0.0348 \end{bmatrix}$	$\begin{bmatrix} -1.2436 \\ -0.2227 \end{bmatrix}$	$\begin{bmatrix} -1.2166 \\ 0.2028 \end{bmatrix}$
μ_{lower}	0.00	0.17	-0.17
μ_{inc} (1/g)	0.0039	0.0040	0.0040

Calculated values of μ_{lower} and μ_{inc} are summarized in the table. Figure 6.13 describes the minimum coefficient of friction to pick up a circular object of mass m . When $c = [0, 4]^T$ cm (Fig. 6.12c), larger coefficient of friction is required. Note that this figure describes the coefficient of friction at the moment that the hand picks up the circular object. More friction is required after the moment.

MATLAB programs are available in https://+++/Chapter_8_MATLAB_Programs.zip. Download and unzip the file, and move to folder ExpandingMembraneHand. File `expanding_membrane.m` simulates membrane deformation (Fig. 6.11). File `expanding_membrane_hand.m` simulates the deformation of a membrane expanding hand grasping a circular object (Fig. 6.12). File `expanding_membrane_hand_forces.m` calculates grasping forces (f_{left} and f_{right}) and friction parameters (μ_{lower} and μ_{inc}) in Table 6.1.

6.4.3 Mechanics of Bending Finger Hand

This section describes calculation of static deformation of a PneuNet actuator (Fig. 6.14a). This actuator is composed of elastic material of Young's modulus $E = 0.1$ MPa and Poisson's ratio $\nu = 0.48$, and involves a series of three air chambers along its left side. Air pressure is applied inside the actuator, expanding the air chambers, which yields the bend deformation of the actuator (Fig. 6.14b, c). Nodal points on the bottom side are fixed to the floor. Actual PneuNet actuators are three-dimensional; junctions between neighboring air chambers and right side of the actuator are connected by front and back elastic covers, resulting that distance between a junction and the right side remains almost constant. So, in this calculation, we impose two additional geometric constraints that the distances between individual junctions and the right side remain constant. The two junction are specified by P_{i1} and P_{i2} , their corresponding nodal points are given by P_{j1} and P_{j2} , respectively. Consequently, we have the following conditional optimization problem:

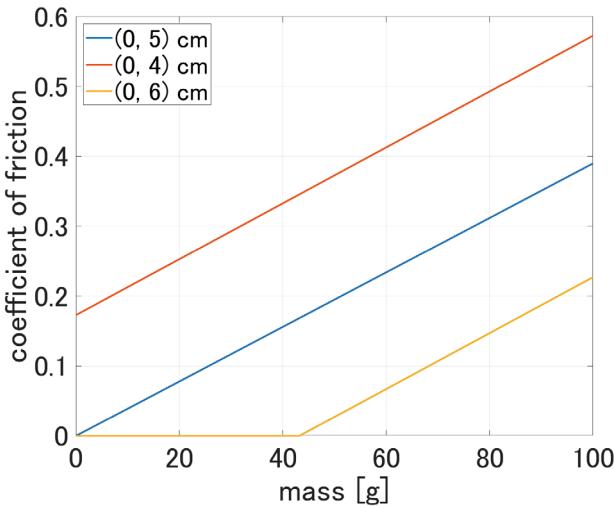


Fig. 6.13 Minimum coefficient of friction to pick up a circular object

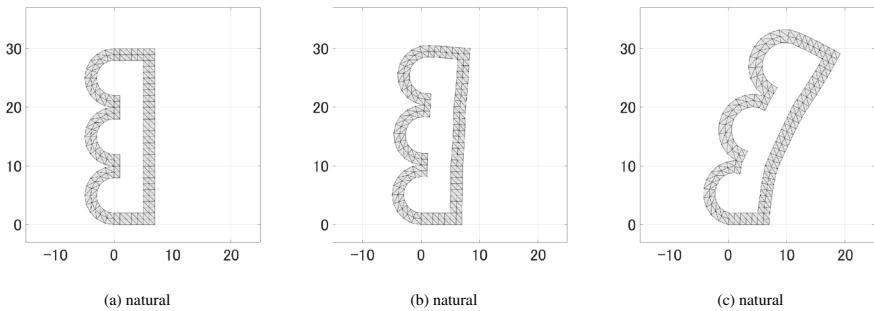


Fig. 6.14 Deformation of PneuNet actuator

$$\begin{aligned}
 & \text{minimize} && I(\mathbf{u}_N) \\
 & \text{subject to} && A^T \mathbf{u}_N = \mathbf{0} \\
 & && R_1 = \|\mathbf{r}_{i1} - \mathbf{r}_{j1}\| - \|\mathbf{x}_{i1} - \mathbf{x}_{j1}\| = 0 \\
 & && R_2 = \|\mathbf{r}_{i2} - \mathbf{r}_{j2}\| - \|\mathbf{x}_{i2} - \mathbf{x}_{j2}\| = 0
 \end{aligned} \tag{6.18}$$

where matrix A originates from constraints imposed on nodal points on the bottom side and $\mathbf{r}_k = \mathbf{x}_k + \mathbf{u}_k$. Figure 6.14b shows the deformation at applied pressure of 500 kPa and Fig. 6.14c shows the deformation at applied pressure of 700 kPa. It turns out that applying 500 kPa pressure causes little deformation but 700 kPa pressure yields much deformation. This computations was performed by MATLAB running on Windows 10, i5–6300U CPU at 2.40 GHz with 8.0 GB memory. Computation time was about 55 min.

Let us impose equilibrium equation to speed up the above computation. Let λ is a set of Lagrange multipliers corresponding to $A^T \mathbf{u}_N = \mathbf{0}$, λ_1 and λ_2 are Lagrange multipliers corresponding to $R_1 = 0$ and $R_2 = 0$. The equilibrium equation is then described as follows:

$$K\mathbf{u}_N - \mathbf{f}_p - A\lambda - \mathbf{g}_1\lambda_1 - \mathbf{g}_2\lambda_2 = \mathbf{0}, \quad (6.19)$$

where K is the stiffness matrix, $\mathbf{f}_p = ph \partial S / \partial \mathbf{u}_N$ denotes a set of nodal forces caused by pressure p , $\mathbf{g}_1 = \partial R_1 / \partial \mathbf{u}_N$, and $\mathbf{g}_2 = \partial R_2 / \partial \mathbf{u}_N$. Since Lagrange multipliers are additional unknowns, we have the following conditional optimization problem:

$$\begin{aligned} & \text{minimize} && I(\mathbf{u}_N) \\ & \text{subject to} && A^T \mathbf{u}_N = \mathbf{0} \\ & && R_1 = \|\mathbf{r}_{i1} - \mathbf{r}_{j1}\| - \|\mathbf{x}_{i1} - \mathbf{x}_{j1}\| = 0 \\ & && R_2 = \|\mathbf{r}_{i2} - \mathbf{r}_{j2}\| - \|\mathbf{x}_{i2} - \mathbf{x}_{j2}\| = 0 \\ & && [K - A - \mathbf{g}_1 - \mathbf{g}_2] \begin{bmatrix} \mathbf{u}_N \\ \lambda \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \mathbf{f}_p \end{aligned} \quad (6.20)$$

The deformed shape of a PneuNet actuator can be calculated by solving the above conditional optimization problem numerically. We obtained the deformed shapes shown in Fig. 6.14. Computation time was less than 50 s, implying that imposing equilibrium equation realizes over 60 times speed up in calculation. MATLAB programs are available in https://+++/Chapter_8_MATLAB_Programs.zip. Download and unzip the file, and move to folder BendingFingerHand. File pneu_net_actuator.m simulates the deformation of a PneuNet actuator (Fig. 6.14).

6.4.4 Embodied Intelligence in Soft Hands

This section describes an example of embodied intelligence in soft hand grasping and manipulation. Embodied intelligence in grasping and manipulation originates from structures and materials of robot hands. Synthesizing structures and materials with control and computation enables better performance in a simple manner. In this section, we focus on the control of the orientation of the target object held by a soft pad hand with two fingers (Fig. 6.9). We demonstrate how softness of soft hands simplifies the control law.

Assume that a sensor measures the orientation of the held object and that the rotation of each finger is controlled based on the measured orientation. Let θ_{obj} be the orientation angle of the object and let θ_1 and θ_2 be the rotational angles of the

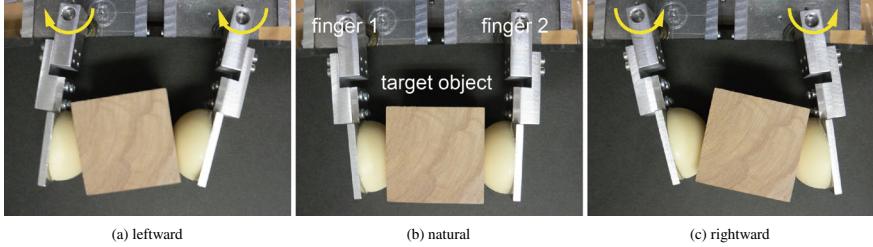


Fig. 6.15 Orientation of object grasped by soft pad hand [8]

left and right fingers, respectively. The rotational angles increased when the fingers moved inward and decreased when they moved outward. Let θ_{obj}^d be the desired angle of the object orientation. The goal of orientation control is to guide θ_{obj} to its desired value, θ_{obj}^d , by controlling the finger angles θ_1 and θ_2 .

Figure 6.15 shows the dependence of object orientation on the rotation of the two fingers. When both fingers rotate leftward (Fig. 6.15a), the object rotates rightward. Namely, when θ_1 decreases and θ_2 increases, the orientation angle θ_{obj} decreases, implying that when $\theta_{\text{obj}} > \theta_{\text{obj}}^d$, we should decrease θ_1 and increase θ_2 to reduce the orientation angle θ_{obj} . By contrast, when both fingers rotate rightward (Fig. 6.15c), the object rotates leftward. That is, when θ_1 increases and θ_2 decreases, the orientation angle θ_{obj} increases, implying that when $\theta_{\text{obj}} < \theta_{\text{obj}}^d$, we should increase θ_1 and decrease θ_2 to increase the orientation angle θ_{obj} .

We introduced virtual desired angles for the right and left fingers based on the aforementioned observations. We assume that the virtual desired angles are updated at a predetermined time interval T . Let $\theta_{1,k}^d$ and $\theta_{2,k}^d$ be the virtual desired angles in $[kT, (k+1)T]$. The updated laws are as follows:

$$\theta_{1,k}^d = \theta_{1,k-1}^d - K_1(\theta_{\text{obj}}(kT) - \theta_{\text{obj}}^d), \quad (k = 1, 2, \dots) \quad (6.21)$$

$$\theta_{2,k}^d = \theta_{2,k-1}^d + K_1(\theta_{\text{obj}}(kT) - \theta_{\text{obj}}^d), \quad (k = 1, 2, \dots) \quad (6.22)$$

where K_1 denotes a positive constant. Their initial values are expressed as follows:

$$\theta_{1,0}^d = -K_1(\theta_{\text{obj}}(0) - \theta_{\text{obj}}^d),$$

$$\theta_{2,0}^d = K_1(\theta_{\text{obj}}(0) - \theta_{\text{obj}}^d).$$

Equations (6.21) (6.22) show that $\theta_{1,k}^d$ decreases and $\theta_{2,k}^d$ increases when $\theta_{\text{obj}} > \theta_{\text{obj}}^d$ whereas $\theta_{1,k}^d$ increases and $\theta_{2,k}^d$ decreases when $\theta_{\text{obj}} < \theta_{\text{obj}}^d$. We apply the following proportional–derivative (PD) control laws to determine the torques for the rotational motions of the right and left fingers during $[kT, (k+1)T]$:

$$u_1 = -K_P(\theta_1 - \theta_{1,k}^d) - K_D\dot{\theta}_1 + \tau_b, \quad (6.23)$$

$$u_2 = -K_P(\theta_2 - \theta_{2,k}^d) - K_D\dot{\theta}_2 + \tau_b, \quad (6.24)$$

where τ_b denotes the positive bias torque that generates the holding force. The aforementioned control law has been demonstrated experimentally, and the object orientation angle θ_{obj} successfully converges to θ_{obj}^d , whereas θ_1 and θ_2 do not converge to their virtual desired angles, $\theta_{1,k}^d$ and $\theta_{2,k}^d$, respectively [8]. This implies that Eqs. (6.22) (6.21) of the control law should accept the deviations of θ_1 and θ_2 . When we apply proportional–integral–differential (PID) control laws rather than PD control laws to guide θ_1 and θ_2 to exactly $\theta_{1,k}^d$ and $\theta_{2,k}^d$, θ_1 and θ_2 may diverge or the contact between the fingers and the object may be lost, resulting in object–orientation control failure. Note that the potential energy should reach its local minimum at θ_{obj}^d such that the orientation angle θ_{obj} converges to θ_{obj}^d in a stable state, and the fingers and object should remain in contact during manipulation. The realization of the virtual desired angles $\theta_{1,k}^d$ and $\theta_{2,k}^d$ may break these conditions, implying that deviations in the control of θ_1 and θ_2 should be allowed.

6.5 Soft Hands for Food Manipulation

6.5.1 Background

Food manipulation in industry and at home depends on human hands. For example, various food items are packed into boxes by human hands in lunch box production. Food items are arranged on dishes by human hands in restaurants and diners. Automatic food manipulation is currently desired to cope with labor shortage and to ensure hygiene [35–37]. However, automatic food manipulation remains difficult up to now. This difficulty originates from variation of shapes and dimensions of individual food items. For example, individual fish fries have different shapes and dimensions. Robotic hands must cope with such geometric variation. Several food items are soft and fragile; it is necessary to avoid excess grasping forces during their grasping and manipulation. Additionally, food items are often randomly placed in a container, implying that robotic hands should cope with uncertainty in location of food items. Soft robotic hands can accept relatively larger geometric variation and uncertainty in location (see Sect. 6.2), suggesting that soft robotic hands are effective to automatic food manipulation. This section introduces soft robotic hands for food manipulation.

6.5.2 Soft Pad Hands for Food Manipulation

Binding hand

Many food materials are soft and easily deformed by external forces. Excessive deformation of food materials damages their quality, which should be avoided during their grasping and manipulation. One approach to reduce the deformation is to

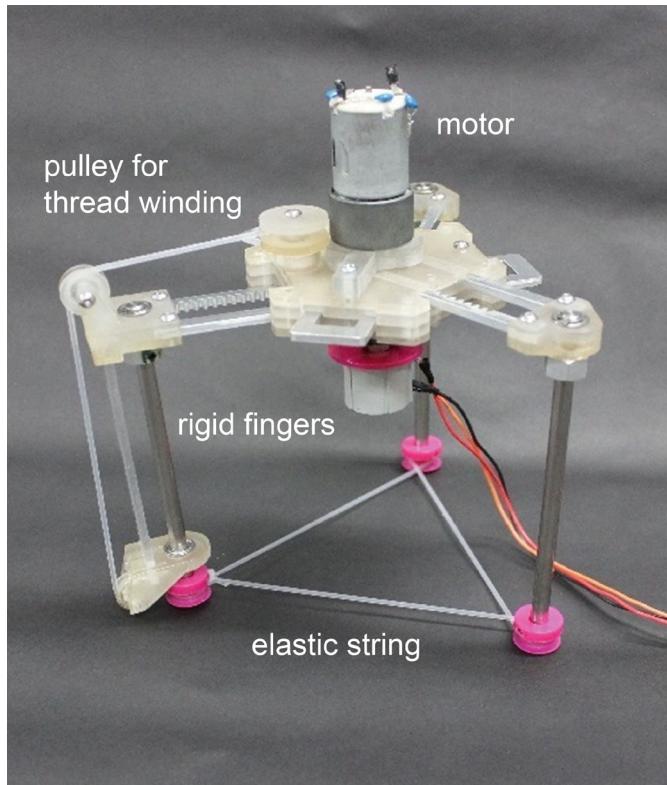


Fig. 6.16 Binding hand

increase contact area between a robotic hand and a grasped object. Binding hands [38] are designed to increase the contact area based on the concept of binding [39]. Figure 6.16 shows a binding hand, which consists of an elastic string and rigid fingers. One motor closes/opens a set of rigid fingers. The elastic string is wound/unwound in conjunction with closing/opening of the fingers. The elastic string contacts with a grasped object instead of rigid fingers to increase the contact area. Additionally, elastic strings can cope with irregular shape and size of food materials as the threads deform according to the shapes.

Figure 6.17 shows a process of binding hand grasping and picking up a paper cup with beans. Initially, the cup is inside the region surrounded by the elastic string (Fig. 6.17a). Then, the motor closes a set of fingers with winding the elastic string so that the string binds the cup (Fig. 6.17b). After binding, the binding hand picks up the bound cup (Fig. 6.17c). Binding hands are categorized into soft pad hands.

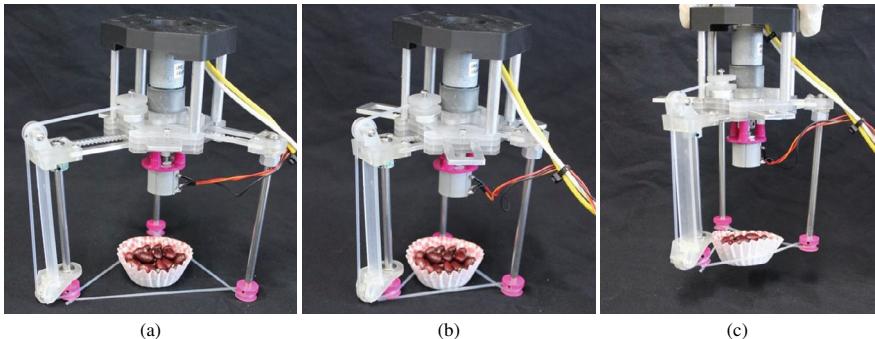


Fig. 6.17 Binding hand grasping and picking up a paper cup with beans [40]

Scooping-binding hand

Scooping-binding hands are designed to grasp slippery food materials [41]. Many fisheries products such as octopus and oysters have soft and slippery surfaces, making it difficult to grasp and manipulate such slippery food materials. A grasped slippery food material might fall during its manipulation due to insufficient friction on the slippery surfaces. Consequently, it is required to support such food materials during their grasping and manipulation so that the grasped food material does not fall. A scooping-binding hand consists of two fingers attached to a parallel gripper (Fig. 6.18). Each finger consists of two rigid rods, elastic strings stretched between the two rods, and a flexible thin plate attached to the bottom of the finger. Elastic strings apply grasping forces and a flexible thin plate support the grasped object, preventing it falling. Scooping-binding hands are categorized into soft pad hands.

Figure 6.19 shows the experimental result of food material grasping, indicating the success rate of five trials. The scooping-binding hand can grasp food materials with slippery surfaces (A: octopus leg, B: pollock roe, C: raw oyster) and irregular-shaped ones (D: meat, E: fried shrimp, F: green pepper) in a high success rate.

6.5.3 Expanding Membrane Hands for Food Manipulation

Planar shell hand

Planar shell hands [42] are categorized in expanding membrane hands. A planar shell hand has multiple fingers, each of which consists of an elastic membrane and a flat rigid shell attached to the membrane (Fig. 6.20a). The membrane is inflated by air pressure applied into the chamber between the membrane and the rigid shell (Fig. 6.20b). Expanding membranes apply contact forces to a grasped object for its grasping. Then, a planar shell hand can grasp and manipulate the grasped object.

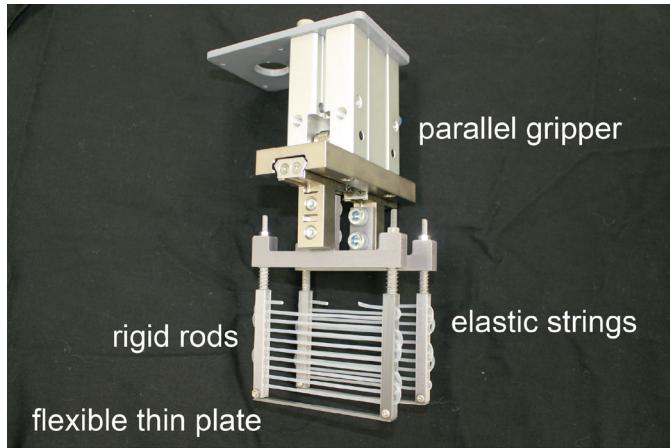


Fig. 6.18 Scooping binding hand

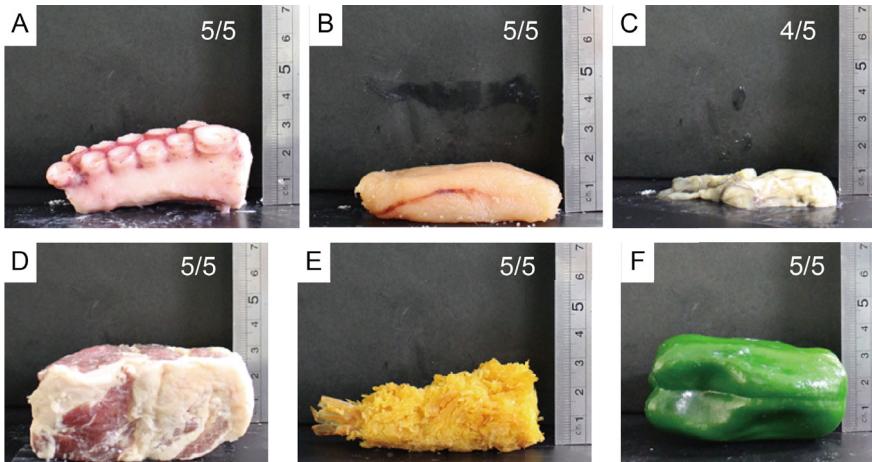


Fig. 6.19 Scooping-binding hand grasping food materials **a** octopus leg, **b** pollock roe, **c** raw oyster, **d** meat, **e** fried shrimp, **f** green pepper [41]

Planar shell hands are applied to packing of cucumbers into a box. Multiple cucumbers are on a conveyer (Fig. 6.21a). A set of several planar shell hands grasps multiple cucumbers simultaneously (Fig. 6.21b) to carton them in a box. Planar shell hands are thin and can be inserted into narrow areas, which contributes simultaneous grasping of multiple cucumbers and their packing.

Circular shell hand

Circular shell hands [33] are categorized in expanding membrane hands. A circular shell hand consists of a circular rigid shell and multiple membranes attached inside

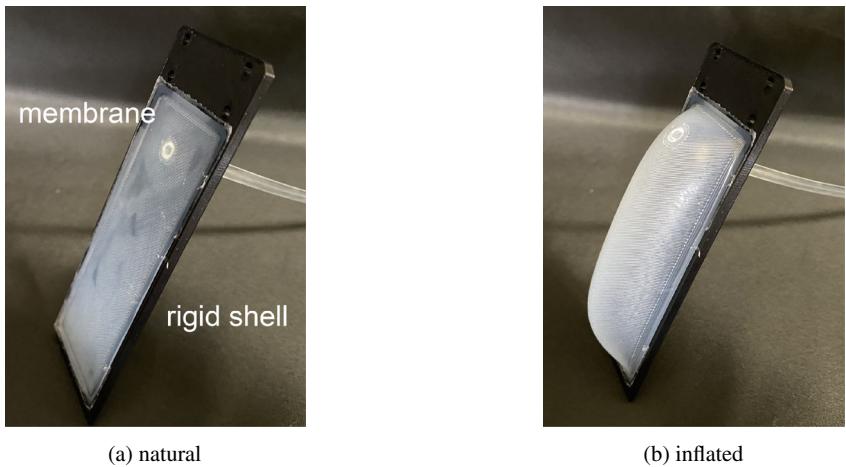


Fig. 6.20 Finger of planar shell hand [42]

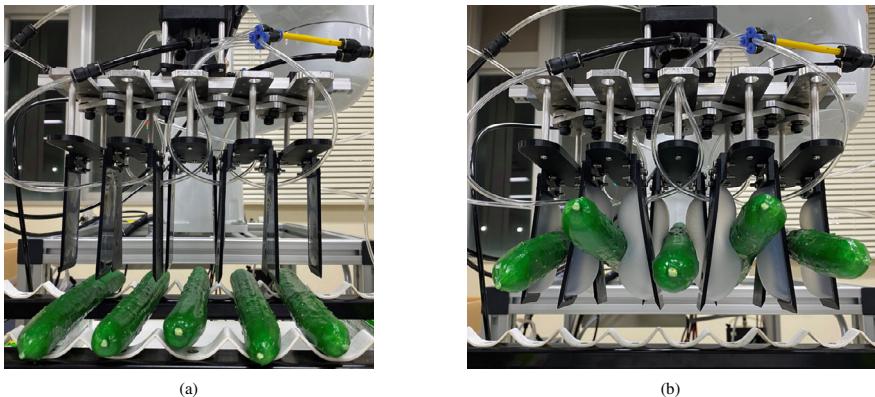


Fig. 6.21 Planar shell hands picking cucumbers [42]

the circular shell (Fig. 6.22a). The membranes expand by air pressure applied into the chambers between the membranes and the rigid plate (Fig. 6.22b). A circular shell hand can grasp a target object inside the circular shell. Contact forces are applied to the target object through expanding membranes around the object. Consequently, relatively large contact forces can be applied to the object, enabling a circular shell hand pick up a heavy target object. Figure 6.23 shows a process that a circular shell hand picks and places a rice cake.



Fig. 6.22 Circular shell hand

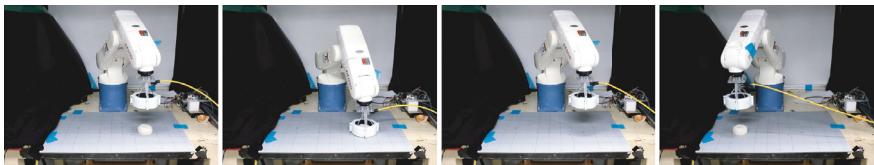


Fig. 6.23 Circular shell hand pick and place a rice cake

6.5.4 *Bending Finger Hands for Food Manipulation*

Curved finger hand

Curved finger hands are designed to grasp thin food items [24]. Normal bending finger hand contacts with a food item at its bent shape. Thus, fingers are often in contact with the food item at the finger edges, yielding instability of grasping. In contrast, a curved finger hand (Fig. 6.24) consists of naturally bent fingers. Fingers are bent outwards in their natural state. When air pressure is applied to the fingers, they bend inward, making them almost straight at a specific air pressure. Figure 6.25a shows a curved finger hand grasping a fried chicken. This curved finger hand can grasp thin food items such as a salmon slice (Fig. 6.25b) and a Japanese rolled omelet (Fig. 6.25c).

Wrapping hand

Wrapping hands [43] are categorized into bending finger hands. Wrapping hands are designed to grasp chopped or granular food materials such as cones and cereals. Usually, there are gaps between neighboring fingers at the closed state of a bending finger hand. Such gaps hinder grasping chopped or granular food materials. Thus,

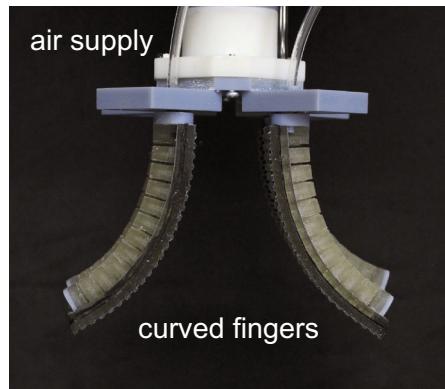


Fig. 6.24 Curved finger hand [24]

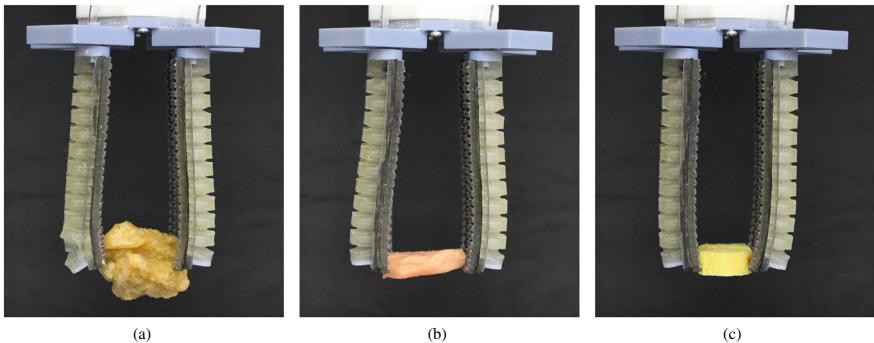


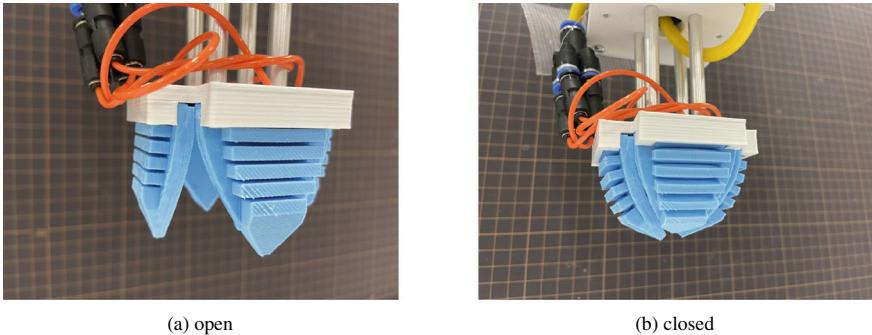
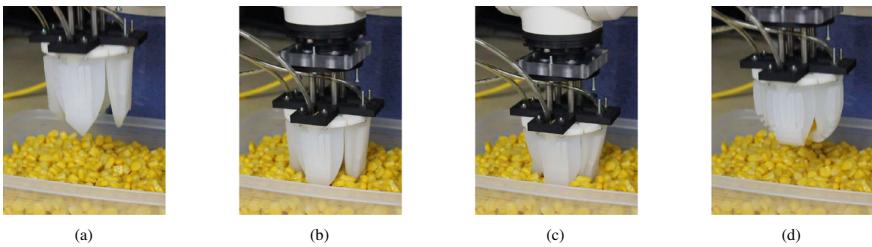
Fig. 6.25 Curved finger hand grasping food items [24]

fingers of a wrapping hand are designed so that gaps are small when fingers are closed. Figure 6.25a shows a wrapping hand consisting of four bending fingers. As shown in Fig. 6.25b, there are little gaps between neighboring fingers when fingers are closed. Figure 6.27 shows a process that a wrapping hand picks up cones. This wrapping hand can pick up cones without their dropping (Fig. 6.26).

6.6 Soft Sensors for Soft Hands

6.6.1 Principles

During human grasping and manipulation, we mainly use visual and tactile perceptions. We often use visual sensing to obtain the location of grasped objects. We can recognize softness of grasped objects through tactile perception. Additionally, we can control grasping forces via tactile sensing. When we grasp a paper cup

**Fig. 6.26** Wrapping hand**Fig. 6.27** Wrapping hand picking up cones [43]

(Fig. 6.28a), we apply relatively small grasping forces to prevent the cup from being broken. When water is poured into a paper cup (Fig. 6.28b), human fingers recognize incipient slip and increase grasping forces, avoiding the cup falling. Human perceptions are realized by sensors in artificial systems. Sensors are devices that transduce physical quantities into electric signals. Robots use sensors to recognize their own states or external environment.

Sensors applied to soft hands must operate normally regardless the deformation of the hands. One method is sensors composed of soft materials, that is, soft sensors. Soft sensors are attached on the surface or embedded inside the body of a soft robot. The attached or embedded sensors should not interfere with the deformation of the soft robot's body. Resistive film sensors and capacitive fiber sensors are involved in this category. Resistive film sensors are composed of thin plastic films, which are bendable but almost inextensible. When a resistive film sensor is applied to a bendable finger, the sensor should be embedded with the finger along its neutral line. Capacitive fiber sensors, which are composed of elastomers, are bendable and extensible. Both bending and extension affect the sensor output.

Another method is marker-detector pairs. Markers are on the surface or inside the body of a soft robot. Detectors measure the motion of the markers caused by the deformation of the soft body. Physical quantities are obtained from the measured marker motion. The detector can be placed outside the soft body, implying that it is

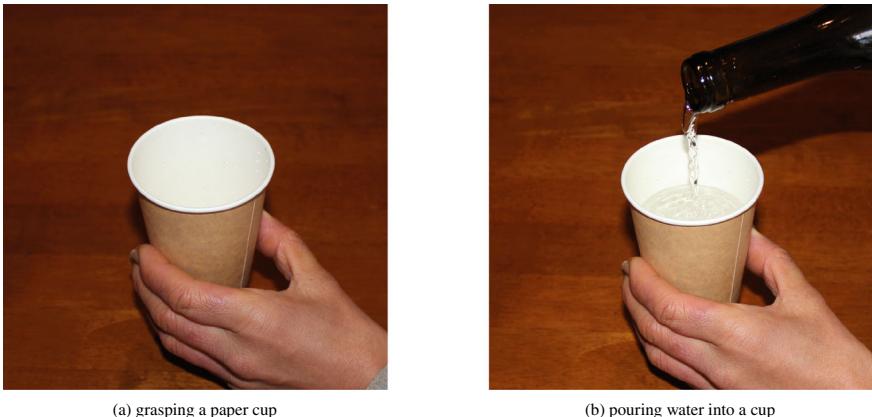


Fig. 6.28 Pouring water into a paper cup grasped by human hand

not necessary for detectors to be deformable. In magnetic sensors [44, 45], permanent magnets or magnetic elastomers work as magnetic markers. In image-based tactile sensors [46–48], image features attached to the surface of a soft body act as visual markers.

6.6.2 Resistive Sensors

Resistive sensors transduce physical quantities such as extensional and bending deformations to resistance. Resistance is then converted into voltage so that computers can read voltage values via analog-digital converters (ADCs).

Resistive extensional sensors

Resistive fiber sensors composed of soft materials can measure extensional deformation. Let us formulate the response of resistive fiber sensors. Assume that a resistive fiber sensor is a cylinder of its length L and cross-sectional area A (Fig. 6.29). The sensor is composed of uniform material of its Young's modulus E . When the sensor extends, length L increases and area A decreases. Assume that change of area A is negligible. Let x be the extension of the sensor. Force applied to the sensor is then described as follows:

$$f = \frac{EA}{L}x$$

Resistance R of the sensor is proportional to area A and is inversely proportional to length L :

$$R = \rho \frac{L}{A} \quad (6.25)$$

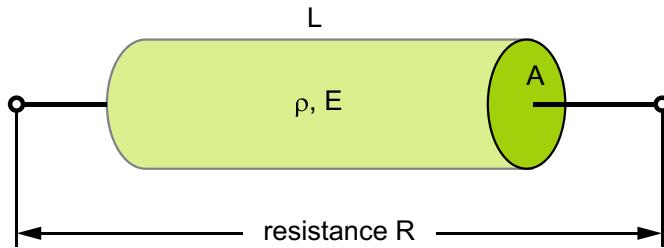


Fig. 6.29 Resistive sensor for extension

where ρ represents volume resistivity of sensor material. Let ΔR be the change of the resistance. Then, we have

$$R + \Delta R = \rho \frac{L + x}{A}$$

which directly yields:

$$\Delta R = \rho \frac{x}{A} \quad (6.26)$$

Consequently, we can estimate extension x by measuring resistance change ΔR .

The response of a resistive sensor depends on Young's modulus E and volume resistivity ρ , which are specific to the sensor material. Let us investigate the response of a resistive sensor of $L = 100$ mm and $A = 1$ mm 2 . Assume that the sensor is composed of an elastomer of Young's modulus $E = 10$ MPa and volume resistivity $\rho = 10^{12} \Omega \cdot \text{m}$. When the sensor extends by $x = 1$ mm, the applied force is $f = 0.1$ N while resistance increases by $\Delta R = 10^{15} \Omega$. The force is acceptable, but the change of the resistance is too large. This suggests that large volume resistivity values of elastomers are not suitable for resistive sensors. Powders of conductive materials are then mixed with elastomers to decrease volume resistivity values. Young's modulus of the mixtures does not increase significantly if the elastomers are dominant in extension. These mixtures are called conductive elastomers. For example, when $\rho = 10 \Omega \cdot \text{m}$, the change of the resistance is $\Delta R = 10^4 \Omega$, which is acceptable. Such conductive elastomers are often applied to resistive sensors.

Let us take the change of area A into formulation. Assume that area A decreases by a when the sensor of length L extends by x . Assume that the volume of the sensor remains despite of the extension. Then, we have $LA = (L + x)(A - a)$, which directly yields

$$a = A\epsilon_L$$

where

$$\epsilon_L = \frac{x}{L}$$

represents the extensional strain of the sensor. From Eq. (6.25), we find

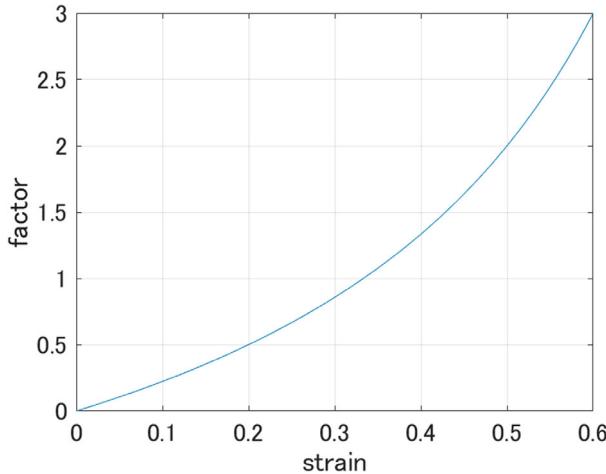


Fig. 6.30 Function $2\epsilon_L/(1 - \epsilon_L)$

$$R + \Delta R = \rho \frac{L + x}{A - a} = R \frac{1 + \epsilon_L}{1 - \epsilon_L}$$

This equation yields

$$\frac{\Delta R}{R} = \frac{2\epsilon_L}{1 - \epsilon_L} \quad (6.27)$$

or

$$\frac{x}{L} = \frac{\Delta R}{2R + \Delta R} \quad (6.28)$$

Consequently, we can estimate extension x by measuring resistance change ΔR . Figure 6.30 shows function $2\epsilon_L/(1 - \epsilon_L)$, which is monotonously increasing.

Resistive bending sensors

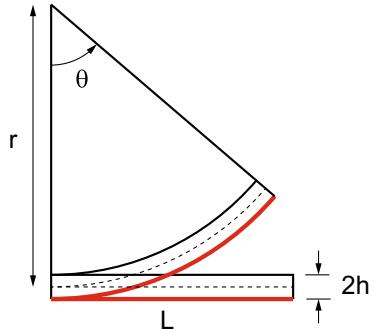
Resistive fiber sensors composed of soft materials can measure bending deformation. A thin film of its length L and thickness $2h$ bends (Fig. 6.31). The film composed of a plastic material is bendable but almost inextensible. A resistive fiber sensor is attached to one side of the film. We assume that the midplane of the film does not extend and that the bending deformation is uniform, implying that the sensor deforms to a circular arc. Let r be the radius of the arc and θ be its angle. Then, we have the following equation:

$$r\theta = L$$

Length of the extended side is described as $(r + h)\theta$. Extension of the sensor is then described as follows:

$$x = (r + h)\theta - L$$

Fig. 6.31 Resistive sensor for bending



which directly yields

$$x = h\theta \quad (6.29)$$

Consequently, we can estimate arc angle θ by measuring sensor extension x . Recall that extension x is estimated by Eq. (6.26) or Eq. (6.28). Curvature along the mid-plane is described as $\kappa = \theta/L$, which determines bending deformation. Note that h corresponds to the distance between the midplane and the sensor, which characterizes the accuracy of the bending measurement.

Converting resistance to voltage

Figure 6.32 shows circuits to convert resistance to voltage. Voltage divider circuit (Fig. 6.32a) consists of a reference resistance R_{ref} and a variable resistance R , which corresponds to a resistive sensor. Reference resistance R_{ref} is constant. Apply a constant input voltage V_{in} and measure output voltage V_{out} . Letting i be current, we have the following equations:

$$V_{\text{in}} = (R_{\text{ref}} + R)i$$

$$V_{\text{out}} = Ri$$

which yields

$$R = \frac{V_{\text{out}}}{V_{\text{in}} - V_{\text{out}}} R_{\text{ref}} \quad (6.30)$$

Note that R_{ref} and V_{in} are predetermined. Thus, we can calculate resistance R by measuring output voltage V_{out} .

Wheatstone bridge circuit (Fig. 6.32b) consists of four resistors R_1 through R_4 . Apply a constant input voltage V_{in} and measure output voltage V_{out} . Let i_{13} be current flowing through R_1 and R_3 , and i_{24} be current flowing through R_2 and R_4 . Then, we have the following equations:

$$V_{\text{in}} = (R_1 + R_3)i_{13}$$

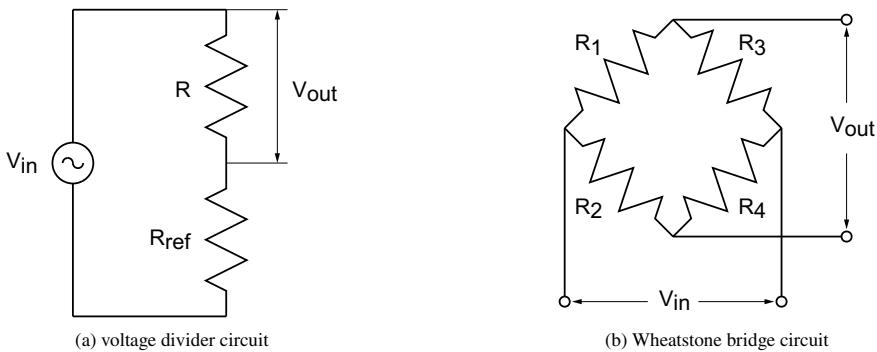


Fig. 6.32 Circuits to convert resistance to voltage

$$V_{\text{in}} = (R_2 + R_4)i_{24}$$

$$V_{\text{out}} = R_1 i_{13} - R_2 i_{24}$$

Eliminating i_{13} and i_{24} , we find

$$R_1 = \frac{s}{1-s} R_3 \quad (6.31)$$

where

$$s = \frac{V_{\text{out}}}{V_{\text{in}}} + \frac{R_2}{R_2 + R_4}$$

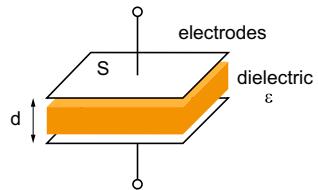
Assume that R_1 corresponds to a resistive sensor and R_2 through R_4 are constants. The above equations imply that we can calculate resistance R_1 by measuring output voltage V_{out} .

6.6.3 Capacitive Sensors

Capacitive sensors transduce physical quantities such as extensional and bending deformations to capacitance. Additionally, capacitive sensors can measure proximity and detect contact.

Response of capacitive sensors

Capacitive sensors consist of a dielectric and two electrodes attached to its both ends. Let us formulate the response of capacitive sensors. Assume that two planar electrodes are parallel each other (Fig. 6.33). Let S be the area of the electrodes and d be the distance between the two electrodes. Capacitance C of the sensor is proportional to area S and inversely proportional to distance d :

Fig. 6.33 Capacitive sensor

$$C = \epsilon \frac{S}{d} \quad (6.32)$$

where ϵ represents permittivity of dielectric material. Force or pressure applied to electrodes changes the distance between the electrodes. Let Δd be the change of distance d and ΔC be the change of the capacitance C . Then, we have

$$C + \Delta C = \epsilon \frac{S}{d + \Delta d}$$

which yields

$$\Delta C = \epsilon S \left\{ \frac{1}{d + \Delta d} - \frac{1}{d} \right\} \quad (6.33)$$

or equivalently

$$\Delta d = \epsilon S \left\{ \frac{1}{C + \Delta C} - \frac{1}{C} \right\} = d \left\{ \left(1 + \frac{\Delta C}{C} \right)^{-1} - 1 \right\} \quad (6.34)$$

Note that ΔC is positive when Δd is negative. From this equation, we can estimate Δd by measuring capacitance change ΔC . Capacitance sensing is affected by stray capacity. Thus, calibration of the capacitive sensors is required to reduce the influence of stray capacity.

Permittivity is specific to materials. Especially, permittivity of vacuum is given by the following equation:

$$\epsilon_0 = 8.854\,187\,813 \text{ pF/m}$$

Ratio between permittivity of a material and permittivity of vacuum is referred to as its relative permittivity. Namely, relative permittivity is defined as follows:

$$\epsilon_r = \frac{\epsilon}{\epsilon_0}$$

Note that relative permittivity is dimensionless. Let us investigate the response of a capacitive sensor with $S = 100 \text{ cm}^2$ and $d = 1 \text{ cm}$. Assume that dielectric is composed of an elastomer of relative permittivity $\epsilon_r = 2$. Capacitance of the nat-

ural state is then given by $C = \epsilon_r \epsilon_0 S/d \approx 17.7 \text{ pF}$. Then, we find $\Delta d \approx \{(1 + \Delta C/17.7 \text{ pF})^{-1} - 1\} \text{ cm}$, as plotted in Fig. 6.34.

Capacitive extensional sensors

Capacitive sensors can measure their extension. Assume that each electrode is a rectangle of length L and width W (Fig. 6.35a). Let d be the distance between the two electrodes. Capacitance between the two electrodes is then described as follows:

$$C = \epsilon \frac{Lw}{d}$$

Distance d and width w decreases when the sensor length L extends. Assume that distance d and width w decreases to αd and βw when length L increases by x (Fig. 6.35b). Then, we have the following equation:

$$C + \Delta C = \epsilon \frac{(L+x)\beta w}{\alpha d} = C(1 + \epsilon_L) \frac{\beta}{\alpha}$$

where $\epsilon_L = x/L$ represents the extensional strain. Especially, when cross-sectional area shrinks uniformly, that is, when $\alpha = \beta$, we find

$$\Delta C = C\epsilon_L = \frac{C}{L}x \quad (6.35)$$

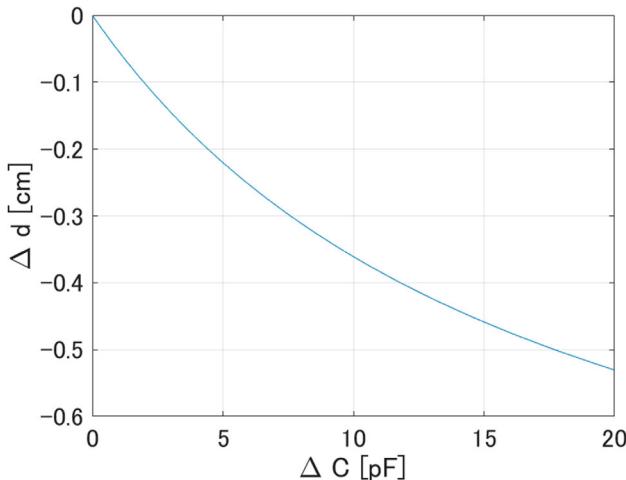


Fig. 6.34 Relationship between ΔC and Δd in capacitive sensor

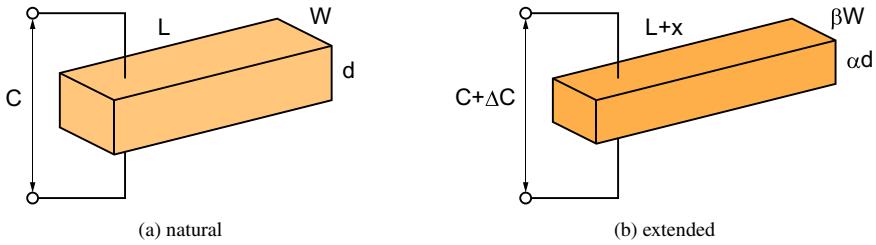


Fig. 6.35 Capacitive extension sensor

From this equation, we can estimate x by measuring capacitance change ΔC . Note that the above formulation requires no assumption that the volume of the dielectric remains constant.

Capacitive proximity sensors

Recall that capacitive sensors can measure proximity and detect contact. When an object is approaching a conductive fabric (Fig. 6.36), capacitance between the object and the fabric increases. Thus, measuring the capacitance, we can detect the object approaching. When the object contacts with the conductive fabric, the capacitance jumps up, which enables contact detection. By adaptively updating sensor references, we can detect proximity and contact robustly [49].

Circuits to measure capacitance

Currently, capacitance-to-digital converter ICs are commercialized. Applying such ICs, computers can read capacitance values directly. This section describes foundation of capacitance measurement.

Figure 6.37a shows a voltage divider circuit for capacitance sensing. This circuit consists of a reference resistance R and a variable capacitance C , which corresponds to a capacitive sensor. Reference resistance R is constant. Apply step input of constant voltage V_0 and measure output voltage V_{out} . Letting i be current, we have the following equations:

$$V_{\text{in}} - V_{\text{out}} - Ri = 0$$

$$V_{\text{out}} = \frac{1}{C} \int_0^t i(\tau) d\tau$$

which yields

$$V_{\text{out}}(t) = V_0 \left\{ 1 - \exp \left(-\frac{1}{RC} t \right) \right\} \quad (6.36)$$

Response of output voltage $V_{\text{out}}(t)$ depends on capacitance C (Fig. 6.37b).

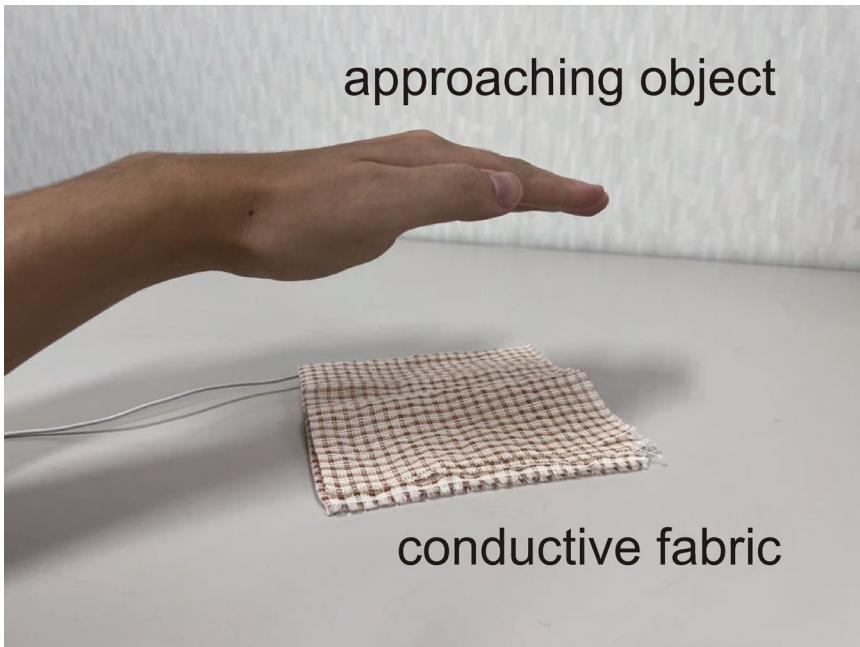


Fig. 6.36 Principle in proximity sensing

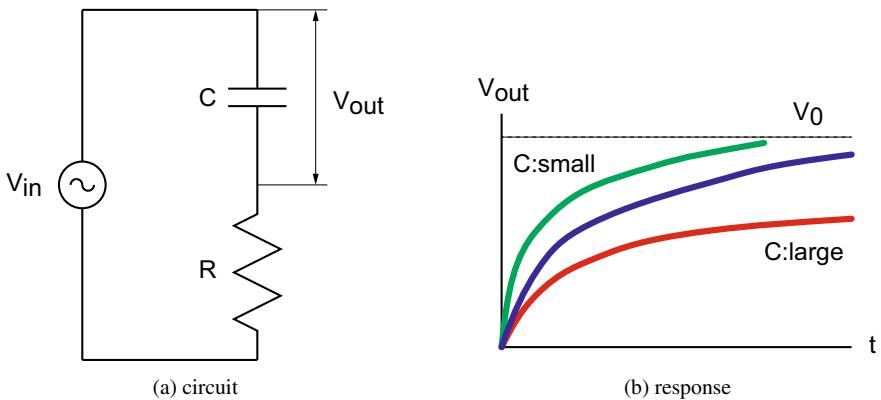


Fig. 6.37 Voltage divider circuit for capacitance sensing

Figure 6.38 shows a circuit to convert capacitance to voltage. This circuit consists of two divider circuits: one for sensor capacitance C and the other for constant reference capacitance C_{ref} . Let $u(t)$ be step function defined as follows:

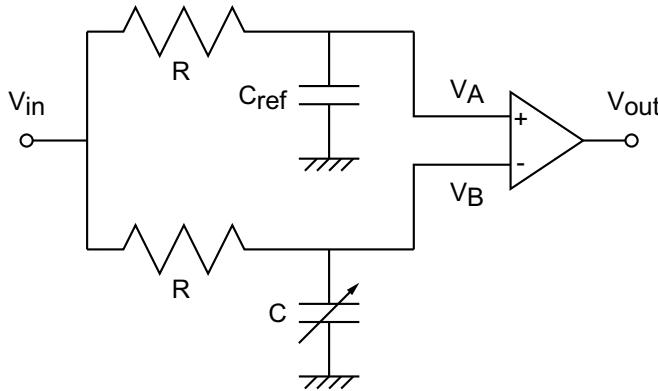


Fig. 6.38 Circuit converting capacitance to voltage

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (6.37)$$

Unit pulse function of time width T is then described as follows:

$$u(t) - u(t - T) = \begin{cases} 0 & t < 0 \\ 1 & 1 \leq t < T \\ 0 & T \leq t \end{cases}$$

Apply pulse input of constant voltage V_0 , which is described as $V_0(u(t) - u(t - T))$, to the two divider circuits. Their output voltages are described as:

$$\begin{aligned} V_A(t) &= V_0 \left\{ \text{expd} \left(t; -\frac{1}{RC_{\text{ref}}}, T \right) - \text{expd} \left(t; -\frac{1}{RC_{\text{ref}}}, 0 \right) \right\} \\ V_B(t) &= V_0 \left\{ \text{expd} \left(t; -\frac{1}{RC}, T \right) - \text{expd} \left(t; -\frac{1}{RC}, 0 \right) \right\} \end{aligned}$$

where

$$\text{expd}(t; a, T) = \begin{cases} e^{a(t-T)} & t \geq T \\ 0 & t < T \end{cases}$$

represents delayed exponential function. Consequently, output voltage is given by

$$V_{\text{out}}(t) = V_A(t) - V_B(t)$$

Figure 6.39 shows response of this circuit with $V_0 = 5$ V, $T = 2$ ms, $R = 1600$ k Ω , and $C_{\text{ref}} = 1000$ pF. Sensor capacitance C is 1200 pF, 1400 pF, or 1600 pF. Note

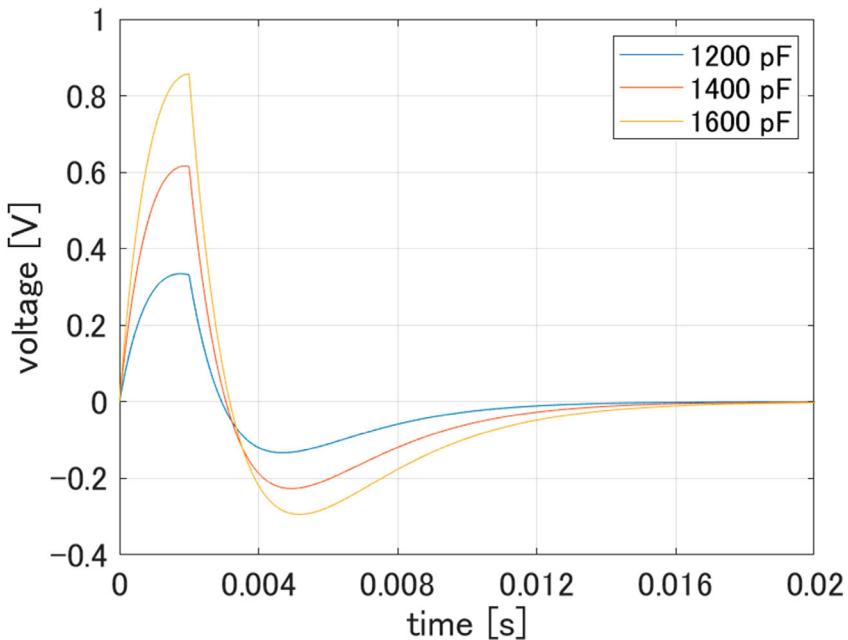


Fig. 6.39 Response of a circuit converting capacitance to voltage

that the maximum voltage depends on the sensor capacitance. Consequently, we can estimate the sensor capacitance through the maximum voltage during a certain period.

6.6.4 Magnetic Sensors

Magnetic sensors transduce physical quantities such as extensional and bending deformations to inductance or magnetic flux density. Magnetic sensors apply marker-detector pairs.

Figure 6.40 shows the sensing based on magnetic flux density. A permanent magnet is embedded with a soft body and a Hall effect device, which transduces magnetic flux density to voltage, is located near the body (Fig. 6.40a). When a finger pushed the body (Fig. 6.40b), the permanent magnet moves, changing the magnetic flux density passing through the sensor. Consequently, from the output voltage of the Hall effect device, we can estimate the pushed distance or the pushing force. In this sensing, a permanent magnet acts as a marker. Applying multiple Hall effect devices, we can estimate pushing location or force components [50–52]. Figure 6.41 shows the sensing based on inductance. The soft body is composed of magnetic elastomer, in which ferromagnetic particles are distributed, and a coil is located near the body

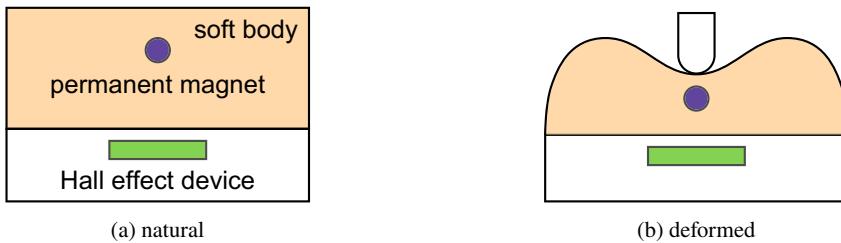


Fig. 6.40 Magnetic flux density based sensing

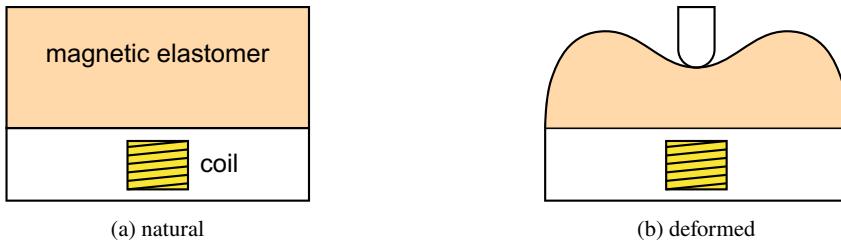


Fig. 6.41 Inductance based sensing

(Fig. 6.41a). When a finger pushed the body (Fig. 6.41b), permeability around the coil changes, resulting that inductance of the coil changes. Consequently, by measuring the coil inductance, we can estimate the pushed distance or the pushing force. In this sensing, magnetic elastomer acts as a marker. Applying multiple coils, we can estimate pushing location or force components [53, 54].

Circuits to measure inductance

Currently, inductance-to-digital converter ICs are commercialized. Applying such ICs, computers can read inductance values directly. This section describes foundation of inductance measurement.

Figure 6.42a shows a voltage divider circuit for inductance sensing. This circuit consists of a reference resistance R and a variable inductance L , which corresponds to an inductive sensor. Reference resistance R is constant. Apply step input of constant voltage V_0 and measure output voltage V_{out} . Letting i be current, we have the following equations:

$$V_{\text{in}} - V_{\text{out}} - Ri = 0$$

$$V_{\text{out}} = L \ddot{i}$$

which yields

$$V_{\text{out}}(t) = V_0 \exp\left(-\frac{R}{L}t\right) \quad (6.38)$$

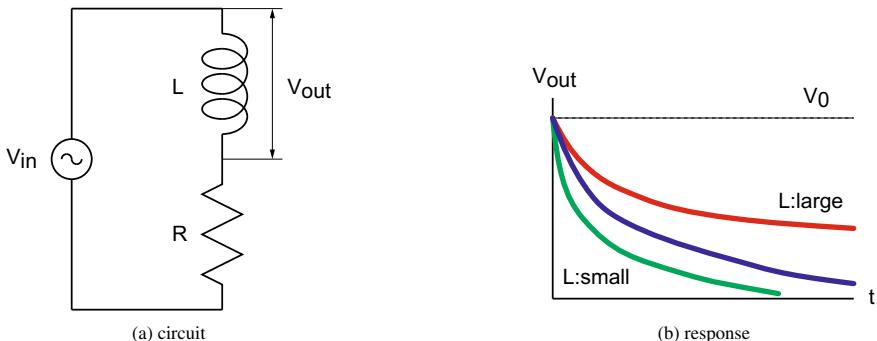


Fig. 6.42 Voltage divider circuit for inductance sensing

Response of output voltage $V_{\text{out}}(t)$ depends on inductance L (Fig. 6.42b). We can obtain inductance L based on this time-response.

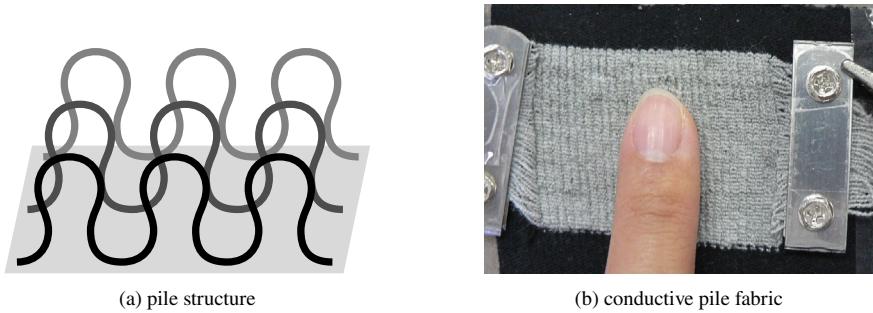
6.6.5 Entropy Reduction in Soft Tactile Sensing

Well designed soft tactile sensors reduce computational cost in signal processing, resulting entropy reduction. This section introduces two examples in sensor design.

Slip detection by pile structure

Pile structure (Fig. 6.43a) of conductive fabrics enables slip detection. Loops of threads of a knitted fabric lie along the fabric surface. On the other side, loops of a pile fabric are outwards of the fabric surface. Pile fabrics are often used for bath towels. Figure 6.43b shows a fabric sensor composed of a pile structured conductive fabric and two electrodes at its both ends. When a finger pushes the fabric sensor, threads contacting with the finger shrink and resistance between the two electrodes decreases. From the change of the resistance, we can detect the contact between the finger and the sensor. When a finger slides over the fabric sensor, the resistance changes as well. However, along with the finger sliding, contacting threads lose their contact and the finger contacts with other threads ahead. This causes vibration of the loops, resulting high-frequency change of the resistance. Consequently, evaluating high-frequency change of the resistance, we can detect the slippage of the finger along the fabric sensor surface.

It is shown that wavelet transform of measured resistance of a pile structured conductive fabric enables slip detection [55]. Detail coefficients of the wavelet become large when slippage occurs while remain small during pushing, making it possible

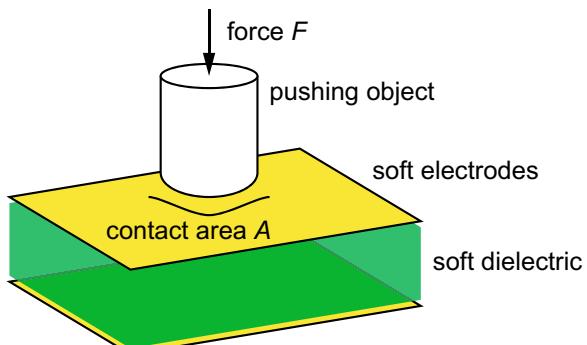
**Fig. 6.43** Fabric sensor

to distinguish slippage and pushing. Note that pile structure is essential in this slip detection. Introduction of a pile conductive fabric simplifies the signal processing.

Material for area-independent force sensing

Assume that a finger pushes one electrode of a capacitive sensor. Applied stress determines the deformation of the dielectric of the sensor, implying that the sensor output depends on not only the applied force but also the contact area between the finger and the electrode. Thus, it is necessary to measure the contact area to compute the applied force from the sensor output, which requires other sensors for contact area measurement and additional computation for image and signal processing.

Output of a capacitive sensor depends on stress-strain relationship of the dielectric material. Namely, the stress-strain relationship determines how sensor output depends on applied force and contact area. Here we design nonlinear stress-strain relationship of the dielectric material so that the dependency of the sensor output against contact area be minimized. When the dependency is small, we can construct a capacitive sensor that can measure the applied force independent of the contact area. Assume that an object pushes a capacitive sensor by force F with contact area S (Fig. 6.44). Let $\sigma = \sigma(\varepsilon)$ be the stress-strain relationship of the dielectric material of

Fig. 6.44 Soft capacitive force sensor [57]

the sensor. Capacitance of the sensor depends on force, contact area, and stress-strain relationship, the capacitance is described as follows:

$$C(F, S; \sigma(\varepsilon))$$

At a specific force F , variation of the capacitance with respect to contact area S is formulated as:

$$C_{\text{var}}(F; \sigma(\varepsilon)) = \int \{C(F, S; \sigma(\varepsilon)) - \mu(F; \sigma(\varepsilon))\}^2 dS$$

where

$$\mu(F; \sigma(\varepsilon)) = \frac{\int C(F, S; \sigma(\varepsilon)) dS}{\int dS}$$

Total variance of capacitance C is then described as:

$$V(\sigma(\varepsilon)) = \int C_{\text{var}}(F; \sigma(\varepsilon)) dF$$

When capacitance is independent of contact area S , that is, $C(F, S; \sigma(\varepsilon)) \equiv C(F; \sigma(\varepsilon))$, we find that $\mu(F; \sigma(\varepsilon)) = C(F; \sigma(\varepsilon))$, which directly yields $V(\sigma(\varepsilon)) = 0$. Consequently, total variance $V(\sigma(\varepsilon))$ reaches to its minimum 0 when capacitance is independent of contact area. Note that total variance depends on stress-strain relationship $\sigma = \sigma(\varepsilon)$. Thus, minimizing total variance $V(\sigma(\varepsilon))$, we obtain stress-strain relationship for area independent force sensing [56].

It is shown that the sensor output is independent of the contact area when the stress-strain relationship of the dielectric material satisfies the following equation:

$$\sigma = \beta \frac{\varepsilon}{1 - \varepsilon}$$

where β is a positive constant. It turns out that several sponges exhibit the stress-strain relationship similar to the above equation. Consequently, capacitive sensors using such sponges can measure the applied forces independent of the contact areas. This demonstrates that materials with stress-strain relationship given in the above equation reduces entropy in tactile sensing to measure applied forces directly in a simple manner.

6.7 Future Applications of Soft Hands for Grasping and Manipulation

Soft robot hands are effective in grasping and manipulation of soft deformable objects or objects with large variations in shape and geometry. Thus, soft robot hands are effective not only for grasping and manipulation of food, but also for grasping and manipulation of living organisms [58–61], suggesting that they can be applied in food industry, agriculture, and fishery. Moreover, soft robot hands accept mechanical interaction with environment, suggesting that they are applicable to physical human-machine interaction involving contact with humans [62, 63]. Integrating soft sensors with soft robotic hands will contribute to delicate mechanical interaction with human or environment.

Acknowledgements This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research on Innovative Areas “Science of Soft Robot” project under Grant Number 19H05337 and JP21H00333, and partly supported by the Cabinet Office (CAO), Cross-ministerial Strategic Innovation Promotion Program (SIP), “An intelligent knowledge processing infrastructure, integrating physical and virtual domains” (funding agency: NEDO).

References

1. Johnson KL (1985) Contact mechanics. Cambridge University Press
2. Popov Valentin L, Markus H, Emanuel W (2019) Handbook of contact mechanics: exact solutions of axisymmetric contact problems. Springer, Berlin, Heidelberg
3. Akella P, Cutkosky M (1989) Manipulating with soft fingers: modeling contacts and dynamics. In: Proceedings, 1989 international conference on robotics and automation, pp 764–769 vol 2
4. Shimoga KB, Goldenberg AA (1992) Soft materials for robotic fingers. In: Proceedings 1992 IEEE international conference on robotics and automation, pp 1300–1305 vol 2
5. Takahiro I, Shinichi H (2006) Elastic model of deformable fingertip for soft-fingered manipulation. *IEEE Trans Robot* 22(6):1273–1279
6. Tahara K, Maruta K, Yamamoto M (2010) External sensorless dynamic object manipulation by a dual soft-fingered robotic hand with torsional fingertip motion. In: 2010 IEEE international conference on robotics and automation, pp 4309–4314
7. Lu Q, He L, Nanayakkara T, Rojas N (2020) Precise in-hand manipulation of soft objects using soft fingertips with tactile sensing and active deformation. In: 2020 3rd IEEE international conference on soft robotics (RoboSoft), pp 52–57
8. Inoue T, Hirai S (2009) Mechanics and control of soft-fingered manipulation. Springer
9. Giacomo M (2007) Theoretical model of the grasp with vacuum gripper. *Mechan Mach Theory* 42(1):2–17
10. Yamaguchi K, Hirata Y, Kosuge K (2013) Development of robot hand with suction mechanism for robust and dexterous grasping. In: 2013 IEEE/RSJ international conference on intelligent robots and systems, pp 5500–5505
11. Hasegawa S, Wada K, Niitani Y, Okada K, Inaba M (2017) A three-fingered hand with a suction gripping system for picking various objects in cluttered narrow space. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1164–1171
12. Koivikko A, Drotlef D-M, Dayan CB, Sariola V, Sitti M (2021) 3d-printed pneumatically controlled soft suction cups for gripping fragile, small, and rough objects. *Adv Intelli Syst* 3(9):2100034

13. Eric B, Nicholas R, John A, Annan M, Erik S, Zakin Mitchell R, Hod L, Jaeger Heinrich M (2010) Universal robotic gripper based on the jamming of granular material. *Proc Nat Acad Sci* 107(44):18809–18814
14. Amend John R, Eric B, Nicholas R, Jaeger Heinrich M, Hod L (2012) A positive pressure universal gripper based on the jamming of granular material. *IEEE Trans Robot* 28(2):341–350
15. Kapadia J, Yim M (2012) Design and performance of nubbed fluidizing jamming grippers. In: 2012 IEEE international conference on robotics and automation, pp 5301–5306
16. Licht S, Collins E, Mendes ML, Baxter C (2017) Stronger at depth: jamming grippers as deep sea sampling tools. *Soft Robot* 4(4):305–316. PMID: 29251570
17. Pushpakath M, Ang Jr MH (2023) Design of a liquid jamming gripper. *Designs* 7(2)
18. Filip I, Mazzeo Aaron D, Shepherd Robert F, Xin C, Whitesides George M (2011) Soft robotics for chemists. *Angewandte Chemie Int Edn* 50(8):1890–1895
19. Jianshu Z, Shu C, Zheng W (2017) A soft-robotic gripper with enhanced object adaptation and grasping reliability. *IEEE Robot Autom Lett* 2(4):2287–2293
20. Wookeun P, Seongmin S, Joonbum B (2019) A hybrid gripper with soft material and rigid structures. *IEEE Robot Autom Lett* 4(1):65–72
21. Tawk C, Gao Y, Mutlu R, Alici G (2019) Fully 3d printed monolithic soft gripper with high conformal grasping capability. In: 2019 IEEE/ASME international conference on advanced intelligent mechatronics (AIM), pp 1139–1144
22. Sylvain A, Teeple Clark B, Wood Robert J (2020) A dexterous soft robotic hand for delicate in-hand manipulation. *IEEE Robot Autom Lett* 5(4):5502–5509
23. Zhang X, Yu S, Dai J, Oseyemi AE, Liu L, Du N, Lv F (2023) A modular soft gripper with combined pneu-net actuators. *Actuators* 12(4)
24. Zhongkui W, Yuuki T, Shinichi H (2017) A prestressed soft gripper: design, modeling, fabrication, and tests for food handling. *IEEE Robot Autom Lett* 2(4):1909–1916
25. Suzumori K, Koga A, Riyoko H (1994) Microfabrication of integrated fmas using stereo lithography. In: Proceedings IEEE micro electro mechanical systems, pp 136–141
26. Batsuren K, Yun D (2019) Soft robotic gripper with chambered fingers for performing in-hand manipulation. *Appl Sci* 9(15)
27. Lu Y, Kim C-J (2003) Micro-finger articulation by pneumatic parylene balloons. In: TRANSDUCERS '03. 12th international conference on solid-state sensors, actuators and microsystems, vol 1, pp 276–279
28. Konishi S, Nokata M, Jeong OKC, Kusuda S, Sakakibara T, Kuwayama M, Tsutsumi H (2006) Pneumatic micro hand and miniaturized parallel link robot for micro manipulation robot system. In: Proceedings 2006 IEEE international conference on robotics and automation, 2006. ICRA 2006, pp 1036–1041
29. Jun S, Samuel R, Bryan S, Dario F, Herbert S (2016) Versatile soft grippers with intrinsic electroadhesion based on multifunctional polymer actuators. *Adv Mater* 28(2):231–238
30. Jianglong G, Chaoqun X, Jonathan R (2018) A soft and shape-adaptive electroadhesive composite gripper with proprioceptive and exteroceptive capabilities. *Mater Design* 156:586–587
31. Pedro P, Ananda C, Rafael PB, Carlos AR, Alexandre BC (2018) Closed structure soft robotic gripper. In: 2018 IEEE international conference on soft robotics (RoboSoft), pp 66–70
32. Kanegae R, Wang Z, Hirai S (2020) Easily fabricatable shell gripper for packaging multiple cucumbers simultaneously. In: 2020 IEEE international conference on real-time computing and robotics (RCAR), pp 188–192
33. Zhongkui W, Ryo K, Shinichi H (2021) Circular shell gripper for handling food products. *Soft Robot* 8(5):542–554 PMID: 32822254
34. Ranasinghe HNB, Kawshan C, Himaruvan S, Kulasekera AL, Dassanayake P (2022) Soft pneumatic grippers for reducing fruit damage during strawberry harvesting. In: 2022 Moratuwa engineering research conference (MERCon), pp 1–6
35. Chua PY, Ilschner T, Caldwell DG (2003) Robotic manipulation of food products—a review. *Int J Indus Robot*

36. Khan ZH, Khalid A, Iqbal J (2018) Towards realizing robotic potential in future intelligent food manufacturing systems. *Innov Food Sci Emerg Technol* 48:11–24
37. Wang Z, Hirai S, Kawamura S (2022) Challenges and opportunities in robotic food handling: a review. *Frontiers Robot AI* 8
38. Iwamasa H, Hirai S (2015) Binding of food materials with a tension-sensitive elastic thread. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 4298–4303
39. Donald B, Gariepy L, Rus D (2000) Distributed manipulation of multiple objects using ropes. In: IEEE international conference on robotics and automation, vol 1, pp 450–457
40. Okada I, Wang Z, Hirai S (2019) Gripping force modeling of a binding hand. In: 2019 2nd IEEE international conference on soft robotics (RoboSoft), pp 671–676
41. Wang Z, Furuta H, Hirai S, Kawamura S (2021) A scooping-binding robotic gripper for handling various food products. *Frontiers Robot AI* 8
42. Aoyama H, Wang Z, Hirai S (2022) Shell gripper inspired by human finger structure for automatically packaging agricultural product. In: 2022 IEEE-RAS 21st international conference on humanoid robots (Humanoids), pp 90–95
43. Kuriyama Y, Okino Y, Wang Z, Hirai S (2019) A wrapping gripper for packaging chopped and granular food materials. In: 2019 2nd IEEE international conference on soft robotics (RoboSoft), pp 114–119
44. Lenz J, Edelstein S (2006) Magnetic sensors and their applications. *IEEE Sensors J* 6(3):631–649
45. Man J, Chen G, Chen J (2022) Recent progress of biomimetic tactile sensing technology based on magnetic sensors. *Biosensors* 12(11)
46. Yuan W, Dong S, Adelson EH (2017) Gelsight: high-resolution robot tactile sensors for estimating geometry and force. *Sensors* 17(12)
47. Shimonomura K (2019) Tactile image sensors employing camera: a review. *Sensors* 19(18)
48. Abad Alexander C, Anuradha R (2020) Visuotactile sensors with emphasis on gelsight sensor: a review. *IEEE Sensors J* 20(14):7628–7638
49. Takahiro M, Zhongkui W, Kaspar A, Shinichi H (2019) Adaptive update of reference capacitances in conductive fabric based robotic skin. *IEEE Robot Autom Lett* 4(2):2212–2219
50. Lorenzo J, Lorenzo N, Giorgio M, Giulio S (2015) Highly sensitive soft tactile sensors for an anthropomorphic robotic hand. *IEEE Sensors J* 15(8):4226–4233
51. Mohammadi A, Xu Y, Tan Y, Choong P, Oetomo D (2019) Magnetic-based soft tactile sensors with deformable continuous force transfer medium for resolving contact locations in robotic grasping and manipulation. *Sensors* 19(22)
52. Rosle MH, Kojima R, Or K, Wang Z, Hirai S (2020) Soft tactile fingertip to estimate orientation and the contact state of thin rectangular objects. *IEEE Robot Autom Lett* 5(1):159–166
53. Takumi K, Takato H, Hisashi I, Minoru A (2018) Flexible tri-axis tactile sensor using spiral inductor and magnetorheological elastomer. *IEEE Sensors J* 18(14):5834–5841
54. Shipeng X, Yuanfei Z, Minghe J, Chongyang L, Qingyuan M (2021) High sensitivity and wide range soft magnetic tactile sensor based on electromagnetic induction. *IEEE Sensors J* 21(3):2757–2766
55. Van Ho A, Makikawa M, Hirai S (2013) Flexible fabric sensor toward a humanoid robot's skin: fabrication, characterization, and perceptions. *IEEE Sensors J* 13(10):4065–4080
56. Takahiro M, Shinichi H (2021) Reducing the influence of the contact area on a soft capacitive force sensor. *IEEE Robot Autom Lett* 6(3):5824–5831
57. Takahiro M, Shinichi H (2022) Analysis of soft contact in force sensing and elastic jumping. *J Robot Mechatron* 34(2):285–287
58. Galloway Kevin C, Becker Kaitlyn P, Brennan P, Jordan K, Stephen L, Dan T, Wood Robert J, Gruber David F (2016) Soft robotic grippers for biological sampling on deep reefs. *Soft Robot* 3(1):23–33. <https://doi.org/10.1089/soro.2015.0019>. PMID: 27625917
59. Domenico M, Manuel B, Giacomo D, Giorgio G, Andrea C, Catalano Manuel G (2018) A soft modular end effector for underwater manipulation: a gentle, adaptable grasp for the ocean depths. *IEEE Robot Autom Mag* 25(4):45–56. <https://doi.org/10.1109/MRA.2018.2871350>

60. Haili L, Xingzhi L, Bo W, Xiaoyang S, Jiantao Y (2023) A fault-tolerant soft swallowing robot capable of grasping delicate structures underwater. *IEEE Robot Autom Lett* 8(6):3302–3309. <https://doi.org/10.1109/LRA.2023.3266668>
61. Morikage O, Wang Z, Hirai S, Nonaka A (2023) A soft gripper for automating split operation of silkworm. In: 2023 IEEE international conference on soft robotics (RoboSoft), pp 1–5. <https://doi.org/10.1109/RoboSoft55895.2023.10122050>
62. Block AE, Kuchenbecker KJ (2019) Softness, warmth, and responsiveness improve robot hugs. *Int J Soc Robot* 11(1):49–64. <https://doi.org/10.1007/s12369-018-0495-2>
63. Jørgensen J, Bojesen KB, Jochum E (2022) Is a soft robot more “natural”? exploring the perception of soft robotics in human-robot interaction. *Int J Soc Robot* 14(1):95–113. <https://doi.org/10.1007/s12369-021-00761-1>

Chapter 7

Malleable Robots



Angus B. Clark, Xinran Wang, Alex Ranne, and Nicolas Rojas

Abstract This chapter is about the fundamentals of fabrication, control, and human-robot interaction of a new type of collaborative robotic manipulators, called malleable robots, which are based on adjustable architectures of varying stiffness for achieving high dexterity with lower mobility arms. Collaborative robots, or cobots, commonly integrate six or more degrees of freedom (DOF) in a serial arm in order to allow positioning in constrained spaces and adaptability across tasks. Increasing the dexterity of robotic arms has been indeed traditionally accomplished by increasing the number of degrees of freedom of the system; however, once a robotic task has been established (e.g., a pick-and-place operation), the motion of the end-effector can be normally achieved using less than 6-DOF (i.e., lower mobility). The aim of malleable robots is to close the technological gap that separates current cobots from achieving flexible, accessible manufacturing automation with a reduced number of actuators.

7.1 Introduction

Reconfigurable robot systems provide several key potential advantages over traditional robots, including increased task versatility by adapting to better suit tasks, and reduced robot cost due to a smaller total number of modules, such as links and joints. As such, there has been significant research into the development of reconfigurable robots, with the most popular approach utilising modularity as the

A. B. Clark · X. Wang · A. Ranne
Imperial College London, London, UK
e-mail: a.clark17@imperial.ac.uk

X. Wang
e-mail: xinran.wang20@imperial.ac.uk

A. Ranne
e-mail: alex.ranne17@imperial.ac.uk

N. Rojas (✉)
The AI Institute, Cambridge, MA, USA
e-mail: nrojas@theaiinstitute.com

method of reconfiguration, as this allows for the interchangeability of parts, leading to self-repair [1, 2]. The reconfigurability feature has specifically been of interest in unstructured and unpredictable environments, characterised by changing operating contexts, which take the most advantage from robots that can adapt their shape and operating mode [3].

An alternative approach for the application of reconfigurable robot manipulators can be found in the industrial field of serial manipulators. In an ideal case, a manipulator would be designed with the exact number and configuration of joints necessary for its expected set of tasks [4]. This is known as task-based optimisation, and requires information to be known about the robot structure [5], collections of working points [6], or end effector regions [4]. However, knowledge of all tasks a robot might encounter in its lifetime can be difficult to determine. Instead, serial manipulators with a higher degree of freedom (DOF) are typically selected, ensuring dexterity across tasks at the expense of an increased robot cost and footprint.

Malleable robots, which are **reduced DOF serial robot arms with changeable geometry**, provide a solution to the task versatility problem through variable relative positioning of the revolute joints [7]. An example of the reconfiguration capability of a 2-DOF malleable robot can be seen in Fig. 7.1. Malleable robots follow a similar process of reconfiguration to modular reconfigurable robots, in that their reconfiguration alters the relative positioning of the active joints of the robot. However, unlike modular reconfigurable robots which achieve this by disassembling and reassembling (in an alternative configuration) the modules of the robot, malleable robots achieve this by transitioning between reconfigurable and rigid modes. Once in a reconfigurable mode, the robot can either reconfigure itself using additional drive systems (intrinsic malleable robots), or it can be manually reconfigured by an external system, such as a user (extrinsic malleable robots) as discussed in [7], where the design of such robots was firstly explored.

This chapter will explore malleable robots, providing further details and discussion on their design and construction, their kinematics, the use of augmented reality to aid their reconfiguration, and their intrinsic control. For each section of this chapter a relevant tutorial is given, providing the reader with the necessary skills and information to become familiar with the full development process of malleable robots.

7.2 Design of Malleable Robots

The key enabling technology behind malleable robots is their malleable link. This is defined broadly as a segment or link that connects two active revolute robot joints at an offset, ensuring their coordinate axes are different. The most important aspect of malleable robots is that this offset between their coordinate axes is **variable** in all 6 degrees of freedom (3 translation, 3 rotation). It can be changed to modify the joints relative positions, which in turn modifies the structure of the robot, resulting in a variable but controllable workspace, which can form specialised topologies

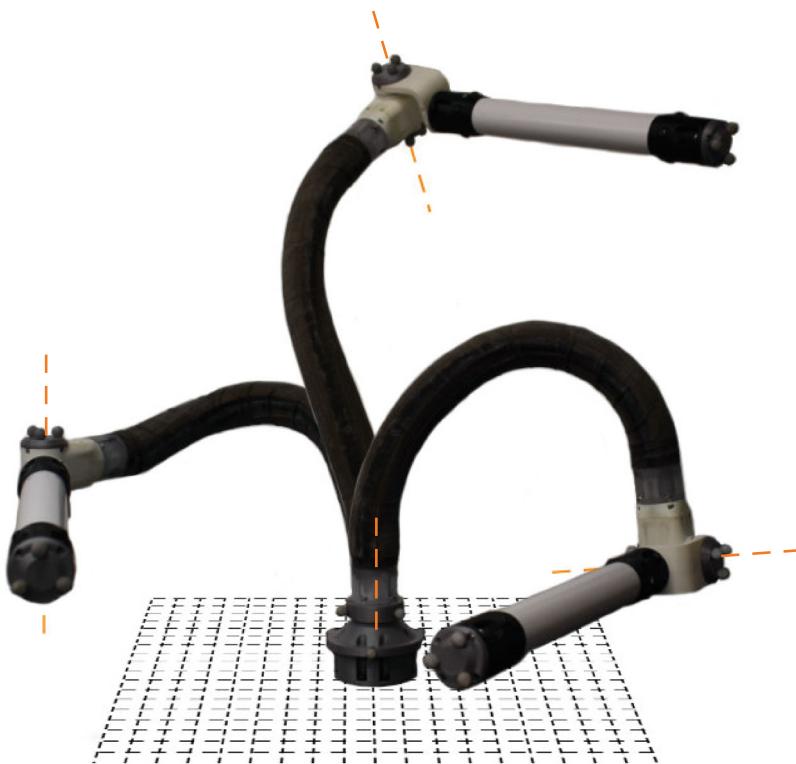


Fig. 7.1 A two-degree-of-freedom (DOF) malleable robot arm, showing various topology configurations it can achieve. The positioning of the revolute axes for each configuration are highlighted

(e.g., SCARA, Spherical, or PUMA-like). In turn, this enables malleable robots to be of lower mobility than typical serial robot manipulators, where the majority are composed of rigid links and joints [8], as they are highly flexible and can adapt to different tasks.

It is also key that this reconfiguration only occurs when desired, and at all other times the structure of the robot is fixed. As described, there is no one specific method for creating a malleable link. Various technologies could be used to provide the variable structure of the link, examples including simple lockable multi-segment links [9] and more complex variable stiffness technologies such as jamming, the latter of which we will examine more herein. Considering how they are reconfigured, malleable links can be classified into two categories: *intrinsic*, where the reconfiguration of the link is performed by the link itself, such as through an internal actuation system, and *extrinsic*, where the reconfiguration is performed by an external factor, such as a human manually reconfiguring the robot.

In this section, we will explore the alternative solutions in the design and construction of malleable robots, and provide a guide for an extrinsic layer jamming based solution.

7.2.1 Variable Stiffness Technologies

The promising performance of soft continuum robots has led to the development of various continuum implementations [10–12], however despite providing increased dexterity they suffer from a number of issues, most notably a lack in holding strength [13, 14]. To solve this, soft continuum robots have integrated variable stiffness technologies within their design, allowing for both soft, highly flexible motion and rigid, high load resistance [15–18]. These technologies are discussed and categorised in depth by Blanc et al. [19].

Variable stiffness technologies provide high flexibility and high stiffness in typically a small space, and have been used to increase the stiffness of small flexible devices. As such they have shown significant development in the medical device sector for increasing the stiffness of medical endoscopes and tools when performing delicate tasks requiring a high force output (and thus a high stiffness) during surgery.

Malleable links are typically larger (>50 mm diameter) than standard medical continuum robots (<15 mm diameter). This has consequences on the variable stiffness technologies that can be used for malleable links, as their increase in size means that certain technologies, like shape memory alloys (SMAs) and low melting point alloys (LMPAs), are unsuitable due to the significant increase in time to transition between rigid and flexible states.

A comparison on different variable stiffness technologies, along with their considerations regarding malleable links, can be seen in Table 7.1.

Of these technologies, longitudinal locking is the most promising for use in malleable links, due to the limited disadvantages of significantly scaling it up. For malleable links, due to their use as a robot manipulator and possible extrinsic reconfiguration, we suggest their diameter is as existing manipulator links at 40–50mm, which are also the optimal diameter for manual manipulation [20]. First proposed by Kim et al. [18], layer jamming is a promising technology that utilises longitudinal locking of thin layers of material wrapped conically along a cylinder to achieve a significant stiffness gain in a very thin and lightweight structure. Thanks to these features, it has shown recent popularity in continuum robotics [21–26].

One of the limitations of longitudinal locking is the stiffness dependence on the structure configuration, such as layer orientation. For continuum structures, this presents as a reduced stiffness as the cylindrical design bends. At extreme curvatures, buckling and unwanted deformation of the structure occurs, resulting in a dramatic loss of structural integrity. Thankfully, through the inclusion of a support structure within the continuum link, the stiffness decrease can be limited, and the buckling can be prevented [27]. It is important that, for such a design, it does not significantly

Table 7.1 Variable stiffness technologies

Method	Example technology	Pros	Cons
Glass transition	SMA	Extensive existing literature and use	Activation time increases significantly with scale Expensive
Phase transition	LMPA	Large stiffness gain	Activation time increases significantly with scale
Rheological fluids	Magnetorheological	Can achieve very fast (millisecond) activation times	Challenging to seal and maintain for long term use
Bulk locking	Granular jamming	Very easy to assemble Quick and simple to control stiffness	Weight of the granules can become a challenge with significant increase in scale
Longitudinal locking	Layer jamming	Lightweight, hollow centre	Complex to assemble
Segments locking	Central wire	Simple to actuate, as can be controlled by a single wire	Backlash in the segments is cumulative, resulting in significant backlash for long links

affect the advantages of the continuum structure (hollow and light), nor does it reduce the available DOF ability.

7.2.2 Manufacturing Flexible Spines and Malleable Links

In this section we detail a method for achieving a variable stiffness layer-jamming-based malleable link, and a method for achieving a multi-DOF neutral-line constant diameter support mechanism, namely a flexible spine, to reduce structural deformation.

7.2.2.1 Malleable Links

The following approach to link design for extrinsic malleable robots focuses on leveraging existing layer jamming technology used in continuum manipulators to create a finite length variable stiffness link, that is reconfigured using manual human deformation. Layer jamming using conically wrapped overlapping sheets of Mylar® (0.18 mm) (polyethylene terephthalate), first presented as a medical manipulator by Kim et al. [18], was used. The laser cut flap pattern, detailed in Fig. 7.2, contained 12

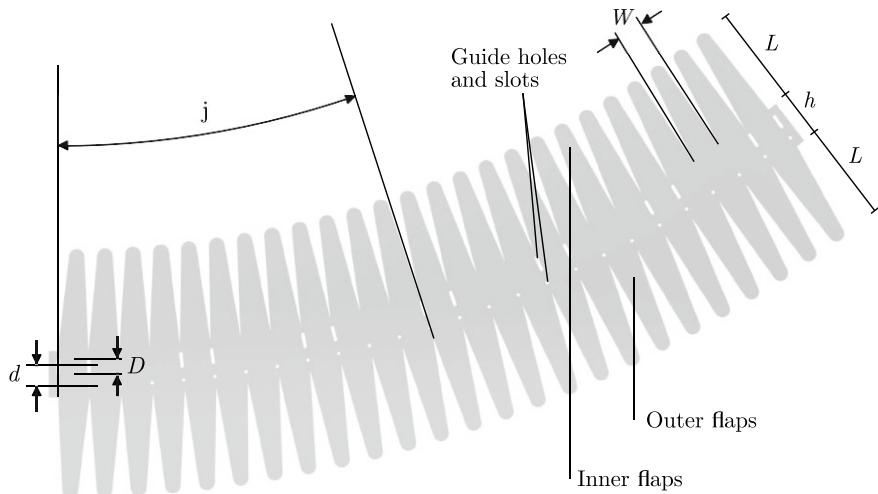


Fig. 7.2 Double-sided flap pattern specifications for layer jamming sheath with guide holes and slots

flaps spanning the circumference of the link, with a minimum of 10 overlapping layers always in contact. This determines the maximum stiffness, which can be calculated from the force F required to separate the layers as $F = \mu n P W L$, where μ is the coefficient of friction, n is the number of overlapping layers, and P is the applied pressure. By varying the pressure in the sealed volume containing the layers using a vacuum pump (BACOENG 220V/50Hz BA-1 Standard), it is possible to vary the stiffness of the link from flexible to rigid as the cumulative friction causes a significant increase in rigidity proportional to the negative pressure applied. Flap parameters used were flap length $L = 45$ mm, flap width $W = 13$ mm, mid length $h = 16$ mm, guide hole distance $d = 9.5$ mm, and inclination angle $\varphi = 12.75^\circ$. The flap pattern was then wrapped conically and contained within two cylinder membranes of latex sheet (0.25 mm), and sealed with link termination ends 3D printed from Vero Clear on a Stratasys Objet 500, which also provided mounting points for an internal structural spine (see next section) to prevent excessive deformation under extreme bending of the link, as well as mounting points to attach the other components of the robot. The components of the malleable link can be seen in Fig. 7.3.

A key consideration of layer jamming is the termination of a finite length of it. While layer jamming provides a high level of stiffness when under compression, the point at which the layer jamming fixes to the rigid joints of the robot (which we call ‘layer termination’) presents a significant reduction in that stiffness. Due to the conical design, the termination point (and therefore fixture location) is not obvious. The traditional solution is to cut the tubular structure to form a flat face, which can then be adhered to. However, this results in a loss of \sim half of the overlapped layers at the termination point, resulting in a significant loss of stiffness. To correct this,

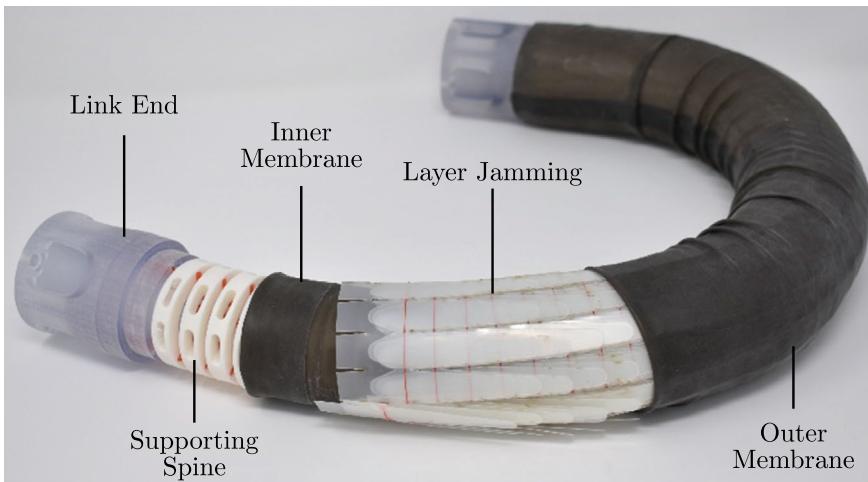


Fig. 7.3 Cut-away malleable link highlighting individual components

passive layers can be used which replace the removed layers, but do not impact the termination.

The layers were manufactured by laser cutting matte surfaced polyethyleneterephthalate film (Mylar®), with a thickness of 0.18 mm and a measured coefficient of friction of 0.4. Extra strong polyester line (Gütermann M 782) with a diameter of 0.20 mm was sewn using a needle along the guide holes, completing a circumference every 12 holes. As the layers wrapped around forming a hollow cylinder, the guide slots of the previous loop were sewn to the following loop every 4 holes, forming connections every 120°.

The slot length and guide hole distance d determine the overall maximum and minimum length of the link through extension and compression, and as such the bending ability of the link (without deforming), where one side of the structure is at maximum extension while the other is at maximum compression. The maximum bending angle θ can be calculated as $\theta = (N - 1) \sinh \frac{D}{\phi}$, where N is the number of layer loops, D is the slot length, and ϕ is the diameter of the layer jamming sheath. The maximum length l_{max} and minimum length l_{min} of the link can be computed as $l_{max}, l_{min} = (N - 1)(d \pm \frac{D}{2}) + h$.

To modulate the stiffening mechanism, an outer and inner tubular membrane were joined at each end around the layers, forming a sealed tubular volume which could be pressurised, compressing the layers together. The membranes were manufactured manually by cutting sheet latex (0.25 mm thickness) to the required length and perimeter, then sealing the latex into tubular membranes using liquid latex adhesive (CopyDex). At each end of the link, the layers and membranes terminated at a link end. These link ends were fabricated by 3D printing—fused deposition modeling (FDM) using ABS material, and incorporated a mounting position for both the membranes and the layers, which were attached using Ethyl 2-cyanoacrylate (Super

Glue). Further, one of the two link ends enabled a connection via a 6 mm PVC tubing to a vacuum pump, allowing access to the sealed volume containing the layers.

7.2.2.2 Flexible Spine

To prevent buckling of the structure from the unmanaged deformation of the layers when bending, a supporting flexible spine was fabricated, which connected to each of the link ends, and maintained the diameter along the link as it underwent bending. The spine properties were identified from the layer jamming tube, making sure that the spine did not impact the existing bending performance of the malleable link. The spine is defined by the parameters neutral gap (G_n), compressed gap (G_c), extended gap (G_e), ligament beam length (B), ligament neutral angle (Θ), central gap (g), segment height (w), and segment diameter (D_s), as shown in Fig. 7.4. To ensure the spine DOF did not affect the layers, an equal or lower rigid:flexible component ratio was required, to allow the spine equal or greater compression and extension than the layers. This was possible by using the maximum and minimum link lengths (l_{max} , l_{min}). This was computed assuming the flexible components were capable of compressing to a length of zero, and extending to the required length of $l_{max} - l_{min}$,

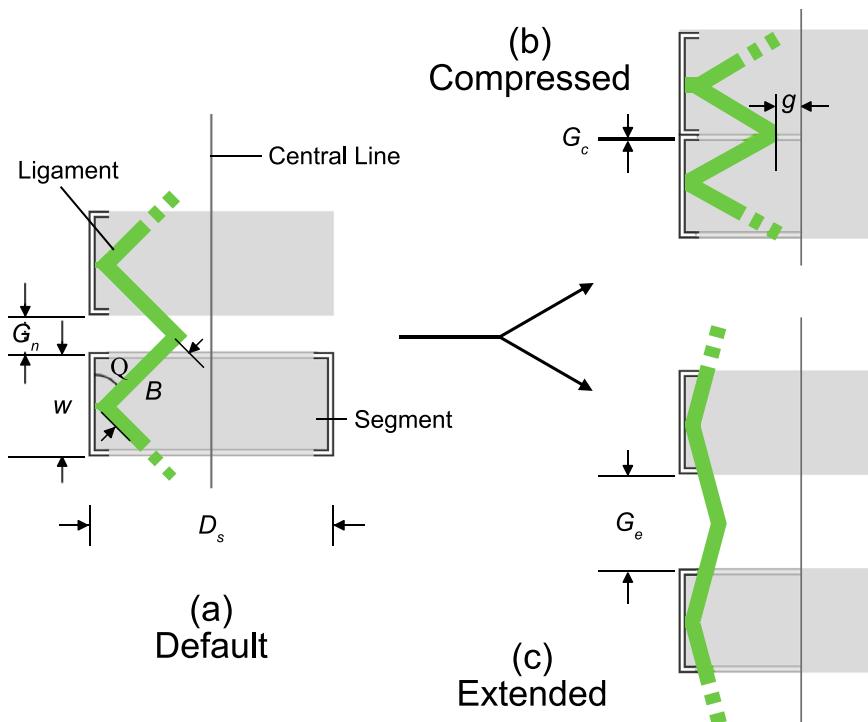


Fig. 7.4 Neutral spine position (a), compressed spine (b), and extended spine (c)

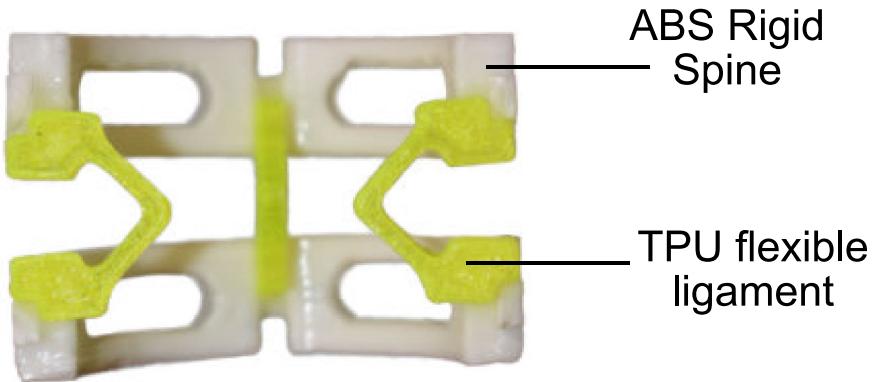


Fig. 7.5 Sectional view of 2-segment spine with integrated flexible ligaments (square planar ligament connections)

implying an increase of 100% from rest position. The spine was constructed from 14 rigid ABS segments, each 8 mm in height and 32.5 mm in diameter, totalling to 132 mm of rigid support when added to the 10 mm of support provided by each link end.

Connecting the segments were 3 magazine spring-inspired TPU ligaments, which were fixed every 120° to the segments and link ends using Ethyl 2-cyanoacrylate (Super Glue). The assembly of the components is demonstrated in Fig. 7.5. The flexible ligaments were composed of connectors and triangular springs. When fully compressed, the central gap g can be computed as

$$g = \frac{D_s}{2} - (B^2 - (G_c + \frac{w}{2})^2). \quad (7.1)$$

With the compression and expansion limits known, we can rearrange Eq. (7.1) to solve for ligament beam length. The ligament neutral angle was set as 45° to allow for equal support to be generated at the stress concentrations of the ligament. The bending capability and 4 DOF of the spine is demonstrated in Fig. 7.6.

7.2.3 Forming the Malleable Robot

A robot manipulator with a very unique link component, malleable robots maybe require other unique components to function effectively.

A 2-DOF malleable robot, formed from two revolute joints, a malleable link, and a rigid link, was developed following work in [7]. Joint 1, positioned at the base of the robot, provides rotation in the z-axis. Joint 2 was positioned at the end of the malleable link, providing rotation in the axis perpendicular to the termination end. Both joints

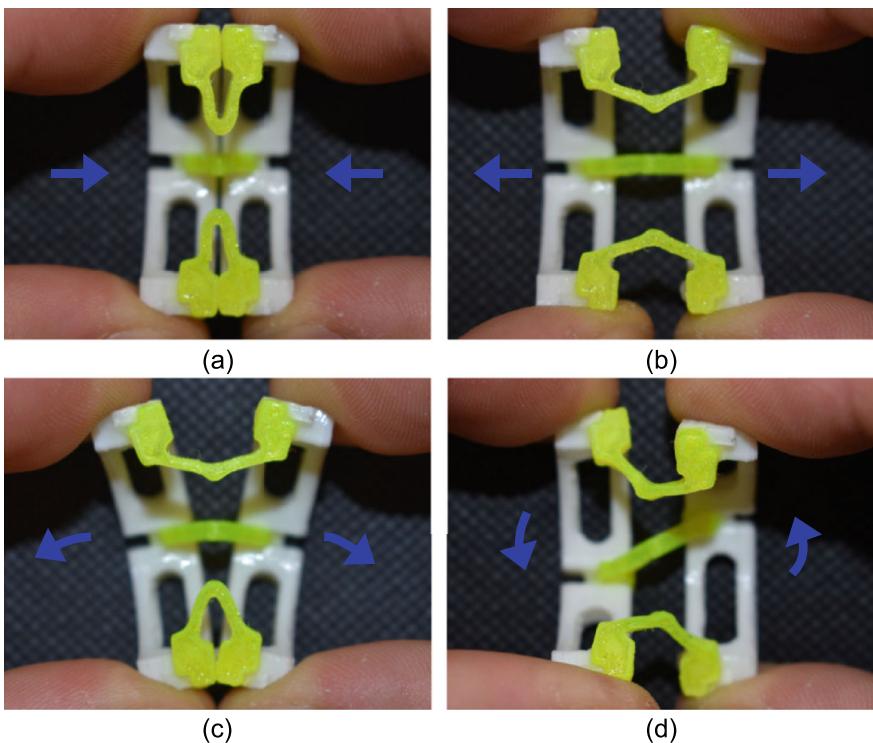


Fig. 7.6 Motion of the flexible spine. Compression (**a**), extension (**b**), bending (**c**), and twist (**d**)

were constructed from a Dynamixel MX-64 servo motor, with a 3D printed ABS housing, and a thrust ball bearing (size 51106) providing force distribution of the motor torque to the output side of the joint. The malleable link is a variable stiffness link that uses Mylar-based layer jamming to transition between rigid and flexible modes, with a maximum length of 700 mm and a minimum length of 550 mm. The rigid link attached to the secondary joint has a length of 370 mm (actual distance of 450 mm between joint axis and end effector). The link was composed of a 42 mm \varnothing polypropylene tube, and was attached to the robot using 3D printed ABS link ends similar to those used on the malleable link. The dimensions of the distal link were selected to be like that of the malleable link, with the shorter length chosen to prevent frequent collisions with the floor plane other.

7.2.4 Tutorial: Layer Jammed Malleable Link

In this subsection we provide a tutorial for creating a single malleable link. In addition, commentary is provided at relevant steps for when considering modifications or

replacements. The design utilises layer jamming as the variable stiffness technology, and a change in pneumatic pressure to activate (ambient to vacuum) and deactivate (vacuum to ambient) the stiffening.

7.2.4.1 Preparation

The necessary components and tools to follow this tutorial are detailed below, with the specific items used shown in brackets. Software versions used at the time of writing are noted. These specific items are only suggestions, and suitable alternatives or newer versions may/should be used instead. **Material:**

- 0.2 mm Mylar[®] (polyester film) sheets
- PETG filament - *Prusament PETG*
- TPU filament - *Sainsmart 95A TPU*
- 0.25 mm sheet latex - *T40 translucent smoky black 0.25mm*
- 4 mm ID x 6 mm OD PVC hose
- 0.6 mm 135lb Braided fishing line

Software:

- CAD package - *SOLIDWORKS 2022-23*
- 3D printer slicing software - *PrusaSlicer 2.5.0*
- Laser control software - *Universal Laser System Control Panel*

Tools:

- FDM 3D printer - *Prusa MK3S*
- Laser cutter - *Universal Laser System VLS 3.50*
- Cyanoacrylate glue - *Gorilla Superglue Gel*
- Epoxy glue - *Gorilla Epoxy*
- Copydex latex adhesive

7.2.4.2 Method

The instructions are broken down into sub-components, followed by a final assembly.

Layer Jamming. The layer jamming is the core component that enables the variable stiffness of the malleable link. It is achieved by overlapping multiple layers of material, here Mylar sheets, which are then sandwiched between two membranes. To provide a strong yet light structure, a tube is typically formed of which the length can be adjusted to suit the user's needs. To form this tube, the flat sheets of Mylar must be cut in a conical design so that they wrap around the circumference of this tube.

The basic structure of such a conical design is described in Fig. 7.7. It consists of a central strip (shown in red) that is curved slightly upwards, and several flaps

(shown in blue) that protrude outwards (vertically up and down) from the central strip. The angle of curvature of the central strip determines the diameter of the tube that is formed, whereas the length and width of the flaps determines the total frictional area, and therefore the overall maximum stiffness. One would think to therefore maximise this value, however an increase in the length and width of the flaps limits the bending ability due to restricted bending points, and it is partially limited by the taper of the flaps as above a certain length they are tapered to an insignificant width. Therefore, a healthy middle ground for the length of the flaps must be selected that balances the flexibility with the maximum stiffness. As this relates directly to the diameter of the tube that is formed and the personal preference in bending ability, it is not detailed here and must be selected by the user. That said, a value that produces 12 flaps per circumference has been shown to work well as a starting value.

The overlapping components of the design (the flaps) must be separated to ensure they do not overlap undesired parts of the design (they should only overlap other flaps directly above and below them, not to the side), and that they do not hinder the bending ability of the design. As the central strip is curved, this results in the flaps being tapered. Note, it is worth considering the manufacturing method here too, as the taper may need to be increased to ensure the flaps cleanly separate (e.g. in the case of laser cutting).

Ideally, a single, conical piece wrapped around would form the tube. This is typically not possible due to the significant size of the unwrapped conical shape, and therefore the design must be made modular. This is achieved using the guide slots and guide holes, which are used to align and join a following piece. In the design provided, two full circumferences is possible in a single piece (guide slots located every 120°), after which another piece is required to join to continue wrapping. You should consider the tools available to you before the design process as the maximum size of the laser cutter will limit the upper bound of the layer jamming size you can create.

To obtain an accurate design, it is suggested to use a 2D parametric design software to create the shape to cut out, here SOLIDWORKS is used (as a single sketch). This allows the use of equations and variables to define the shape, which once setup allow easy modification of the design. Once created, the design can then be exported as DXF, a 2D vectorised design. Note, make sure to only export the lines that you wish to cut, and ensure to remove items such as construction lines (in SOLIDWORKS you are provided with a menu to select the lines you would like to include when you export as DXF, see Fig. 7.8). An example SOLIDWORKS part file is made available as a base design to start with and modify. The resulting exported DXF file is also made available (if any changes are made to the SOLIDWORKS file, a new exported DXF will need to be generated following the above instructions.).

According to the labelling in Fig. 7.7, the provided CAD file and DXF have the initial dimensions shown in Table 7.2, which can be modified easily using either the Equation Manager in SOLIDWORKS or directly.

Once exported, it is now time for manufacture. The laser cutter is used for this task, cutting the Mylar sheet material. It is not possible to reliably cut multiple stacked sheets, therefore it is recommended to cut a single sheet at a time. Once

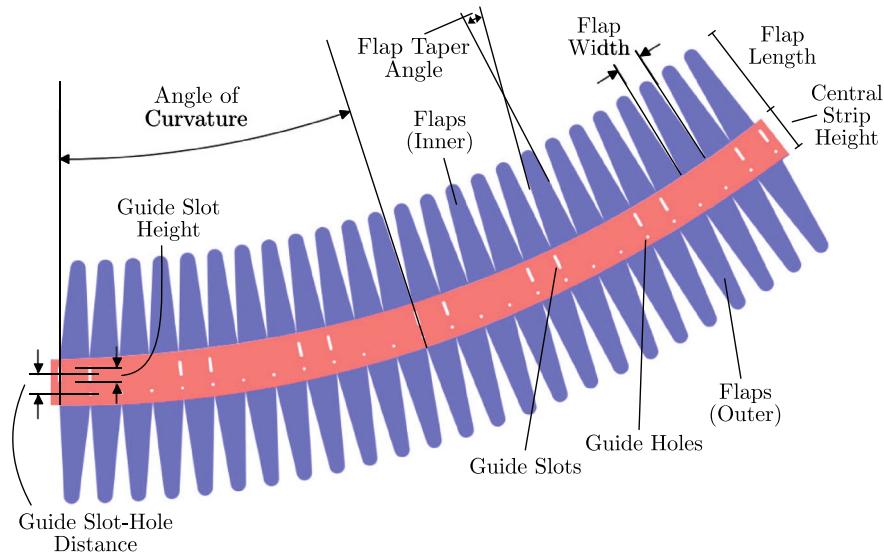


Fig. 7.7 Diagram highlighting the key elements that compose the layer jamming structure

Table 7.2 Specifications of layer jamming pattern parameters

Value	Dimension
Angle of curvature	11°
Central strip height	16 mm
Flap length	45 mm
Flap taper angle	8°
Flap width	12 mm
Guide slot height	5 mm
Guide slot-hole distance	9.5 mm

the DXF file is imported to the laser software, ensure that the entire design is set to ‘CUT’, and not ‘ENGRAVE’ or ‘RASTER’. The laser cutter should come preloaded with presets for a variety of materials. For the ULS VLS 3.50 used here, there is a polypropylene/Mylar option which can be used to apply the correct speed/power settings of the laser to guarantee a clean cut. As the layers are modular, it is necessary to calculate the number of pieces needed to cut to achieve the desired tube length.

Once cut, the final stage of the layer jamming is to assemble the layers from their 2D form into a 3D tube. This is achieved using either thread or a cable, which is sewn along the guide holes and guide slots around the circumference of the tube. Here, braided fishing line is used as it is extremely strong and quite easy to tie off. Wire has the advantage of increased strength, but limited flexibility and as such is more of a challenge during assembly. To form the 3D shape, the line is first fixed at one end of the bottom row of the guide holes. This is then sewn back and forth along

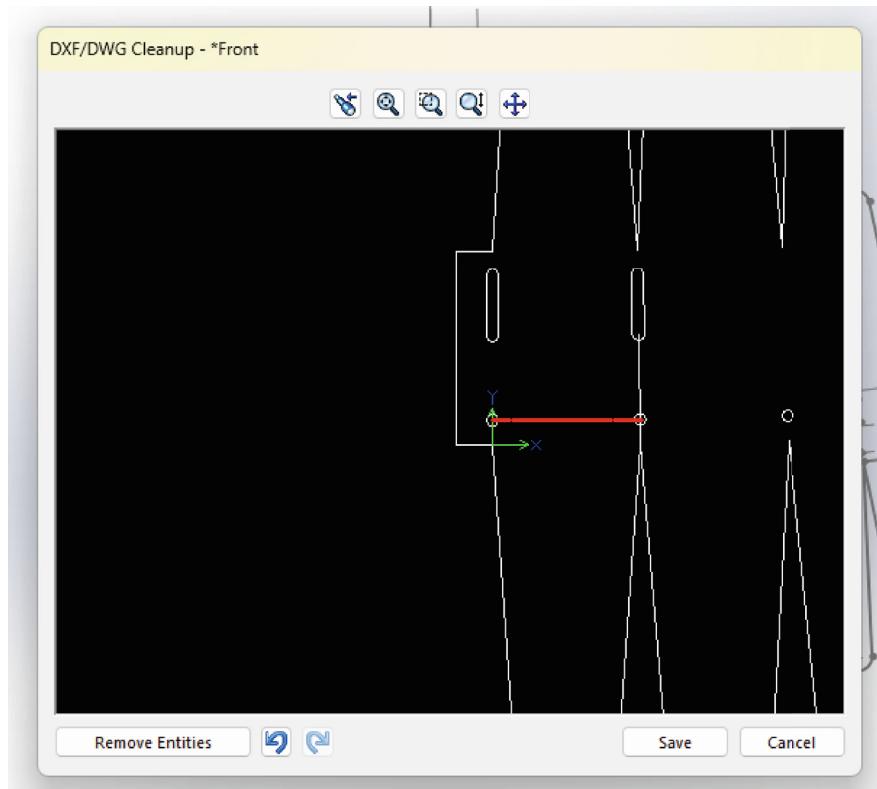


Fig. 7.8 Cleanup options when exporting a DXF from SOLIDWORKS

the guide holes for a single circumference. The layers are then curved to the desired diameter, aligning the guide slots with the guide holes, and the sewing continues, ensuring that the line enters and exits at the inside of the layer tube for the guide slots. A diagram showing the line route is shown in Fig. 7.9. Once complete, it should be tightened enough to secure the structure shape, while allowing the layers to move over each other. Finally, the additional flaps at the top and bottom of the cylinder can be removed (cut off) to provide a smooth flat surface at each end of the cylinder.

Membranes. To apply a vacuum to the layer tube, it must be sealed within a membrane. The easiest way to achieve this is by using two tubes, one larger and the other smaller than the layer tube. These can then be placed outside and inside the layer tube respectively, and when sealed at the ends form an airtight volume. The membranes must be flexible such that they do not hinder the bending curvature of the layer tube nor break under tension, and strong enough to resist the pressures of a high vacuum. From our testing, 0.25 mm thick latex sheet is ideal. This can easily be formed into tubes of custom dimensions by joining two edges of a flat quadrilateral using latex glue (such as Copydex latex adhesive). When connecting the edges, it is

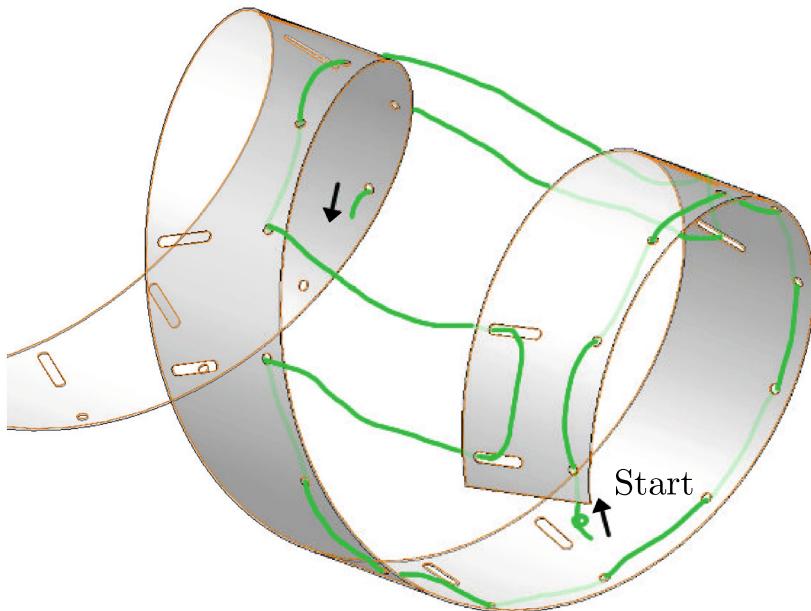


Fig. 7.9 Diagram showing the sewing route (green) for the layer jamming. External flaps are not shown to increase visibility

imperative to ensure an airtight seal. Overlapping a large portion of the edges helps to achieve this.

Supporting Spine. With the layer tube as it is now, under extreme bending angles the structure will buckle and collapse in on itself. There is also a chance of buckling while in rigid mode when under extreme loads. To prevent this, a supporting spine structure is required down the centre of the tube. This supporting structure should have a rigid component (that resists the buckling) and a flexible component (that allows the support structure to follow the curvature of the layer tube). The layer tube can bend in 4 degrees of freedom: Bending in two dimensions, twisting, and extending/compressing. The supporting structure must also be capable of these.

There are many designs that would satisfy the requirements, and one such design is presented here as an example. The design uses acrylonitrile butadiene styrene (ABS) as the rigid material, and thermoplastic polyurethane (TPU) as the flexible material. Both materials can be 3D printed, which eases manufacturing. Here, the rigid components are hollow discs, which provide a strong outer edge to support the layers, and a hollow centre to allow for items to pass through the length of the tube. This can be key for when integrating the malleable link into a larger design, such as a malleable robot that requires cables and tubes to run throughout it. The flexible TPU component is shaped with a zig-zag pattern such to prevent self-collisions when the tube bends, and to increase the ease of bending. The spacing of the rigid/flexible components is dependent on the layers bending radius. When printing using FDM

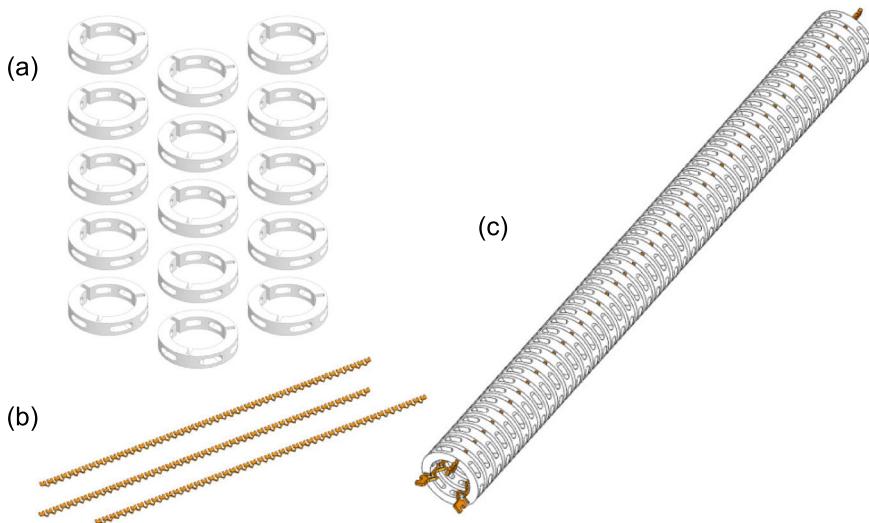


Fig. 7.10 The parts of the spine: **a** the rigid component, **b** the flexible components, and **c** the assembled spine

printers, components are often weakest in the vertical Z axis. To ensure maximum strength of the TPU component, it is printed on its side. Once printed, the ABS and TPU components can be connected using an adhesive (such as superglue). The components along with the assembled spine are shown in Fig. 7.10. Using more advanced multi-material 3D printers, it is possible to avoid the assembly step by printing both components pre-assembled in a single print. Considerations must be taken here to ensure that the rigid and flexible components bond securely, and that the design still maintains the flexibility. For example, is the design printed in an expanded state (requiring a lot of support material), or in a compressed state? Using a dissolvable material for the support material could also provide advantages here, at an increased cost. The spine design shown here is made available to the reader in both editable CAD files (SOLIDWORKS, both part and assembly) and printable 3MF files.

Link Terminations. With the components so far, we can make a tubular structure. However, this structure is currently open at both ends, not forming a sealed volume as needed. Also, the only exposed surface on the structure is a rubber membrane, which is difficult to form a strong, solid connection to. To correct these problems, we must introduce ‘malleable link terminations’. These are rigid components that are attached to each end of the tubular structure that provide a solid connection point for integrating the malleable link into a larger assembly and form an airtight seal on the structure. They also provide a singular location for all of the other components to connect together, as so far they are next to each other but not connected (such as the membrane and layers, or layers and spine). We must also have an access point

to the internals of the sealed volume to evacuate it, which can be achieved using a pre-cut hole in the termination, into which a 6 mm PVC tube can be inserted. Here, the terminations can simply be 3D printed out of the same rigid material used for the spine. For the design, it is key to have the outer membrane connect further down the termination than the layers/inner membrane, otherwise it is very difficult to form a complete seal. It is also key to consider the location of the tube/s to the sealed volume, as the radial distance between the inner and outer membrane and the membrane positioning will limit their size and placement. As with the other components, the design is made available to the reader in both editable CAD files (SOLIDWORKS) and printable 3MF files.

Final Assembly. The final step is to assemble everything together. Starting from the link termination with the vacuum tube insert, the PVC tube must be inserted and fixed in position (superglue). We can now assemble from the inside-out, starting with the spine. The spine is fixed to the termination using the pre-defined slots for the flexible components. These can also be fixed using the superglue. Ensure that only one of the two terminations is attached! The other must be left until later. Next, we slide the inner membrane over the spine, and fix it to the termination. Instead, this time we use the slower drying epoxy, which can be spread over the connection area on the termination before placing the membrane on it. Due to the translucency of the membrane, you can check the airtight seal is formed by ensuring that the dark band formed by the epoxy is seen in a continuous line around the entire circumference. Care must be always taken with both membranes to make sure no damage is caused to them, affecting the integrity of the seal. Once dry, we can now slide the layers over the inner membrane and fix them in place using more epoxy. Care must be taken here so that the epoxy is only applied to the desired area, and does not accidentally get into the rest of the layers, which once set would limit the flexibility of the link. Note that alternative methods to connect the layers could be taken here, such as using more thread or a clamping system. Once set, the outer membrane can be fixed in place using the same method as the inner membrane. Now all the components are fixed to one of the termination ends, we can attach the other. This must be fixed from the centre of the tube working outwards, using the necessary adhesives for each components.

You can check the assembly was successful by applying a vacuum down the expose 6 mm tube, which should cause the link volume to evacuate and the link to transition to its rigid state. By removing the vacuum and returning the volume to ambient pressure, the link should transition to the flexible state. Here, the link should be manipulated in all four bending directions to confirm everything is fixed adequately, and nothing is restricting the motion of the link. If any problems arise, it is most likely an incomplete seal, which can often be fixed using more adhesive. If all the above tests are successful, congratulations! You have now assembled your own malleable link.

7.3 Workspace Computation and Kinematics Computation

7.3.1 Overview of Distance Geometry

Distance geometry, first defined by Leonard Blumenthal in first half of the 20th century [28], is a relatively new branch of mathematics that focuses on the study of geometries through the use of their metrics – i.e. their distances. It avoids the need to define arbitrary reference frames, and allows for translations and rotations to be separately computed [29], compared to typical kinematic modelling using Denavit-Hartenburg (D-H) parameters. As such, distance geometry has successfully been used to intrinsically characterise Euclidean spaces [30]. In this chapter, distance geometry is used as an alternative to other methods for computing the kinematics and topology reconfiguration, such as typical D-H parameters or Screw theory, as it avoids further hardware and software complexities. For example, it may be challenging to directly obtain the relative joint positions and angles using internal sensors in the malleable robot, which is an ongoing area of research in continuum robots [31]. In contrast, distance geometry only requires the localisation of points distributed along the robot, which can simply be achieved with external motion tracking. This is key for the computation of workspaces of malleable robots as traditional strategies based on the attachment of reference frames to the robot joints, such as those that make use of the D-H convention [32, 33], cannot be employed since both link dimensions and the relative orientation of the joints can change. An alternative is to perform the workspace analysis using screw theory [34, 35] or distance geometry [29] as in these approaches the parameterisation does not depend on relative angles and distances between joint reference frames. We make use of a distance-geometry-based method herein as the technique has been shown to simplify the computation of the workspace equation of complex mechanisms [36, 37].

7.3.1.1 Notation

In what follows, we denote a point in \mathbb{E}^3 as P_i , $\mathbf{p}_{i,j} = \overrightarrow{P_i P_j}$ denotes the vector from P_i to P_j , $\mathbf{p}_{i,j,k} = \mathbf{p}_{i,j} \times \mathbf{p}_{i,k}$ denotes the cross product between vectors $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,k}$, and $s_{i,j} = \|\mathbf{p}_{i,j}\|^2 = d_{i,j}^2$ denotes the squared distance between P_i and P_j , with vector coordinates arranged as column vectors. The vectors $\mathbf{p}_{i,j}$, $\mathbf{p}_{i,k}$, and $\mathbf{p}_{i,j,k}$ in general represent a non-orthogonal reference frame that is denoted by the column vector of nine components $\mathbf{q}_{i,j,k} = (\mathbf{p}_{i,j}^T, \mathbf{p}_{i,k}^T, \mathbf{p}_{i,j,k}^T)^T$.

The tetrahedron defined by points P_i , P_j , P_k , and P_l is denoted as $\triangle_{i,j,k,l}$, with its *origin* located at P_i , its *base* given by the triangle $\triangle_{i,j,k}$ with area $A_{i,j,k}$, *base vectors* $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,k}$, and *output vectors* $\mathbf{p}_{i,l}$, $\mathbf{p}_{j,l}$, and $\mathbf{p}_{k,l}$. This notation is shown in Fig. 7.11 [38, 39].

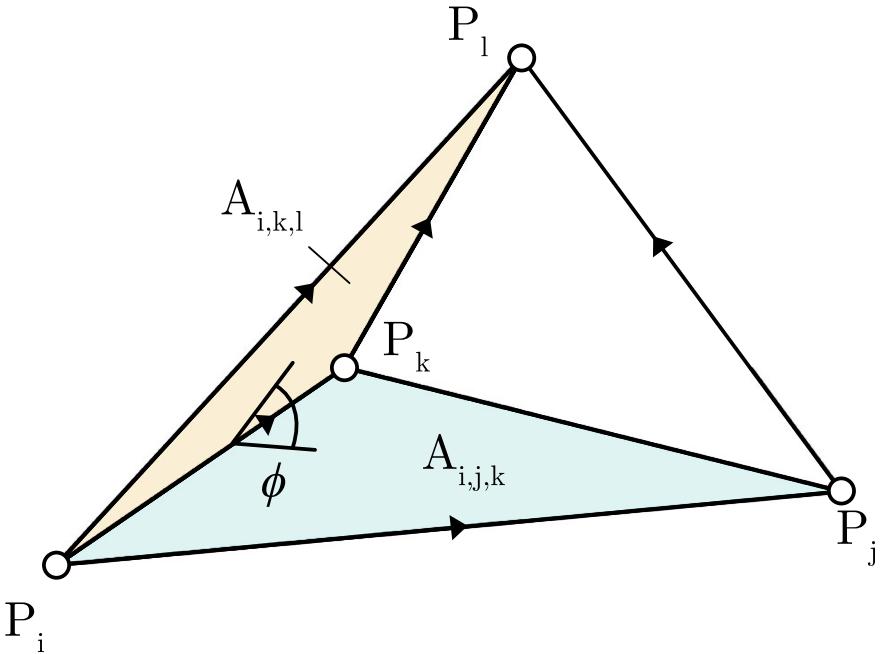


Fig. 7.11 A dihedral angle ϕ of the tetrahedron $\Delta_{i,j,k,l}$ defined by the two triangles $\Delta_{i,j,k}$ and $\Delta_{i,k,l}$. Base vectors $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,k}$ and output vectors $\mathbf{p}_{i,l}$, $\mathbf{p}_{j,l}$, and $\mathbf{p}_{k,l}$ are shown. In this case, $V_{i,j,k,l} > 0$

7.3.1.2 Cayley-Menger Determinants

The *Cayley-Menger bideterminant* of two sequences of n points, $[P_{i,1}, \dots, P_{i,n}]$ and $[P_{j,1}, \dots, P_{j,n}]$, is defined as [29]

$$D(i_1, \dots, i_n; j_1, \dots, j_n) = 2 \left(\frac{-1}{2} \right)^n \begin{vmatrix} 0 & 1 & \cdots & 1 \\ 1 & s_{i_1, j_1} & \cdots & s_{i_1, j_n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & s_{i_n, j_1} & \cdots & s_{i_n, j_n} \end{vmatrix}.$$

When the two point sequences are the same, $D(i_1, \dots, i_n; i_1, \dots, i_n)$, this is abbreviated as $D(i_1, \dots, i_n)$, known as the *Cayley-Menger determinant*. For example, for the 5 points $D(P_1, \dots, P_5)$ this is

$$D(1, 2, 3, 4, 5) = -\frac{1}{16} \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} \\ 1 & s_{1,2} & 0 & s_{2,3} & s_{2,4} & s_{2,5} \\ 1 & s_{1,3} & s_{2,3} & 0 & s_{3,4} & s_{3,5} \\ 1 & s_{1,4} & s_{2,4} & s_{3,4} & 0 & s_{4,5} \\ 1 & s_{1,5} & s_{2,5} & s_{3,5} & s_{4,5} & 0 \end{vmatrix}. \quad (7.2)$$

For the general point sequence P_1, P_2, \dots, P_n , the Cayley-Menger determinant gives $(n-1)!$ times the squared hypervolume of the simplex spanned by the points in \mathbb{E}^{n-1} [40]. Hence, $D(1, 2, 3, 4, 5) = 0$ in \mathbb{E}^3 . Similarly, for $n = 3$, we have [41],

$$D(i, j, k) = 4A_{i,j,k}^2 = \| (P_j - P_i) \times (P_k - P_i) \|^2, \quad (7.3)$$

which is the Heron's formula relating the area $A_{i,j,k}$ of triangle $\triangle_{i,j,k}$. This can also be expressed purely in interpoint distances as:

$$A_{i,j,k} = (p_{i,j,k}(p_{i,j,k} - \|\mathbf{p}_{i,j}\|)(p_{i,j,k} - \|\mathbf{p}_{i,k}\|)(p_{i,j,k} - \|\mathbf{p}_{j,k}\|))^{\frac{1}{2}}, \quad (7.4)$$

where $p_{i,j,k}$ is half the perimeter of the triangle $\triangle_{i,j,k}$ defined as

$$p_{i,j,k} = \frac{1}{2} (\|\mathbf{p}_{i,j}\| + \|\mathbf{p}_{i,k}\| + \|\mathbf{p}_{j,k}\|). \quad (7.5)$$

For $n = 4$ we obtain the orientated volume $V_{i,j,k,l}$ of the tetrahedron $\triangle_{i,j,k,l}$ as

$$D(i, j, k, l) = 36V_{i,j,k,l}^2. \quad (7.6)$$

It is defined as positive if $|\mathbf{p}_{i,j}, \mathbf{p}_{i,k}, \mathbf{p}_{i,l}| > 0$, and negative otherwise [38]. For *Cayley-Menger bideterminants*, for $n = 3$ we have

$$\begin{aligned} D(i, j, k; i, k, l) &= 4A_{i,j,k}A_{i,k,l} \cdot \cos(\phi_{i,j,k,l}) \\ &= ((P_i - P_k) \times (P_j - P_k)) \cdot ((P_i - P_l) \times (P_k - P_l)), \end{aligned} \quad (7.7)$$

where $\phi_{i,j,k,l}$ is the dihedral angle between the two planes defined by the triangles $\triangle_{i,j,k}$ and $\triangle_{i,k,l}$. This can be seen in Fig. 7.11.

7.3.1.3 Trilateration

Trilateration trilateration is a method for computing the location of an unknown point using known distances of the point from 3 different known sites. For example, given a tetrahedron $\triangle_{i,j,k,l}$ (Fig. 7.11), we can compute the output vector $\mathbf{p}_{i,l}$ as [38]

$$\mathbf{p}_{i,l} = \mathbf{W}_{i,j,k,l} \mathbf{q}_{i,j,k}, \quad (7.8)$$

where

$$\mathbf{W}_{i,j,k,l}^T = \frac{1}{4A_{i,j,k}^2} \begin{pmatrix} -D(i, j, k; i, k, l)\mathbf{I} \\ D(i, j, k; i, j, l)\mathbf{I} \\ 6V_{i,j,k,l}\mathbf{I} \end{pmatrix}, \quad (7.9)$$

with \mathbf{I} being the 3×3 identity matrix.

7.3.2 Workspace Definition

The workspace of a robot arm is defined as the region (or surface/volume) within which every point can be reached by the end effector, and is one of the most important specifications for both robot designers and users [42]. We can either compute the workspace given the structure (analysis), or alternatively determine the robot structure from a desired workspace (synthesis) [43]. For the case of malleable robots, traditional strategies for the computation of workspaces based on the attachment of reference frames to the robot joints, such as those that make use of the Denavit-Hartenberg convention [32, 33], cannot be employed since both link dimensions and the relative orientation of the joints can change. An alternative is to perform the workspace analysis using screw theory [34, 35] or distance geometry [29] as in these approaches the parameterisation does not depend on relative angles and distances between joint reference frames. We make use of a distance-geometry-based method herein as the technique has been shown to simplify the computation of the workspace equation of complex mechanisms [36, 37].

We define our 2-DOF malleable robot with a vertical rotary joint at the base, connected co-linearly to one end of a malleable link, which terminates at a second rotary joint mounted perpendicularly to the other end of the malleable link. A second, rigid link is then attached to the second rotary joint, also perpendicularly, which then terminates at the end effector. Since a link connecting two skew revolute axes can be modelled as a tetrahedron by taking two points on each of these axes and connecting them all with edges, and a rigid link connected to a revolute axis can be modelled as a triangle by taking two points on the axis and a point at the end of the link and connecting them all with edges [36], we can model the developed 2-DOF malleable robot using distance geometry as a bar-and-joint framework of 6 points and 12 edges, shown in Fig. 7.12, with P_5 corresponding to the end-effector, and P_1 corresponding

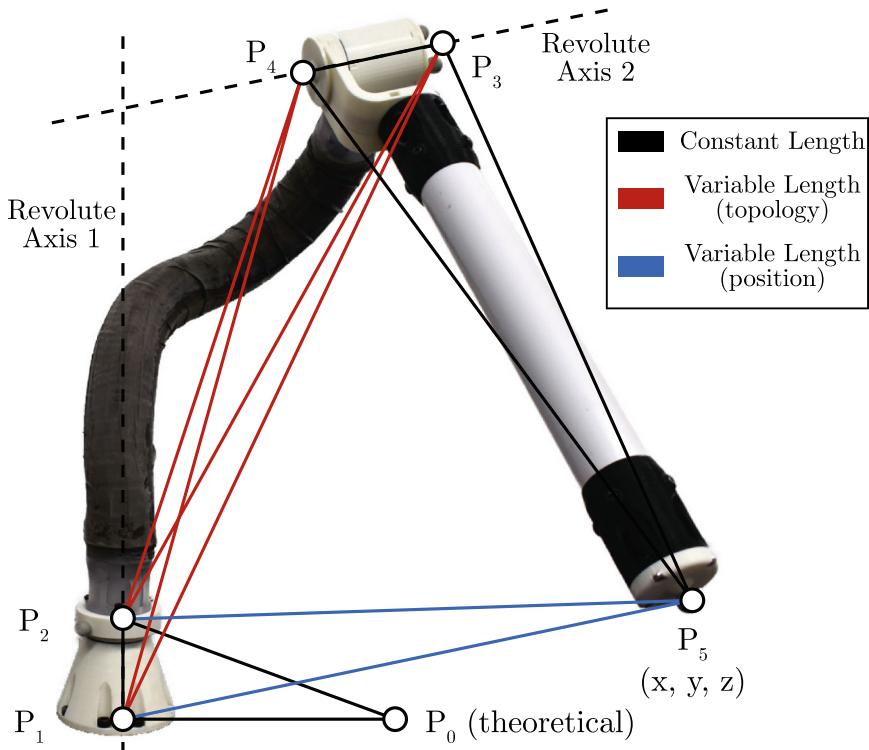


Fig. 7.12 The 2-DOF malleable robot arm can be modelled as a bar-and-joint framework formed by connecting 5 points: P_1 and P_2 , which define the first axis; P_3 and P_4 , which define the second axis; and P_5 , which corresponds to the centre of the end effector. An additional offset point P_0 is added, which is used to define the angle of the first axis. The distances between points can then be categorised as constant distances that do not change (**black**), distances that vary based on the topology configuration of the robot (**red**), and distances that vary based on the joint positioning of the robot (**blue**)

to the robot origin. The axes of the revolute joints are defined by the points P_1 and P_2 for joint 1, and P_3 and P_4 for joint 2. An additional fixed point P_0 is defined offset from the origin at P_1 , which is necessary for the forward and inverse kinematics, but not for the workspace definition or topology reconfiguration. The interpoint distances can further be categorised as:

1. Distances with constant length, that do not change with variation in robot topology and positioning ($\mathbf{p}_{0,1}$, $\mathbf{p}_{0,2}$, $\mathbf{p}_{1,2}$, $\mathbf{p}_{3,4}$, $\mathbf{p}_{3,5}$, $\mathbf{p}_{4,5}$).
2. Distances that vary with changes in robot topology, but not robot positioning ($\mathbf{p}_{1,3}$, $\mathbf{p}_{1,4}$, $\mathbf{p}_{2,3}$, $\mathbf{p}_{2,4}$).
3. Distances that vary with changes in robot positioning, but not robot topology ($\mathbf{p}_{1,5}$, $\mathbf{p}_{2,5}$).

For defining the workspace, we can represent this as a Cayley-Menger determinant of 5 points (not including P_0), shown in Eq. 7.2. Since $D(1, 2, 3, 4, 5) = D(4, 3, 2, 1, 5) = 0$, using properties of the determinant of block matrices [44], it can be shown that this condition can be compactly expressed using 3×3 matrices as

$$D(1, 2, 3, 4, 5) = 2 s_{1,2} s_{1,5} s_{2,5} \det(\mathbf{A} - \mathbf{BCB}^T) = 0, \quad (7.10)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & s_{3,4} \\ 1 & s_{3,4} & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ s_{2,4} & s_{1,4} & s_{4,5} \\ s_{2,3} & s_{1,3} & s_{3,5} \end{bmatrix}, \text{ and}$$

$$\mathbf{C} = \frac{1}{2} \begin{bmatrix} -\frac{s_{1,5}}{s_{1,2}s_{2,5}} & \frac{1}{s_{1,2}} & \frac{1}{s_{2,5}} \\ \frac{1}{s_{1,2}} & -\frac{s_{1,2}}{s_{1,2}s_{1,5}} & -\frac{s_{1,5}}{s_{1,2}} \\ \frac{1}{s_{2,5}} & \frac{1}{s_{1,5}} & -\frac{s_{1,5}}{s_{1,5}s_{2,5}} \end{bmatrix}.$$

Following the notation of Fig. 7.12, Eq. (7.10) is solely satisfied in the points in \mathbb{E}^3 where a 2-DOF malleable robot can physically exist. This fact can be exploited to compute the Cartesian equation of the robot workspace, say $\Gamma(x, y, z)$, by deriving the locus of point P_5 , the end effector, the coordinates of which are $\mathbf{p}_5 = (x, y, z)$ in a particular reference frame. To simplify this computation we can assume, without loss of generality, that P_1 equals the origin of the global reference frame and that P_2 is located in the positive side of the z -axis, such that $\mathbf{p}_1 = (0, 0, 0)$ and $\mathbf{p}_2 = (0, 0, d_{1,2})$. Therefore,

$$\begin{aligned} s_{1,5} &= x^2 + y^2 + z^2 \\ s_{2,5} &= x^2 + y^2 + z^2 - 2d_{1,2}z + s_{1,2}. \end{aligned} \quad (7.11)$$

Substituting Eq. (7.11) into Eq. (7.10), fully expanding the result and rearranging terms, we get

$$\begin{aligned} \Gamma(x, y, z) &\stackrel{\text{def}}{=} q_0(x^2 + y^2 + z^2)^2 + q_1 d_{1,2} z (x^2 + y^2 + z^2) \\ &\quad + q_2 x^2 + q_3 y^2 + q_4 z^2 + q_4 d_{1,2} z + q_5, \end{aligned} \quad (7.12)$$

where q_i , $i = 0, \dots, 5$ are polynomials in $s_{1,2} = d_{1,2}^2, s_{1,3}, s_{1,4}, s_{2,3}, s_{2,4}, s_{3,4}, s_{3,5}$, and $s_{4,5}$. $\Gamma(x, y, z)$ is an algebraic surface of degree 4 (a quartic surface) that corresponds to the workspace surface, traced by the end effector (point P_5), of a 2-DOF malleable robot. The expressions of the polynomials q_i are not included here due to their length, however these polynomials can be easily reproduced using a computer algebra system following the steps given above.

By providing constraints to the two revolute axes of the malleable robot, we can define certain workspace categories belonging to specific robot configurations (topologies). Malleable robots are a general purpose serial robot, and so follow

similar applications where the task workspace defines the configuration. The robot configurations we define are spherical, PUMA-like, SCARA, and general articulated. The constraints for each of them are discussed next.

7.3.2.1 Spherical (or Variable Radius) Case

In a spherical robot configuration, the two revolute axes of the robot coincide at the base, such that, according to the notation of Fig. 7.12, points P_1 and P_3 are coincident. Thus, $s_{1,3} = 0$, $s_{2,3} = s_{1,2}$, and $s_{3,4} = s_{1,4}$. Substituting these values into (7.12), we obtain

$$\Gamma_A(x, y, z) \stackrel{\text{def}}{=} x^2 + y^2 + z^2 - s_{3,5} = 0, \quad (7.13)$$

which corresponds to the equation of a sphere of radius $d_{3,5}$ centred at P_1 . Observe that in this case the radius $d_{3,5}$ is not constant, it can be adjusted according to need. An example of this workspace can be seen in Fig. 7.13a.

7.3.2.2 PUMA-Like (or Variable Centre and Radius) Case

In a PUMA-like robot configuration, the two revolute axes of the robot are perpendicular and coincide at a point located in the positive side of the z -axis, such that points P_2 and P_4 are coincident, and the angle $\angle P_1 P_2 P_3$ is $\frac{\pi}{2}$. Thus, $s_{2,4} = 0$, $s_{1,4} = s_{1,2}$, $s_{3,4} = s_{2,3}$, and $s_{1,3} = s_{1,2} + s_{2,3}$. Substituting these values into (7.12), we get

$$\Gamma_B(x, y, z) \stackrel{\text{def}}{=} x^2 + y^2 + (z - d_{1,2})^2 - s_{4,5} = 0, \quad (7.14)$$

which corresponds to the equation of a sphere of radius $d_{4,5}$ centred at P_2 . Observe that in this case the centre $(0, 0, d_{1,2})$ and radius $d_{4,5}$ are not constant, they can be adjusted according to need. The same equation is obtained when the perpendicularity of the two axes is relaxed. An example of such a workspace can be seen in Fig. 7.13b.

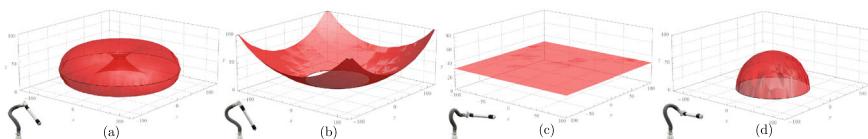


Fig. 7.13 Simulated example workspaces for each type of robot topology achievable by a 2-DOF malleable robot: **a** Spherical, **b** PUMA-like, **c** SCARA, and **d** General Articulated [7]

7.3.2.3 SCARA (or Planar) Case

In a SCARA robot configuration, the two revolute axes of the robot are parallel. Using projective geometry arguments, this implies that there exist a point in the second axis, say P_3 , such that the distance between it and the xy -plane is δ , with $\delta > 0, \delta \rightarrow \infty$. Hence, $d_{1,3} = \delta, d_{2,3} = d_{1,2} + \delta, d_{3,4} = z_4 + \delta, d_{3,5} = z_5 + \delta$, being z_i the distance between P_i and the xy -plane. Substituting these values into (7.12), we obtain an equation that can be written as a quadratic polynomial in δ , say $\Omega = k_2(x, y, z)\delta^2 + k_1(x, y, z)\delta + k_0(x, y, z) = 0$. By factoring out δ^2 in this polynomial, we get $\Omega = \delta^2(k_2(x, y, z) + \frac{k_1(x, y, z)}{\delta} + \frac{k_0(x, y, z)}{\delta}) = 0$. Since $\delta \rightarrow \infty$, then $\Omega = k_2(x, y, z) = 0$.

Since the two revolute axes of the robot are parallel, we have to include additional constraints in $\Omega = k_2(x, y, z) = 0$, that is, $P_2 = P_4 = P^\infty$. This implies that $s_{2,4} = 0$ and $d_{1,4} = d_{1,2}$. Substituting these values into $\Omega = k_2(x, y, z) = 0$, we get $(z_4 - d_{1,2})s_{1,2}\Phi(x, y, z) = \Phi(x, y, z) = 0$. We can then include the final constraint $z_4 = d_{1,2}$ (as $P_2 = P_4$) in the result ($\Phi(x, y, z)$). This yields,

$$(z - z_5)(x^2 + y^2 + z^2 - 2d_{1,2}z + d_{1,2}^2 - s_{4,5}) = 0.$$

Following a similar procedure in the above equation to that done for δ , but in this case for $d_{1,2}$ ($d_{1,2} \rightarrow \infty$ since $P_2 = P_4 = P^\infty$), we finally get

$$\Gamma_C(x, y, z) \stackrel{\text{def}}{=} (z - z_5) = 0, \quad (7.15)$$

which corresponds to the equation of a plane parallel to the xy -plane. Observe that z_5 , the distance between the end effector and the xy -plane, is not constant and can be adjusted according to need. An example of this workspace can be seen in Fig. 7.13c.

7.3.3 General Articulated

We define the general articulated robot configuration as any robot configuration that does not comply with any of the constraints of the 3 other defined robot configurations, thus the form of the workspace surface in this case is $\Gamma(x, y, z) = 0$ (Eq. (7.12)). An example of this workspace, which corresponds to a torus, can be seen in Fig. 7.13d.

7.3.4 Forward Kinematics

In computing the forward kinematics of the malleable robot, we specify the joint angles (in this case, the dihedral angles—use of the physical joint values is addressed later), and obtain the new position of P_5 .

We assume the robot is in a fixed topology, and that the current point positions (and therefore their interpoint distances) are known. Starting from the origin, we define $\phi_{1,0,2,3}$ as the joint 1 dihedral angle between the fixed triangle $\triangle_{0,1,2}$ and the current topology defined triangle $\triangle_{1,2,3}$. We then define $\phi_{3,2,4,5}$ as the joint 2 dihedral angle between the current topology defined triangle $\triangle_{2,3,4}$ and the constant length triangle $\triangle_{3,4,5}$.

We first compute the new location of P_3 , defined by the new dihedral angle $\phi_{1,0,2,3}$ value. Using the interpoint distances we can calculate the areas of both triangles ($A_{0,1,2}, A_{1,2,3}$) using Eq. 7.3. Substituting into Eq. 7.7, we can solve for the new distance $\mathbf{p}_{0,3}$. With all distances for the tetrahedron $\triangle_{1,0,2,3}$ known, we can compute the new position of P_3 using trilateration.

After using Eq. 7.6 to compute the orientated volume $V_{1,0,2,3}$, we can use the now known tetrahedron $\triangle_{1,0,2,3}$ distances to compute the new position of point P_3 :

$$P_3 = P_1 + \mathbf{W}_{1,0,2,3} \mathbf{q}_{1,0,2}. \quad (7.16)$$

With the new position of P_3 known, and the constant distance $\mathbf{p}_{3,4}$, we can compute the position of P_4 using the points P_1 , P_2 , and P_3 and their known interpoint distances:

$$P_4 = P_1 + \mathbf{W}_{1,3,2,4} \mathbf{q}_{1,3,2}. \quad (7.17)$$

Finally, we can compute the new location of P_5 using the same process for P_3 , using the new positions of P_3 and P_4 and the new dihedral angle $\phi_{3,2,4,5}$ for joint 2:

$$P_5 = P_3 + \mathbf{W}_{3,2,4,5} \mathbf{q}_{3,2,4}. \quad (7.18)$$

As the orientation of the volume of tetrahedra requires all distances to be known (Eq. 7.6), in determining P_3 and P_5 we instead define the orientation based on the dihedral angles, where it is defined as positive if $\phi_{i,j,k,l} < 180^\circ$ and negative otherwise, where $0^\circ \leq \phi_{i,j,k,l} \leq 360^\circ$.

7.3.5 Inverse Kinematics

For the inverse kinematics of the robot, we provide the end effector position (P_5) and compute the required dihedral angles necessary to obtain it. It is assumed the current topology of the robot (its positions and interpoint distances) are all known.

We first calculate the angle of joint 2, $\phi_{3,2,4,5}$. Using the new position of P_5 , we know the new distance $\mathbf{p}_{2,5}$. Rearranging Eq. 7.7 to solve for $\phi_{i,j,k,l}$ we obtain

$$\cos(\phi_{i,j,k,l}) = \frac{D(i, j, k; i, k, l)}{4A_{i,j,k}A_{i,k,l}},$$

$$= \frac{D(i, j, k; i, k, l)}{D^{\frac{1}{2}}(i, j, k)D^{\frac{1}{2}}(i, k, l)}. \quad (7.19)$$

Using Eq. 7.19 along with Eq. 7.4, we can calculate the angle of joint 2 $\phi_{3,2,4,5}$ using only interpoint distances. This procedure returns the smallest value of the dihedral angle, limited to the range $0^\circ \leq \phi_{i,j,k,l} \leq 180^\circ$. We can extend this value to the full 360° range to find the two valid values of the dihedral angle, where it is $360^\circ - \phi_{i,j,k,l}$ or $\phi_{i,j,k,l}$ otherwise. These are known as the *elbow up* and *elbow down* configurations of an arm.

Next, we can compute the new position of P_3 as performed in the forward kinematics using Eq. 7.8, using the current topology points of the robot, along with the new position of P_5 . Thus,

$$P_3 = P_5 + \mathbf{W}_{5,1,2,3}\mathbf{q}_{5,1,2}. \quad (7.20)$$

With the new position of P_3 known, we can repeat the procedure for calculating $\phi_{3,2,4,5}$ for $\phi_{1,0,2,3}$. If desired, the new location of P_4 can also be computed using Eq. 7.8 and the calculated dihedral angle $\phi_{1,0,2,3}$.

7.4 Augmented Reality for Collaborative Reconfiguration

7.4.1 Introduction

The malleable robot, as introduced in the previous sections, offers a versatile and robust solution for a new class of low-cost reconfigurable robots. With the ideation of the malleable link, enabled through layer jamming, the user is offered the option of re-configuring the link into wide range of operation shapes, hence eliminating the need for designing a new robotic system in order to satisfy the new task. This results in a low DOF robot having a significantly increased task versatility, due to the variable workspace of the robot. Naturally, this raises a new set of challenges. For extrinsic malleable robots, where their reconfiguration is performed externally by reshaping the malleable link, the user must understand how to do so such that the resulting position allows the robot to perform a task. This involves conducting path planning and motion control without the help of classical forward and inverse kinematics frameworks, while also accounting for the presence of a human operator, and work towards an automated positioning system.

Human-in-the-loop robotic assistance and control, or otherwise commonly known as a form of human robot interaction, is a well-studied field that acts as an important cornerstone for semi-autonomous robotic systems. Having originated in the 1980s with the emergence of intelligent robots and artificial intelligence [45], HRI systems were explored for industrial automation and mobile robotics under different settings, paving the way for modern day systems. 30 years later, researchers began



Fig. 7.14 The developed augmented reality-assisted reconfiguration system of a malleable robot. The user is shown interacting with the AR assisted points placement scene overlaid on top of the robot base, with the presence of a transparent configuration space. The lectern to the right prints out the instructions for the task the user needs to conduct

to incorporate human like features into such systems, including facial expressions in the case of humanoid and assistive robots, as well as gestures and spatial sound in the case of visual assisted systems [46]. In recent years, HRI research has continued to evolve, with a focus on developing robots that can adapt to the needs and preferences of individual users. Additionally, there has been increasing interest in the development of collaborative robots (cobots) that can work alongside humans in various settings. However, despite the progress made, under an industrial setting, robots still excel at performing precise, accurate and repetitive tasks while being kept away from humans, since it is difficult to visualise their intentions in real time and communicate that to a nearby user. In this section, we will first explore this challenge from the perspective of HRI and investigate the available tools from literature that are suitable to us, as well as any necessary innovation aspects that are needed, then we will provide a guided walkthrough of the development process (Fig. 7.14).

7.4.2 Human Robot Interaction

In this work, we have focused on the development of a physical human-robot collaboration system, which is defined as the situation where humans, robots and the environment come into contact in order to accomplish a task, forming a tightly coupled dynamic system. What is important about this definition is that each component must be able to observe, and estimate the contribution of the other counter parts

through sensory information. Once the changes in the environment is understood, the system then reacts to the change or enhances the observation to improve the collective performance [47]. One of the most common HRI systems take the form of fusing assistive environment or cues with the reality, and suitable reactions from the system can be provided. Applications of the fusion system is especially important in safety critical applications, such as during maintenance or surgery. In Neges et al.'s work [48], the team utilised natural markers observed from the maintenance scene in order to better assist engineers and perform indoor navigation, thus significantly improving the efficiency of the work performed. Other works by Lee et al. [49] has integrated augmented reality (AR) with facility management system, thus providing real time monitoring information of critical components in a building. Other examples include robotic surgery, where collaborative systems have been developed to assist with registration between the surgical scene and the medical instruments. For example, in the realm of orthopaedic surgery, Iqbal et al. [50] demonstrated a precise AR registration system that provides holographic feedback to the user depending on their position of the tool, where a guidance cylinder was used to provide the user with feedback on the quality of their tool alignment. On the other hand, HRI system may be control-based, where a specialised controller is designed to help the user in a passive or active manner in order to collaboratively achieve the goal. Notably, active control may take the form of feedforward [51], where the system predicts the user's intentions, or feedback, which reacts to inputs of the system and minimises the error between the expectation and reality. An example of HRI in the form of dynamic active constraints can be found in Bowyer et al.'s work, where passive control was used to monitor the user's actions, while providing a feedback to the user when a condition is met (eg. The user is deviating from a predefined metric [52], such as virtual boundaries). Despite its widespread adoption in the field, such algorithms were not adopted in our designed system, thus will not be discussed further. However, the reader is encouraged refer to [52] for more details.

As the malleable robot does not possess any intrinsic actuation, the study of HRI will primarily focus on evaluating solutions for human supervisory control [53]. With that said, most researchers agree that there needs to be a shared understanding of the robot's workspace in order to initiate the configuration process, such that effective collaboration can take place. Although this method of interaction is still novel, a study by Solymán et al. [54] with 2D workspace visualisation has demonstrated the potential of these workflows to accelerate robot assembly tasks. The system provides an interface that allows the user to define objects in the scene to move, perform trajectory planning and check for collision. However, performing tasks with 2D supervision is not desirable when the task is more complex, for example when manipulating objects in free space. If similar solutions were adopted, the user must master the ability to extrapolate from 2D images and reconstruct the 3D scene mentally. Such challenges have been commonly faced in surgery, where slices of 2D ultrasound or x-ray was used to perform complex procedures, such as in endovascular aneurysm repair [55], albeit surgeons have undergone extensive specialist training in order to master that skill. On some occasions, the lack of information on image depth may even compromise patient safety [56]. Overall, these challenges were commonly encountered in

2D workspace visualisation studies in literature, thus demonstrating the importance of a 3D visualisation experience. With that said, 3D visualisation can take the form of full immersion (in the case of virtual reality, VR) and in the form of digital-reality fusion (in the case of augmented reality, AR). Under VR, the ability of the system to decouple the user from reality simplifies the problem of mapping from a user to a robot's reference frame. This translates the problem into the teleportation domain, where the user is effectively away from the scene of operation. However, this has led to problems such as poor user experience, difficulty in using the system, and inaccurate bi-directional mapping [57]. Lipton et al. has attempted a novel idea of using an intermediate "VR control room" in order to give the user the same point of view as the robot, making the experience more realistic [58]. Nevertheless, VR technology is limited to teleoperation tasks, where the robot is instructed from afar about how it should move. On the other hand, AR has gained popularity in the past decade, with its benefit of providing hands-free 3D guidance for a wide range of tasks. AR systems have matured over the past decade with the introduction of Microsoft HoloLens, an HMD system which capitalises on posture, gesture, vision, and voice to enrich the user experience, while also offering a vibrant AR developer's community that supports its fast growing user base. AR systems have been used in the past for applications in motion planning [59], in control [60], and visualisation [61]. Recently, AR has also been applied to the field of computer assisted neurosurgery, where mean target errors of 3.44 mm on average [62] were achieved. The reader is directed towards the review by Makhataeva et al. [46] for more information of existing surgical systems. While enhancing the performance of precision medicine, AR systems have also received appraisal from their users. Rosen et al. performed a study that used an AR predictive system to inform the user about the fate of a trajectory and whether collisions will occur [59]. Data was collected from two user groups, with one being exposed to AR support and another without. At the end of the test, the accuracy of the two groups' predictions was evaluated, with the result indicating that the users of HMD yielded higher performance (75% vs 65%). These studies demonstrate the pinnacle of state of the art AR systems and what they are capable of achieving. In this work, we therefore explore the use of visual feedback for HRI in the form of workspace visualisation and reconfiguration guidance, applied specifically to reconfigurable robots (Fig. 7.15).

7.4.3 Augmented Reality System Design and Development

To demystify the process of design and architectural building for the reader, this subsection will begin with a brief introduction into the fundamental building blocks of an AR software, including, but not limited to, the Unity game engine, the C# language, the elements of a scene that constitutes an AR experience, as well as the existing interaction strategies that the software engages with the user.

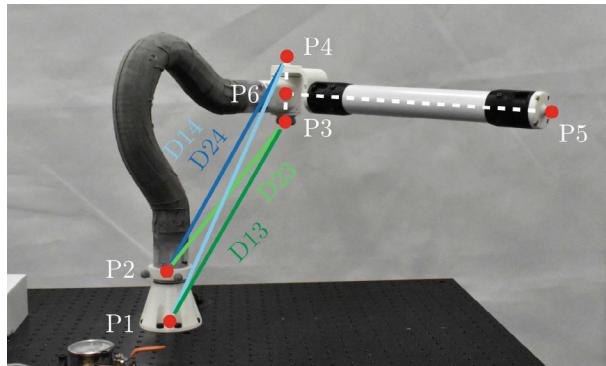
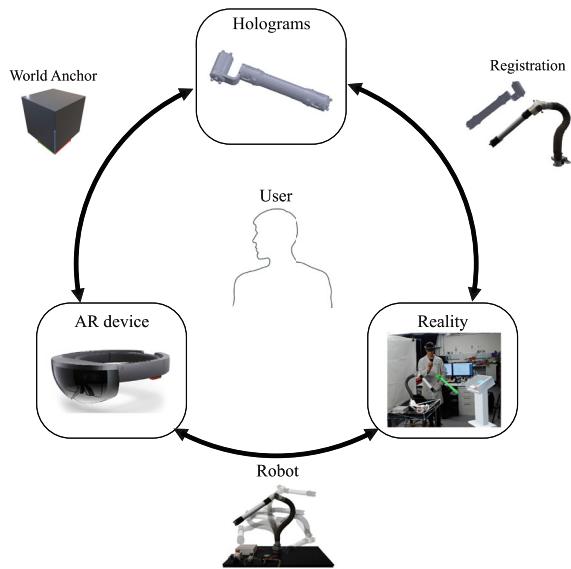


Fig. 7.15 The previously developed 2-DOF malleable robot to be reconfigured, highlighting the point ($P_1 - 5$) representation and interpoint distances ($D_{13}, D_{14}, D_{23}, D_{24}$) that define the malleable link topology

Fig. 7.16 Key AR components



The AR application designed in this section was for a malleable robot system presented in Sect. 7.2, and preliminary discussed in [7]. It is a 2-DOF reconfigurable robot, formed of a vertical revolute joint at the base, a malleable link connecting this base to a second revolute joint, and a rigid link connecting the second revolute joint to the end-effector. This is an extrinsic malleable robot, and thus we believe that there is potential for an AR system to recommend on the end-effector placement (Fig. 7.16).

With any AR interface, there consists of three key components:

- AR device (Microsoft Hololens): An HRI interface consisting of a display (head-mounted (HMD) or simply a device with a display) that facilitates user interactions. Such devices are not only capable of displaying virtual holographic objects, but also have the capability of receiving user intentions and actions through different modalities (Eg. Voice commands, gestures, device relative directions and positions...)
- Holograms (Unity game engine): These are objects that exist in the virtual world, but are displayed through the AR device. Such objects may take a shape which resembles the same object in the real world, or might be a fictitious one, eg. a virtual interactive panel.
- Reality (Malleable robot/Reconfigurable robot): This is the world that we as humans interact with on a day to day basis, and where the robotic system is located. Guidance from holograms and the virtual world should aid actions in the real world.

These components are highly intertwined with each other, and rely on each other's information in order to be defined or operate in harmony. For example, the definition of a hologram is dependent on the world anchor/origin, which is defined in the virtual world with respect to the AR device. When defining other holograms, the world anchor (or origin point) is used to define their positions. In order to make the position definitions as easy as possible, it is often convenient to define the world anchor with respect to a fixed object of interest, for example the base joint of a robot, as that is easy for the HMD as well as the user to see. The world anchor then needs to be registered to the real world, in order for the holograms to be defined at the correct locations. This can be done using optical tracking systems that consists of infrared cameras that outputs the absolute positions of fiducial markers with respect to its own origin, through markerless registration using surface point clouds or object landmarks with learning based approaches [63], or simply via manual alignment. Although the topic of registration is not the focus of this chapter, the reader may refer to [46, 64] for further details of such algorithms. However, it should also be recognised that the importance of registration in the context of human robot interaction is debatable, as it cannot achieve the same degree of accuracy that a robot can. In addition, in the context of a malleable robot, the malleable link and the layer jamming mechanism limits the degree of accuracy achieved by the robot due to its flexibility and mechanical backlash. Finally, there is the malleable robot that the user will manipulate based on the holographic guidance generated. Having previously discussed the engineering design of the malleable robot, in the following section we focus on the main components within the virtual/holographic space, followed by a walkthrough of the main reconfiguration workflow.

7.4.4 Holographic Components

7.4.4.1 The Table and the Robot Base

In this application, the location and orientation definitions of the robot base is crucial, since all of the robot points are defined with respect to it (known also as the world anchor). The coordinate system of our scene can be represented by Fig. 7.17.

We denote the important frames of Hololens as H , the end-effector as E , the base as B and the world origin as W . The Hololens, with its spatial awareness enables by its four tracking cameras, defines its spatial relation with respect to the robot base, and the world anchor., while the end effector is defined with respect to the robot base. The relative relations of the Lectern L is also seen by the Hololens, with its position defined by the user. Conveniently, the Unity game engine defines the relative relation between all objects and a world origin point $\{W\}$. In order to define everything with respect to $\{B\}$ instead, a simple transformation needs to be done. For example, to find $^B T_E$, the transformation from the base to the end-effector, given the relationship between the end-effector $^W T_E$ with origin, and the base $^W T_B$ with the origin:

$$^B T_E = {^W T_E}^{-1} {^W T_B} \quad (7.21)$$

Under Unity, this procedure can be accomplished by defining the local position of virtual objects with respect to the base frame $\{B\}$ using a parent-child relationship. We use the robot base to perform the initial alignment between the holographic world with the real world. To allow the user to deploy the system in any environment, we

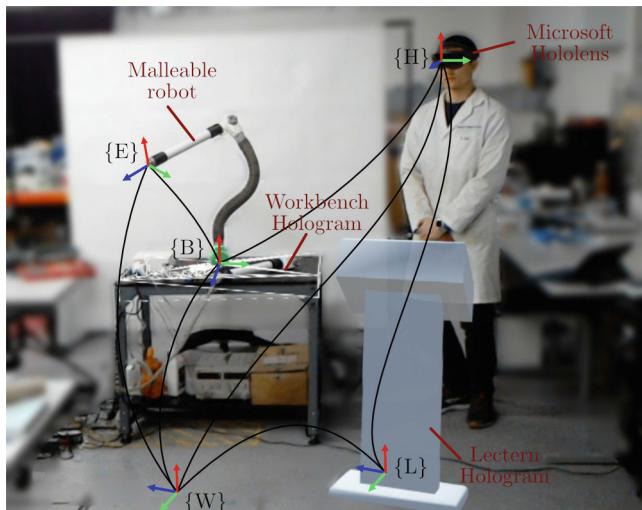


Fig. 7.17 The Coordinate system of the scene, where $\{H\}$, $\{B\}$, $\{L\}$, $\{W\}$, $\{E\}$ represents the frames for the Hololens, the robot base, the lectern, the world origin and the end-effector, respectively

provide the user the freedom to drag and drop the holograms onto the real-world object, such that a manual alignment is performed. Automated solutions using fiducial markers such as QR codes, reflective markers or performing scene understanding are standard alternatives, but sending such information to the Hololens will inevitably increase the computation load, and we prioritized in reducing the load applied to the Hololens to perform such tasks. In order to make the alignment more convenient, the robot base is attached to a larger object, the table, such that the user can manipulate the holograms more conveniently. We also added gesture controls to the object to allow for easy placement in free space, and voice commands to allows hand's free “anchoring” and “releasing” of the world anchor. Furthermore, the table and robot base holograms are freely movable in position, but with its rotation in x and y fixed. This was done to improve the comfort of the alignment process, but more importantly it limits the alignment error and its effects on the system. From a holistic standpoint, the robot base also constitutes a class of objects that is grouped under the robot base module. This consists of the table, the end-effector mesh models, the reconfiguration workspace, and the post-reconfiguration workspace. These objects will only be displayed during when conditions are triggered on the lectern, and when the task requires them.

7.4.4.2 The Lectern

The lectern is the main interface that the user interacts with. It consists of a lectern hologram and a dynamic instruction panel on top. By pressing the buttons on the instruction panel, the user can navigate through the self-explanatory instructions, while the holographic scene changes with respect to the user progression through the tasks. During initial development phases, the ergonomics of the lectern and a floating “tag-along” canvas was compared. The main difference between the two comes down to the preference of the user, however from preliminary user studies, we have found that having the instructions out of the way improves the workflow, as it does not accidentally obstructs the user’s view. In addition, the slanted design of the lecture allows the user to interact with it conveniently while they are standing up. In our case, it can be placed next to the working environment to take over the job of a traditional 2D display.

7.4.4.3 The Reconfiguration Workspace

Given the unique flexible nature of the malleable robot, there exists an infinite number of robot topologies (also known as reconfigurations) that can be achieved within a maximum reachable space [65]. In order to narrow the reconfigurations down to a finite set of solutions, the system should respond to the specific requirements of the task, classify the nature of the task in terms of its feasibility through the use of distance geometry, then display this information back to the user with visual or other feedback mediums. In our application, the “reconfigurable workspace” (or otherwise

known as the maximum reachable workspace) is defined as the volume in which all points within this space can be reached by the malleable robot through the tuning of stiffness of its link, such that there is no joint movement of the robot end-effector involved. The reconfigurable workspace is limited by the physical constraints of the robot, or more specifically, by the maximum reach (L_{max}) and the minimum radius of curvature (r_{min}) of the malleable link, which corresponds to the involute and cycloid curves. This can be defined as follows:

$$x = r(1 - (\cos(t) + (t - a)\sin(t))) \quad (7.22)$$

$$y = r(\sin(t) - (t - a)\cos(t)), \quad (7.23)$$

and the cycloid curve is defined as:

$$x = r(1 - \cos(t)) \quad (7.24)$$

$$y = r(t - \sin(t)) + L_{min}, \quad (7.25)$$

where a is the maximum bending angle of the malleable link, t is the variable curvature of the malleable link, and r is the radius of curvature. Values of $r_{min} = 110$, $L_{max} = 700$, $L_{min} = 550$, and $a = 5$ rad were used. However, such curves still does not take into account the rigid distal link, which further extends this reconfigurable workspace. By offsetting this curve by the length of the distal link, then sweeping the 2D area enclosed by the involute and cycloid curves, we were able to generate the final reconfiguration volume for the interface (Fig. 7.18).

The reconfigurable workspace acts as a key decision-making tool for the user, as it directly represents all the possible reachable options that the robot can comply. Following several iterations of development and user feedback, the reconfigurable workspace was found to perform best when it is transparent and interactive, but constantly in the view of the user, as it can speed up the decision-making process.

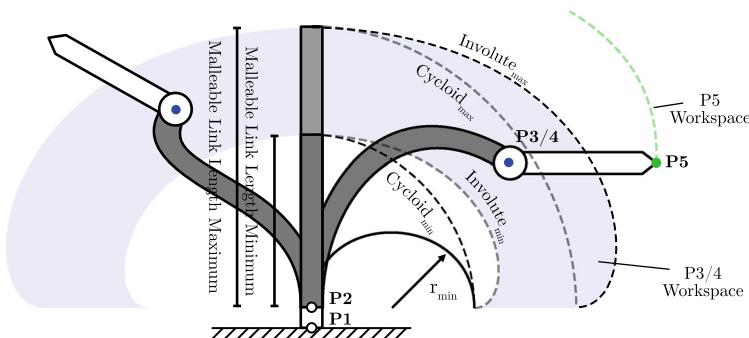


Fig. 7.18 Involute and Cycloid curves traced by the malleable link in both maximum and minimum lengths, demonstrating how the reconfigurable space of P3/4 (blue) and P5 (green) were theoretically calculated

Additional checking algorithms were also implemented in the form of color feedback, such that the user can constantly verify the validity of their desired end-effector solution using collision detection algorithms from Unity. When a valid solution is defined, the end-effector flashes green to indicate that the user can progress to the next step, or turns red if otherwise.

7.4.4.4 The Configured Workspace and the Choice of End Effector Solutions

Following the selection of a target end effector pose, three tasks remain: the generation of a range of topologies which satisfies the user's input, the selection of the suitable configuration that consists of a suitable workspace, the assistance of user alignment to desired configuration. As discussed in section (insert Angus' section on distance geometry), given the desired configuration, the topology computation equation computes a range of achievable solutions in the form of the 4 distances D_{13} , D_{14} , D_{23} , and D_{24} , which defines the distances between P_1 , P_2 , P_3 , and P_4 , corresponding to the robot base origin, directly above the origin, and the two sides of the distal joint respectively. Knowing those distances provides a deterministic pose of the end-effector. Visualising solely interpoint distances is difficult, and using them for reconfiguration is challenging even for experts in robotics. Nevertheless, in our previous work on end-effector alignment, a non-AR, tracking based alignment strategy was proposed [7]. An OptiTrack (Corvallis, Oregon, USA) system with 6 cameras was setup around the malleable robot. The user was given real-time feedback on the error between the expected position and the current position during an alignment, despite it being extremely difficult for the user to use such errors to correct their movement as there are no feedback regarding which direction this error is pointing towards. One alternative approach could be to use the AR interactive environment to mark the desired P3-5 positions in the user's vision, and physically attaching such points on the robot before performing the alignment. However, such system design will impact comfort and ease of use, since having only the 4 points means the user must deduce the position and orientation of the distal link. To ease the user when performing alignment, we applied an additional level of abstraction by directly transforming the end effector using those 4 distances, such that the new robot topology is represented using the mesh model of the distal link, located at the final optimal position defined by the 4 distances. In order to do so, a simple transformation method was developed. Initially, the points P_3 and P_4 were defined using the distances, and P_3 was used as the origin point $[0, 0, 0]$, where the z axis aligns to the line connecting P_3 and P_4 . Then, we align the mesh of the distal link to P_3 , and rotate the model around P_3 until P_4 is aligned. For each combination of points P_3 and P_4 , there exists a unique reconfigured workspace. Please note that this differs from the reconfiguration workspace, since it is the volume that the robot can reach given a particular end-effector reconfiguration. This workspace, is a subset of the larger reconfiguration workspace, defined when the malleable link is rigid, with the end effector placed at a unique pose. During the rotation, multiple end-effector

poses are generated, each possessing its unique workspace. Once generated, the mesh models of the end effector, and of the workspaces are loaded back into the AR scene to complete the reconfiguration. Using the interactive lectern, the user now loops through each of these poses, before selecting one that is most suitable for the task at hand. Finally, the user is guided with interactive holographic waypoints in the form of colored arrows, to depressurize the malleable link, and locking it in its desired place.

7.4.4.5 Summary of Workflow

The main ethos of the application is simplicity and ease of use, hence all of its elements and the instructions it provide are designed to be self-explanatory. Initially, the user performs an alignment between the real robot base and table with its holographic counter part, using voice commands such that they can adjust the manual alignment if the initial attempt was not successful. The user instructs the robot base to ‘anchor’ via voice command, which fixes the base in place. Thereafter, the user follows a standard procedure to perform the alignment in two distinct phases, definition and reconfiguration. In definition, the user begins to configure a scene by observing the reconfiguration space from Sect. 7.4.4.3, instantiating a mesh model of the end effector, then use hand gestures to orientate it to its desired pose. Once complete, the user confirms the position by pressing “check” on the lectern, where an end effector checking algorithm verifies the holograms lie within the reconfiguration space, and prevents invalid hologram transforms to be defined (eg. Outside the reconfiguration space). During checking, the end effector changes color to “green” indicating a valid topology, or “red” vice versa. Before the conditions for “check” is fulfilled, the user is prevented from advancing further, providing a fail safe. At this stage, the user then confirms the coordinates of the distal link points P5 and P6, then proceed to saving the configuration in the scene, and send these information to the workstation for computation. The workstation is capable of computing the optimal topologies of the malleable robot, defined by distance geometry using a sampling of dihedral angles between the triangles formed by the point in question (P4/5/6), the orientation of the end effector and the fixed robot topology, and returns the desired end effector poses, along with their workspaces (as discussed in Section x). The offloading of the mathematical computation allows the Hololens to perform at high frame rates, needed for real time guidance. The final results from the workstation is now transferred to the Hololens. In this implementation, such transfer is performed by saving the scene, closing the application and redeploying to the Hololens. The obvious disadvantage of doing so is the disruption made to the workflow, and the interruption introduced during redeployment. On the other hand, TCP/IP communication is a clear alternative, where a seamless transition between the two phases of workflow is ensured. However, this decision was made in consideration of the hardware limitations of the Hololens, where its computation power is simply not sufficient to handle a mass transfer of mesh models on the fly, without substantially affecting the user experience (eg. Freezing, very low fps rate) (Fig. 7.19).

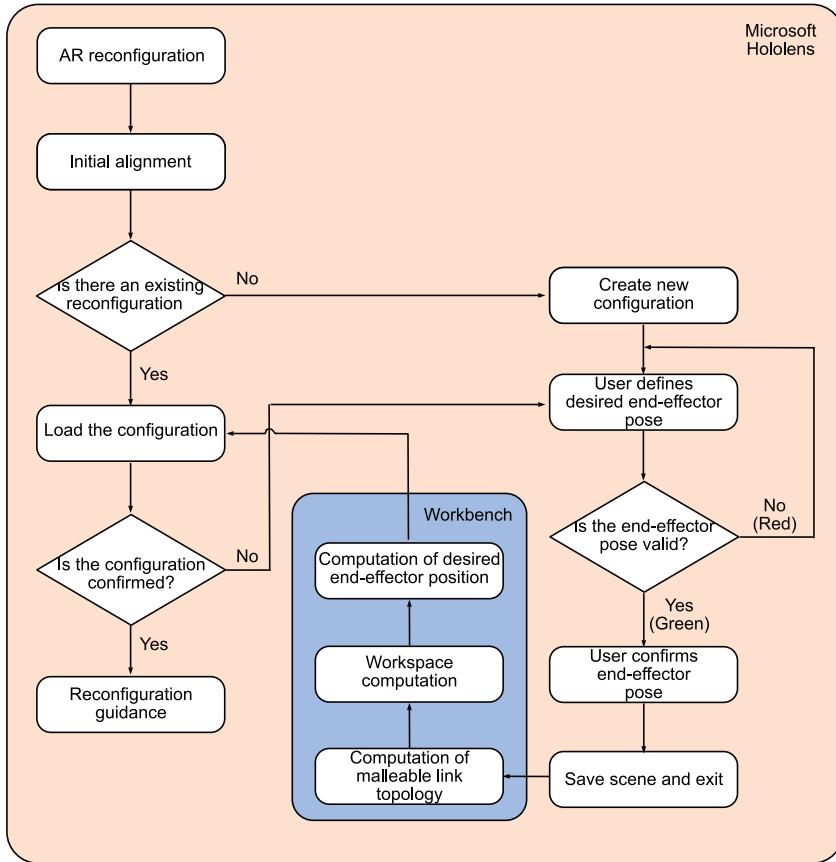


Fig. 7.19 A summary of the reconfiguration algorithm

7.5 Autonomous Reconfiguration

7.5.1 Introduction to Control of Flexible Spines

Malleable robot design incorporates a flexible spine and layer jamming components. The author first investigated controlling the flexible spine of malleable robots using the tendon-driven approach. Unlike the traditional continuum robot with a centre spine, the flexible spine has four soft spines spaced equally around the disk and is compressible. The advantage of using flexible spines is that it allows cables to pass through the center of the malleable robot.

Traditional control of continuum robots mainly relies on an accurate mathematical model. Beam theory, Cosserat rod model and Constant curvature model are commonly used in modelling the continuum robot. Cosserat rod model is the most

accurate among all the continuum robot models with consideration of external loads while being computationally expensive [66, 67]. Alternatively, a learning approach using neural network to model system of soft continuum robots is popular due to non-linear characteristics of soft materials. Machine learning approaches like gaussian process regression, Kalman filter, and Koopman operator model have been used in controlling continuum robots trained with real-world data [68–71]. Deep learning methods are divided into model-free and model-based learning methods. Model-free methods like deep Q-learning utilize neural network and gradient descent to find optimal policy to control continuum robots. Model-based learning methods have also been proposed, mainly relying on a math model. When these deep learning methods trained in simulation, there is a sim-to-real gap needed to fill. Meanwhile, training them in the real-world is not data-efficient.

This section focuses on solving the problem of data-hungry in learning-based methods for controlling continuum robots. A real-world data-efficient model-based learning framework has been proposed, requiring only 100 real-world data to achieve comparable or better performance compared to control policies trained with thousands of real data.

7.5.1.1 Flexible Spine Design and Continuum Robot

The continuum robot we experimented with has four compressible spines. The spine is 3D-printed using thermoplastic polyurethane (TPU). The disks are 3D-printed using PETG material. We connect the four spines with 12 plastic disks with 0.01 m space between them in the relaxed state. The overall length of the continuum robot is 0.22 m. This type of continuum robot is easy to manufacture, and inherently safe to work with due to its flexibility. There are four cables passing through the continuum robot, each controlled by a stepper motor. Meanwhile, the position control is performed by adding position switches to reset each time the robot is powered on.

As shown in Fig. 7.20b, a unique connector was created to join tendons and a lead screw (b). A cable tensioner was included into the 3D-printed connector. By pushing the cable seat with M4 bolts and nuts in the connector, the total length of the cable is shortened. Tendon tension is increased, and the cable is held in place. The incompressible Teflon tubes serve as the continuum robot's guiding routes for the steel-wire (tendons).

7.5.2 Mathematical Modelling Through Cosserat Rod Model

The continuum robot with soft spine is easy to deform under external loads. In this case, cosserat rod model provides us the advantage of accurate modelling while we are able to adjust the model to capture the effect of compression that reduces the overall length of the continuum robot.

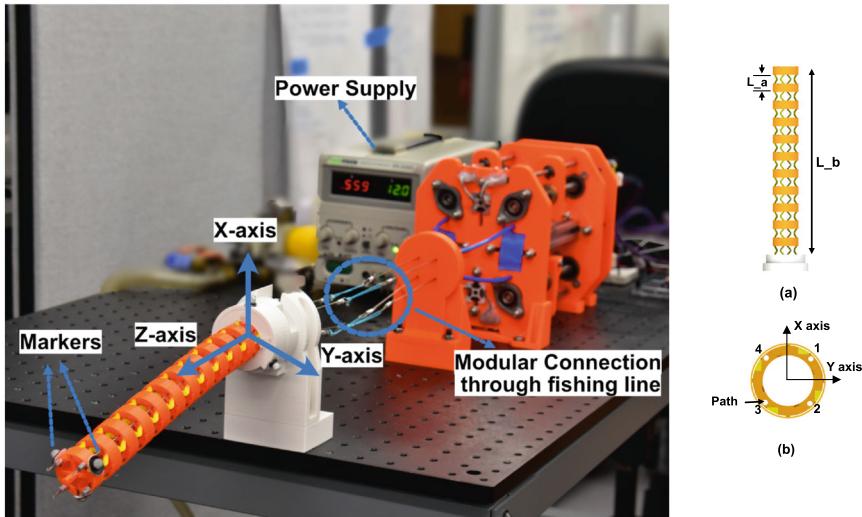


Fig. 7.20 **a** Design of flexible spine. **b** Tendon path and cross-section area in flexible spine

The idea of cosserat rod model is to divide the continuum robot into finite small sections and apply force and moment balancing equation along these small sections (can reference Chap. 2). With known boundary value conditions, we can use the shooting method to solve such problem. In this case, our continuum robot is fixed at one end. And on the other end, assuming we have known its position and orientation to simplify the problem, this becomes a boundary value problem we can solve by having different initial guesses of moments and forces acting on the continuum robot.

When the robot is tendon actuated, the system's given tendon lengths limit the position and orientation of the distal end of the continuum robot. In order to convert this into a new boundary condition, we must integrate the tendon route length to match the given tendon length while making guesses about the moments and forces acting on the continuum robot using the shooting approach.

The Cosserat rod model was first developed in the static model based on the kinematics of a rod. Along the centre-line of the rod, we have defined the position $p(s)$, orientation $R(s)$, the change rate of linear velocity $v(s)$ and angular velocity $u(s)$ at the length s of the rod. For any arbitrary section of a rod, static equilibrium equations of force and moment are applied. Using linear constitutive equations to map internal force $n(s)$ and moment $m(s)$ of a rod, the explicit Cosserat static rod model can be represented as:

$$\begin{aligned}\dot{p} &= Rv, v = K_{se}^{-1}R^Tn + v^* \\ \dot{R} &= R\hat{u}, u = K_{bt}^{-1}R^Tm + u^* \\ \dot{n} &= -f \\ \dot{m} &= -\dot{p} \times n - l\end{aligned}\tag{7.26}$$

where v^* and u^* are reference configurations of a straight rod, in our case, v^* is $[0;0;1]$ and u^* is $[0;0;0]$. And K_{se} is the stiffness matrix for shear and extension and K_{bt} is the stiffness matrix for bending and torsion. They can be represented as:

$$\begin{aligned} K_{se}(s) &= \text{diag}(GA(s), GA(s), EA(s)) \\ K_{bt}(s) &= \text{diag}(EI_{xx}(s), EI_{yy}(s), EI_{zz}(s)) \end{aligned} \quad (7.27)$$

where $G(s)$ is the shear modulus, $A(s)$ is the cross-section area of the rod, $E(s)$ is Young's modulus, I_{xx} and I_{yy} are the second moments of the in the corresponding axis. $I_{yy} = I_{xx} + I_{yy}$.

7.5.2.1 Tendon Model

The tendon path is defined as radius relative to the centre of the spine $r_i(s) = [x_i(s) \ y_i(s) \ 0]^T$ knowing the x, y coordinates of each tendon, assuming the longest dimension of continuum robot inline with z axis. For example, in Fig. 7.20, the cross-section of the continuum robot stays constant along the length s of the continuum robot. r_i becomes fixed value across the continuum robot. Otherwise, it is a function of length s of the continuum robot. The tendon position relative to the centre of the spine with displacement using previous radius and current orientation of the rod can be represented as $p_i(s) = R(s)r_i(s) + p(s)$,

In this tendon force model, we assumed no friction between the tendon and the path it travels to simplify the model. Using static equilibrium equations in Eq. (7.26), we represent the internal force and moment distribution caused by tendons as following:

$$f_t = - \sum_{i=1}^n \tau_i \frac{\hat{p}_i^2}{\|\dot{p}_i\|^3} \ddot{p}_i, \quad l_t = - \sum_{i=1}^n \tau_i (\hat{R}r_i) \frac{\hat{p}_i^2}{\|\dot{p}_i\|^3} \ddot{p}_i \quad (7.28)$$

7.5.2.2 Tendon-Driven Cosserat Model

Now we incorporate the tendon model in f_t and l_t with a static Cosserat rod model. The derived tendon-control static model can be represented as:

$$\begin{aligned} \dot{p} &= Rv, \quad v = K_{se}^{-1} R^T n + v^* \\ \dot{R} &= R\hat{u}, \quad u = K_{bt}^{-1} R^T m + u^* \\ \dot{n} &= -f_t - \bar{f} \\ \dot{m} &= -\dot{p} \times n - l_t - \bar{l} \end{aligned} \quad (7.29)$$

where, $-f_t$ represents distributed tendon tension forces, $-l_t$ represents distributed moment caused by tendon forces. In addition, $-\bar{f}$ and $-\bar{l}$ represent the distributed forces and moments caused by the continuum robot and external loading.

7.5.2.3 Boundary Conditions for Tendon-Length Control using Static Cosserat Model

When comes to using tendon length control, we now guess tendon forces in each tendon and reduce the tendon length error in Error term for given tendon length value for each tendon using a shooting method. The boundary conditions are described as the moment and forces at the tip of the continuum of robots. We first have an initial guess on $v(t, 0)$, $u(t, 0)$, which are the velocity and angular velocity for \forall time t and at length 0 of the rod. Then we applied the shooting method to reduce residual terms for given tendon forces τ_i in each tendon. The shooting method is solved in MATLAB software's fsolve function. Now the initial guess and residual term represented as following:

$$\begin{aligned} \text{Guess} &= [v(t, 0)^T \ u(t, 0)^T \ \tau_1(t) \ \dots \ \tau_n(t)] \\ \text{Error} &= [(E^F)^T \ (E^M)^T \ E_1^l \ \dots \ E_n^l] \\ E_n^l &= (1 + \tau_i)(l^* - l_i + \beta_i) \\ l_i &= l_i^q + l_i^B \\ \beta_i &= \begin{cases} 0, & \tau_i > 0 \\ \gamma^2, & \tau_i < 0 \end{cases} \\ \tau_i &= \begin{cases} \gamma^2, & \tau_i > 0 \\ 0, & \tau_i < 0 \end{cases} \end{aligned} \tag{7.30}$$

where l_i^q is a fixed length from the end of the continuum robots to the motor, l_i^B is an integrated length along the backbone of the continuum robots, and the β_i is a slack that existed in each tendon.

The slack is defined as if the tendon force τ_i is great than zero, then there will be no slack in the tendon. Otherwise, it will have a positive amount of tendon slack, which lets the shooting method optimize other tendons with positive tendon forces.

It is worth noticing that the residual terms need to change according to the number of actuating tendons. If there is only one tendon actuating, only E_1^l is needed to include in the Error term along with the initial guess of the tendon force τ_1 . Otherwise, it is restricting three other tendons from exceeding the original length of the continuum robots. If two tendons are actuating the continuum robot, E_1^l , E_2^l and initial guess of τ_1 , τ_2 are included corresponding to those two tendons and so on.

7.5.2.4 Compressible Spine Modelling

Tendon tensions, which are compression forces, account for the majority of internal forces in the Cosserat rod model. We have taken into consideration the internal forces that compress the spine under the circumstance of low external stress and the continuum robot's lightweight. In this case, we have assumed that the robot is under compression mostly from internal forces generated by tendon tensions.

To characterize the spine compression under tendon tension, we have conducted multiple trials of experiments to determine the length change of our compressible soft spine with respect to the compression force. Firstly, we used a two-section specimen of our compressible spine for testing. A simplified linear equation has been used to represent the relationship between our specimen's changing length and the various force applied on top of it:

$$l_c = c_{spine} \times \|n(s)\|, \quad (7.31)$$

where $\|n(s)\|$ represents the magnitude of internal normal forces acting on the robot. The experiments conclude the coefficient has a value of $0.2 \frac{mm}{N}$ with a limit compression force of 20 Newtons.

As illustrated in Fig. 7.20b, our continuum robot consists of eleven sections of compressible segments. The weight and external load effects on the interior forces $n(s)$ at these 11 sections are calculated using linear constitutive law:

$$n(s) = R[K_{se}(v - v^*)], \quad (7.32)$$

where K_{se} is the stiffness matrix for shear and extension and v^* is the straight rod reference configuration with a value of $[0;0;1]$. Instead of simply reducing the total length of the continuum robots, we iteratively reduced the length of each segment of the continuum robots according to the internal force at the length s of the rod within the shooting method of the Cosserat rod model. The algorithm is described in detail in Algorithm 1. The shooting method continues to optimize the function using MATLAB function fsolve until it reaches a desired error threshold ϵ_t . We discretize the continuum robot into many small sections, $ds(i)$ represents the section length, which is the total length of the continuum robot divided by the number of sections.

7.5.3 Neural Network and Gaussian Process

We introduce a model-based learning method by training a Cosserat rod mathematical model using a neural network. A Gaussian process regression (GPR) is used to model the error between the simulated model and real model. This error model will later be incorporated into the control algorithm to achieve data-efficient control. Three control policies based on the GPR error model and a neural network trained in the

Algorithm 1 Calculating the reduced length of the continuum robot with compressible spine

```

while Error is greater than  $\epsilon_t$  do
    Solving system ODEs for  $\dot{p}$ ,  $\dot{R}$ ,  $\dot{n}$ ,  $\dot{m}$ 
    Using Euler's method to calculate  $p$ ,  $R$ ,  $n$ ,  $m$ 
    for each compressible section  $i$ ,  $i = 1, 2, \dots, 11$  do
        for each Spatial Point  $j$  inside section  $i$  do
            Calculate  $n(i)$  at segment  $i$  with Eq. (7.31)
            Calculate  $l_c(i)$  using Eq. (7.31)
            Update  $ds(i)$  as  $ds(i) = ds(i) - l_c(i)$ 
            Update  $p(j)$ ,  $R(j)$ ,  $n(j)$ ,  $m(j)$  using Euler's Integration method
        end for
    end for
end while

```

simulation are formed to provide a real-world data-efficient framework to control the continuum robot.

7.5.3.1 Neural Network Modelling for the System

As a universal approximator, neural networks have been extensively used to model robotic systems. In terms of continuous robot control, neural networks have outperformed many computationally costly methods, such as finite elements and finite differences, in terms of computational efficiency. In both kinematic and dynamic modeling of continuum robotics, neural network-based approaches have demonstrated comparable or lower error prediction than traditional mathematical models.

To use neural network to simulate such continuum robot system with soft spines, a feed-forward neural network or recurrent neural network was typically used. By providing the control inputs of the systems and resulted end effector positions, the neural network could predict the positions of the continuum robot system after training. The difference between a feed forward neural network and a recurrent neural network is presented in the Fig. 7.21. The benefits of recurrent neural network are to incorporate history into the training, which allows for better capture of dynamic systems. To simulate a static continuum robot, a feed-forward network is typically used in this case.

For modeling the continuum robot with a feed-forward neural network, there are few observable variables in the system. The continuum robot is controlled through four tendons L_1, L_2, L_3, L_4 through the spine, while the end effector position p_i in X, Y, Z axes of the continuum robot is captured using the motion tracking system. By providing the control inputs parameters and corresponding end-effector positions, we can specify the system like this:

$$\begin{aligned} u_i &= (L_1, L_2, L_3, L_4) \\ (u_i, p_i) &\rightarrow p_{i+1} \end{aligned} \tag{7.33}$$

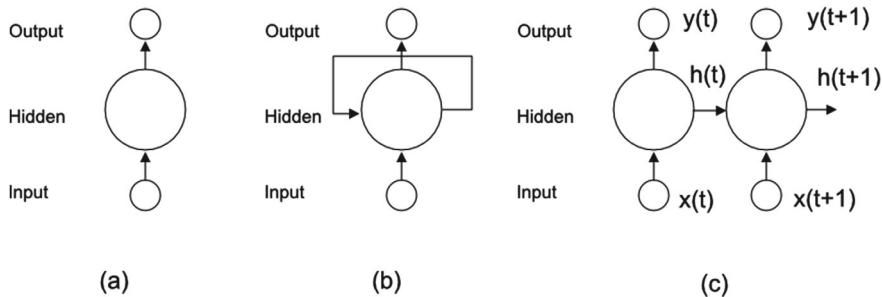


Fig. 7.21 **a** Feedforward neural network. **b** Recurrent neural network. **c** Unfolded recurrent neural network

For each given control input, and the current position of the end effector, we can predict the resulting end-effector position using a neural network. The structure of the neural network can be constructed as an input layer, hidden layers, and an output layer. Taking feed-forward neural network as an example, the hidden layers can be represented as Dense layers using Keras API. In the case of recurrent neural network, the hidden layers is easily represented using simple RNN layers or LSTM (Long Short Term Memory networks) layers.

7.5.3.2 Data Generation for Neural Network Training

To train such neural networks, we first need to generate data from Cosserat rod mathematical model. Given different control inputs u_i and initial guess of velocity $v(t, 0)$, and angular velocity $u(t, 0)$ for \forall time t and at length 0 of the rod, we can use shooting method to estimate the end-effector position of the continuum robot.

In the calibration process of the mathematical model, we use a small number of real position data and corresponding control inputs to adjust Young's modulus (E) of the Cosserat rod model. To perform this optimization process that minimizes the error of simulated end-effector positions and real position data, we can use MATLAB function:fminsearch.

A random motor bubbling technique is used during this data generation process. This method is to add random offsets to the control inputs and generates the corresponding mathematical position data points. By generating data this way, it helps to better capture the system and the trained neural network is less likely to overfit. 10000 data points have generated in this way for neural network training. The training of the neural network is straightforward with a choice of “adam” optimizer, “mean squared error” in the Keras sequential model, and the option of “EarlyStopping” to avoid overfitting.

Since this continuum robot has a soft spine, the calibrated mathematical model alone still has relatively large errors compared to the traditional continuum robot with a rigid spine. The same is true for neural networks trained from this calibrated

mathematical model. In the following section, we present methods that incorporate Gaussian process regression with our trained neural network to achieve accurate modelling while being data-efficient.

7.5.3.3 Gaussian Process Regression

Gaussian process regression has been demonstrated to be data-efficient in model-based system control with fewer data points [69]. It can provide continuous mean value and covariance value estimation for unknown data points using observed data points. The Gaussian process has two main components: a mean function and a Kernel function / Covariance function. Normally, the mean value is constant and assumed to be zero. The covariance function encodes information regarding the expected shape and structure of the function. A simple and popular example is the square exponential (SE) kernel. This also implies if two points in the input space are close to each other, then the outputs (or function values) at those points are likely to be similar as well. GPR could approximate any nonlinear function $y_i \sim f(x_i) + \epsilon$, where ϵ is a white Gaussian noise with zero mean and variance σ_n^2 . The GBP and SE kernel is represented as:

$$\begin{aligned} f(x_i) &\sim GP(m(x_i), k(x_i, x')) \\ k(x_i, x') &= \sigma_s^2 \exp\left(-\frac{1}{2}(x_i - x')^T \lambda(x_i - x')\right) \end{aligned} \quad (7.34)$$

Now we consider an input data $X = x_i$ and output data $Y = y_i$ for $i = 1, 2, \dots, N$. We can select a GP prior distribution on \mathbf{f} for the function $\mathbf{f} = f(X)$ as follows:

$$p(\mathbf{f}|X) = N(\mathbf{f}; \mu(X), K(X, X)) \quad (7.35)$$

Now given some observed data points $D = (X, \mathbf{f})$, we want to make predictions on points X_* . We can write the joint distribution between the known training values $f(X) = \mathbf{f}$ and testing function values $f(X_*) = \mathbf{f}_*$ as follows:

$$p(\mathbf{f}, \mathbf{f}_*) = N\left(\begin{bmatrix} f \\ f_* \end{bmatrix}; \begin{bmatrix} (X) \\ (X_*) \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \quad (7.36)$$

Then the posterior distribution to make predictions for input set x_* can be expressed as follows:

$$p(\mathbf{f}_*|X_*, D) = N(\mathbf{f}_*; m_{f|D}(X_*), K_{f|D}(X_*, X_*)) \quad (7.37)$$

$$m_{f|D}(x) = m(x) + K(x, X)(K(X, X) + \sigma_n^2 I)^{-1}(y - m(X)) \quad (7.38)$$

$$K_{f|D}(x, x') = K(x, x') - K(x, X)(K(X, X) + \sigma_n^2 I)^{-1}K(X, x') \quad (7.39)$$

There is also a drawback, for a more complex system, more data points will be needed in the training process. While GPR becomes more computation expensive with larger amount of training data points. To overcome this drawback, we have selected GPR to model the error term between the data generated from Cosserat rod model and real experiemtn data. The detailed inputs and outputs of our GPR model is described in the following equation:

$$\begin{aligned} x_i &= (L_1, L_2, L_3, L_4, p_{sim}) \\ f(x_i) &= p_{sim} - p_{exp} = e_i \end{aligned} \quad (7.40)$$

Where p_{sim} represents the end-effector position given control inputs of L_1, L_2, L_3, L_4 from the Cosserat rod model. In this way, we could dramatically reduce the number of real data points to only 100 real data points to get an accurate GPR model as explained in [72]. By providing observed data points, the GPR now is ready to make predictions on new data points. The detailed implementation of a Jupiter notebook can also be found on our remote repository.

7.5.4 Data Efficient Control Policies

7.5.4.1 Inverse Mapping of the Continuum Robot

In the Sect. 7.5.3.2, we used a neural network to model the forward kinematics of the continuum robot system. Now in order to control the continuum robot to reach certain positions, we need an inverse mapping of the continuum robot. In the motor bubbling data generation stage, we recorded, control input u_i , current position p_i and next position p_{i+1} . In this case, we just need to rearrange the inputs and train the neural network. The structure of the neural network is the same.

$$(p_i, p_{i+1}) \rightarrow u_i \quad (7.41)$$

To apply the trained inverse mapping neural network, we have a current position and desired position, then we could easily obtain corresponding control input in this case. The representation can be written like this:

$$(p_i, p_{target}) \rightarrow u_i \quad (7.42)$$

7.5.4.2 Control Policy A

The inverse mapping of the continuum robot is trained using simulation data, but there is still a sim-to-real gap. The neural network trained directly from the mathematical model still has noticeable errors. GPR error model in this case can be used to fill the

Algorithm 2 Update $p_{s\text{-target}}$ with iterative approach in closed-loop control

Require: $p_{real\ target}$, $p_{initial}$, time-step: i

initial guess: $p_{s\text{-target}} = p_{real\ target}$

while $\|p_{s\text{-target}}(i-1) - p_{s\text{-target}}(i)\| > 0.001$ **do**

- Get control input $(p_{initial}, p_{s\text{-target}}(i-1)) \rightarrow L(i)$
- Predict using GPR $e_{gp}(i) = f((L(i), p_{s\text{-target}}(i-1)))$
- Obtain tip position $p_{real}(i)$ from tracking system
- Update current position $p_{initial} = p_{real}(i) + e_{gp}(i)$
- $p_{s\text{-target}}(i) = p_{real\ target} + e_{gp}(i)$

end while

sim-to-real gap.

$$\begin{aligned} e_{gp} &= f((L, p_{s\text{-target}})) \\ p_{s\text{-target}} &= p_{real\ target} + e_{gp} \end{aligned} \quad (7.43)$$

Given a target position $p_{real\ target}$ for the continuum robot to reach in the real-world space, we want to find the corresponding control inputs. We have an inverse mapping neural network trained from mathematical equations(simulation). However, in order to allow this neural network to generate correct control output. This $p_{real\ target}$ needs to be compensated with an error term e_{gp} , such that we could obtain a simulated target position $p_{s\text{-target}}$.

Similarly, the error term e_{gp} also depends on this simulated target position. This will need an iterative approach that continuously updates this $p_{s\text{-target}}$ value. The initial guess, we can assume $p_{s\text{-target}} = p_{real\ target}$. Later, this value will be updated as the continuum robot move close to the target position.

When given a real target point $p_{real\ target}$ in the real workspace, now we need to estimate the target point $p_{s\text{-target}}$ in the simulated workspace. Term e_{gp} is also dependent on the input value of $p_{s\text{-target}}$. It then becomes an optimization problem of finding suitable $p_{s\text{-target}}$ value. We take an initial guess of $p_{s\text{-target}} = p_{real\ target}$. feeding it into the RNN and receive tendon length control outputs. Then feeding the control outputs and $p_{s\text{-target}}$ into the GPR model to generate e_{gp} . This error term will be added to $p_{s\text{-target}}$ as a correction of initial guess.

For open-loop control, the current position remains unchanged during the control time step. In contrast, for closed-loop control, the current position is continuously updated based on position feedback. In the meantime, as an input for our inverse mapping model, we assume the current position is from the simulated workspace rather than the actual workspace. The current position will also need to be compensated with an error term e_{gp} from the GPR model as well. As a result, the error term becomes accurate as the robot reaches the target point in the closed-loop control. The controller is presented in Algorithm 2 and Fig 7.22.

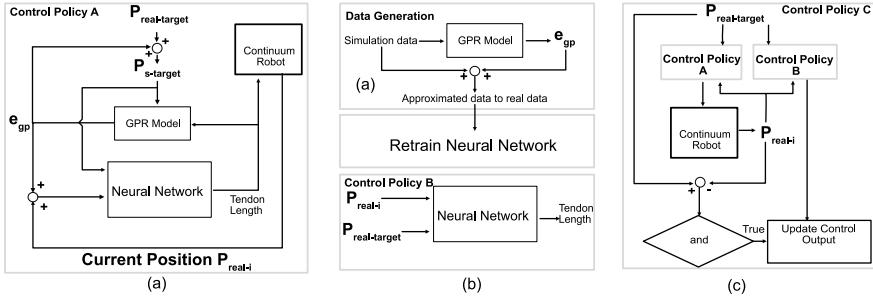


Fig. 7.22 **a** Control policy A. **b** Control policy B. **c** Control policy C

7.5.4.3 Control Policy B

The GPR error model could also be used to generate more data given simulated inputs as the GPR model could predict continuously mean and covariance values for different unseen inputs. In this way, we use Eq. (7.40) to generate corresponding error term e_{gp} . Assuming that each e_{gp} is accurate, we will obtain accurate real workspace positions by utilizing the following equation to approximate real position data for given control inputs:

$$P_{exp} = P_{sim} - e_{gp} \quad (7.44)$$

$$(P_{real-i}, P_{real-target}) \rightarrow u_i$$

With updated training data, we can now retrain the inverse mapping neural network. This method is more straightforward, the four tendon lengths are directly obtained from the retrained neural network given target position in the real workspace. However, control policy A has slightly better performance based on experiment data [72].

7.5.4.4 Control Policy C

This control policy incorporates elements of control policies A and B to create a hybrid model. The A control policy relies on a precise simulation model, while the GPR model compensates for the error. Typically, the simulation model is a good approximation of the actual robot, and when it is accurate, it performs well.

Due to the simulation model's simplifications and numerous assumptions, certain physics could not be accurately represented. This issue becomes more critical for our application as we incorporate softer materials TPU. Using the same data generated by the GPR model, we employ control policy B to learn the mechanics that are absent or severely distorted in the actual robot. Although the approximated data generated by the GPR model may be less precise than actual data, it will outperform control policy A when the simulation model is inaccurate.

Using the final position tracking error as a criterion and checking L_A and L_B to prevent an infinite cycle, we have presented the control policy C to connect policies A and B. Presented in Fig. 7.22, if the absolute position error is greater than a constant value e_c , we will switch to the control output generated by policy B presuming the simulation model is not accurate at this location. And e_c is selected based on policy A's average goal-reaching error: 0.015m. Currently, policy C relies on feedback to verify e_c and operates in a closed-loop configuration.

7.5.5 Conclusion

In this section, we have demonstrated the methods that incorporate neural networks and Gaussian process regression to obtain fast, accurate, and data-efficient control of a continuum robot with soft spine. With advancement of the neural network techniques and a combination of traditional machine learning techniques, controlling such complex and soft robots like malleable manipulators becomes fast and accurate without requiring extensive real-world data calibration.

Acknowledgements The work in Chap. 7 was supported in part by UK Research and Innovation (UKRI) under grant EP/N509486/1, project reference 2286512.

References

- Yim M (2007) Modular self-reconfigurable robot systems: challenges and opportunities for the future. *IEEE Robot Automat Mag* 10:2–11
- Seo J, Paik J, Yim M (2019) Modular reconfigurable robotics. *Annual Rev Control Robot Auton Syst* 2:63–88
- Valente A (2016) Reconfigurable industrial robots: a stochastic programming approach for designing and assembling robotic arms. *Robot Comput Integr Manuf* 41:115–126
- Kereluk JA, Emami MR (2017) Task-based optimization of reconfigurable robot manipulators. *Adv Robot* 31(16):836–850
- Brandstötter M, Gallina P, Seriani S, Hofbaur M (2018) Task-dependent structural modifications on reconfigurable general serial manipulators. In: International conference on robotics in Alpe-Adria Danube region, pp 316–324. Springer
- Guilin Yang and I-Ming Chen (2000) Task-based optimization of modular robot configurations: minimized degree-of-freedom approach. *Mechanism Mach Theory* 35(4):517–540
- Clark AB, Rojas N (2020) Design and workspace characterisation of malleable robots. In: 2020 IEEE international conference on robotics and automation (ICRA), pp 9021–9027. IEEE
- Majidi C (2014) Soft robotics: a perspective—current trends and prospects for the future. *Soft Robot* 1(1):5–11
- Aghili F, Parsa K (2009) A reconfigurable robot with lockable cylindrical joints. *IEEE Trans Robot* 25(4):785–797
- Kang R, Branson DT, Zheng T, Guglielmino E, Caldwell DG (2013) Design, modeling and control of a pneumatically actuated manipulator inspired by biological continuum structures. *Bioinspiration Biomimetics* 8(3):036008

11. Grissom MD, Chitrakaran V, Dienno D, Csencits M, Pritts M, Jones B, McMahan W, Dawson D, Rahn C, Walker I (2006) Design and experimental testing of the octarm soft robot manipulator. In: Unmanned systems technology VIII, vol 6230, page 62301F. International Society for Optics and Photonics
12. McMahan W, Jones BA, Walker ID (2005) Design and implementation of a multi-section continuum robot: air-octor. In: 2005 IEEE/RSJ international conference on intelligent robots and systems, pp 2578–2585. IEEE
13. Robertson MA, Sadeghi H, Florez JM, Paik J (2017) Soft pneumatic actuator fascicles for high force and reliability. *Soft Robot* 4(1):23–32
14. Burgner-Kahrs J, Rucker DC, Choset H (2015) Continuum robots for medical applications: a survey. *IEEE Trans Robot* 31(6):1261–1280
15. Jiang A, Xynogalas G, Dasgupta P, Althoefer K, Nanayakkara T (2012) Design of a variable stiffness flexible manipulator with composite granular jamming and membrane coupling. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, pp 2922–2927. IEEE
16. Santiago JLC, Godage IS, Gonthina P, Walker ID (2016) Soft robots and kangaroo tails: modulating compliance in continuum structures through mechanical layer jamming. *Soft Robot* 3(2):54–63
17. Kim Y-J, Cheng S, Kim S, Iagnemma K (2013) A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery. *IEEE Trans Robot* 29(4):1031–1042
18. Kim Y-J, Cheng S, Kim S, Iagnemma K (2012) Design of a tubular snake-like manipulator with stiffening capability by layer jamming. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, pp 4251–4256. IEEE
19. Blanc L, Delchambre A, Lambert P (2017) Flexible medical devices: review of controllable stiffness solutions. In: Actuators, vol 6, p 23. Multidisciplinary Digital Publishing Institute
20. McDowell TW, Wimer BM, Welcome DE, Warren C, Dong RG (2012) Effects of handle size and shape on measured grip strength. *Int J Indus Ergon* 42(2):199–205
21. Yoon H-S, Yi B-J (2009) A 4-dof flexible continuum robot using a spring backbone. In: 2009 international conference on mechatronics and automation, pp 1249–1254. IEEE
22. Mehling JS, Diftler MA, Chu M, Valvo M (2006) A minimally invasive tendril robot for in-space inspection. In: The First IEEE/RAS-EMBS international conference on biomedical robotics and Biomechatronics, 2006. BioRob 2006, pp 690–695. IEEE
23. Blessing M, Walker ID (2004) Novel continuum robots with variable-length sections. *IFAC Proc* 37(14):55–60
24. Kim Y-J, Cheng S, Kim S, Iagnemma K (2013) A stiffness-adjustable hyperredundant manipulator using a variable neutral-line mechanism for minimally invasive surgery. *IEEE Trans Robot* 30(2):382–395
25. Santiago JLC, Walker ID, Godage IS (2015) Continuum robots for space applications based on layer-jamming scales with stiffening capability. In: 2015 IEEE aerospace conference, pp 1–13. IEEE
26. Langer M, Amanov E, Burgner-Kahrs J (2018) Stiffening sheaths for continuum robots. *Soft Robot* 5(3):291–303
27. Clark AB, Rojas N (2019) Stiffness-tunable limb segment with flexible spine for malleable robots. In: 2019 international conference on robotics and automation (ICRA), pp 3969–3975. IEEE
28. Blumenthal LM (1953) Theory and applications of distance geometry
29. Rojas N (2012) Distance-based formulations for the position analysis of kinematic chains. PhD dissertation
30. Havel T (1991) The use of distances as coordinates in computer-aides proofs of theorems in euclidean geometry. *J Symbolic Comput* 11:579–593
31. Wang L, Nurzaman SG, Iida F (2017) Soft-material robotics. *Found Trends Robot* 5(3):191–259
32. Kung Y-S, Shu G-S (2005) Development of a fpga-based motion control ic for robot arm. In: 2005 IEEE international conference on industrial technology, pp 1397–1402. IEEE

33. Li Z, Du R, Lei MC, Yuan SM (2011) Design and analysis of a biomimetic wire-driven robot arm. In: ASME 2011 international mechanical engineering congress and exposition, pp 191–198. American Society of Mechanical Engineers Digital Collection
34. Coppola G, Zhang D, Liu K (2014) A 6-dof reconfigurable hybrid parallel manipulator. *Robot Comput Integr Manuf* 30(2):99–106
35. Xie F, Liu X-J (2015) Design and development of a high-speed and high-rotation robot with four identical arms and a single platform. *J Mech Robot* 7(4):041015
36. Rojas N, Dollar AM (2016) The coupler surface of the rsrs mechanism. *J Mech Robot* 8(1)
37. Rojas N, Dollar AM, Thomas F (2015) A unified position analysis of the dixon and the generalized peaucellier linkages. *Mechan Mach Theory* 94:28–40
38. Rojas N, Dollar AM (2017) Distance-based kinematics of the five-oblique-axis thumb model with intersecting axes at the metacarpophalangeal joint. In: 2017 international conference on rehabilitation robotics (ICORR), pp 1331–1336. IEEE
39. Rojas N, Thomas F (2018) Forward kinematics of the general triple-arm robot using a distance-based formulation. In: Computational kinematics, pp 257–264. Springer
40. Menger K (1931) New foundation of euclidean geometry. *American J Mathemat* 53(4):721–745
41. Thomas F, Ros L (2005) Revisiting trilateration for robot localization. *IEEE Trans Robot* 21(1):93–101
42. Tsai YC, Soni AH (1983) An algorithm for the workspace of a general nr robot. *J Mechan Transmiss Autom Design* 105(1):52–57
43. Gupta KC, Roth B (1982) Design considerations for manipulator workspace. *J Mechan Design* 104(4):704–711
44. Powell PD (2011) Calculating determinants of block matrices. arXiv preprint [arXiv:1112.4379](https://arxiv.org/abs/1112.4379)
45. Dautenhahn K (2007) Socially intelligent robots: dimensions of human-robot interaction. *Philos Trans Royal Soc B Biol Sci* 362(1480):679–704
46. Zhanat Makhataeva and Huseyin Atakan Varol (2020) Augmented reality for robotics: a review. *Robotics* 9(2):21
47. Ajoudani A, Zanchettin AM, Ivaldi S, Albu-Schäffer A, Kosuge K, Khatib O (2018) Progress and prospects of the human–robot collaboration. *Auton Robots* 42:957–975
48. Neges M, Koch C, König M, Abramovici M (2017) Combining visual natural markers and imu for improved ar based indoor navigation. *Adv Eng Inform* 31:18–31
49. Lee SG, Ma Y-S, Thimm GL, Verstraeten J (2008) Product lifecycle management in aviation maintenance, repair and overhaul. *Comput Indus* 59(2-3):296–303
50. Iqbal H, Tatti F, Rodriguez F, Baena Y (2021) Augmented reality in robotic assisted orthopaedic surgery: a pilot study. *J Biomed Inf* 120:103841
51. Magnanimo V, Saveriano M, Rossi S, Lee D (2014) A bayesian approach for task recognition and future human activity prediction. In: The 23rd IEEE international symposium on robot and human interactive communication, pp 726–731. IEEE
52. Bowyer SA, Davies BL, Rodriguez F, Baena Y (2013) Active constraints/virtual fixtures: a survey. *IEEE Trans Robot* 30(1):138–157
53. Sheridan TB (2016) Human–robot interaction: status and challenges. *Human Factors* 58(4):525–532
54. Solymar AE, Ibrahim KM, Atia MR, Saleh HI, Roman MR (2020) Perceptive augmented reality-based interface for robot task planning and visualization. *Int J Innov Comput Inf Control* 16(5):1769–1785
55. Doelare SAN, Smorenburg SPM, van Schaik TG, Blankenstein , Wisselink W, Nederhoed JH, Lely RJ, Hoksbergen AWJ, Yeung KK (2021) Image fusion during standard and complex endovascular aortic repair, to fuse or not to fuse? a meta-analysis and additional data from a single-center retrospective cohort. *J Endovascular Therapy* 28(1):78–92
56. Ma L, Zhao Z, Chen F, Zhang B, Ligong F, Liao H (2017) Augmented reality surgical navigation with ultrasound-assisted registration for pedicle screw placement: a pilot study. *Int J Comput Assisted Radiol Surgery* 12(12):2205–2215
57. Mine MR, Brooks Jr FP, Sequin CH (1997). Moving objects in space: exploiting proprioception in virtual-environment interaction. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp 19–26

58. Lipton JI, Fay AJ, Rus D (2017) Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robot Autom Lett* 3(1):179–186
59. Rosen E, Whitney D, Phillips E, Chien G, Tompkin J, Konidaris G, Tellex S (2019) Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *Int J Robot Res* 38(12–13):1513–1526
60. Ostanin M, Yagfarov R, Klimchik A (2019) Interactive robots control using mixed reality. *IFAC-PapersOnLine* 52(13):695–700
61. Frank JA, Moorhead M, Kapila V (2017) Mobile mixed-reality interfaces that enhance human–robot interaction in shared spaces. *Frontiers Robot AI* 4:20
62. Kersten-Oertel M, Gerard I, Drouin S, Mok K, Sirhan D, Sinclair DS, Collins DL (2015) Augmented reality in neurovascular surgery: feasibility and first uses in the operating room. *Int J Comput Assisted Radiol Surgery* 10(11):1823–1836
63. Hu X, Liu H, Rodriguez F, Baena Y (2021) Markerless navigation system for orthopaedic knee surgery: a proof of concept study. *IEEE Access* 9:64708–64718
64. Andrews CM, Henry AB, Soriano IM, Southworth MK, Silva JR (2020) Registration techniques for clinical applications of three-dimensional augmented reality devices. *IEEE J Trans Eng Health Med* 9:1–14
65. Clark AB, Rojas N (2021) Malleable robots: Reconfigurable robotic arms with continuum links of variable stiffness. *IEEE Trans Robot*. Submitted 11/06/21
66. Rao P, Peyron Q, Lilge S, Burgner-Kahrs J (2021) How to model tendon-driven continuum robots and benchmark modelling performance. *Frontiers Robot AI* 7:630245
67. Till J, Alois V, Rucker C (2019) Real-time dynamics of soft and continuum robots based on cosserat rod models. *Int J Robot Res* 38(6):723–746
68. Fang G, Wang X, Wang K, Lee K-H, Ho JDL, Fu H-C, Chung Fu DK, Kwok K-W (2019) Vision-based online learning kinematic control for soft robots using local gaussian process regression. *IEEE Robot Autom Lett* 4(2):1194–1201
69. Deisenroth M, Rasmussen CE (2011) Pilco: a model-based and data-efficient approach to policy search. In: Proceedings of the 28th International Conference on machine learning (ICML-11), pages 465–472. Citeseer
70. Sabelhaus AP, Majidi C (2021) Gaussian process dynamics models for soft robots with shape memory actuators. In: 2021 IEEE 4th International conference on soft robotics (RoboSoft), pp 191–198. IEEE
71. Li M, Kang R, Branson DT, Dai JS (2017) Model-free control for continuum robots based on an adaptive kalman filter. *IEEE/ASME Trans Mechatron* 23(1):286–297
72. Wang X, Rojas N (2022) A data-efficient model-based learning framework for the closed-loop control of continuum robots. In: 2022 IEEE 5th international conference on soft robotics (RoboSoft), pp 247–254. IEEE

Chapter 8

Continuum Soft Robots with Applications in the Construction Industry



Manu Srivastava and Ian D. Walker

Abstract In this chapter, we discuss some practical issues in deploying soft robots. Specifically, we describe the practical advantages and problems arising from the conversion of industrial concrete hoses into soft continuum hose robots, for 3D printing of cement in the construction industry. “Robotizing” the inherent flexibility and maneuverability of hoses offers the potential for the creation and high fidelity repair of more complex structures than currently feasible. However, the inherent softness present in the hose structures presents practical problems. Hose compression can reduce the accuracy of the concrete print. We discuss this phenomenon, along with practical measures taken to address the issue.

In this Chapter, we discuss the potential and challenges of applying soft robots to a specific application of significance in the construction industry. Specifically, we examine and detail issues in deploying a soft robot in the 3D printing of cement. The concept is to create a robotic version of the industrial hose typically used to deposit cement in construction sites. The goal is to combine the inherent compliance—i.e. the “softness”—of the hose with an enhanced, i.e. robotic, ability to orient the hose tip. Improved dexterity at the hose tip could enable the automated construction of more complex advanced cement-based structures than feasible with current techniques.

We describe the construction and operation of a robotic version of the cement hose, realized as a tendon driven continuum robot. Testing of the system identified challenges due to un-modeled axial compliance in the hose. We discuss the nature and underlying causes of these challenges, and further describe the approach taken to address them.

We begin with a background of 3D concrete printing in the construction industry, followed by a review of continuum robots and their inherent qualities, focusing on continuum robot hoses actuated by tendons. Potential applications for such robots, and in particular the applications in construction using such hoses for 3D printing

M. Srivastava
Clemson University, Clemson, USA
e-mail: manus@g.clemson.edu

I. D. Walker (✉)
The University of Wyoming, Laramie, USA
e-mail: iwalker2@uwyo.edu

of concrete which motivated this work, are discussed. These ideas are based on the notions of compliance in continuum robots, with potential applications in the construction industry.

8.1 Background: 3D Printing in the Construction Industry

Construction with 3D printed structures has the potential to be cheaper, quicker, and ecologically friendlier than the traditional process by reducing the need for manual labor in the forming of blocks and transporting them to the construction site. Another advantage is that it allows building freeform structures without erecting complex and expensive formwork. In addition, reducing manual labor reduces the potential for workplace injuries and can result in long term cost savings. It is estimated that the benefits offered by 3D printing can translate into cost-savings of 35–60% of the overall costs of concrete structures [1].

3D Concrete printing, which is also referred to as cement-based additive manufacturing, has been under development for the past 10 years and is a topic of active research. The world's largest 3D printed house was built in New York in 2019. ARCS took 48 h to build a 1900-square-foot home, for \$6000 in material costs. Multiple labor-intensive processes were automated [2].

8.1.1 Types of 3D Concrete Printing Systems

Current 3-D printers can be classified as shown in Fig. 8.1 and research in each of the areas is summarized below.

8.1.1.1 Gantry Based Contour Crafting

Gantry based techniques involve an end-effector mounted on an external frame. In these systems, concrete is extruded through the nozzle and deposited in layers along a certain pattern. Two trowels are attached to this nozzle which support the material deposition by improving the form and smoothness of the final part.

8.1.1.2 Gantry Based D-Shape Printing

D-shape printing shares similarities with inkjet printing, as both employ powdered material to create the desired object. In this method, a mixture of dry sand and silica is chosen as the base material and evenly distributed on a flat surface. The binder, a liquid substance, is then precisely applied to the specific areas where the final component is needed. The binder liquid seeps into the mixture, solidifying the layer in the desired shape while leaving the surrounding sand unaffected. This procedure

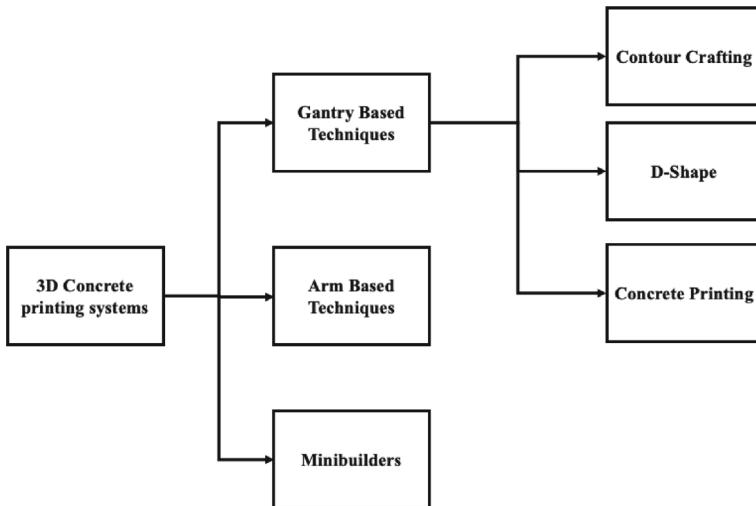


Fig. 8.1 Classification of 3DCP techniques [1]

is repeated until the entire object is formed, and typically, it takes around a day for the entire part to fully harden [1].

8.1.1.3 Gantry Based Concrete Printing

Gantry based concrete printing is very similar to the contour crafting but without the trowels. It can produce more complex geometries due to higher deposition resolution but is slower.

8.1.1.4 Arm Based Techniques

Arm based techniques are newer and are most related to our research, particularly techniques involving arms mounted on mobile bases. Rigid-link robot arms with a hose for concrete flow along their body terminated at their end effector, enable tangential continuity of the print nozzle allowing smoother transition of print layers and therefore giving a more aesthetic look. Commercial systems can cost in the \$200000 range however, and their workspace needs to be open to avoid collisions with the robot structure.

Arms mounted on a fixed base have a hemispherical workspace limited by the reach of the arm and the need for an egress location. These limitations led to designs with a mobile base and the “print-on-arrival” paradigm. Here, the base moves to a new location that is conducive to the deposition of concrete thereby increasing the possible print area. Current advances in collaborative robotics have enabled division of the

printing work between a team of robots improving scalability and time efficiency. Two robots have been used to print a single concrete structure in tandem [3]. The concrete was laid in overlapping layers at the interfaces between printed sections to ensure adequate bond strength.

In general, print-on-arrival structures are constrained by the necessity of an egress location. These problems are addressed by the “print-while-moving” paradigm which enables printing single-piece structures of arbitrary sizes with a single robot. A simple motion planning algorithm that resolved redundancy in 2 ground based DoFs allowing for simultaneous motion of the base and the arm relative to the base was developed in [4]. The algorithm prioritized base movement and relied on differential inverse kinematics to find the trajectories of the arm and it was possible for the algorithm to not return a solution for the arm’s joints trajectory. The physical implementation on the robot required localization/pose estimation and model predictive control (MPC) of the base and could achieve concrete deposition accuracy of the order of 1cm at a nozzle speed of 10cm/sec. The workspace of the robot was constrained by the maximum allowable hose length as determined by the pump pressure, hose motion, and marker visibility. Also, ground evenness played a key role in determining base mobility.

Similar robots have been used collaboratively to print while moving [5]. However, their implementation only had the robotic arm moving in the z direction with respect to the base. The focus of their work was on-board multi robot aware planning and task allocation tasks to perform concrete deposition tasks. They used an auction-based system called MURDOCH designed to allocate and coordinate manipulation tasks in a fault-tolerant manner among the robots. Software was built using Robot Operating System (ROS) and MoveIt! motion planning framework.

8.1.1.5 Minibuilders

Another notable printing approach for 3D concrete printing is Minibuilders, which is composed of three small mobile robots. The first robot prepares a short concrete wall by following pre-installed strips. The second robot is placed on the top of the wall and continues building it while moving through the contour of the wall. The last robot uses suction cups and pressurized air to build vertically and reinforce the printed structure which previously had only horizontal layers.

8.1.1.6 Cable-Driven Parallel Robots (CDPR)

An emerging, and so far, noncommercial, category of wide workspace three dimensional concrete printing robots, CDPR, is easier to deploy on site than a gantry system of similar scale. These robots use at least 6 cables which connect a platform to a frame, reeled in and out by winches as actuation members, the other elements being easily stacked for easy relocation and reconfiguration, which is critical for

Fig. 8.2 A continuum robot grasping and manipulating a fragile (glass) object, using its compliant backbone to adapt to the object's shape [10]



on-site construction. It also removes the height constraint plaguing arm-based systems. The cable length is adjusted synchronously to provide control and stability in all 6DoF in a large workspace. In [6], the authors developed a robotic system with a workspace of $13.6 \times 9.4 \times 3.3 \text{ m}^3$ that can hold a load of up to 500kg accurately enough within 1cm tolerance that it could be used for concrete deposition without external sensing.

Since large curvatures introduced vibrations, a G-Code post processor module was designed using Rhinoceros3D and Grasshopper3D that modified regular CNC G-Code to ensure the trajectory curvatures were lesser than an empirically determined value at a given printing speed. The wrench closure workspace of the tip of the extruder rises near the drawing points.

This robot exhibited severe accumulation of material at the start and stop points of the layers even though a stop time is not programmed. Another problem was that the quality of the print reduced with increasing height as demonstrated by uneven thickness and width of the material layer due to position errors resulting from the design of the controller, operating in joint space with dual-space feedforward control, combined with low reduction ratio on the motors. The extruder tip position errors were reconstructed from the joint errors measured at motor encoders using the small displacement Jacobian matrix. These errors also resulted in the extruder tip colliding with the printed layer occasionally which in turn provided a reactionary force to the CDPR increasing the actual end effector height for subsequent printing.

However, no efforts to date have considered making the concrete hose itself an active robotic element. A robotic hose potentially obviates the need for a heavy robotic arm and can be added to either gantry systems or CDPRs to provide the non-vertical 3D printing capability of a robotic arm used when a complex 3D contour needs to be described. This is especially beneficial in space and weight constrained construction applications.

Since the robotic hose would be a continuum robot, we briefly review the history, types, and applications of continuum robots in the next section.

8.2 A Brief Review of Continuum Robots

Rigid link robot arms do well in structured environments, e.g. a factory floor. Exploiting the well understood kinematics of serially connected rigid links, they can place their end effector (EE) with very high accuracy and precision. This gives them an unmatched capability to perform prechoreographed repetitive tasks [7].

However, they are ill suited to operate outside the confines of structured environments or engineered task spaces. First, rigid link robots can present a mortal danger to humans when they are within the robot's task space. Second, they only perform tasks using their EE with the rest of their body structure used to position the EE for the task to be performed. Their links, being rigid, can not physically adapt to unanticipated or unplanned contact with the environment.

A different paradigm exists—Continuum robots are a type of robot manipulator that has a continuous form or backbone, rather than being formed from rigid links [8, 9]. This allows them to bend, extend, contract, and sometimes twist at any point along their structure, providing them with capabilities beyond those of traditional rigid-link manipulators.

In Fig. 8.2, a continuum robot [10] gently adapts its shape to wrap around a fragile object (a glass lampshade) without causing breakage or damage. This type of operation would be difficult for a conventional rigid-link manipulator to perform, as it would require active force sensing to present a sufficiently compliant interface and a priori knowledge of the task along with specialized tooling since a parallel jaw gripper cannot conform to the shape of an arbitrary object. Continuum robots can also use their compliant, actively controlled continuous structure to easily navigate tight spaces and negotiate complex environmental geometries [11]. The field of continuum robots focuses on realizing the potential of this novel design in useful hardware.

8.2.1 *History and Design Types*

Interest in continuous backbone robots dates back to the 1960s, with the first prototype, the Tensor Arm, being developed for underwater applications [12]. However, it proved difficult to control due to the complex relationship between inputs and shapes and was never deployed. There was a lull in research on continuum robots until the late 1980s and early 1990s, when Chirikjian published the first works on the kinematics and dynamics of these robots [13], and Hirose published innovative results on the evolution of their shapes [14]. In the 1990s, research on hardware for these robots focused on the extension of original designs and the development of pneumatically actuated backbones. Robinson and Davies, while working on pneumatically actuated backbones, coined the term “continuum robot” in 1999 [15]. In parallel with these hardware developments, the 1990s saw progress in modeling continuum robot kinematics, with the development of “bottom-up” approaches that were directly matched

to hardware constraints and enabled real-time computer control of continuum robot shapes [11].

A key characteristic of a continuum robot is a continuously curving, flexible backbone that can be controlled to change its shape [16]. The backbone is also compliant, meaning that it can smoothly yield to external loads. (This feature causes continuum robots to fall into the more general category of soft robots.) These features allow a continuum robot to adapt its shape to navigate complex environments and to conform to a variety of objects, making it more versatile than traditional rigid-link robots.

Designing structures without rigid elements is a new concept for many robot designers, but there are several fundamental principles that can be learned by studying biological structures like tongues, trunks, and tentacles. In particular, “muscular hydrostats,” [17] which include octopus arms [18], elephant trunks [19], squid tentacles [20], and mammalian tongues, have provided valuable insight for continuum robot designers [10]. These structures are made up almost entirely of their own actuators (muscles), with some additional fluid and connective tissue, and can bend, twist, and extend. Some initial work attempted to mimic this concept in continuum robots using artificial muscle technologies, but these technologies have not yet been able to combine the necessary bending and force generation for practical continuum robots. If artificial muscle technologies improve in the future, it could greatly impact the design and operation of these robots, although other challenges such as packaging, power consumption, and wiring would still need to be addressed.

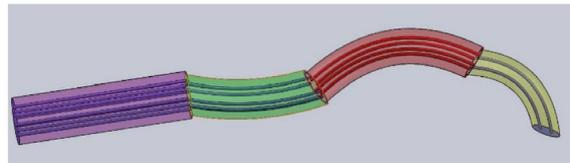
In the absence of technologies that can easily mimic biological inspirations, designers have followed alternative paths to achieve active bending and extension using continuous backbone structures that possess helpful internal energy properties. Three fundamental design strategies have emerged, and these are summarized in the following subsections.

8.2.1.1 Concentric Tube Designs

A type of extrinsically actuated (with actuators all outside, or extrinsic, to the robot) continuum robot that has emerged is based on a backbone made up of concentric compliant tubes that can rotate and translate relative to each other and be actuated at the base of the robot [21–25]. The structure can extend and contract by the longitudinal sliding of the tubes and achieve local rotation by rotational sliding of the tubes. While this design directly provides both extension and torsion, it does not inherently enable backbone bending. One approach to add bending is to use precurved compliant tubes [26] (relative rotation between two precurved tubes allowing change in overall curvature, shown in Fig. 8.3), while another is to use tendons to bend the tubes, although this increases the complexity of the design.

Concentric tube continuum robots have a thin, clean design and offer direct correspondence between actuator values and backbone shape variables, but they require an external actuator package and do not inherently support actively controlled bending. They are especially suitable for minimally invasive medical procedures due to their

Fig. 8.3 Concentric tube continuum robot concept [11]



small size [27] and high compliance and have been used for tasks such as navigation through the human lung [28] and in neurosurgery [29].

8.2.1.2 Intrinsic Actuated Designs

Locally actuated (intrinsic) continuum robots are characterized by having their actuators directly incorporated into the backbone of the robot. This design is similar to the structure of biological continuum structures, leading to their classification as “intrinsically actuated”. These robots are typically made using pneumatic “McKibben” muscles [30–32] although some versions using other actuators, e.g. shape memory alloys, have also been created [33]. The backbone is made up of independently actuated sections, each of which is typically composed of three independently actuated muscles. The muscles can either be “extenders” (which increase in length with increased pressure) or “contractors” (which decrease in length with increased pressure). When pressure changes are evenly applied to all three actuators of a straight section, the section will either extend or contract. Applying different pressures to the actuators will cause the section to bend into a curve with roughly constant curvature, with the plane of the curve determined by the pressures applied. The shape of the backbone is therefore made up of a series of approximately constant curvature segments connected in series, with the end tangents of each segment coinciding [34].

The specific actuators for intrinsically actuated continuum robots can come from any available type of artificial muscle. Pneumatic [35–37] or hydraulic actuators as artificial muscles have been used in projects like the “Octarm” (Fig. 8.4, [32]) and “European Octopus” [38]. Locally actuated continuum robots that use shape memory alloys (SMA) [39] as their actuators have been used in medical procedures such as steering active cannulae and in systems designed for single port access surgery. SMA based robots have also been developed for endoscopic stitching in surgical obesity treatment and for colonoscopic insertion [40].

Recently, “growing” robots [41–44] have been developed using inflatable structures that have been pre-designed to achieve the task space objectives. The key aspect of most such growing robots is the principle of eversion, in which the body is initially rolled up and then unfurled, typically using pneumatic internal pressure.

Unlike extrinsic actuation, locally actuated continuum robots have the ability to provide extension, bending, and torsion capabilities in the backbone structure. However, they have relatively low force generation capabilities and typically require

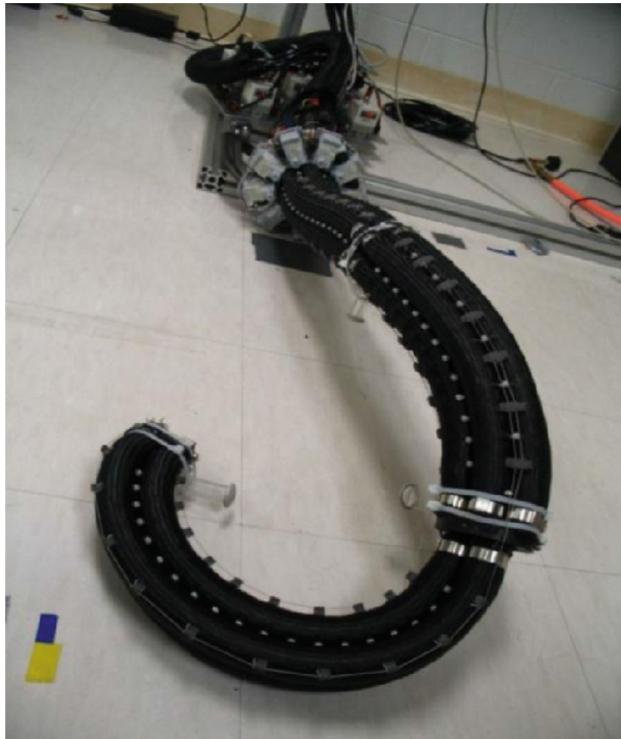


Fig. 8.4 Intrinsically actuated three-section, nine-degree-of-freedom pneumatic “Octarm” continuum manipulator [10]

external pressure regulation equipment and a compressor. Additionally, they tend to have complex tube routing and valving.

8.2.1.3 Tendon-Based Designs

The most widely used approach to bending a continuous robotic structure is to use remotely actuated tendons [45–47]. The tendons are routed along the backbone and terminated at specific points, and the forces (tensions) applied to them at the base produce torques at the termination points, causing the backbone to bend [48].

The Tensor Arm [12] used this approach, routing tendons through rigid spacer elements called vertebrae, to produce bending in several sections along the backbone. A spring-based backbone, as used in the Tendril robot by NASA Johnson Space Center, [49] provides natural compliance but proved difficult to control due to compression of the spring. The design is relatively simple, but it can be challenging to control due to the loss of control effort in compression. To address this issue, some designs use

an incompressible rod as the backbone element, which provides more predictable behavior but does not allow for extension.

In recent times, active stiffness control of continuum robot backbones has been explored [50, 51]. There are several approaches that have been proposed to increase the stiffness of a backbone in a robot. One of these approaches is called material jamming, which involves using a membrane filled with granular media or thin sheets [52]. By applying a vacuum, the membrane collapses on the filler, increasing its density and stiffness. This method is simple and fairly reliable, but it may be difficult to scale up without significantly increasing the size of the robot. Another technology is thermal activation, which involves using materials with low melting points that can be rapidly heated to change phase and modulate stiffness [53]. This method is time consuming and less reliable than material jamming and may not provide as much stiffness, but it does not require as much of an increase in the size of the robot.

Tendon-based continuum designs have a finite number of sections defined by the tendon termination points along the backbone, and they can generate relatively high forces [45]. There is, however, a coupling between the tendons which needs to be resolved in order to actuate a multisection tendon driven continuum robot (TDCR). These designs require a method to prevent slack and backlash in the tendons, and they typically use a bulky actuator package at the base of the robot. The location of the actuator package outside the backbone categorizes these designs as extrinsically actuated continuum robots. Tendon-actuated continuum robots have been developed for space operations and medical procedures [54, 55] including sinus surgery [56], ACL surgery [57], laryngeal surgery [58, 59], and underwater use [60].

8.2.2 *Emerging Applications of Continuum Robots*

The ability of continuum structures, especially those that are long and thin, to navigate and move through congested environments makes them well-suited for use in inspection applications, such as those requiring active borescopes and endoscopes [61]. The many applications of continuum robots in minimally invasive surgery have been noted and more are emerging [29]. This section lists other emerging applications for them.

One example is the use of slender continuum robots for the in-situ repair of aeroengines. In an industrial demonstration, a 1.2 m long continuum robot with cameras and exchangeable grinding tools at the tip was able to access and repair a low-pressure compressor in an aeroengine from the front [62]. A 5 DOF continuum robot has also been demonstrated for teleoperated aerofoil repair, enabling a fast response to fleet repair needs around the world with the help of a specialist operating remotely. Multiple continuum arms, each handling a 3000F flame, have been used for collaborative operations in aeroengines, such as the thermal barrier coating repair of a commercial aeroengine combustor through narrow inspection holes. Finally, a 5 m long continuum robot has been demonstrated for long-reach inspection and maintenance in aeroengines [63].

Thin continuum robots have the potential to be used for in-space inspection tasks, such as inspecting equipment racks in the International Space Station. Testing for this in a full-scale mock-up of the Space Station at NASA has been performed [64]. In addition to inspection, continuum robots could potentially be used as “active hooks” or long, thin “tendril-like” robots to explore tight obstacle fields, such as crevasses, lava tubes, or skylights [65]. These capabilities could be useful for supporting exploration on planetary bodies and addressing key scientific questions in these environments.

In locomotion, multicontinuum-limbed robots have been demonstrated to walk and trot [66]. Prototypes for underwater [67] and aerial [68] use have been developed.

8.2.3 *A Review of Continuum Hose Robots*

A natural further application of continuum robots is as “active hoses” [69], which involve using the interior of the backbone of the robot to pump fluids through it. This concept, the focus of this work, has gained increasing interest in recent years as various groups have explored its potential applications. In the case of the construction-focused application discussed in this chapter, the material being pumped is concrete. However, other fluidic materials have been pumped using continuum robots, as briefly reviewed below.

The first continuum robot used as an active hose appears to be the EMMA robot. This large-scale system [70], suspended from a parallel cable structure, was designed to be deployed in underground radioactive waste storage tanks, and used to sluice out waste by pumping high pressure water through its continuum hose. Although the remediation program for which EMMA was developed was canceled, the demonstrations illustrated the potential of the approach. Figure 8.5 is a montage of later efforts. Research aimed at shore to ship refueling, pumping fuel through a continuum hose robot at the end of a rigid-link robot, has been reported [71]. Other efforts have discussed and demonstrated the pumping of air through continuum robots: in a concept demonstration of removing dust from solar panels [72], and using negative pressure, to negatively pump, i.e. suck, air to power artificial suckers at the tip and on the body of continuum robots [32]. Another approach to robotic hoses used reaction forces from water jets to build a fire-fighting hose [73]. Note however that in most of the above-mentioned pumping applications, high accuracy has not been a key requirement. Such is not the case for pumping of cementitious materials in general.

8.3 Applications of Continuum Robots to 3D Cement Printing

We propose a novel alternative in which we adapt the concrete hose , available at almost every construction site, to become a continuum robot [74, 75], as shown in Fig. 8.6. This is especially beneficial in space and weight constrained construction applications, e.g., at hard-to-reach places.

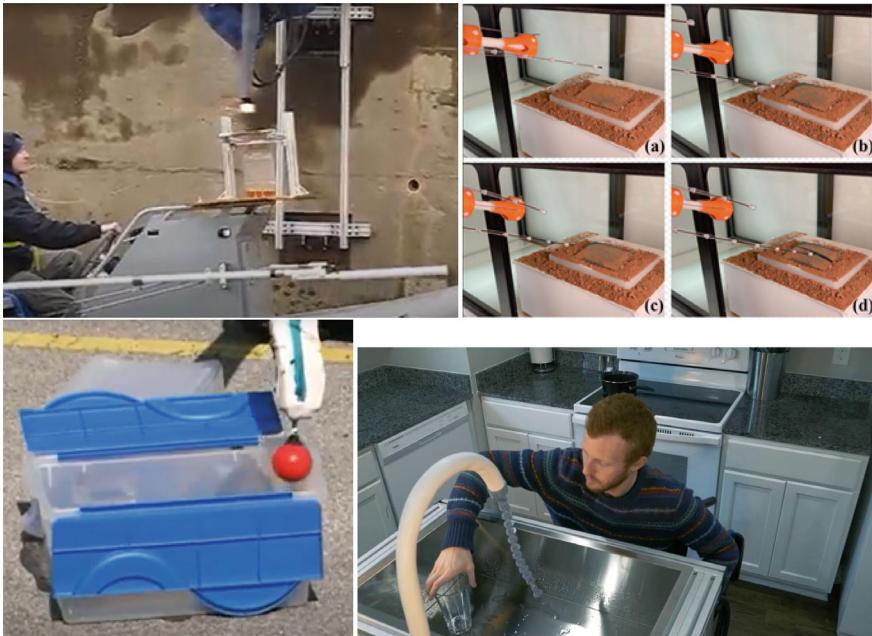


Fig. 8.5 Examples of continuum hoses (clockwise) 1. Refueling a boat [71] 2. Cleaning the dust on top of solar panels [72], 3. Continuum sink faucet, 4. Vacuum gripper [32]

A robotic hose could provide existing gantry based systems, i.e. cable driven parallel robots [76], and aerial platforms the ability to deposit concrete at a range of angles including but not limited to vertically downwards. Note that the structural integrity of special 3D printed curved structures such as domes is improved when the orientation of the deposition nozzle is variable which enables bonding of layers with many layers underneath as opposed to the single layer bond created when printing vertically downwards. This, however, requires a new physical printing capability as well as a new slicing algorithm that can exploit variable tip orientation.

Another potential use-case is bridge/structural repair which involves depositing concrete in small damaged areas previously inaccessible using traditional methods and therefore requiring demolition and rebuilding of the entire structure. The snake-like movement capability of the hose enables navigation through small crevices to deliver the concrete to the affected area.

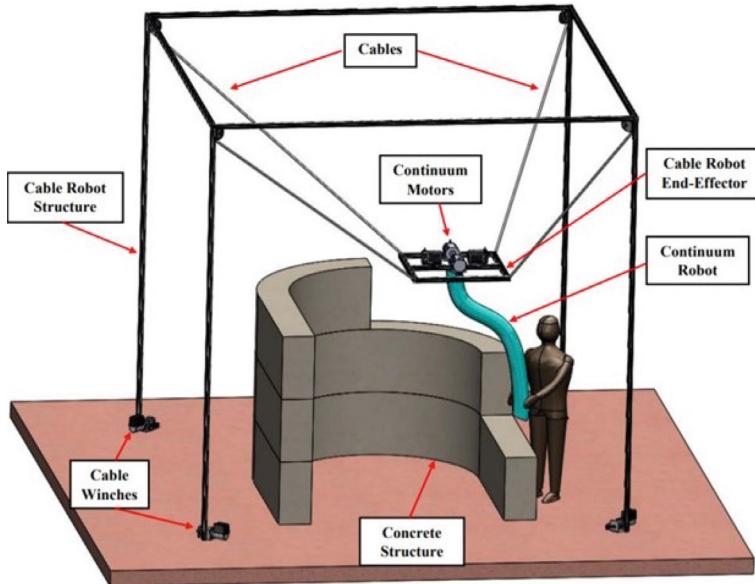


Fig. 8.6 Continuum hose (green element) in the construction environment [74]

8.3.1 Challenges in Large-Scale, High Payload Applications of Continuum Robots

3D printing with a continuum robotic hose requires greater accuracy than most large-scale continuum robot applications so far, presenting challenges. The challenges can be divided into the following categories: design, modeling, and sensing (and control), each of which is briefly discussed below.

Design challenges, common to all types of continuum robots, involve evaluating the Degrees of Freedom (DoFs) of the end effector (EE). For a non-extending continuum section with uniform and symmetrical tendon routing to avoid torsion, the EE has 2 DoFs, best visualized in the shape space as the plane of bending and amount of bending. Each additional section (sections are normally connected serially, i.e. end to end) provides two additional shape space DoFs at the cost of actuation complexity and weight. Notably, the additional shape space degrees do not map to additional task space positional DOFs directly—bending explicitly changes orientation, but position implicitly.

The design goals for the 3D concrete printing application could be achieved by a two section continuum robot augmented with a platform capable of translation in the vertical direction. To increase the workspace of the combined platform and continuum robot system, we additionally (via mounting the hose to an (x,y,z) translation base) provided our platform with two additional positional DOFs enabling it to translate in the horizontal plane. The overall system thus possessed 7 DoF (3 translational via

the base, and 4 in the two hose sections). This given an extra degree of freedom for traditional 6 DoF EE tasks (3 DoF for tip position, and 3 DoF for tip orientation) and poses the interesting possibility of redundancy resolution between the two systems, although that issue is not addressed further here.

Modeling challenges of large scale continuum robots involve consideration of gravity which precludes the use of traditional kinematics, a geometric modeling approach that does not consider forces. Each successive section loads the tip of the section preceding it and this loading changes with motion of the distal sections. Material properties such as backbone and tendon elasticity (and hysteresis), which are ignored in kinematic models, significantly affect the robot's pose and their effect is compounded due to the robot's length. Additionally, these elastic effects are inherently anisotropic when a finite number of equally spaced tendons are used to actuate the section due to the radial variation in the location of the resultant force and result in different EE speeds and curvature in different bending planes, as we shall describe in detail shortly.

Sensing presents a third set of challenges. Tension sensing for continuum robots using motor current based approaches is inapplicable when there is significant gear reduction on motor outputs due to friction losses. Therefore, additional sensors that often require redesigning of the actuator package are required for tension sensing. Additionally, control algorithms based on tension sensing are quite complex and often need high computational capabilities for use in real time. In the construction application in particular, Simultaneous Localization And Mapping (SLAM) of the robot is a hard problem using external sensors such as cameras and LIDAR due to the changing construction environment. Dust and occlusion add to this difficulty. IMUs, which can be attached to the robot body, are susceptible to changing magnetic interference in such environments.

Modeling and compensating for dynamics in continuum robots is a topic of much current research [8]. However, in the 3D concrete printing application discussed in this Chapter, while we expected to find significant dynamical effects involved with pumping of concrete through the actuated hose, we found that the hose hardware we developed did not exhibit vibrations or noticeable changes to its shape when cement was being pumped through it. We attribute that to the steady and slow flow speed of cement in 3D printing applications. This allowed us to neglect dynamic effects in our models and experiments. However, even without the presence of significant dynamic effects, modeling and operation of the developed hose robot presented significant new challenges requiring novel solutions, the details of which are described and discussed in the following. First, we provide an overview of the underlying issues faced—issues we faced in creating the hose robot we used to print concrete- in designing tendon-driven continuum robots.

8.3.2 Design Considerations for Tendon-Driven Continuum Robots

The fundamental functionality of a tendon-driven continuum robot section arises from the bending of its backbone due to torques applied at its tip via forces applied at an offset from that tip, those forces supplied by tensions in tendons terminated at that offset from the tip. As such, the core elements of the design space for such robots are the backbone stiffness, section length, and tendon offset (distance from the backbone curve of the tendon).

Of these elements, the one that most affects the behavior of the robot is the backbone stiffness. The section needs to bend, but also possess sufficient bending stiffness to provide resistance to the effects of the tendons. Most—though not all [30]—tendon-actuated continuum backbones have been selected to be initially straight, i.e. possess zero curvature in the unactuated state, and also have uniform bending stiffness in all directions radial to the backbone. In this way, the section is “trying to bend back to straight”, with consistent behavior in all bending planes.

Numerous approaches to realizing the underlying backbone stiffness have been demonstrated. A typical approach is to utilize a central beam (for planar sections) [77] or rod/tube [78]/[64] (for sections bending in 2D) of a homogeneous flexible material (for example, thin steel plates [77] or plastic rods/carbon fiber tubes [78]/[64], respectively). In these cases the backbone is periodically augmented with external “spacers”, with holes through which the tendons are routed.

An alternative approach is to arrange multiple thin flexible rods lengthwise in parallel, connecting them periodically by passing them through holes in the outer edges of thin spacers oriented parallel to the rods [63]. The backbone stiffness is then provided by the composite stiffnesses of the rods. This approach has the advantage of leaving the central core open, allowing, for example, the passing of power and signal cables for sensors mounted at the section’s tip. When moving freely in the spacers but fixed in the most distal one, the rods can be extended and contracted to effectively serve as tendons!

Other researchers have used springs, combining the stiffnesses of the springs to achieving the backbone stiffness. For example, one approach was to fix springs along and around the exterior of a zero-stiffness backbone structure (a set of u-joints connected in series) so that the unactuated system had zero curvature, but the springs combined to provide a predetermined stiffness in any radial direction about the backbone [19]. Another approach has been to connect a combination of extension and compression springs in series, and simply use that structure as the backbone [49]. A further approach is to use tendons to bend pressurized tubes [36].

In whatever way the backbone and its inherent stiffness is constructed, a further design choice is the offset of the tendons from the backbone center. This distance, at the tip or where the tendon is terminated, specifies the moment arm which transfers the force applied by the tension in the tendon to the moment at the section tip which bends the backbone. Physically, this quantity is realized by the radius of the spacers (or, more precisely, the radial distance from the backbone center to the hole in the

spacer through which the tendon passes). While in concept it might seem desirable to make this distance as large as possible, in most cases there are practical constraints on how wide the spacers can be, since many applications of continuum robots require a thin overall profile.

In almost all cases, the spacers have been identical in size, i.e. creating a constant offset of the tendons along the backbone. In some recent research however [79], it has been shown that tapering the distance of the tendon from the backbone, i.e. making the spacers smaller as they approach to tip, can improve the performance of the section.

The length of the section (functionally, the distance from its base to point of termination of the tendons) is usually selected to match the specified or anticipated application tasks of the robot [63]. For example, in the design of tendon-actuated continuum robots targeted at inspection operations in Space and on the International Space Station, long (greater than one meter) and thin (external spacer radius less than one centimeter) “tendril” robots were developed [64]. For medical applications [29], sections appropriate to the average size of human anatomy to be traversed are used.

A secondary design issue is how many sections are needed. Each section essentially provides a “c-bend in space”. So, if a given application requires the robot to bend in an s-shape, two sections will be required. For a “double s-bend”, four sections would be required, and so on. In a multi-section tendon-actuated continuum robot, the tendons for distal sections need to be routed through the spacers of the proximal sections. This constraint precludes in practise the creation of such robots with a great many (say, 5 or greater) sections. However, interestingly from the perspective of matching sections to tasks, a study of elephants and how they use their trunks indicated that the animals almost always select the shape of their trunks to form three sections [80]. Perhaps this is a biological “existence proof” that three sections is in some sense optimal. Most continuum robots to date have featured a maximum of three sections.

Coupled with the above design choices is the selection of actuators. Clearly the actuators must have sufficient power to overcome the bending stiffness of the backbone and generate the desired amount of bending. (Typically, the capability of a section is evaluated by the maximum achievable angle between tangents to the backbone at base and tip—90° for it to “look around a corner”, 180° for it to “look back on its base, etc.). However, actuators which overpower the system tend to be large and massive with respect to the—typically slender and light—robot, so actuators are usually selected to have “just enough” power for the desired bending.

Another key—and often neglected in the literature—design decision concerns tendon transmission/cable routing. Performance of the robot will be directly and strongly affected by how effectively the tendons are routed between actuators and the section tip. Typically the actuators are electric motors inherently providing rotational motion, so tendons are routed across one or more capstans to convert this rotational motion into the linear motion required to bend the section.

A key advantage of tendon-based actuation is the potential of generating high forces, but tendon-based systems are often plagued by intermittent slack in the tendons. This slack, caused by, e.g., local tendon separation from capstans during a change in motion direction, results in temporary loss of tension in the tendon. This will in turn cause a “dead zone” when the robot, although intended to be in motion, remains still (i.e. unactuated) until the slack is taken up, and consequently moves at a later time than anticipated. This phenomenon, which is notoriously hard to model and predict, causes havoc with control systems and can significantly reduce the accuracy and response time of the robots.

Consequently, significant effort is often devoted to arranging the actuator package at the base of the robot in a way that minimizes the number and angular magnitude of turns the tendons make between the actuator and section tip. Some systems incorporate mechanical approaches to handling slack, e.g. spring-loaded capstans to attempt to maintain positive tension on the tendons [19].

Secondary issues in realization of tendon-actuated continuum robots include the selection of tendon material. Depending on the tension required to bend the backbone, materials ranging from fishing line to steel cable have been used. Friction between the tendons and the spacers can sometimes be an issue [46], particularly in applications which require high bending radii, and this has resulted in the use of low-friction materials such as Teflon-coated nylon and Nitinol wires.

8.3.3 Compliant Continuum Robot Concrete Hose Hardware

For our concrete printing application, we designed and constructed a two section, six tendon driven continuum hose prototype with, necessarily, an steel mesh reinforced 1” ID industrial shotcrete hose as its backbone (Fig. 8.7). This specific hose was selected due to its versatility in the construction industry. Instead of the originally envisioned CDPR shown in Fig. 8.6, we ultimately used an xyz table that provided the external platform.

We chose to use two sections because that is the least number of sections required to control the end effector’s orientation while the first section is bent, and under the assumption that the external platform is controlling the tip position. Adding more sections adds complexity in controlling the hose because there is kinematic and mechanical coupling between sections while not providing additional capabilities to a hose with a 3D mobile base. A limited few types of concrete structures could, however, be built with the continuum robot mounted to a platform restricted to vertical axis control, e.g. a cylinder and a wall.

The driving assembly of the hose is attached to the external base and acts upon the hose from above. Each section of the hose has length 0.7m and is actuated by three steel tendons separated by 120°, and the tendons are attached to the hose by five evenly spaced collars per section. The 3D printed collars are held on to the hose via hose clamps, and friction is reduced by routing the cables through offset bearings



Fig. 8.7 Industrial cement hose (black/green) modified to continuum robot section. Note external tendons (not shown in this image) pass through specially designed spacers (silver) [74]

in the spacers (Fig. 8.8). The tendons for the distal section (containing the nozzle) pass through the proximal section's collars at an offset of 60° from its tendons.

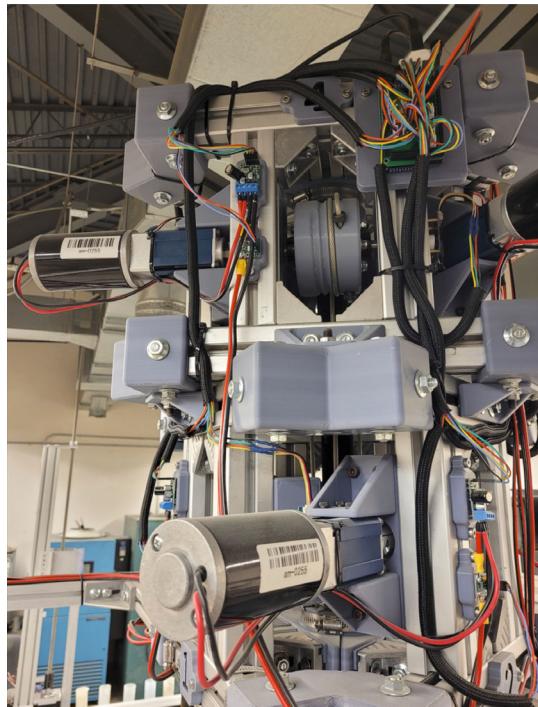
Due to the weight of the hose imposing a high-power requirement, each tendon is actuated using a separate 110V AC-12V DC 40A power supply connected to a 25A MOSFET H-bridge motor driver, DC motor (5300RPM, 133A stall current), and 100:1 drive reduction gearbox attached to a 3-D printed capstan on which the tendon is wound (Fig. 8.9). An Arduino microcontroller is used to implement position-control of tendon lengths calculated by a linear transform of the shaft angle sensed by absolute encoders.

Given our application of 3d printing concrete, and the associated needs of pumping, the cementitious mixture's rheological and stiffening properties [81] informed



Fig. 8.8 Detail of spacer design and tendon routing for continuum cement hose. Tendons pass through bearing assembly within collars (a), which are fixed to the cement hose backbone with hose clamps (b). Each section is actuated by three steel tendons, equally spaced apart around the circumference of the hose [74]

Fig. 8.9 Drive assembly. Motors (two shown, at left and bottom of image), connected through gearboxes, rotate capstans around which the actuating tendons are wound. One capstan is shown center top, on which a tendon and its termination point on the capstan can be seen [74]



the robot's design. Initial tests confirmed the robot's ability to bend to the desired configurations and at the speeds desired for the proposed application.

However, in designing and realizing the hardware of this robot, we made the assumption that the backbone hose would be incompressible under the loading required and applied to bend it. This assumption proved incorrect, resulting in issues critically compromising the accuracy of the system in operation. These issues are illustrated in the Chapter via a specific cement printing experiment with the robot.

8.4 Axial Hose Compression

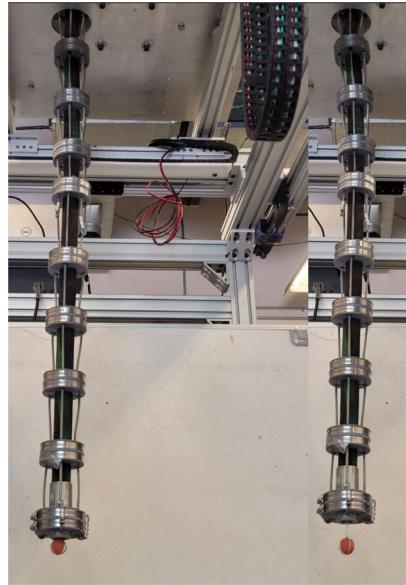
We illustrate these problems, their causes, and solutions to the problems, via the example of printing a circular/cylindrical concrete structure. This required that the continuum section maintain uniform curvature while rotating through the full 360° range of bending planes. We noticed that the EE slowed down in the tendon planes and dropped in curvature in between tendon planes. This resulted in a triangular ‘Adobe’ logo type of printed structure shown in Fig. 8.10. This is a major problem when accurate and repeatable control of the EE is desired.

In these experiments, we observed that the length of the robot hose, assumed up to this point to be constant, appeared to vary. Since the tendons in our robot were made of solid steel and the backbone was a steel mesh reinforced rubber hose, we inferred that this was caused by elasticity in the backbone, i.e. it was exhibiting axial

Fig. 8.10 Effect of implementation of purely kinematic modeling on a printed structure. The desired structure was a cylinder [74]



Fig. 8.11 Extent of axial compression of the continuum hose shown by the protruding ball suspended on a string routed through the center of the hose. The picture on the left is at initial configuration with tendons barely under any tension and the picture on the right shows the extent of axial compression [82]



compression. We verified this by running a line through the center of the hose, and observing its tip extending and contracting from the hose nozzle as the hose was re-positioned (see Fig. 8.11). We observed that the amount of protrusion varied with the bending plane suggesting variable axial compression.

What this means in practise is that the configuration (shape) of each section of the hose is defined by three variables: degree of bending θ ; plane in which that bending occurs ϕ ; and backbone length s , with this last quantity now observed to vary due to axial compression. See Fig. 8.12. The first two of these are directly controlled by tendon lengths l_i , can be described via kinematics, and in the case of inextensible sections are (in the absence of gravity) a sufficient set of variables to describe the system. In the case where extension/contraction of the section length is also directly controlled, the backbone length variable combines with the other two variables to form a complete set, describable by kinematics. However in the case where extension/contraction are due to axial compression, which is not directly controlled and is due to applied cable tension, kinematics, which are based on geometric models, are insufficient to describe the system.

Hence to model and control the system, it became necessary to use more general models which take tendon tensions into account. Our hose prototypes did not have any tension sensing capability, which precluded existing mechanics models from being used. We also did not have knowledge of the hose material properties and these would have been hard to measure post development of the prototype.

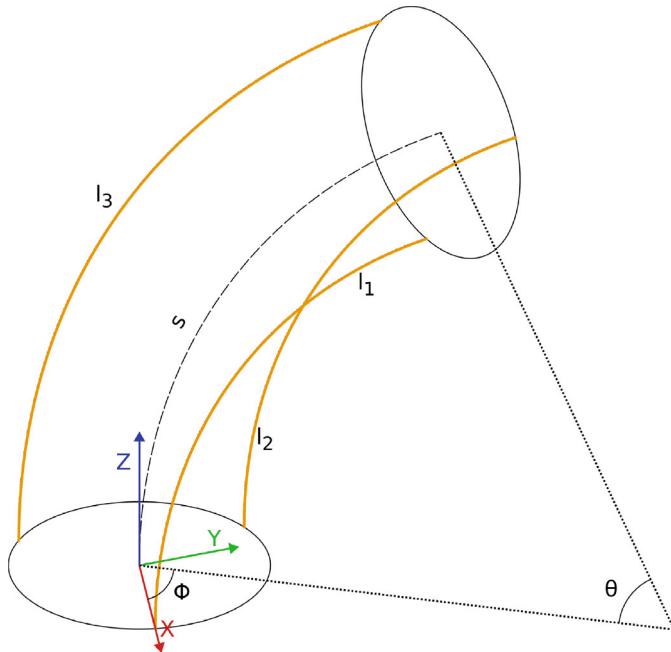


Fig. 8.12 Notation for single section robot. A section of nominal length s is actuated by three tendons with lengths l_i , to bend in a plane ϕ , with bending angle θ . Reference frame at the base (proximal) end of the section is shown, with z axis coincident with the tangent to the section backbone at the base [83]

8.4.1 Understanding the Effects of Axial Compression

To address the above issues, we developed [75] a mechanics based model that does not require tension sensing or the knowledge of material properties and worked with a variable curvature backbone by identifying and exploiting the variation of axial compression with bending plane for TDCRs with finite number of tendons.

Consider a compressible continuum backbone being actuated by a single tendon, capable of half-planar bending, without torsion. Since there is no torsion, the TDCR bends in the plane of the tendon. Assuming: 1. linearity of strain; and 2. superposition of strain, this system is governed by the equilibrium equations [84]:

$$RT = M = K_b \frac{\theta_{actual}}{s} \quad (8.1)$$

$$T = K_a \epsilon_a = K_a \frac{\Delta s}{s} \quad (8.2)$$

where the previously undefined symbols above are defined in Table 8.1.

Table 8.1 Symbol definitions

Symbol (units)	Definition
R (mm)	Distance of tendon to the center of the backbone
T (N)	Tendon tension
M (Nmm)	Bending moment at the center, $M = R \times T$
r (mm), k (mm^{-1})	Bend radius and curvature. $k = 1/r$
K_b (Nmm^2)	Bulk bending stiffness. $M = K_b * k$
ϵ_b	Pure bending strain, $\epsilon_b = k * \text{distance from center}$
θ_{actual} (rad)	Actual (reduced) EE pitch
θ_{des} (rad)	Desired EE pitch
s (mm)	Unactuated section length
Δs (mm)	Change in backbone length due to compression
ϵ_a	Uniform axial compression strain, $\epsilon_a = \Delta s / s$
K_a (N)	Bulk axial stiffness. $T = K_a \epsilon_a$

Substituting (8.2) in (8.1) results in:

$$\Delta s = \frac{K_b}{K_a} \frac{\theta_{actual}}{R} \quad (8.3)$$

These equations above are valid differentially and are therefore valid for backbones exhibiting variable curvatures by averaging the relevant properties—curvature, T, and ϵ_a . Since the goal of our analysis is controlling EE pitch, the “plane sections remain plane and perpendicular to the neutral axis after bending” assumption in Euler Bernoulli beam theory allows the use of averaged properties.

In the absence of axial compression, $K_a \rightarrow \infty$, and the tendon length change determined by a tendon length (position) controller using well established inverse kinematics would (assuming accurate sensing and control of the tendon lengths, a reasonable assumption in practise [8, 9, 11, 16]) result in the desired EE pitch, θ_{des} . However, when $K_a < \infty$, the length of the backbone reduces as EE pitch increases. Suppose that the tendon length controller were unaware that the continuum section exhibits axial compression, and actuates the section using traditional compression unaware kinematics (CUK), with the unmodeled backbone compression resulting in lesser than desired EE pitch, $\theta_{actual} < \theta_{des}$, as shown in Fig. 8.13. Linearity and superposition of strain provide the kinematic constraint:

$$\Delta l = R\theta_{des} = R\theta_{actual} + \Delta s \quad (8.4)$$

where Δl once again denotes the change in tendon length from the straight configuration.

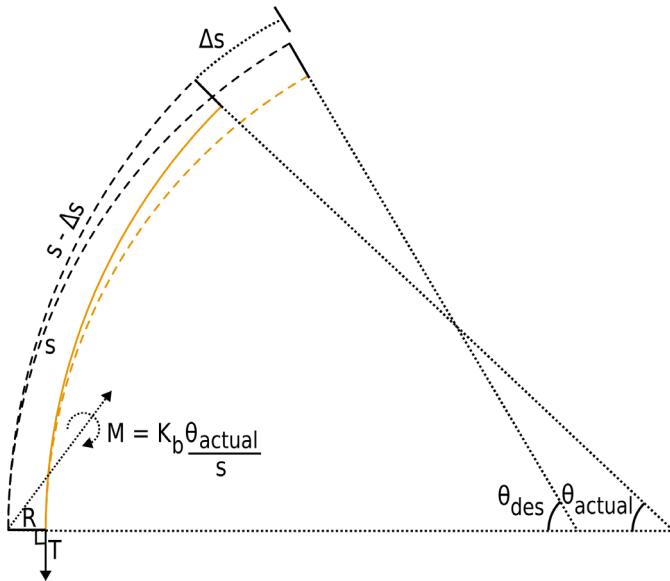


Fig. 8.13 Section bending in plane of one of the tendons with the effects of axial compression illustrated. The tendon is retracted by an amount $\Delta l = R\theta_{des}$ and tension, T , develops in it as the backbone compresses. This tension is lesser than the tension developed if the backbone were incompressible [83]

Using (8.3) in (8.4) produces:

$$\theta_{des} = \left(1 + \frac{K_b}{K_a R^2}\right) \theta_{actual} \quad (8.5)$$

Let us now consider the case when two tendons (labeled 1 and 2) separated by 120° actuate a compressible backbone. See Fig. 8.14, in which the radial connection points of three tendons are shown at the top, and the new variable ϕ , used to model the plane of bending of the section, is introduced and defined. The local coordinate system to which the subsequent analysis is referenced is located and illustrated at the base of the section.

To establish that θ_{actual} varies spatially (i.e. as a function of ϕ), we will consider the case when bending occurs at the midplane, 60° , as shown in Fig. 8.14, where tendons 1 and 2 are actuated and tendon 3 is slack (or equivalently non-existent).

Assuming linearity and superposition as before, we write the governing resultant moment (8.6) and axial compression (8.7) equilibrium conditions:

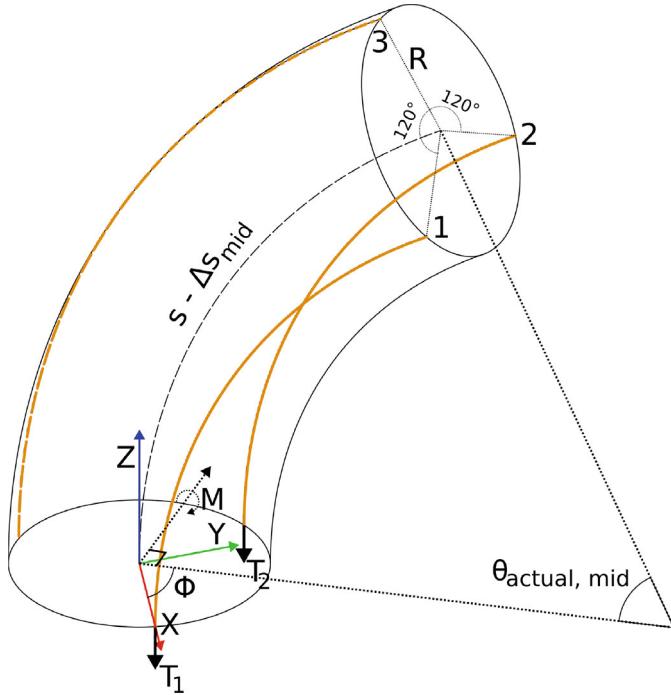


Fig. 8.14 Compressed backbone state while bending in the midplane between two tendons. If tendon 3's length is controlled using compression unaware kinematics, it would be slack $\forall \phi \in [0, 2\pi/3]$ including at the midplane, $\phi = \pi/3$ [83]

$$\begin{aligned} RT_1 \cos\left(-\frac{\pi}{3}\right) + RT_2 \cos\left(\frac{2\pi}{3} - \frac{\pi}{3}\right) &= K_b \frac{\theta_{actual,mid}}{s} \\ RT_1 \sin\left(-\frac{\pi}{3}\right) + RT_2 \sin\left(\frac{2\pi}{3} - \frac{\pi}{3}\right) &= 0 \end{aligned} \quad (8.6)$$

$$T_1 + T_2 = K_a \epsilon_{a,mid} = K_a \frac{\Delta s_{mid}}{s} \quad (8.7)$$

where the numeric subscript indicates the tendon number and 'mid' refers to the property value at the mid plane. Note that in (8.6) each side of the first equation gives the moment, at the center of the backbone and orthogonal to the plane of bending, that provides the bend (pitch) θ , at the plane of the mid point between tendons, i.e. equally spaced by $\pi/3$ from the tendons at 0 and $2\pi/3$ about the periphery, respectively. The second equation in (8.6) is the (necessarily zero, as the section is assumed to bend only in the ϕ plane) moment in the plane of bending, at the center of the backbone. The $R \cos()$ and $R \sin()$ terms provide the moment arms r_i (vectors from the tendon termination locations to the backbone center) to generate these moments via $M = r \times T$.

Equations (8.6) and (8.7) give:

$$\Delta s_{mid} = \frac{2K_b}{K_a} \frac{\theta_{actual,mid}}{R} \quad (8.8)$$

Using a controller based only on kinematics, linearity and superposition of strain, with bending shown in Fig. 8.14 and the radial offset of the tendons depicted in Fig. 8.15, the binding kinematic constraint for each tendon is:

$$\Delta l_{1,2} = \frac{R}{2} \theta_{des} = \frac{R}{2} \theta_{actual,mid} + \Delta s_{mid} \quad (8.9)$$

Using (8.8) and (8.9),

$$\theta_{des} = \left(1 + \frac{4K_b}{K_a R^2}\right) \theta_{actual,mid} \quad (8.10)$$

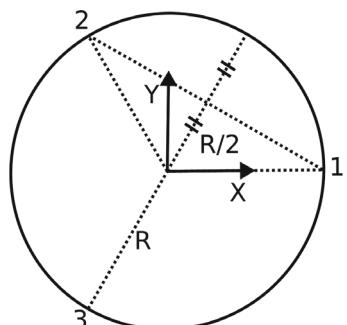
Comparing (8.10) and (8.5) shows that $\theta_{actual,mid} < \theta_{actual,0}$ proving that CUK-based algorithms must necessarily result in spatial variation of the EE pitch as the plane of bending ϕ varies, causing inaccurate bending under kinematic control when the backbone is compressible.

We have also shown [75] that the tip speed varies when backbone compression is present, due to slack which results from the unmodeled compression. Together, these two results provide an understanding of the causes of the practical problems illustrated in Fig. 8.10. However, these problems still need to be addressed in practise. In the following, we discuss how further use of the model in this section can generate an approach to compensate for the issues.

8.4.2 Elasticity Compensation (EC) Model

The underlying problem with unmodeled axial compression, as discussed above, is that when tendons are tensioned based on models that do not take into account forces

Fig. 8.15 Radial offset of the tendons while bending at the midplane [83]. Note that the moment arm through which the tendon tensions here create the moment at the center is half that of the moment arm when the bending is directly in the plane of a tendon



(i.e. pure kinematics), the backbone compresses in unanticipated ways. This allows slack in the tendons, and produces undesired configurations and trajectories of the backbone.

To compensate for slackness in tendons, a standard approach is to preload i.e. pre-tension the tendons and this precompresses the hose. However, this is typically done in an ad hoc manner at the implementation stage, resulting in more than necessary or variable (when lesser than adequate preload is applied, <2 in (8.14)) amounts of compression and loading on the structure, and introducing error into kinematics-based controllers. Preloading is the solution to the axial compression problem, but there is a minimum amount of preload required and current practise is to empirically search for the amount of it until the TDCR behaves as expected. In this sense the axial compression (or tendon extension) problem is being solved implicitly, by observing slack tendons and intuitively increasing tension in the system. Thus, currently, preloading is rather an unspoken practice that TDCR operators come to realize is necessary but is accomplished on an ad hoc basis.

In order to compensate for the axial compression in a more rigorous and predictable manner, we therefore fix the total tension in a way that ensures no slack is possible, i.e. the tendon tensions always remain non-negative, but at a minimal value calculated using our model. The total tension $\Sigma T_i(\phi)$ can only be fixed at its maximum value which occurs, when moment arm is the shortest, at the midplane. Fixing it at a lesser value implies that, at and near the midplane, tendons 1 and 2 “ease up”, requiring tendon 3 to push at the midplane to achieve the desired bending, and ropes (tendons in this situation) cannot push!

The total tension could be fixed at larger value, but this would result in subjecting the robot to unnecessary forces. Therefore, we elect to fix the total tension $\Sigma T_i(\pi/3)=2T$, where T is the tension when a single tendon pulls to achieve a specified θ , which we term θ_{\max} . When only one tendon is pulling, $T = K_b(\theta/R_s)$, where K_b is the bulk bending stiffness, therefore we set the total tension as:

$$\Sigma T_i = 2T = 2K_b(\theta_{\max}/R_s) \quad (8.11)$$

The resultant moment equilibrium equations when 3 tendons are engaged (a generalization to three tendons and variable ϕ of (8.6)) is:

$$\begin{aligned} T_1 \cos(\phi) + T_2 \cos\left(\frac{2\pi}{3} - \phi\right) + T_3 \cos\left(\frac{4\pi}{3} - \phi\right) &= K_b \frac{\theta}{R_s} \\ -T_1 \sin(\phi) + T_2 \sin\left(\frac{2\pi}{3} - \phi\right) + T_3 \sin\left(\frac{4\pi}{3} - \phi\right) &= 0 \end{aligned} \quad (8.12)$$

The coupled resultant moment and axial compression equilibrium equation with fixed maximum total tension is (combining (8.11) and (8.12)):

$$\begin{bmatrix} \cos(\phi) & \cos(\frac{2\pi}{3} - \phi) & \cos(\frac{4\pi}{3} - \phi) \\ -\sin(\phi) & \sin(\frac{2\pi}{3} - \phi) & \sin(\frac{4\pi}{3} - \phi) \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} K_b \frac{\theta}{R_s} \\ 0 \\ 2K_b \frac{\theta_{\max}}{R_s} \end{bmatrix} \quad (8.13)$$

Abbreviating, $A(\phi)$

If tension sensing and system property information (knowledge of K_b) were available, the above could be used to achieve any given ($\theta \leq \theta_{\max}$, ϕ) combination at a fixed backbone length using the tensions returned by:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = A^{-1}(\phi) \begin{bmatrix} \theta/\theta_{\max} \\ 0 \\ 2 \end{bmatrix} K_b \frac{\theta_{\max}}{R_s} \quad (8.14)$$

with guaranteed slack avoidance, $T > 0$, for all tendons at any bending plane for $\theta < \theta_{\max}$.

However, in the cement hose robot (and most tendon driven continuum robots) tension sensing is not available. Therefore, in [75, 83], we extend the model to create a linear relation used to calculate tendon length changes as a function of tendon tensions. This results is a direct method to calculate the (actively controlled) tendon lengths as a function of the desired orientation of the hose (ϕ and θ_{\max}) in the above equation.

The key idea behind our approach [83] was to identify that the distance of the neutral axis was dependent on the plane of bending. The neutral axis is an axis that runs parallel to the centroidal axis of continuum section, and does not undergo length change during bending when axial compression is present. When axial compression is not present, the neutral axis and the centroidal axis coincide, but when it is present, the centroidal axis is shorter than the neutral axis. Kinematic models do not make the distinction between these two axes.

The next step involved synthesizing the linear tendon length/tendon tension relationship to constrain the neutral axis location at its furthest point in all directions, removing the dependence on the bending plane. We did this by resolving the moments created due to tendon tensions at the centroid and considering axial compression separately [83]. They could be analysed separately due the principle of superposition of strain. Our formulation always generates non-negative tendon tensions.

8.4.3 Printing Using the EC Model

The application of our EC model to account for axial compression in the cement hose hardware successfully resolved the key issues observed when using purely kinematic models. We illustrate the performance of the approach when deploying the hardware to print a cylindrical trajectory, as shown in Fig. 8.16.

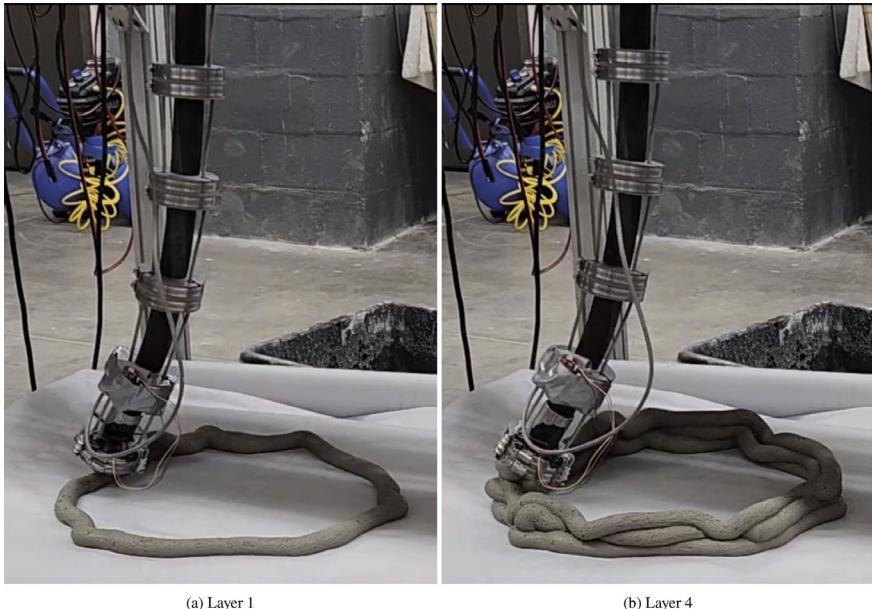


Fig. 8.16 3D printing a cylinder using the EC model. We see that the actual trajectory is similar to the desired circular trajectory. The continuum section is able to maintain relatively constant EE pitch and speed at all bend planes which can be inferred from the circular shape and evenness of material deposition [83]

Though not a perfect circle, Fig. 8.16 shows that the EC model produces a much better circular structure than kinematics in the initial experiment, Fig. 8.10. The effects of loss of curvature θ as plane of curvature ϕ varies, and the variation of EE speed, are seen to be greatly reduced, demonstrating the ability of the EC approach introduced in this work to model and compensate for their causes. This enabled us to print a cylindrical concrete structure with uniform EE speed though all the bending planes using open loop control.

8.5 Summary

The transition of soft robots from the research environment to practical applications is an exciting but challenging prospect. The compliance inherent in the structures of soft robots offers enhanced and/or previously inaccessible performance, as discussed herein and elsewhere in this book. However, issues in controlling the same compliance that motivates the use of soft robots can make their application difficult.

In this Chapter, we have discussed a specific type of robot incorporating compliance, specifically tendon driven continuum hose robots, and discussed challenges

in the application of such robots to 3D printing of concrete in construction applications. The inability of traditionally used kinematics-based approaches to account for the softness of the robot—in this case axial compression of the hose—has been demonstrated and discussed. Using a model specifically developed to model axial compression in tendon driven continuum robots, we explained the cause of the problems introduced when the robot is “soft” in this sense. We further described the use of the model to generate an approach to overcome the issues introduced by the softness/compliance. It is hoped that this case study will serve as a useful example of both the benefits and challenges involved in the practical application of soft robots.

Acknowledgements The work in this chapter was supported in part by US National Science Foundation (NSF) under Grant No. 1924721.

References

1. Nekkanti H (2020) Influence of selected supplementary cementitious materials on properties of 3D printable cementitious mixture for application in additive manufacturing. PhD thesis. Clemson University
2. SQ4D (06 Apr 2021) World’s largest permitted 3d printed home. <https://www.prnewswire.com/news-releases/sq4d-builds-the-worlds-largest-3d-printed-home-300984356.html>
3. Zhang X, Li M, Lim JH, Weng Y, Tay YWD, Pham H, Pham Q-C (2018) Large-scale 3d printing by a team of mobile robots. *Autom Constr* 95:98–106
4. Tiriyaki ME, Zhang X, Pham Q-C (2019) Printing-while-moving: a new paradigm for large-scale robotic 3d printing. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2286–2291
5. Sustarevas J, Tan KXB, Gerber D, Stuart-Smith R, Pawar VM (2019) Youwasps: towards autonomous multi-robot mobile deposition for construction. In 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2320–2327
6. Izard Jean-Baptiste, Dubor Alexandre, Hervé Pierre-Elie, Cabay Edouard, Culla David, Rodriguez Mariola, Barrado Mikel (2017) Large-scale 3d printing with cable-driven parallel robots. *Constr Robot* 1:69–76
7. Murray RM, Li Z, Shankar Sastry S (2017) A mathematical introduction to robotic manipulation. CRC Press
8. Russo M, Hadi Sadati SM, Dong X, Mohammad A, Walker ID, Bergeles C, Xu X, Axinte DA (2023) Continuum robots: an overview. *Adv Intell Syst* 5:1–25
9. Webster RJ, Jones BA (2010) Design and kinematic modeling of constant curvature continuum robots: a review. *Int J Robot Res* 29(13):1661–1683
10. Walker ID, Dawson D, Flash T, Grasso F, Hanlon RJ, Hochner B, Kier WM, Pagano C, Rahn CD, Zhang Q (2005) Continuum robot arms inspired by cephalopods. In: Proceedings SPIE conference on unmanned ground vehicle technology VII, pp 303–314
11. Walker ID (2013) Continuous backbone: “Continuum” robot manipulators. *ISRN Robot* 1–19:2013
12. Anderson VC, Horn RC (1967) Tensor arm manipulator design. *Trans ASME* 67-DE-57:1–12
13. Chirikjian GS, Burdick JW (1994) A modal approach to hyper-redundant manipulator kinematics. *IEEE Trans Robot Autom* 10(3):343–354
14. Hirose S (1993) Biologically inspired robots. Oxford University Press
15. Robinson G, Davies JBC (1999) Continuum robots—a state of the art. In: IEEE International conference on robotics and automation, pp 2849–2854

16. Trivedi D, Rahn CD, Kier WM, Walker ID (2008) Soft robotics: biological inspiration, state of the art, and future research. *Appl Bionics Biomech* 5(3):99–117
17. Kier WM, Smith KK (1985) Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats. *Zool J Linnean Soc* 83:307–324
18. Sumbre Germán, Fiorito Graziano, Flash Tamar, Hochner Binyamin (2006) Octopuses use a human-like strategy to control precise point-to-point arm movements. *Curr Biol* 16(8):767–772
19. Hannan MW, Walker ID (2003) Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *J Robot Syst* 20(2):45–63
20. Wilson JF, Li D, Chen Z, George RT (1993) Flexible robot manipulators and grippers: relatives of elephant trunks and squid tentacles. In: *Robots and biological systems: toward a new bionics*. NATO, pp 474–479
21. Burgner-Kahrs J, Gilbert HB, Granna J, Swaney PJ, Webster RJ (2014) Workspace characterization for concentric tube continuum robots. In: *Proceedings IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 1269–1275
22. Lyons LA, Webster III RJ, Alterovitz R (2010) Planning active cannula configurations through tubular anatomy. In: *Proceedings IEEE international conference on robotics and automation*, pp 2082–2087
23. Lock J, Laing G, Mahvash M, Dupont PE (2010) Quasistatic modeling of concentric tube robots with external loads. In: *Proceedings IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 2325–2332
24. Kim C, Ryu SC, Dupont PE (2015) Real-time adaptive kinematic model estimation of concentric tube robots. In: *Proceedings IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 3214–3219
25. Torres LG, Baykal C, Alterovitz R (2014) Interactive-rate motion planning for concentric tube robots. In: *Proceedings IEEE international conference on robotics and automation*, pp 1914–1920
26. Wang J, Ha J, Dupont PE (2019) Steering a multi-armed robotic sheath using eccentric pre-curved tubes. In: *Proceedings IEEE international conference on robotics and automation*, pp 9834–9840
27. Butler EJ, Hammond-Oakley R, Chawarski S, Gosline AH, Codd P, Anor T, Madsen JR, Dupont PE (2012) Robotic neuro-endoscope with concentric tube augmentation. In: *Proceedings IEEE/RSJ international conference on intelligent robot systems (IROS)*, pp 2941–2946
28. Swaney PJ, Mahoney AW, Remirez AA, Lamers E, Hartley BI, Feins RH, Alterovitz R, Webster III RJ (2015) Tendons, concentric tubes, and a bevel tip: Three steerable robots in one transoral lung access system. In: *IEEE international conference on robotics and automation*, pp 5378–5383
29. Burgner-Kahrs J, Rucker DC, Choset H (2015) Continuum robots for medical applications: a survey. *IEEE Trans Robot* 31(6):1261–1280
30. Grzesiak A, Becker R, Verl A (2011) The bionic handling assistant—a success story of additive manufacturing. *Assem Autom* 31(4)
31. Mahl T, Hildebrandt A, Sawodny O (2014) A variable curvature continuum kinematics for kinematic control of the bionic handling assistant. *IEEE Trans Robot* 30(4):935–949
32. McMahan W, Pritts M, Chitrakaran V, Dienno D, Grissom M, Jones B, Csencsits M, Rahn CD, Dawson D, Walker ID (2006) Field trials and testing of OCTARM continuum robots. In: *Proceedings IEEE international conference on robotics and automation*, pp 2336–2341
33. Allen EA, Swensen JP (2020) Design of a highly-maneuverable pneumatic soft actuator driven by intrinsic SMA coils (PneuSMA actuator). In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 8667–8672
34. Jones BA, Walker ID (2006) Kinematics for multisection continuum robots. *IEEE Trans Robot* 22(1):43–55
35. Suzumori K, Iikura S, Tanaka H (1991) Development of flexible microactuator and its applications to robotic mechanisms. In: *IEEE international conference on robotics and automation*, pp 1622–1627

36. McMahan W, Jones BA, Walker ID (2005) Design and implementation of a multi-section continuum robot: air-octor. In: Proceedings IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3345–3352
37. Nguyen PH, Mohd IB, Sparks C, Arellano FL, Zhang W, Polygerinos P (2019) Fabric soft poly-limbs for physical assistance of daily living tasks. In: Proceedings IEEE international conference on robotics and automation, pp 8429–8435
38. Laschi C, Mazzolai B, Mattoli V, Cianchetti M, Dario P (2009) Design of a biomimetic robotic octopus arm. *Bioinspiration Biomim* 1:1–8
39. Martinez RV, Branch JL, Fish CR, Jin L, Shepherd RF, Nunes RMD, Suo Z, Whitesides GM (2013) Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Adv Mater* 25(2):205–212
40. Xu K, Zhao J, Geiger J, Shih AJ, Zheng M (2011) Design of an endoscopic stitching device for surgical obesity treatment using a n.o.t.e.s. approach. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 961–966
41. Ataka Ahmad, Abrar Taqi, Putzu Fabrizio, Godaba Hareesh, Althoefer Kaspar (2020) Model-based pose control of inflatable eversion robot with variable stiffness. *IEEE Robot Autom Lett* 5(2):3398–3405
42. Greer JD, Blumenschein LH, Okamura AM, Hawkes EW (2018) Obstacle-aided navigation of a soft growing robot. In: Proceedings IEEE international conference on robotics and automation, pp 4165–4172
43. Greer JD, Blumenschein LH, Alterovitz R, Hawkes EW, Okamura AM (2019) Robust navigation of a soft growing robot by exploiting contact with the environment. [arXiv:1908.08645](https://arxiv.org/abs/1908.08645)
44. Putzu F, Abrar T, Althoefer K (2018) Plant-inspired soft pneumatic eversion robot. In: 2018 7th IEEE international conference on biomedical robotics and biomechatronics (Biorob), pp 1327–1332
45. Neumann M, Burgner-Kahrs J (2016) Considerations for follow-the-leader motion of extensible tendon-driven continuum robots. In: IEEE international conference on robotics and automation (ICRA), pp 917–923. IEEE
46. Subramani G, Zinn MR (2015) Tackling friction—an analytical modeling approach to understanding friction in single tendon driven continuum manipulators. In: IEEE international conference on robotics and automation (ICRA), pp 610–617. IEEE
47. Tonapi M, Godage IS, Vijaykumar AM, Walker ID (2015) Spatial kinematic modeling of a long and thin continuum robotic cable. In: Proceedings IEEE international conference on robotics and automation, pp 3755–3761
48. Dalvand MM, Nahavandi S, Howe RD (2018) An analytical loading model for n-tendon continuum robots. *IEEE Trans Robot* 34(5):1215–1225
49. Mehling JS, Diftler MA, Chu M, Valvo M (2006) A minimally invasive tendril robot for in-space inspection. In: BioRobotics 2006 conference, pp 690–695
50. Mahvash M, Dupont PE (2010) Stiffness control of a continuum manipulator in contact with a soft environment. In: *IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp 863–870
51. Shiva A, Stilli A, Noh Y, Faragasso A, De Falco I, Gerboni G, Cianchetti M, Menciassi A, Althoefer K, Wurdemann HA (2016) Tendon-based stiffening for a pneumatically actuated soft manipulator. *IEEE Robot Autom Lett* 1(2):632–637
52. Santiago JL, Gonthina P, Godage IS, Walker ID (2016) Soft robots and kangaroo tails: modulating compliance in continuum structures via mechanical layer jamming. *Soft Robot* 3(2):54–63
53. Sadati SMH, Sullivan L, Walker ID, Althoefer K, Nanayakkara T (2018) A 3d-printable thermoactive helical interface with decentralized morphological stiffness control for continuum manipulators. *IEEE Robot Autom Lett* 3(3):2283–2290
54. Degani A, Choset H, Wolf A, Zenati M (2006) Highly articulated robotic probe for minimally invasive surgery. In: Proceedings IEEE international conference on robotics and automation, pp 4167–4172

55. Zhang Z, Dequidt J, Back J, Liu H, Duriez C (2019) Motion control of cable-driven continuum catheter robot through contacts. In: IEEE robotics and automation letters, presented at IEEE international conference on robotics and automation
56. Yoon H-S, Oh SM, Jeong JH, Lee SH, Tae K, Koh K-C, Yi BJ (2011) Active bending robot endoscope system for navigation through sinus area. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 967–972
57. Watanabe H, Kanou K, Kobayashi Y, Fujie MG (2011) Development of a “steerable drill for ACL reconstruction to create the arbitrary trajectory of a bone tunnel. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 955–960
58. Liu N, Abdelaziz EMK, Shen M, Yang G (2018) Design and kinematics characterization of a laser-profiled continuum manipulator for the guidance of bronchoscopic instruments. In: Proceedings IEEE international conference on robotics and automation, pp 25–31
59. Simaan N, Taylor R, Flint P (2004) A dexterous system for laryngeal surgery. In: Proceedings IEEE international conference on robotics and automation, pp 351–357
60. Wang H, Xu H, Yu F, Li X, Yang C, Chen S, Chen J, Zhang Y, Zhou X (2020) Modeling and experiments on the swallowing characteristics of an underwater continuum manipulator. In: IEEE international conference on robotics and automation (ICRA), pp 2946–2952
61. Lastinger MC, Rice A, Kapadia A, Walker ID (2020) Variable topology “tree-like” continuum robots for remote inspection and cleaning. In: Proceedings IEEE aerospace conference, pp 1–10
62. Wang M, Palmer D, Dong X, Alatorre D, Axinte D, Norton A (2018) Design and development of a slender dual-structure continuum robot for in-situ aeroengine repair. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5648–5653
63. Dong X, Axinte D, Palmer D, Cobos S, Raffles M, Rabani A, Kell J (2017) Development of a slender robotic system for on-wing inspection/repair of gas turbine engines. Robot Comput Integr Manuf
64. Wooten M, Frazelle C, Walker ID, Kapadia A, Lee JH (2018) Exploration and inspection with vine-inspired continuum robots. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1–5
65. Walker ID (2013) Robot strings: long, thin continuum robots. In: IEEE aerospace conference, pp 1–12
66. Godage IS, Nanayakkara T, Caldwell DG (2012) Locomotion with continuum limbs. In: Proceedings IEEE/RSJ international conference on intelligent robot systems (IROS), pp 293–298
67. Kazakidi A, Zabulis X, Tsakiris DP (2015) Vision based 3d motion reconstruction of octopus arm swimming and comparison with an 8-arm underwater robot. In: IEEE international conference on robotics and automation, pp 1178–1183
68. Hou T, Yang X, Su H, Jiang B, Chen L, Wang T, Liang J (2019) Design and experiments of a squid-like aquatic-aerial vehicle with soft morphing fins and arms. In: IEEE international conference on robotics and automation, pp 4681–4687
69. Tsukagoshi H, Kitagawa A, Segawa M (2001) Active hose: an artificial elephant’s nose with maneuverability for rescue operation. In: IEEE international conference on robotics and automation, pp 2454–2459
70. Bostelman RV, Albus JS, Graham RE (1997) RoboCrane and EMMA applied to waste storage tank remediation. In: American nuclear society seventh topical meeting on robotics and remote systems
71. Scott GP, Glen Henshaw C, Walker ID, Willimon B (2015) Autonomous robotic refueling of an unmanned surface vehicle in varying sea states. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 1664–1671
72. Lastinger MC, Verma S, Kapadia AD, Walker ID (2019) Tree: a variable topology, branching continuum robot. In: Proceedings IEEE international conference on robotics and automation (ICRA), pp 5365–5371
73. Ando H, Ambe Y, Ishii A, Konyo M, Tadakuma K, Maruyama S, Tadokoro S (2018) Aerial hose type robot by water jet for fire fighting. IEEE Robot Autom Lett 3(2):1128–1135

74. Srivastava M, Ammons J, Peerzada AB, Krovi VN, Rangaraju P, Walker ID (2022) 3d printing of concrete with a continuum robot using variable curvature kinematics. In: IEEE international conference on robotics and automation (ICRA), pp 3216–3222
75. Srivastava M (2023) Accurate orientation control of tendon driven continuum robots that exhibit elasticity. PhD dissertation, Department of Electrical and Computer Engineering, Clemson University, USA
76. Chesser PC, Wang PL, Vaughan JE, Lind RF, Post BK (2022) Kinematics of a cable-driven robotic platform for large-scale additive manufacturing. *J Mech Robot* 14(2)
77. Gravagne IA, Rahn CD, Walker ID (2003) Large deflection dynamics and control for planar continuum robots. *IEEE/ASME Trans Mechatron* 8(2):299–307
78. Gravagne IA, Walker ID (2002) Manipulability, force, and compliance analysis for planar continuum manipulators. *IEEE Trans Robot Autom* 18(3):263–273
79. Oliver-Butler K, Till J, Rucker DC (2019) Continuum robot stiffness under external loads and prescribed tendon displacements. *IEEE Trans Robot* 35(2):403–419
80. Martin F, Niemitz C (2003) How do African elephants (*Loxodonta Africana*) optimize goal-directed trunk movements? *Jahresversammlung der Dt. Zool. Ges. und der Dt. Ges. f. Parasitologie*, 96:159
81. Peerzada AB (2022) Active rheological and stiffening control of cementitious systems for additive manufacturing. PhD dissertation, Department of Civil Engineering, Clemson University, USA
82. Srivastava M, Walker ID (2023) On tendon driven continuum robots with compressible backbones. In: IEEE international conference on robotics and automation (ICRA), pp 669–675
83. Srivastava M, Walker ID (2023) On backbone compression and tendon extensibility in tendon driven continuum robots. *Robot Autom Lett* (submitted)
84. Yip MC, Camarillo DB (2014) Model-less feedback control of continuum manipulators in constrained environments. *IEEE Trans Robot* 30(4):880–889

Chapter 9

Origami Robots



Cynthia Sung and Jamie Paik

Abstract This chapter discusses origami robotics, that is, robots that are fabricated as 2D sheets and folded into their 3D form. Folding is an elegant solution to assembly wherein thin sheets morph to produce a wide variety of structures with complex geometry, kinematics, and mechanical response. By incorporating actuators, sensing, and computation directly into the sheet, full robots can be fabricated, self-assembled, and deployed in a single uniform process. Moving joints are fabricated in exactly the same way as static, structural elements, opening opportunities for end-to-end manufacturing of multi-functional robotic devices with fully embedded behaviors and control. We discuss methods for modeling, designing, controlling, and fabricating origami robots, and we highlight how these principles have been applied in a wide variety of applications, including manipulation, locomotion, self-reconfiguration, and human-robot interaction.

9.1 Introduction

In the push for more capable, robust, and versatile technologies, there is a pressing need for a novel design paradigm that goes beyond conventional approaches. The symbiotic relationship between humans and machines necessitates a shift from rigid, utilitarian robotic designs to those that seamlessly integrate into our lives, enhancing our capabilities and adapting to our dynamic environments. Enter the realm of origami-inspired robot design—an arena that not only challenges existing paradigms but also harnesses the potential of origami-based mechanical metamaterials to advance robotics.

C. Sung (✉)
University of Pennsylvania, Philadelphia, PA, USA
e-mail: crsung@seas.upenn.edu

J. Paik
École Polytechnique Fédérale De Lausanne, Lausanne, Switzerland
e-mail: jamie.paik@epfl.ch

Origami robots are machines whose morphologies and functions are created by folding flat sheets. The term *origami* refers to the ancient art of paper folding. It is derived from the Japanese *ori* meaning “to fold” and *kami* meaning “paper.” In robotics, this “paper” has expanded in scope to include flat sheets of almost any material, including multilayered multimaterial composites that integrate sensing, actuation, and computation. When the sheet is folded, these active elements are aligned into a complex, three-dimensional (3D) form, enabling actuation and sensing elements to operate in the required directions despite their two-dimensional (2D) origin. Folding as a fabrication and assembly strategy reduces design complexity since all of the robotic components are integrated directly in sheet form, and no additional wiring or mounting is required. The strategy enables rapid fabrication of active structures where folds are used as joints or as structural support in the robot’s body.

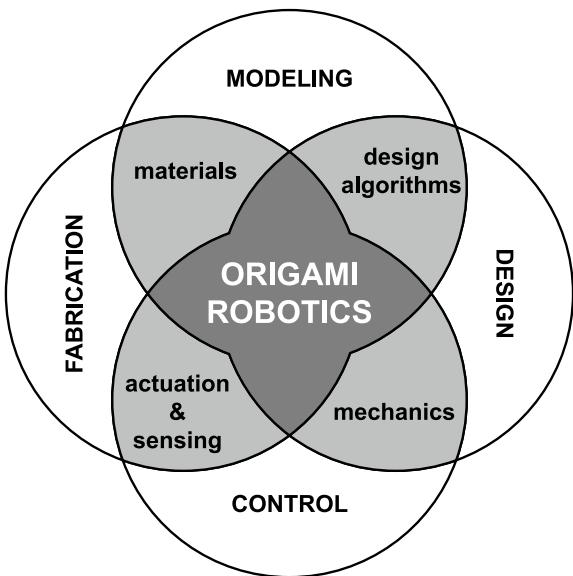
Folding thin sheets naturally gives rise to soft, compliant, and adaptable robots. Origami robots fall into the class of semi-rigid robots. They are soft in the sense that they are made of thin materials that bend and twist, especially around fold lines. At the same time, in many cases, origami robots can be treated as stiff faces rotating about soft hinges. Origami robots thus span the spectrum between more traditional rigid robots and completely soft robots, with behavior that can be tuned by manipulating the fold pattern itself. By placing folds strategically in a 3D robot, the body can be designed to locally stretch, shrink, shear, and bend, enabling robots to self-assemble and deploy reliably and to exhibit anisotropic compliance to both conform to and resist loads as needed.

Capabilities of Origami Machines

Origami robots have been successfully demonstrate in a broad array of applications [1–3] in manipulation, locomotion, reconfiguration, and human-computer interaction tasks, leading to the growth of a new field in origami robot design. The primary strengths of origami robots include the following:

1. By taking advantage of a uniform, mostly planar fabrication process, origami robots with fully integrated computation, actuation, and sensing can be manufactured quickly and inexpensively, often within hours or days.
2. Origami structures are inherently compliant with main modes of deformation that can be tuned by changing the geometry of the fold pattern, making them highly customizable.
3. The same fold pattern can often fold in multiple different ways, making origami robots self-reconfigurable even after assembly. Thus origami robots can change their geometries, their kinematic structures, or their mechanical responses depending on the task requirements.
4. Origami structures are thin-shell structures and have comparatively high strength-to-weight relative to their 3D counterparts. Such thin-shell structures provide design advantages since more mass and volume can be devoted to active components such as actuation, power, and computation.

Fig. 9.1 Overview of engineering subfields that contribute to origami robotics



5. Since origami robots are constructed as multimaterial multifunctional sheets, the robot's bodies affect their behaviors, and the robot's physical response and controller are tightly connected. Origami robots are thus prime platforms for demonstrating embodied intelligence and for navigating the spectrum of physical and computational control.

Major Components of Origami Robots

Origami robots are tightly integrated systems that incorporate new advancements in modeling, design, control, and fabrication processes. Figure 9.1 outlines these components and how they interact in the context of an origami robot. In the sections below, we outline the major results in each of these spaces, concluding with open problems and future outlooks for the field at their intersections.

9.2 Fundamentals: Modeling and Designing Origami as Kinematic Structures

At its core, folding is the process of adding rotational hinges into a flat sheet. These rotational hinges provide the structure with more degrees of freedom and also add kinematic constraints. Geometrically, it is thus possible to analyze and design origami patterns as purely kinematic structures consisting of rigid plates connected at rotational joints. The original models and designs developed for origami structures treated them in this way.

9.2.1 Basic Origami Patterns and Mechanisms

Taking this viewpoint, there are a few constructions that appear repeatedly and as the basis of many origami robot designs (ref. Fig. 9.2).

Straight crease

The most basic fold pattern in origami design is the single straight crease. The fold turns a flat sheet into two flaps that can rotate relative to each other. These flaps are referred to as *faces*. When the crease is folded so that the sheet forms an “V,” this is called a *valley fold*. When the crease is folded so that the sheet forms a “Λ,” this is called a *mountain fold*. The 3D form in Fig. 9.2 shows a mountain fold.

When faces are assumed to be rigid and the only deformation of the sheet is at the crease, the shape of the *rigid origami* can be modeled using rigid body transformations. Let us represent a fold pattern as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n_v}\}$ are the vertex locations and the edges are $\mathcal{E} = \{(\mathbf{v}_i, \mathbf{v}_j)\}$ such that the line segment $\mathbf{v}_i\mathbf{v}_j$ is either on the boundary of the sheet or is a crease line, as shown in Fig. 9.3. Edges on the interior of the sheet are *creases* $\mathbf{c}_i \in \mathcal{E}$ and each have associated with them a fold angle θ_i . Then the faces \mathcal{F}_i can also be represented as the list of either vertices or edges forming their boundaries. It is standard convention to order these vertices or edges in counterclockwise order.

Now consider a single crease \mathbf{c}_i with a corresponding fold angle θ_i . Decompose the crease line into its direction $\hat{\mathbf{c}}_i$ and its midpoint $\bar{\mathbf{c}}_i$. Hold one of the faces neighboring the crease fixed and let \mathbf{p} be a point on the interior of the other neighboring face. Then the location of the point \mathbf{p} in the folded state can be computed as

$$\mathbf{p}' = \text{Fold}_{\mathbf{c}_i, \theta_i}(\mathbf{p}) = \text{Rot}(\hat{\mathbf{c}}_i, \theta_i)(\mathbf{p} - \bar{\mathbf{c}}_i) + \bar{\mathbf{c}}_i \quad (9.1)$$

where \mathbf{p}' is the 3D location of the transformed point, \mathbf{p} is its original untransformed location in the 2D flat sheet, and $\text{Rot}(\hat{\mathbf{c}}, \theta)$ is a rotation matrix corresponding to a rotation about the axis $\hat{\mathbf{c}}$ by an angle θ . Note that the resulting 3D shape will depend on the sign of θ and the direction of \mathbf{c} . Similarly to how we describe faces, it is

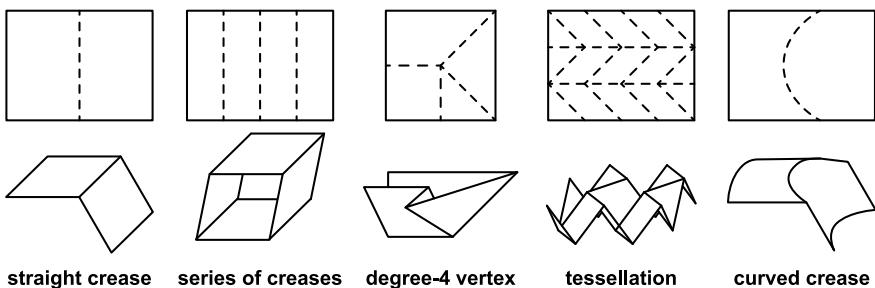


Fig. 9.2 Basic folds and origami patterns used in mechanisms and robot design

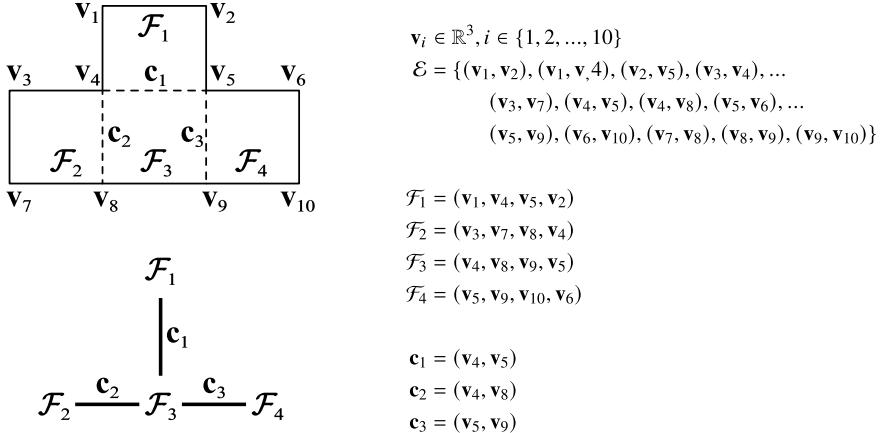


Fig. 9.3 Representation of a fold pattern. The fold pattern is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_{n_v}\}$ are the vertices of the fold pattern in the 2D sheet, and \mathcal{E} are the boundary and crease edges. The creases \mathbf{c} separate the sheet into faces \mathcal{F} . The pattern can also be simplified into a graph of faces and creases in order to compute its kinematics

standard convention to take \mathbf{c} in the direction counterclockwise around the boundary of the face being transformed (i.e., right-hand about the face's normal). Using this convention, the fold angle θ remains the same for both faces on either side of the crease, and the resulting 3D shape remains the same regardless of which face is being transformed, up to a global rotation and translation.

When the fold angle θ is left free, then the straight crease forms the simplest type of origami joint, with one degree-of-freedom rotation up to 360° between the two faces.

Series of Creases

Multiple creases can be added to a sheet to increase the complexity of the geometry, change the number of degrees of freedom, or add kinematic constraints. In the simplest case, when none of the creases touch each other (i.e., their endpoints are all on the boundaries of the sheet), then the creases form independent joints, and the kinematics of the origami pattern is that of a rigid-body serial kinematic chain. The geometry of a folded structure given fold angles can thus be computed by composing homogeneous transformation matrices as is standard procedure when computing forward kinematics for serial chains (see, for example, [4]). For the pattern in Fig. 9.3, holding face \mathcal{F}_2 fixed, the location of a point $\mathbf{p} \in \mathcal{F}_4$ can be computed as

$$\mathbf{p}' = \text{Fold}_{\mathbf{c}_2, \theta_2} (\text{Fold}_{\mathbf{c}_3, \theta_3} (\mathbf{p})) \quad (9.2)$$

since creases \mathbf{c}_2 and \mathbf{c}_3 are on the path from \mathcal{F}_2 to \mathcal{F}_4 . The sequence of creases that must be included in such a computation can be identified via, for example, breadth-first search.

Degree-4 Vertex

Creases that touch on the interior of the sheet have more complicated kinematics since the all of the faces connected by those creases form a cycle and impose kinematic constraints. For an interior vertex, the number of creases that form the vertex is called the *degree* of the vertex. The number of faces that surround the vertex is equal to the degree.

The faces around an interior vertex form a spherical linkage mechanism where the vertex where the creases meet is the sphere center [5, 6]. To visualize this equivalency, imagine cutting out the vertex of a fold pattern so that the cut out portion is a circle with the single vertex of interest in the center and all of the creases are now the same length. Now imagine folding the circle at the creases. Since lengths are preserved during folding, the outer edges of the sheet must always lie on a sphere with radius equal to the lengths of the creases.

Using this equivalency, we find that the degree-4 vertex is the smallest fold pattern for an interior vertex that admits motion of the folded sheet. In particular, Grubler's formula applied to spherical linkages indicates that the number of degrees of freedom for a degree-4 vertex is

$$\#dof = 3(N - 1 - j) + j = 3(4 - 1 - 4) + 4 = 1 \quad (9.3)$$

where $N = 4$ is the number of faces and $j = 4$ is the number of crease. Fewer creases yields a rigid structure (unless two of the creases happen to be in line with each other), and more creases produce more degrees of freedom. Thus, the geometry of the degree-4 vertex can be determined by controlling a single fold angle. Analytical expressions for the relationships between fold angles in an arbitrary degree-4 origami vertex have been derived in [7].

Tessellation

Combining multiple creases and vertices increase the complexity of the fold pattern and its resulting motion. Locating these creases is the essence of origami design.

Tessellations are origami patterns consisting of repeated units arranged in a regular tiling [8]. These patterns were some of the first origami patterns to find practical use in engineering. Perhaps the most famous of these is the Miura-ori [9, 10] (Fig. 9.2), which was originally proposed for collapsing large sheets into a small area for applications such as deployable solar arrays. The pattern is a tiling of parallelogram faces. At each of the parallelogram corners is a degree-4 vertex. Due to the arrangement of parallelograms, the entire fold pattern has one degree of freedom; that is, the entire fold pattern theoretically can fold with a single actuator.

Tessellations are appealing for deployable structures since their regular arrangement admits a simplified modeling and analysis approach. It is necessary to analyze

only a few vertices to capture the behavior of the entire sheet. The motion of these vertices are subject to symmetry constraints and, once determined, can be propagated through the rest of the pattern. The Miura-ori pattern is a specific instance of a large class of tessellations consisting of quadrilaterals connected by degree-4 vertices. Other popular variations include the Kresling and Yoshimura patterns inspired by buckling tubes, as well as the general quadrilateral mesh [11]. It has been shown that tuning the angle of each quadrilateral in the pattern produces one-degree-of-freedom deployable surfaces with designable curvature [12], meaning that tessellations apply not only to flat arrays but to potentially arbitrary surfaces.

Curved Crease

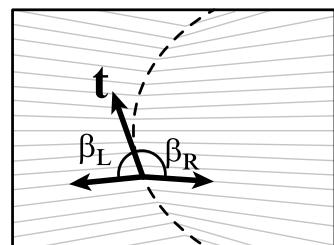
While most origami structures are composed of straight creases, it is also possible to fold along creases that are curved. Compared to straight creases, curved creases bend the folded sheet and generate curved surfaces [13]. More recently, there has been growing interest in how to design [14–18] and model [19–21] these forms.

In particular, structures folded from flat sheets fall into a class of surfaces known as *developable* surfaces, which means that they remain locally flat everywhere, except at the creases. The locally flat constraint implies that there exist lines on the sheet, called *ruling lines*, that remain straight on the folded form and along which the tangent plane to the folded form is constant [20] (ref. Fig. 9.4). If the ruling lines are known, then the fold angles along each curve can be computed directly. Parameterize the curved crease by the arclength parameter s and let $\mathbf{t}(s)$ denote the tangent to the curve at s . Let $\beta_R(s)$ and $\beta_L(s)$ denote the angles in 2D between the curved crease and the ruling lines on the right and left sides, respectively, and let $\kappa_{2D}(s)$ be the curvature of the crease when flat. Then, in the 3D form, the folded curve must satisfy [13, 16]:

$$\frac{d\alpha}{ds} = \frac{1}{2}\kappa_{2D}(s) \tan \alpha(s) (\cot \beta_L(s) - \cot \beta_R(s)) \quad (9.4)$$

where $\alpha(s)$ is half the fold angle. Note that the fold angle $2\alpha(s)$ is not constant along the curve. The 3D form can then be computed by using the geometric relationships:

Fig. 9.4 Curved crease pattern with ruling lines drawn in gray. In order to compute the folded state, the angles between the tangent to the curved crease \mathbf{t} and the ruling lines on the left and right of the curve must be known



$$\kappa(s) \cos \alpha(s) = \kappa_{2D}(s) \quad (9.5)$$

$$\tau(s) = \frac{1}{2} \kappa_{2D}(s) \tan \alpha(s) (\cot \beta_L(s) + \cot \beta_R(s)) \quad (9.6)$$

where $\kappa(s)$ and $\tau(s)$ are the curvature and the torsion of the curved crease in 3D, respectively.

While these equations have been used to design a number of curved crease origami structures [16, 18], in practice, the exact placement of the ruling lines depends on a number of factors, including the material properties of the sheet and external constraints composed by contact or collision. Thus, the number of degrees of freedom in a curved crease is potentially infinite, and the problem of systematic design of curved crease origami structures remains open.

Kirigami

Traditionally, “origami” refers only to foldings of sheets without any holes. In many instances, however, cutting and adding holes to a folded structure has practical benefits, increasing the number of degrees of freedom without adding folding complexity. More recently, there has been a growing interest in these so-called “kirigami” patterns (*kiri* being the Japanese word “to cut”), and in how combinations of cutting and folding can be used to create more versatile and complex shapes.

In many cases, cuts and holes are used simply as a method to remove extra folds that would be required to fold an intact sheet but serve no structural or functional purpose. Patterns in [22–24] are examples of this strategy. In these cases, the main folds required to assemble the structure and for it to continue to deform afterward are kept in their required locations in the sheet, and holes are cut out to remove the excess.

At the same, introducing cuts changes the kinematics and the mechanics of the structure and expands the space of foldable and thin-sheet designs. Similarly to origami tessellations, kirigami designs often consist of regular, repeated patterns, creating a new class of folded *metamaterial* whose properties and behavior are determined not only by the sheet being folded but also by the precise arrangement of cuts and folds. Models such as those in [25] study how thin strips of material interact when a kirigami sheet is stretched or deformed in ways that would not be kinematically possible if the sheet were rigid. These ideas can be applied to create springs with programmable stiffness [26], as well as creating textures for such applications as collecting condensation [27] or generating directional friction for snake-like locomotion [28].

9.2.2 Algorithmic Design of Geometry and Kinematics

Attempts to design folded structures algorithmically often focus on geometric design. In this space, theoreticians have found that static simple polyhedra of any geometry can be formed [29–31]. Approaches such as Treemaker [30], Origamizer [29], and

others [32] produce fold patterns for simple and complex polyhedral surfaces with a focus on the outer geometry of the structure, often ignoring the number of layers, number of folds, or excess of paper on the inside of the structure. Work in [33] characterized these types of fold patterns and provided necessary conditions for the patterns to be foldable. Many of these design algorithms are based on folded tucks [34–36], which induce kinematic constraints that can lock the folded structure. This is good for creating structural rigidity but not always for generating motion.

Robots fall into the category of *action origami* (i.e., origami that moves) since robots must be able to move and perform work after folding. The number of action origami designs in art and engineering is large, and many can be analyzed as compositions of spherical four-bar linkages. Other than an attempt to classify existing action origami patterns [6], very little is known about designing this class of fold patterns. Work in fold pattern composition has shown that arbitrary linkage mechanisms are able to be constructed through folding [23, 37, 38], but the resulting fold pattern is complex. Kirigami-inspired approaches have simplified the resulting patterns [39–41], leading to more compliant structures with large degrees of freedom. Once the fold pattern is known, [42] also provided necessary and sufficient conditions for the structure to be able to be actuated to fold into the desired structure. Still, the design problem for generating a particular motion remains challenging.

A common strategy to origami design involves modularizing the structure, that is, combining simple designs with known forms and fold patterns into more complicated ones. This approach is particularly powerful for trusses and linkage mechanisms, which can be broken down into combinations of structural links and joints. In this case, the design problem reduces to three questions:

1. What modules are required to cover the space of desired designs?
2. For a given design problem, which modules should be combined into the solution? and
3. How should modules be combined in a way that is correct geometrically and kinematically?

Modular approaches to origami mechanism design [43–47] primarily tackle question 1, focusing on building libraries of expressive crease patterns. These patterns are generated through in-depth studies of action origami [6, 48], buckling and bellows patterns [49–51], and high-degree-of-freedom tessellations of both the origami [52] and kirigami variety [22, 23]. For question 2, a number of systems provide design guidelines for how to choose parts from a database of parameterized origami links [44, 45], joints [51, 53], or both [43, 46]. All of these approaches target interactive design and rely on the user to specify both the modules to be joined as well as how they should be combined. It has been shown that for certain classes of mechanisms – e.g., serial kinematic chains [38] – the choice of modules can be fully automated, but more work must be done to understand the full extent of formalizable results in origami mechanism design. Finally, for question 3, partial automation of origami module compositions has been developed for shapes in [37, 43, 44], including algorithmic resolution of geometric conflicts. Systems in [54–56] provide methods for users to visualize, simulate, and design the resulting fold patterns.

9.3 Origami as Soft Structures

Because origami structures are formed from thin sheets, they allow for bending, twisting, and other soft-body deformations. Folds in the sheet impose kinematic constraints that then change the structure's degrees of freedom and its mechanical response. By taking advantage of the highly customizable and programmable mechanics of origami, robots can simplify their controllers and computational requirements by embedding desired behaviors directly into their mechanical structure [57]. Multiple examples of folded robot designs explicitly strive to minimize the compliance of the robot by introducing internal bracing or additional layers of material [58, 59] to reduce this discrepancy, but in actuality, they miss an opportunity to take advantage of deformation in the sheets for additional degrees of freedom. In this section, we outline efforts to model soft origami structures, which can be incorporated into designs for desired behaviors in Sect. 9.4.

9.3.1 Modeling Folds as Springs

The simplest model of an origami structure as a compliant body is the pseudo-rigid-body model [60] (Fig. 9.5a), where faces are approximated as rigid and folds as torsion springs. In this model, the compliance of the pattern comes exclusively from the small amount of material that is bending at the folds. Given a fold made of thin-sheet material with Young's modulus E , fold length ℓ , material thickness t , and fold width w , the bending stiffness of the fold can be written as:

$$k_f = E \frac{\ell t^3}{12w} \quad (9.7)$$

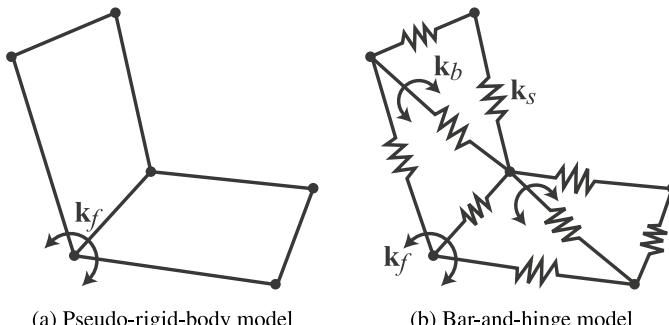


Fig. 9.5 Origami structures consist of faces and folds, which impose a natural discretization of the structure. Common models for these structures either **a** treat the structure as rigid faces connected at perfect spring-like hinges or **b** model facial deformation using a combination of linear and torsional springs for stretch, shear, and bending

and the bending moment in response to a fold angle θ as:

$$\tau_f = k_f \theta = E \frac{\ell r^3}{12w} \theta \quad (9.8)$$

The rest of the structure must obey the kinematic constraints of the equivalent rigid-body mechanism according to the equations in Sect. 9.2. These equations capture the behavior of complex, multifold joints such as those in [23]. By adding inertia [61] and actuation forces [62, 63], they also extend naturally to dynamical systems. At the same time, since the fold patterns are still required to satisfy theoretical kinematic constraints imposed by the folds, they are restricted to *rigid origami* models, that is, origami models whose faces experience negligible deformation during folding.

9.3.2 Modeling Face Deformation

Non-rigid origami patterns are those that require face deformation in order to fold and unfold. These patterns are often difficult for pseudo-rigid-body models because the kinematic constraints associated with the fold pattern geometry lock the structure. Instead, models that incorporate face deformation must be used. Predictions for compliance [52, 64, 65] and other properties [66, 67] can be computed using finite element simulations.

A simplified version of this idea is the bar-and-hinge model [54, 68, 69] (Fig. 9.5b), in which elements are at the scale of the faces in the fold pattern. In this model, all of the faces are triangulated, and each of the edges of the triangulated mesh is approximated as a linear spring, which captures the stretching and shearing of the sheet. The stiffnesses of the springs are functions of the material's Young's modulus E , the material thickness t , and the edge length ℓ as [69]:

$$k_s = \frac{EA_s t}{\ell} \quad (9.9)$$

The area A_s includes the portion of the face that belongs to this edge and can be approximated as

$$A_s = \frac{2A_f}{(1 - \nu) \sum_i \ell_i} \quad (9.10)$$

where A_f is the total area of the polygonal face that the edge bounds, ν is the material's Poisson's ratio, and $\sum_i \ell_i$ is the total length of all the linear springs in this face. Folds are modeled as torsion springs with an equilibrium at the desired fold angle and a stiffness similar to Eq. (9.7). In addition, because the faces are triangulated, additional torsion springs are included on the interiors of the faces to capture bending resistance within a face using the same model as the folds $k_b = k_f$.

To capture the full deformation of the fold pattern under load, the net spring force acting at each vertex of the fold pattern is computed, and the entire model is integrated forward in time until it reaches its equilibrium folded state. From an energy perspective, this is the state that minimizes spring potential energy.

9.4 Physical Computation and Control

The success of soft and semi-rigid robots is due in part to their ability to accommodate variable stiffness and non-uniform stiffness distributions throughout their physical bodies for tasks including versatile grasping [70, 71], manipulation [72, 73], impact resistance [74, 75], and error tolerance in navigation [76]. Recent work in origami mechanics has demonstrated that origami mechanisms can also be designed to produce desired soft behaviors, including anisotropic stiffness or heterogeneous stiffness distributions. Further, for origami structures, the mechanical response depends as much on geometry as on materials. These results reveal new opportunities for generating a desired behavior for a soft origami robot by tuning its geometric design.

9.4.1 Programmable Compliance

Origami models are combinations of flexible hinges and stiffer faces, and their compliance can be manipulated by changing the placement of these elements. To illustrate this idea, consider a thin-walled tube. This tube can be constructed as an origami pattern where two opposite sides of the paper are glued together. Sheets folded into hollow tubes in this way [44, 77] can act as nearly rigid struts, providing high strength and load-bearing capabilities in folded structures up meters in length [78]. Patterning these tubes with tessellations such as the Miura-ori pattern [79] enables them to collapse and expand. Further, by interleaving the resulting zigzag tubes in a “zipper” fashion, the tubes retain their high bending stiffness [52] with little added weight. The resulting structures provide anisotropic stiffness, producing structures that deploy in reliable and predictable ways with minimal resistance to actuation and without sacrificing payload capacity.

Dynamical tasks or tasks requiring structures to conform to contacts require greater control over the structure’s local stiffness. One approach to programming the stiffness of an origami structure is to use non-rigid origami design. Under the assumptions of Sect. 9.2.1, many origami structures are kinematically locked and have zero degrees of freedom, meaning that they are theoretically infinitely stiff. Practically, however, because the sheet material forming the origami structure can bend and stretch, the structures retain the ability to deform. Thus, non-rigid origami designs enable programmable compliance by controlling the amount by which the pattern would need to deform in order to achieve the desired configuration.

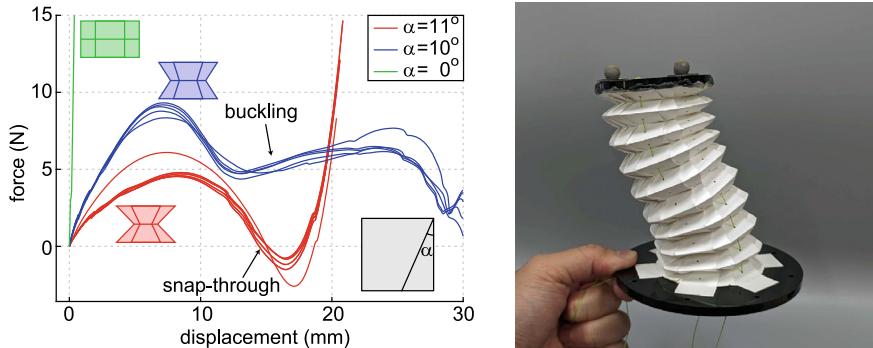


Fig. 9.6 Example of a bellows origami pattern with programmable stiffness. Left: Force vs. displacement of REBO bellows [80] folded from 0.13 mm thick PET with 16 sides, 1 segment, and an outer radius of 60 mm. Fold pattern unit is shown as a gray inset. Right: Photo of 12-sided bellows with 9 segments. The compliance of the bellows structure allows it to bend into an S-shape under tendon-driven actuation [51]

Consider, for example, the pattern in Fig. 9.6 [51, 80]. The pattern is a tessellation of straight and zigzag pleats that, when folded, produce a 3D multi-layered bellows. The columns on the left and right sides are glued together to form a ring. The main interesting feature here is that changing the individual geometric parameters in the tessellation units provides a wide range of control over the shape of the individual layers and the overall structure. For example, small changes to a parameter α can have large consequences on the structural compliance. Figure 9.6 shows measurements from a compression test on the bellows pattern folded from 0.13 mm thick PET, with 16 identical columns and an outer radius of 60 mm. Changing the parameter α_i by only 1° doubles the stiffness from 1.5 to 2.8 kN/m. This can be explained as a change in magnitude of face versus fold deformation effects. In general, within a particular layer, let $A = 2 \sum_i \alpha_i$ over all of the columns i . When $A = 360^\circ$, the folded state is a flat polygon with little resistance to axial forces and stiffness drops to essentially zero, producing a structure that can not even lift its own weight. This is because the flat folded configuration relies on torsional stiffness in the folds, and these stiffnesses are usually small. In contrast, when $A < 360^\circ$, the gluing constraint between the left-most and rightmost columns induces an angular defect in the layer, causing it to fold into a truncated pyramid. Compressing or expanding axially now requires stretching and bending deformation in the faces themselves, and resistance to these modes of deformation, depending on the geometry of the fold pattern, is high. When all $\alpha_i \rightarrow 0$, the folded structure reduces to a thin-walled polygonal cross-section beam, which maximizes stiffness at 44 kN/m. The bellows is thus able to span a continuous spectrum of compliance, from very soft structures (when $A \rightarrow 360^\circ$) to very stiff ones (when $A \rightarrow 0$), and reveals different mechanical instabilities, e.g., buckling and snap-through depending on the face geometry. Changing other parameters, such as the number of columns or the height of each layer further impacts the shape of the force-displacement curve, as well as nesting of multiple tubular origami shapes with

different parameters. These design parameters enable lightweight $\sim 50\text{ g}$ folded bellows to store and release enough energy to perform dynamical tasks such as repeated hopping in a kilogram-scale robot [81]. Bellows designs such as this one make up a large component of origami design. Popular patterns include the Yoshimura [82, 83], Kresling [47], magic ball [84], and other bellows patterns [85, 86], all of which are non-rigid origami structures but are able to produce deformations such as stretching and shrinking, bending, and twisting.

Finally, while most applications of origami in robotics use traditional straight-crease design, there is growing interest in how curved creases can also produce a wide variety of geometries and behaviors. Similarly to straight creases, curved creases can form deployable structures [16–18] or produce desired bending stiffness [87]. There are indications that curved creases may be able to achieve a similarly wide range of geometries, mechanics [14], and motions [14, 15] as straight creases, but with fewer creases and simpler designs. Since relatively few examples of robots using curved crease design exist, the potential for curved crease origami and curved structures in robotics remains to be determined.

9.4.2 Mechanical Logic

Our ability to precisely design the mechanical response of an origami structure creates new opportunities for specialized robots that exhibit physical intelligence by responding automatically to external forces through bodily change. In particular, origami structures that exhibit snap-through response produce binary signals and enable a new type of physical logic.

In mechanics, snap-through is a phenomenon by which a system can transition rapidly from one state to another. Bistable systems, for example, have two stable states. These systems reject small disturbances around their stable equilibria but have the ability to transition to the other equilibrium when a disturbance is large. The bistable grasper in [88] has two stable states—open and closed—that it transitions between. The snap-through instability is produced using a rotationally symmetric pattern consisting of an even number of creases folded in alternating mountain and valley folds. When the pattern is kept symmetric, [42] has shown that it can fold along two distinct trajectories in state space. Transitioning between these two trajectories requires the structure either to unfold back to its flat state or to stretch and bend in non-rigid ways. When the second strategy is applied, the structure resists stretching and bending in the neighborhood of the nominal undeformed trajectories, snapping to whichever trajectory is closer in the state space. The result is a gripper that automatically opens and closes in response to contact with other objects, without any required actuation or control. Similar ideas have been applied for deploying and collapsing manipulator arms [47, 89], triggering jumping [63, 85, 90], reconfiguring kinematics of a multi-joint hand [91, 92], detecting light contacts [93], and programming desired gait sequences [94].

When combined with each other, snap-through origami mechanisms enable structures capable of more complex logic and decision-making. For example, [95] showed that a Miura tessellation, when certain units are popped into different states, locally stiffens or softens. By selectively choosing which units to pop, a robot can produce different deformation modes and, for example, swimming gaits [14]. Instrumenting these patterns with actuators to switch modes in response to signals such as humidity or heat additionally allow the structures to compute on the resulting logical signals using physical implementations of logic gates [96, 97] and even nonvolatile memory [97].

9.5 Fabrication and Assembly of Robot Components

One of the great advantages of origami robots lies in their ability to embed complex functionalities in a relatively simple fabrication process. Origami robots are fabricated using planar manufacturing processes and folded into their 3D form. Unlike 3D manufacturing processes such as machining or 3D printing, planar manufacturing processes are relatively fast both at the prototyping stage (e.g., laser cutting, vinyl cutting) and at large scale (e.g., stamping, roll-to-roll).

9.5.1 Fabrication Processes

In their simplest form, origami robots are made by cutting and folding a sheet of passive material. Such structures have been folded out of plastic [98–100], metal [45, 101], and carbon fiber [102], among others. They are most often folded manually, and electronics are attached during or after folding to actuate the robot. When a thicker material is used, special care must be taken with designing the fold since excessive folding may cause the material to fracture. Hinge designs have been evaluated in [103], with different geometries excelling at stabilizing the structure under compressive, shear, or torsion loads.

Because of the difficulty of fabricating a robust hinge out of the same material as a stiff face, many origami robots are fabricated as laminate structures. By layering multiple materials and selectively removing material from particular layers, the local mechanical properties can be manipulated. For example, flexible and robust hinges may be created by layering a fabric with rigid metal faces, which enforce kinematics constraints of the fold pattern. This idea has been used successfully to create structures from the centimeter [104–106] to meter [107] scales.

A laminate fabrication approach brings additional advantages in that layers may serve not only a structural purpose but also a functional one. Figure 9.7 illustrates how these structures are formed. Passive layers are stacked with active layers (e.g., smart materials) and active circuitry. The layers are sealed together via intermediate adhesive layers or heat press. When gaps are cut in different layers to expose the

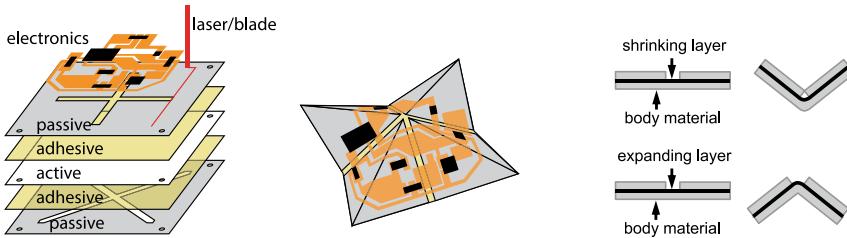


Fig. 9.7 Laminate fabrication process developed to attach detailed circuitry on folded structures. Layers are fused into a single electromechanical sheet. These layers may include electronics, passive plastic or metal substrates, or active materials such as shape-memory polymers, hydrogels, dielectric elastomer actuators, and more. Depending on the geometry, gaps are cut into the different layers to expose material and create hinges. Right: Cross-section of the laminate showing gaps in material layers and the resulting folding direction

actuation or sensing layers, the pattern can change its geometry in response to external or computed signals. The key is proper alignment between all of the layers to ensure that small gaps and traces are placed properly in the folded structure. Alignment is typically done using pins [108] or a fold [109, 110] in the stacked layers. This fabrication process is largely independent of the material used in the device. Other than the adhesive, which must adhere to the neighboring layers with sufficient strength, any passive or active material or electronic component can be included, meaning that with proper design, the fabrication approach produces fully integrated electromechanical devices with the ability to compute, sense, and act in response to environmental and task performance requirements.

9.5.2 Actuation

Actuators serve two primary purposes in an origami robot: (1) to fold the design from 2D into its 3D form, and (2) to move the robot after folding so that it can perform its task. Similarly to traditional robots, it is possible to add actuators onto the sheet after fabrication to actuate the design. For example, DC motors were used in [81, 98, 111] for the purpose of helping the origami robot to locomote after assembly. Piezoelectric actuators have also been recruited for legged locomotion in a similar manner [112, 113]. At the same time, however, by taking advantage of the planar print-and-fold fabrication process and embedding actuators directly into the flat sheet, it is possible to use the same actuators for self-assembly and for task performance, saving space and resulting in more efficient designs.

One approach to actuating an origami robot is to add an active material layer to the laminate sheet as shown in Fig. 9.7. The right subfigure shows a cross-section of a laminate sheet with one layer of active material sandwiched between two layers of passive material. Gaps on one of the passive layers exposes the active material.

When the middle layer activates, the asymmetry in the cross-section forces the sheet to bend at the gap. Shrinkage in the middle layer causes the sheet to fold in the direction of the gap, while expansion of the middle layer causes it to fold in the opposite direction. *Self-folding* origami structures using this idea have been made to respond to a variety of stimuli [114, 115], including heat [59], humidity [116], chemical signals [117, 118] and light [119].

Thermal Actuation

The most common signal used for self-assembly and control of an origami robot is heat. Shape memory alloys (SMA) and shape memory polymers (SMP) are materials that can be programmed to remember a particular shape. During robot fabrication, the material is deformed out of its preprogrammed shape and placed on the origami sheet. When exposed to heat, it then returns to its remembered state, deforming the part of the sheet to which it has been attached.

Self-assembling robots that self-fold at high temperatures generally make use of SMP layers embedded into a trilayer laminate. Materials such as polystyrene (PSPS) [120–122], polyvinyl chloride (PVC) [109, 123], and others [124] exhibit non-reversible shrinking in the presence of heat, allowing the origami structure to fold into the desired shape in one step through global heating. For these designs, care must be taken to ensure that the shape of the final folded structure is correct independent of the folding sequence. When the precise sequence of folding is important, multiple polymers that respond at different temperatures can be embedded into the laminate instead [125]. As the structure is heated slowly up from room temperature, the low-temperature folds activate first, followed by the high-temperature folds.

For more precise control, SMA layers combined with resistive heating offer the ability to fold and unfold using electrical signals. A popular SMA is a nickel titanium (NiTi) alloy. NiTi actuators operate through a phase transition where the crystal structure of the material changes depending on its temperature and stress. At low temperatures, the material is in a low-stiffness martensite state that can easily be plastically deformed. At high temperatures, the material transitions into an austenite phase, which is approximately twice as stiff and much harder to plastically deform. Embedding SMA bending actuators into the folds of an origami structure allow it to fold for self-assembly [126], locomotion [127], or a combination of both [128]. Since SMA actuators are bidirectional [129] and electrically conductive, they can be activated and controlled electrically through resistive heating, producing precise and controlled folding sequences that can be used to create fully reconfigurable structures that can fold into multiple different structures [59]. However, because these actuators rely on cyclic heating and cooling, they are often slow, with actuation frequency less than 3 Hz [115].

Fluidic Actuation

Fluids such as water or air are inherently soft and provide natural actuation strategies for folded structures. At very small scales, capillary forces [130] can be used to attach a drop of fluid to the origami structure, manipulate the faces into a folded state, and

glue edges to each other to produce closed shapes with high reliability. At larger scales, hydrogels can be embedded directly into an origami laminate as the active layer. Hydrogels are polymers that swell when wet and shrink when dried. Patterning the laminate with hydrogels at the folds causes the structure to fold when exposed to water [131, 132]. Fluid absorption can be controlled using an electric field [116] for reversible actuation.

Fluids can also be pumped internally through the folded structure for faster and more controllable actuation. For example, plastic layers on the laminate sheet can be fused to create pneumatic channels, with pouches located at particular folds. For these “pouch motors” [133, 134], pressurizing the pouch causes it to shrink in the direction perpendicular to the fold. By designing the channels appropriately, air can be routed to sets of folds for sequential actuation. Taking this idea to the extreme, [135] has shown that enclosing an entire pattern in a pouch and applying a vacuum results in simultaneous folding of all folds in the origami structure within seconds. Further, the vacuum adds support to the origami structure, stiffening it and helping it to withstand external loads. This idea has been applied to the magic ball tessellation to create a gripper that conforms to and maintains a grasp on an irregular object [136].

Magnetic Actuation

Magnetic forces are a convenient method for folding and actuating origami robots without requiring direct contact between the actuator and the robot itself. This approach is useful for small-scale applications where the robot is unable to carry its own power and computation. In its simplest form, magnetic actuation consists of attaching a permanent magnet to the robot and applying an external magnetic field. The resulting torques on the magnet force the robot to rotate or bend in the desired direction. These forces can be used to control robot locomotion [109, 123] or to selectively fold and deploy parts of a multi-part fold pattern [47, 137]. Permanent magnets and electromagnetic coils embedded directly into a laminate sheet produce low-profile linear actuators [138, 139] that can also be used to selectively lock particular joints [140].

Integrating a magnetic actuation approach directly into a folded structure involves programming magnetic dipoles into the origami sheet. In most cases, this requires precise placement of units with known magnetic polarity. Since programming a part of a structure with an arbitrary magnetic profile can be difficult, a common strategy is to program small magnetic units using a uniform magnetic field and then physically combine them to create a more complex profile [141–143]. Once the sheet is assembled, the magnetic response can be changed by physically reorienting magnetic particles [144] or by suppressing magnetic response with another material, for example a light-responsive shape-memory polymer [145].

9.5.3 Sensing

Similarly to actuation, sensing can be incorporated into an origami robot by layering sensors into the laminate sheet. The most important sensing signal for an origami robot is the fold angle of active folds. One method for obtaining these signals is by adding resistive surfaces, for example by adding a carbon-ink layer that produces a resistance signal that changes with strain [128, 146, 147], or embedding liquid-metal filled channels into the structure [148]. Capacitive signals can also be used. Since laminate sheets are parallel plate structures, adding electrically conductive material into the layers produces structures whose capacitance changes with bend angle [149, 150]. These ideas also extend beyond single folds. Folding entire origami patterns from conductive materials produces folded sensors that produce resistance, capacitance, or inductance changes in response to large-scale deformation in the pattern [151].

9.6 Applications

Origami robots have been successfully applied in a variety of applications and settings. The ease of assembly and high customizability of origami machines lowers barriers to prototyping and demonstration. In many cases, full robots can be constructed and tested within a few hours or days, enabling designers to engage in a rapid prototyping and design cycle. Below, we outline some of the use cases for origami robot designs.

Reconfiguration and Deployables

It should come as no surprise that one of the most popular applications of origami robotics is in developables and reconfigurable robots. Folding naturally produces structures with the ability to change their geometries, expanding, collapsing, and folding flat.

Folding for shape transformation is most commonly used for fabrication and assembly of deployable structures. Folds are embedded into the robot itself [109, 121, 123, 126, 152] or into support structures that fold the robot into form using pop-up like design techniques [108, 153]. By tuning the mechanics of the origami pattern, these structures can expand and contract under active control and then passively maintain their shape under load [47, 78, 89], thus reducing volume required for storage and transportation of the robot without sacrificing functional stiffness. For greater versatility, origami robots can be fabricated as individual connected modules [154–159]. For example, [156] demonstrates a modular system consisting of triangular tiles connected at the hinges by DC motors. The tiles snap together to form fully actuated, 3D folding structures, with all the same kinematic constraints as a regular triangular mesh. Similar approaches in [22, 100] connect multiple folded modules together to form more complex reconfigurable structures.

Post-assembly, continuing to manipulate the fold angles morphs the robot into different forms. It has been shown that universal origami tessellations exist, that is, that there are origami patterns that can fold into any shape subject to the fold pattern resolution [59]. Thus, the complexity of a robot's transformation is limited only by fabrication and assembly constraints. Functional patterns can be as simple as a four folds [15] or can take the form of full tessellations with hundreds of folds. Shape changing patterns such as the magic ball tessellation enable robots to change their dimensions, for example, trading off large wheels that are better for fast travel against small wheels that are better for traction and balance [101, 107]. Collapsible designs that flatten the robot's body are also useful for moving the robot through tight spaces [160] or stacking them for transportation. Aside from geometric effects, shape change is often accompanied by a change in the robot's behavior. In [161], for example, a bistable pattern is used to collapse a quadrotor's arm when impacts a hard surface. The pattern serves dual purpose here in that (1) geometric contraction of the quadrotor arms provides the vehicle with more space to maneuver, and (2) the impact energy is absorbed directly into the arm's fold pattern rather than transferred to the rest of the vehicle, providing the vehicle with greater passive stability. The bistable waterbomb pattern in [162] serves a similar purpose, switching between its two stable states to transition between tumbling and stable descent through the air.

Manipulation

Origami robots are inherently compliant, and thus origami structures have also been proposed as designs for soft manipulators and hands. While it is possible to model and design an origami manipulator as a rigid serial kinematic chain [38], practically, bending in the folds and the folded sheet enable the origami structure to deform, providing the compliance of the more popular elastomeric manipulator but with the light weight of a thin shell. Origami gripper designs take advantage of this compliance to produce stable, adaptive grasps with relatively simple fold patterns. Graspers in [88] and [163], for example, use the kinematics of the degree-4 vertex to fold their fingers inward during grasping while also relying on material bending to conform. For manipulator arms, which require larger scale fold patterns, a number of designs have been proposed for serially composable compliant joints with rotational [38, 43], translational [51, 89], twisting [23], and bending [82] degrees of freedom. To approximate continuum manipulators, origami tessellations patterned along the length of a thin shell tube produce continuous bending. Common patterns for producing this type of deformation include the twisted tower [164], the Kresling pattern [47, 165], and the Yoshimura [82].

Locomotion

Folding as a fabrication approach for lightweight, millimeter- to centimeter-scale robots has given rise to a number of mobile platforms spanning ground, aerial, and aquatic vehicles. Some of the earliest origami robots were legged crawlers. In 2008, [106, 166] showed how a process called smart composite microstructures (another method for forming multi-material laminates) could be used to fabricate an origami

hexapod at the centimeter-scale. The robot combined a stiff composite as the rigid faces with flexible polymer hinges and included folded four bar linkages to move the legs in a desired gait. Since this work, laminate fabrication approaches have been advanced and simplified to address a broad range of origami designs. Legged robots such as [98, 113, 126, 167–169] use a similar strategy of forming four-bar or other canonical linkages to generate cyclical leg motions for robot gaits, but origami-inspired approaches to robotic locomotion are not restricted to these types. Tubular tessellations have also found a place in ground locomotion, where embedding actuators to cyclically compress and extend segments on the tube allow these designs to inch forward similarly to biological worms [28, 170–172] or jump using the tubular origami structures as a spring [81]. Beyond ground locomotion, origami structures have also been used to construct frames for flying robots [173, 174] and compliant bodies for swimming robots [14, 84].

Human Interaction

Since origami structures are naturally soft, they are a strong candidate for new interfaces for human-robot and human-computer interaction. The Ori-Pixel module [175] uses a simplified waterbomb pattern to produce combined translation and bending. When three such links are connected in parallel to a tile, actuators on each link control the 3D position and orientation of the tile. Arrays of these Ori-Pixel modules yield a reconfigurable 3D display. By incorporating force sensing and feedback to respond to human touch, the module can also provide real-time haptic feedback to simulate physical interactions with a virtual environment [176]. Since origami is lightweight, these devices can be attached directly to the body as a wearable [177]. The same advantages have also been applied on the face for non-intrusive assistive facial rehabilitation [178] or on the trunk as limb support [179]. Inside the human body, medical devices are also a natural application domain for origami robots since origami provides the robot with the ability to collapse into a small form and then deploy under remote control in the body. In this space, multiple small-scale designs have been demonstrated for such purposes as drug delivery [180], navigation [181], and in-body manipulation [182, 183].

9.7 Summary and Future Outlook

In the field of soft robotics, robotic origami has ushered in an era of rapid prototyping and innovation, culminating in the creation of intelligent robots and machines constructed from thin sheets of functional materials. Origami provides unique opportunities for designing complex and customizable behaviors for multifunctional machines that are lightweight, strong, and simple to fabricate and assemble. Origami robots benefit from a fabrication process where sensing, actuation, and computation are directly integrated into the folded body of the robot for fully embodied intelligence and control. Over the last several decades, we have seen a large growth in origami

design, demonstrating that these robots are capable of a wide variety of tasks ranging from rapid deployment and reconfiguration to compliant manipulation to high-energy locomotion. Looking to the future, we can identify several areas of research that will push origami robots to heretofore unseen abilities and performance.

In the realm of modeling, numerous origami designs have been discovered with a great many capabilities. However, the majority of these models and designs have focused on straight crease design since the kinematic analysis is a relatively straightforward extension of traditional rigid-body kinematics. Curved crease design offers new opportunities to produce complex, natural-looking geometries and motions using simple crease patterns with a minimal number of folds. Further, curved surfaces naturally admit bending and twisting degrees of freedom that have so far not been explored in origami robots and could generate new functionality in applications such as crawling [15] or mechanical reconfiguration [14]. Understanding how to fabricate, actuate, and sense on curved surfaces is also relatively unexplored and introduces new opportunities to bridge the gap between soft origami robots and more standard elastomeric soft robots.

In the space of actuation and control, origami robots have shown large benefits in strength and load-bearing. As an origami robot scales, its weight increases as the length squared, compared to the cubic scaling typical of 3D structures. As a result, embedding sheet actuators onto origami designs has been able to produce robots with greater specific work capacities and costs of transport [98] than their 3D counterparts. However, there is still a gap in material actuators capable of high power or bandwidth. The laminate fabrication process often used for origami robots is advantageous for embedding material actuators in an distributed fashion throughout a robot's body. Further advancements on sheet or filamentous actuators with high power densities that are able to actuate folds at high frequencies will enable future origami robots to perform dynamical tasks and to take greater advantage of their reconfigurability to adapt, react, and respond to task changes as needed in real-time.

Zooming out, origami robots poses large challenges in design. The innate multimaterial-based design freedom of origami robots endows them with unparalleled functional versatility. However, this very versatility brings forth a unique set of challenges. The parameters that define the design of these origami robots transcend conventional norms. Their mechanical layout and fabrication require attention to their 2D and 3D geometries, the integration of compliant mechanisms, the incorporation of functional-material-specific components, and the utilization of advanced 2D fabrication methods. In contrast to the linear and coherent design process that characterizes conventional robots, the construction of origami robots mandates a comprehensive understanding of their multifaceted dimensions. The interplay of 3D and 2D geometries, coupled with compliant mechanisms and specific functional materials, underscores the complexity of this robotic design approach.

In order to design a complete origami robot system, a holistic approach to designing each of these components is needed [46]. Since origami robots are folded machines, designing these structures requires models for how 2D sheets fold in 3D space, kinematics of folded linkages, and mechanical responses for thin-shell structures. These models depend on the fabrication method used to instantiate the

origami design and inform the mechanical design of the folded structure. Mechanical design is typically performed over the geometry and motion of the robot but requires an understanding of the control strategy for the robot. Since origami robots are inherently soft, certain behaviors (compliance to contact, robust deployment, rejection of noise) can be implemented either through actuation or by tuning the robot's mechanical response. The choice of which behaviors to embed into the robot's body and which to actively control requires navigating fabrication tradeoffs in how complex certain mechanical behaviors may make the fold pattern and where and how well actuation and sensing can be integrated into the physical body. Thus, it is clear that origami robots pose a complex co-design problem. Since the entire robot is fabricated directly as a folded multimaterial sheet, it is often impossible to decouple different robot subcomponents from each other. New approaches to systematic design for origami robots that maps their required tasks to their bodily structure will revolutionize the ability for origami robotics to produce multifunction, customizable, robust, and capable robots to support the complex needs for humans and machines.

Acknowledgements The work in this chapter was supported in part by US National Science Foundation (NSF) under Grant No. 2322898 and by the Army Research Office (ARO) under MURI award #W911NF1810327.

References

1. Rus D, Tolley MT (2018) Design, fabrication and control of origami robots. *Nat Rev Mater* 3(6):101–112
2. Rus D, Sung S (2018) Spotlight on origami robots. *Sci Robot* 3(15)
3. Bezzo N, Mehta A, Onal CD, Tolley MT (2015) Robot makers: the future of digital rapid design and fabrication of robots. *IEEE Robot Autom Mag* 22(4):27–36
4. Waldron KJ, Schmiedeler J (2016) Kinematics. In: Siciliano B, Khatib O (eds) Springer handbook of robotics, chapter 2, pp 11–36
5. Balkcom DJ, Mason MT (2008) Robotic origami folding. *Int J Robot Res* 27(5):613–627
6. Bowen LA, Grames CL, Magleby SP, Howell LL, Lang RJ (2013) A classification of action origami as systems of spherical mechanisms. *ASME J Mech Des* 135(11)
7. Foschi R, Hull TC, Ku JS (2022) Explicit kinematic equations for degree-4 rigid origami vertices. Euclidean and non-Euclidean. *Phys Rev E* 106(5):055001
8. Lang RJ (2019) Twists, tilings, and tessellations. A K Peters/CRC Press
9. Koryo M (1969) Proposition of pseudo-cylindrical concave polyhedral shells. *Inst Space Aeronaut Sci Univ Tokyo* 34(9):141–163
10. Miura K, Lang RJ (2009) The science of miura-ori: a review. *Origami* 4(3653):87
11. Evans TA, Lang RL, Magleby SP, Howell LL (2015) Rigidly foldable origami gadgets and tessellations. *Roy Soc Open Sci* 2(9):150067
12. Dudte LH, Vouga E, Tachi T, Mahadevan L (2016) Programming curvature using origami tessellations. *Nat Mater* 15(5):583–588
13. Dmitry F, Serge T (1999) More on paperfolding. *Am Math Monthly* 106(1):27–35
14. Zirui Z, Yong W, Ken L, Wu L, Hanqing J (2020) In situ stiffness manipulation using elegant curved origami. *Sci Adv* 6(47):1–10
15. Feshbach D, Wu X, Vasireddy S, Beardell L, To B, Baryshnikov Y, Sung C (2023) Curvequad: a centimeter-scale origami quadruped that leverages curved creases to self-fold and crawl with one motor. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)

16. Tomohiro T (2013) Composite rigid-foldable curved origami structure. *Transformables* 1:1–6
17. Lee N, Pellegrino S (2014) Self-supporting membrane structures with curved creases for smooth packaging and deployment. In: Spacecraft structures conference, p 1037
18. Martin K, Aron M, Mitra NJ (2017) String actuated curved folded surfaces. *ACM Trans Graph* 36(4):1
19. Huffman DA (1976) Curvature and creases: a primer on paper. *IEEE Trans Comput C-25*(10):1010–1019
20. Duncan JP, Duncan JL (1982) Folded developables. *Proc Roy Soc Lond A Math Phys Sci* 383(1784):191–205
21. Rabinovich M, Hoffmann T, Sorkine-Hornung O (2019) Modeling curved folding with freeform deformations. *ACM Trans Graph* 38(6)
22. Overvelde JTB, Weaver JC, Hoberman C, Bertoldi K (2017) Rational design of reconfigurable prismatic architected materials. *Nature* 541(7637):347–352
23. Cynthia S, Daniela R (2015) Foldable joints for foldable robots. *ASME J Mech Robot* 7(2):021012
24. Hongying Z, Jamie P (2022) Kirigami design and modeling for strong, lightweight metamaterials. *Adv Funct Mater* 32(21):2107401
25. Yaoye H, Yinding C, Wu S, Yanbin L, Yong Z, Jie Y (2022) Boundary curvature guided programmable shape-morphing kirigami sheets. *Nat Commun* 13(1):530
26. Amir F, Tatsuya H, Keisuke N, Kyujin C, Jamie P (2020) Stretchable kirigami components for composite meso-scale robots. *IEEE Robot Autom Lett* 5(2):1883–1890
27. Li J, Ran R, Wang H, Wang Y, Chen Y, Niu S, Arratia PE, Yang S (2021) Aerodynamics-assisted, efficient and scalable kirigami fog collectors. *Nat Commun* 12(1):5484
28. Rafsanjani A, Zhang Y, Liu B, Rubinstein SM, Bertoldi K (2018) Kirigami skins make a simple soft actuator crawl. *Sci Robot* 3(15):eaar7555
29. Demaine ED, Tachi T (2017) Origamizer: a practical algorithm for folding any polyhedron. In: International symposium on computational geometry (SoCG)
30. Lang RJ (1996) A computational algorithm for origami design. In: ACM annual symposium on computational geometry, pp 98–105
31. Benbernou NM, Demaine ED, Demaine ML, Ovadya A (2011) Universal hinge patterns to fold orthogonal shapes. In: Origami 5: fifth international meeting of origami science mathematics and education, pp 405–419
32. Mitani J (2009) A design method for 3D origami based on rotational sweep. *Comput-Aided Des Appl* 6(1)
33. Belcastro S, Hull TC (2002) Modelling the folding of paper into three dimensions using affine transformations. *Linear Algebra Appl* 348(1):273–282
34. Demaine ED, O'Rourke J (2007) Geometric folding algorithms: linkages, origami, polyhedra. Cambridge University Press
35. Mitani J (2011) A design method for axisymmetric curved origami with triangular prism protrusions. In: Origami 5: proceedings of the 5th international meeting of origami science, mathematics, and education, pp 437–447
36. Cheng H, Cheong KH (2012) Designing crease patterns for polyhedra by composing right frusta. *Comput-Aided Des* 44(4):331–342
37. Sung C, Demaine ED, Demaine ML, Rus D (2013) Edge-compositions of 3D surfaces. *ASME J Mech Des* 135(11)
38. Wei-Hsi C, Woohyeok Y, Lucien P, Koditschek DE, Sung CR (2023) Kinogami: algorithmic design of compliant kinematic chains from tubular origami. *IEEE Trans Robot* 39(2):1260–1280
39. Yellowhorse A, Howell LL (2016) Creating rigid foldability to enable mobility of origami-inspired mechanisms. *ASME J Mech Robot* 8(1):011011
40. Sussman DM, Cho Y, Castle T, Gong X, Jung E, Yang S, Kamien RD (2015) Algorithmic lattice kirigami: a route to pluripotent materials. *Proc Natl Acad Sci* 112(24):7449–7453
41. Yang S, Choi I, Kamien RD (2016) Design of super-conformable, foldable materials via fractal cuts and lattice kirigami. *MRS Bull* 41(2):130–138

42. Tachi T, Hull TC (2017) Self-foldability of rigid origami. *ASME J Mech Robot* 9(2):021008
43. Mehta AM, Rus D (2014) An end-to-end system for designing mechanical structures for print-and-fold robots. In: IEEE international conference on robotics and automation (ICRA)
44. Faal SG, Chen F, Tao W, Agheli M, Tasdighikalat S, Onal CD (2016) Hierarchical kinematic design of foldable hexapedal locomotion platforms. *ASME J Mech Robot* 8(1)
45. Adriana S, Cynthia S, Andrew S, Wei Z, Robin C, Eitan G, Daniela R, Wojciech M (2017) Interactive robogami: an end-to-end system for design of robots with ground locomotion. *Int J Robot Res* 36(10):1131–1147
46. Zhenishbek Z, Jamie P (2018) Design methodology for constructing multimaterial origami robots and machines. *IEEE Trans Robot* 34(1):151–165
47. Wu S, Ze Q, Dai J, Udupi N, Paulino GH, Zhao R (2021) Stretchable origami robotic arm with omnidirectional bending and twisting. *Proc Natl Acad Sci* 118(36):e2110023118
48. Amir F, Jamie P (2017) An under-actuated origami gripper with adjustable stiffness joints for multiple grasp modes. *Smart Mater Struct* 26(5):055035
49. Santoso J, Skorina EH, Luo M, Yan R, Onal CD (2017) Design and analysis of an origami continuum manipulation module with torsional strength. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 2098–2104
50. Hiromi Y, Tomohiro T, Mia L, Jinkyu Y (2017) Origami-based tunable truss structures for non-volatile mechanical memory operation. *Nat Commun* 8(1):1–7
51. Chen WH, Misra S, Gao Y, Lee YJ, Koditschek DE, Yang S, Sung CR (2020) A programmably compliant origami mechanism for dynamically dexterous robots. *IEEE Robot Autom Lett* 5(2):2131–2137
52. Filipov ET, Tachi T, Paulino GH (2015) Origami tubes assembled into stiff, yet reconfigurable structures and metamaterials. *Proc Natl Acad Sci* 112(40)
53. Qiu C, Aminzadeh V, Dai JS (2013) Kinematic analysis and stiffness validation of origami cartons. *ASME J Mech Des* 135(11):111004
54. Ghassaei A, Demaine ED, Gershenfeld N (2018) Fast, interactive origami simulation using gpu computation. *Origami* 7:1151–1166
55. Gray S, Zeichner N, Yim M, Kumar V (2011) A simulator for origami-inspired self-reconfigurable robots. In: Origami 5: proceedings of the 5th international conference of origami science, mathematics, and education, pp 323–333
56. Ida T, Takahashi H, Marin M, Kasem A, Ghourabi F (2009) Computational origami system Eos. *Origami 4: proceedings of the 4th international meeting of origami science, mathematics, and education*, pp 285–293
57. Suyi L, Hongbin F, Sahand S, Priyanka B, Kon-Well W (2019) Architected origami materials: how folding creates sophisticated mechanical properties. *Adv Mater* 31(5):1805282
58. Onal CD, Wood RJ, Daniela R (2013) An origami-inspired approach to worm robots. *IEEE/ASME Trans Mechatronics* 18(2):430–438
59. Hawkes E, An B, Benbernou NM, Tanaka H, Kim S, Demaine ED, Rus D, Wood RJ (2010) Programmable matter by folding. *Proc Natl Acad Sci* 107(28):12441–12445
60. Larry L (2001) *Howell. Compliant mechanisms*. John Wiley & Sons, New York, NY, USA
61. Mete M, Huang J-L, Paik J (2023) Dynamic modeling of an origami prismatic joint. In: IEEE international conference on soft robotics (RoboSoft), pp 1–6
62. Zhenishbek Z, Kazuaki M, Koh H, Jamie P (2019) Designing minimal and scalable insect-inspired multi-locomotion millirobots. *Nature* 571(7765):381–386
63. Zhakypov Z, Belke CH, Paik J (2017) Tribot: a deployable, self-righting and multi-locomotive origami robot. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5580–5586
64. Silverberg JL, Na J-H, Evans AA, Liu B, Hull TC, Santangelo CD, Lang RJ, Hayward RC, Cohen I (2015) Origami structures with a critical transition to bistability arising from hidden degrees of freedom. *Nat Mater* 14(4):389–393
65. Faber JA, Arrieta AF, Studart AR (2018) Bioinspired spring origami. *Science* 359(6382):1386–1391

66. Katia B, Vincenzo V, Johan C, van Martin H (2017) Flexible mechanical metamaterials. *Nat Rev Mater* 2(11):17066
67. Rafsanjani A, Bertoldi K, Studart AR (2019) Programming soft robots with flexible mechanical metamaterials. *Sci Roboti* 4(29):eaav7874
68. Schenk M, Guest SD et al (2011) Origami folding: a structural engineering approach. *Origami* 5:291–304
69. Liu K, Paulino GH (2018) Highly efficient nonlinear structural analysis of origami assemblages using the merlin2 software. *Origami* 7:1167–1182
70. Dollar AM, Howe RD (2010) The highly adaptive SDM hand: design and performance evaluation. *Int J Robot Res* 29(5):585–597
71. Martinez RV, Branch JL, Fish CR, Jin L, Shepherd RF, Nunes R, Suo Z, Whitesides GM (2013) Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Adv Mater* 25(2):205–212
72. Daniela R, Tolley MT (2015) Design, fabrication and control of soft robots. *Nature* 521(7553):467–475
73. Lee C, Kim M, Kim YJ, Hong N, Ryu S, Kim HJ, Kim S (2017) Soft robot review. *Int J Control, Autom Syst* 15(1):3–15
74. Shepherd RF, Ilievski F, Choi W, Morin SA, Stokes AA, Mazzeo AD, Chen X, Wang M, Whitesides GM (2011) Multigait soft robot. *Proc Natl Acad Sci* 108(51):20400–20403
75. Tolley MT, Shepherd RF, Mosadegh B, Galloway KC, Wehner M, Karpelson M, Wood RJ, Whitesides GM (2014) A resilient, untethered soft robot. *Soft Robot* 1(3):213–223
76. Hawkes EW, Blumenschein LH, Greer JD, Okamura AM (2017) A soft robot that navigates its environment through growth. *Sci Robot* 2(8):eaan3028
77. Liu C, Orlofsky A, Kitcher CD, Felton SM (2019) A self-folding pneumatic piston for mechanically robust origami robots. *IEEE Robot Autom Lett* 4(2):1372–1378
78. Melancon D, Gorissen B, García-Mora CJ, Hoberman C, Bertoldi K (2021) Multistable inflatable origami structures at the metre scale. *Nature* 592(7855):545–550
79. Cheung KC, Tachi T, Calisch S, Miura K (2014) Origami interleaved tube cellular materials. *Smart Mater Struct* 23(9):094012
80. Yuan H, Pikul J, Sung C (2018) Programmable 3-D surfaces using origami tessellations. In: 7th international meeting on origami in science, mathematics, and education (OSME), vol 3, pp 893–906
81. Chen W-H, Misra S, Caporale JD, Koditschek DE, Yang S, Sung CR (2020) A tendon-driven origami hopper triggered by proprioceptive contact detection. In: IEEE international conference on soft robotics (RoboSoft), pp 373–380
82. Santoso J, Onal CD (2021) An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics. *Soft Robot* 8(4):371–386
83. Laura P, Gunjan A, Jamie P (2016) Design and analysis of a soft pneumatic actuator with origami shell reinforcement. *Soft Robot* 3(3):109–119
84. Zhiyuan Y, Dongsheng C, Levine DJ, Sung C (2021) Origami-inspired robot that swims via jet propulsion. *IEEE Robot Autom Lett* 6(4):7145–7152
85. Carlson J, Friedman J, Kim C, Sung C (2020) REBOund: untethered origami jumping robot with controllable jump height. In: IEEE international conference on robotics and automation (ICRA), pp 10089–10095
86. Evandro B, Stéphane V (2022) Design of an origami bendy straw for robotic multistable structures. *ASME J Mech Des* 144(3):033301
87. Woodruff SR, Filipov ET (2021) Curved creases redistribute global bending stiffness in corrugations: theory and experimentation. *Meccanica* 56(6):1613–1634
88. Yasuda H, Johnson K, Arroyos V, Yamaguchi K, Raney JR, Yang J (2022) Leaf-like origami with bistability for self-adaptive grasping motions. *Soft Robot* 9(5):938–947
89. Kim SJ, Lee D-Y, Jung G-P, Cho K-J (2018) An origami-inspired, self-locking robotic arm that can be folded flat. *Sci Robot* 3(16)
90. Sun Y, Wang J, Sung C (2022) Repeated jumping with the REBOund: self-righting jumping robot leveraging bistable origami-inspired design. In: International conference on robotics and automation (ICRA), pp 7189–7195

91. Liu C, Felton SM (2018) Transformation dynamics in origami. *Phys Rev Lett* 121(25):254101
92. Zuliani F, Liu C, Paik J, Felton SM (2018) Minimally actuated transformation of origami machines. *IEEE Robot Autom Lett* 3(3):1426–1433
93. Jiang Y, Li Y, Liu K, Zhang H, Tong X, Chen D, Wang L, Paik J (2023) Ultra-tunable bistable structures for universal robotic applications. *Cell Rep Phys Sci* 4(5), 2023
94. Novelino LS, Ze Q, Wu S, Paulino GH, Zhao R (2020) Untethered control of functional origami microrobots with distributed actuation. *Proc Natl Acad Sci* 117(39):24096–24101
95. Silverberg JL, Evans AA, McLeod L, Hayward RC, Hull T, Santangelo CD, Cohen I (2014) Using origami design principles to fold reprogrammable mechanical metamaterials. *Science* 345(6197):647–650
96. Benjamin T, Andrew G, Philip B, Richard V (2018) Origami mechanologic. *Proc Natl Acad Sci* 115(27):6916–6921
97. Wenzhong Y, Shuguang L, Mauricio D, Zhaoliang Z, Daniela R, Ankur M (2023) Origami-based integration of robots that sense, decide, and respond. *Nat Commun* 14(1):1553
98. Onal CD, Tolley MT, Wood RJ, Rus D (2014) Origami-inspired printed robots. *IEEE/ASME Trans. Mechatronics* 20(5):2214–2221
99. Cynthia S, Daniela R (2015) Automated fabrication of foldable robots using thick materials. *Int Symp Robot Res (Springer Proc Adv Robot)* 2:253–266
100. Gao W, Ramani K, Cipra RJ, Siegmund T (2013) Kinetogami: a reconfigurable, combinatorial, and printable sheet folding. *ASME J Mech Des* 135(11):111009
101. Lee D-Y, Kim J-K, Sohn C-Y, Heo J-M, Cho K-J (2021) High-load capacity origami transformable wheel. *Sci Robot* 6(53):eabe0201
102. Sainaghi P, Plecnik M (2022) Methods for folding linkages out of carbon fiber. In: USCToMM symposium on mechanical systems and robotics, pp 100–115
103. Delimont IL, Magleby SP, Howell LL (2015) Evaluating compliant hinge geometries for origami-inspired mechanisms. *ASME J Mech Robot* 7(1):011009
104. Wujoon C, Campbell MF, Popov GA, Stanczak CH, Estep AK, Steager EB, Sung CR, Yim MH, Igor B (2020) Microfabricated foldable wings for centimeter-scale microflyers. *J Microelectromech Syst* 29(5):1127–1129
105. Wujoon C, Luke K, Campbell MF, Calenza TJ, Popov GA, Jeremy W, Sung CR, Mark Y, Igor B (2021) Carbon fiber-aluminum sandwich for micro-aerial vehicles and miniature robots. *MRS Adv* 6:477–481
106. Hoover AM, Steltz E (2008) Fearing Rs. RoACH: an autonomous 2.4 g crawling hexapod robot. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 26–33
107. Dae-Young L, Sa-Reum K, Ji-Suk K, Jae-Jun P, Kyu-Jin C (2017) Origami wheel transformer: a variable-diameter wheel drive robot using an origami structure. *Soft Robot* 4(2):163–180
108. Sreetharan PS, Whitney JP, Strauss MD, Wood RJ (2012) Monolithic fabrication of millimeter-scale machines. *J Micromech Microeng* 22(5):055027
109. Miyashita S, Guitron S, Ludersdorfer M, Sung CR, Rus D (2015) An untethered miniature origami robot that self-folds, walks, swims, and degrades. In: IEEE international conference on robotics and automation (ICRA), pp 1490–1496
110. Addison L, Mykell J, Sung C (2022) Increasing reliability of self-folding of the origami hyper. *ASME J Mech Robot* 14(6):061003
111. Nima M, Alihan B, Onur Ö (2020) Miniature modular legged robot with compliant backbones. *IEEE Robot Autom Lett* 5(3):3923–3930
112. Jayaram K, Shum J, Castellanos S, Helbling EF, Wood RJ (2020) Scaling down an insect-size microrobot. HAMR-VI into HAMR-Jr. In, IEEE international conference on robotics and automation (ICRA), pp 10305–10311
113. Goldberg B, Zufferey R, Doshi N, Helbling EF, Whittredge G, Kovac M, Wood RJ (2018) Power and control autonomy for high-speed locomotion with an insect-scale legged robot. *IEEE Robot Autom Lett* 3(2):987–993
114. Leanza S, Wu S, Sun X, Qi HJ, Zhao RR (2023) Active materials for functional origami. *Adv Mater* 2302066

115. Zhang J, Sheng J, O'Neill CT, Walsh CJ, Wood RJ, Ryu JH, Desai JP, Yip MC (2019) Robotic artificial muscles: current progress and future perspectives. *IEEE Trans Robot* 35(3):761–781
116. Morales D, Palleau E, Dickey MD, Velez OD (2014) Electro-actuated hydrogel walkers with dual responsive legs. *Soft Matter* 10(9):1337–1348
117. Leong TG, Randall CL, Benson BR, Bassik N, Stern GM, Gracias GH (2009) Tetherless thermobiochemically actuated microgrippers. *Proc Natl Acad Sci* 106(3):703–708
118. Miskin MZ, Cortese AJ, Dorsey K, Esposito EP, Reynolds MF, Liu Q, Cao M, Muller DA, McEuen PL, Cohen I (2020) Electronically integrated, mass-manufactured, microscopic robots. *Nature* 584(7822):557–561
119. Liu Y, Shaw B, Dickey MD, Genzer J (2017) Sequential self-folding of polymer sheets. *Sci Adv* 3(3):e1602417
120. Tolley MT, Felton SM, Miyashita S, Xu L, Shin BH, Zhou M, Rus D, Wood RJ (2013) Self-folding shape memory laminates for automated fabrication. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4931–4936
121. Tolley MT, Felton SM, Miyashita S, Aukes D, Rus D, Wood RJ (2014) Self-folding origami: shape memory composites activated by uniform heating. *Smart Mater Struct* 23(9):094006
122. Yichao T, Yanbin L, Yaoye H, Shu Y, Jie Y (2019) Programmable active kirigami metasheets with more freedom of actuation. *Proc Natl Acad Sci* 116(52):26407–26413
123. Miyashita S, Guitron S, Li S, Rus D (2017) Robotic metamorphosis by origami exoskeletons. *Sci Robot* 2(10):eaao4369
124. Ge Q, Dunn CK, Jerry Qi H, Dunn ML (2014) Active origami by 4D printing. *Smart Mater Struct* 23(9):094007
125. Mao Y, Yu K, Isakov MS, Wu J, Dunn ML, Qi HJ (2015) Sequential self-folding structures by 3d printed digital shape memory polymers. *Sci Rep* 5(1):13616
126. Samuel F, Michael T, Erik D, Daniela R, Robert W (2014) A method for building self-folding machines. *Science* 345(6197):644–646
127. Bena RM, Nguyen X-T, Calderón AA, Rigo A, Pérez-Arancibia NO (2021) Smarti: a 60-mg steerable robot driven by high-frequency shape-memory alloy actuation. *IEEE Robot Autom Lett* 6(4):8173–8180
128. Firouzeh A, Paik J (2015) Robogami: a fully integrated low-profile robotic origami. *ASME J Mech Robot* 7(2):021009
129. Paik JK, Wood RJ (2012) A bidirectional shape memory alloy folding actuator. *Smart Mater Struct* 21(6):065013
130. Rogers J, Huang Y, Schmidt OG, Gracias DH (2016) Origami MEMS and NEMS. *MRS Bullet* 41(2):123–129
131. Guan J, He H, Hansford DJ, Lee LJ (2005) Self-folding of three-dimensional hydrogel microstructures. *J Phys Chem B* 109(49):23134–23137
132. Shigemune H, Maeda S, Hara Y, Koike U, Hashimoto S (2015) Kirigami robot: making paper robot using desktop cutting plotter and inkjet printer. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1091–1096
133. Niijima R, Sun X, Sung C, Byoungkwon A, Daniela R, Sangbae K (2015) Pouch motors: printable soft actuators integrated with computational design. *Soft Robot* 2(2):59–70
134. Zhakypov Z, Mete M, Fiorentino J, Paik J (2019) Programmable fluidic networks design for robotic origami sequential self-folding. In: IEEE international conference on soft robotics (RoboSoft), pp 814–820
135. Li S, Vogt DM, Rus D, Wood RJ (2017) Fluid-driven origami-inspired artificial muscles. *Proc Natl Acad Sci* 114(50):13132–13137
136. Li S, Stampfli JJ, Xu HJ, Malkin E, Diaz EV, Rus D, Wood RJ (2019) A vacuum-driven origami “magic-ball” soft gripper. In: IEEE international conference on robotics and automation (ICRA), pp 7401–7408
137. Ma X, Vogtmann D, Bergbreiter S (2016) Dynamics and scaling of magnetically folding multi-material structures. In: IEEE international conference on robotics and automation (ICRA), pp 1899–1906

138. Salerno M, Firouzeh A, Paik J (2017) A low profile electromagnetic actuator design and model for an origami parallel platform. *ASME J Mech Robot* 9(4):041005
139. Salerno M, Zuliani F, Firouzeh A, Paik J (2017) Design and control of a low profile electromagnetic actuator for foldable pop-up mechanisms. *Sens Actuators A: Phys* 265:70–78
140. Zuliani F, Paik J (2022) Electromagnetic actuator design for distributed stiffness. *Smart Mater Struct* 31(11):115023
141. Tang D, Zhang C, Sun H, Dai H, Xie J, Jianzhong F, Zhao P (2021) Origami-inspired magnetic-driven soft actuators with programmable designs and multiple applications. *Nano Energy* 89:106424
142. Yi S, Wang L, Chen Z, Wang J, Song X, Liu P, Zhang Y, Luo Q, Peng L, Wu Z, Guo CF, Jiang L (2022) High-throughput fabrication of soft magneto-origami machines. *Nat Commun* 13(1), 2022
143. Dong Y, Wang L, Xia N, Yang Z, Zhang C, Pan C, Jin D, Zhang J, Majidi C, Zhang L (2022) Untethered small-scale magnetic soft robot with programmable magnetization and integrated multifunctional modules. *Sci Adv* 8(25):eabn8932
144. Song H, Lee H, Lee J, Choe JK, Lee S, Yi JY, Park S, Yoo JW, Kwon MS, Kim J (2020) Reprogrammable ferromagnetic domains for reconfigurable soft magnetic actuators. *Nano Lett* 20(7):5185–5192
145. Ha M, Bermúdez GSC, Liu JAC, Mata ESO, Evans BA, Tracy JB, Makarov D (2021) Reconfigurable magnetic origami actuators with on-board sensing for guided assembly. *Adv Mater* 33(25):2008751
146. Firouzeh A, Sun Y, Lee H, Paik J (2013) Sensor and actuator integrated low-profile robotic origami. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4937–4944
147. Mete M, Paik J (2021) Closed-loop position control of a self-sensing 3-dof origami module with pneumatic actuators. *IEEE Robot Autom Lett* 6(4):8213–8220
148. Paik JK, Kramer RK, Wood RJ (2011) Stretchable circuits and sensors for robotic origami. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 414–420
149. Dorsey KL, Huang H, Wen Y (2022) Origami-patterned capacitor with programmed strain sensitivity. *Multifunct Mater* 5(2):025001
150. Shin B, Felton SM, Tolley MT, Wood RJ (2014) Self-assembling sensors for printable machines. In: IEEE international conference on robotics and automation (ICRA), pp 4417–4422
151. Miyashita S, Meeker L, Tolley MT, Wood RJ, Rus D (2014) Self-folding miniature elastic electric devices. *Smart Mater Struct* 23(9):094005
152. Jiaqi L, Yuchong G, Young-Joo L, Shu Y (2020) Responsive and foldable soft materials. *Trends Chem* 2(2):107–122
153. Whitney JP, Sreetharan PS, Ma KY, Wood RJ (2011) Pop-up book mems. *J Micromech Microeng* 21(11):115021
154. Yao M, Xiao X, Tian Y, Cui H, Paik J (2020) An actuation fault tolerance approach to reconfiguration planning of modular self-folding robots. In: IEEE international conference on robotics and automation (ICRA), pp 8274–8280
155. Yao M, Xiao X, Belke CH, Cui H, Paik J (2019) Optimal distribution of active modules in reconfiguration planning of modular robots. *J Mech Robot* 11(1):011017
156. Belke CH, Paik J (2017) Mori: a modular origami robot. *IEEE/ASME Trans Mechatronics* 22(5):2153–2164
157. Belke CH, Paik J (2019) Automatic couplings with mechanical overload protection for modular robots. *IEEE/ASME Trans Mechatronics* 24(3):1420–1426
158. Robertson MA, Kara OC, Paik J (2021) Soft pneumatic actuator-driven origami-inspired modular robotic “pneumagami.” *Int J Robot Res* 40(1):72–85
159. Yao M, Belke CH, Cui H, Paik J (2019) A reconfiguration strategy for modular robots using origami folding. *Int J Robot Res* 38(1):73–89
160. Karras JT, Fuller CL, Carpenter KC, Buscicchio A, McKeeby D, Norman CJ, Parcheta CE, Davydchev I, Fearing RS (2017) Pop-up Mars rover with textile-enhanced rigid-flex PCB body. In: IEEE international conference on robotics and automation (ICRA), pp 5459–5466

161. Mintchev S, Shintake J, Floreano D (2018) Bioinspired dual-stiffness origami. *Sci Robot* 3(20):eaau0275
162. Johnson K, Arroyos V, Ferran A, Villanueva R, Yin D, Elberier T, Aliseda A, Fuller S, Iyer V, Gollakota S (2023) Solar-powered shape-changing origami microfliers. *Sci Robot* 8(82):eadg4276
163. Liu C, Wohlever SJ, Ou MB, Padir T, Felton SM (2021) Shake and take: fast transformation of an origami gripper. *IEEE Trans Robot* 38(1):491–506
164. Jeong D, Lee K (2018) Design and analysis of an origami-based three-finger manipulator. *Robotica* 36(2):261–274
165. Zhang C, Zhang Z, Peng Y, Zhang Y, An S, Wang Y, Zhai Z, Yan X, Jiang H (2023) Plug & play origami modules with all-purpose deformation modes. *Nat Commun* 14(1):4329
166. Wood RJ, Avadhanula S, Sahai R, Steltz E, Fearing RS (2008) Microrobot design using fiber reinforced composites. *J Mech Des* 130(5):052304
167. Hoover AM, Burden S, Fu X-Y, Sastry SS, Fearing RS (2010) Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot. In: IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics, pp 869–876
168. Askari M, Karakadioglu C, Ayhan F, Ozcan O (2017) MiniIAQ-II: a miniature foldable quadruped with an improved leg mechanism. In: IEEE internatinonal conference on robotics and biomimetics, pp 19–25
169. De Rivaz SD, Goldberg B, Doshi N, Jayaram K, Zhou J, Wood RJ (2018) Inverted and vertical climbing of a quadrupedal microrobot using electroadhesion. *Sci Robot* 3(25):eaau3038
170. Onal CD, Wood RJ, Rus D (2012) An origami-inspired approach to worm robots. *IEEE/ASME Trans Mechatronics* 18(2):430–438
171. Pagano A, Yan T, Chien B, Wissa A, Tawfick S (2017) A crawling robot driven by multi-stable origami. *Smart Mater Struct* 26(9):094007
172. Fang H, Zhang Y, Wang KW (2017) Origami-based earthworm-like locomotion robots. Bioinspiration. *Biomimetics* 12(6):065003
173. Jafferis NT, Helbling EF, Karpelson M, Wood RJ (2019) Untethered flight of an insect-sized flapping-wing microscale aerial vehicle. *Nature* 570(7762):491
174. Chen Y, Zhao H, Mao J, Chirarattananon P, Helbling EF, Hyun NP, Clarke DR, Wood RJ (2019) Controlled flight of a microrobot powered by soft artificial muscles. *Nature* 575(7782):324–329
175. Salerno M, Paik J, Mintchev S (2020) Ori-pixel, a multi-dofs origami pixel for modular reconfigurable surfaces. *IEEE Robot Autom Lett* 5(4):6988–6995
176. Mintchev S, Salerno M, Cherpillod A, Scaduto S, Paik J (2019) A portable three-degrees-of-freedom force feedback origami robot for human-robot interactions. *Nat Mach Intell* 1(12):584–593
177. Giraud FH, Joshi S, Paik J (2021) Haptigami: a fingertip haptic interface with vibrotactile and 3-dof cutaneous force feedback. *IEEE Trans Haptics* 15(1):131–141
178. Firouzeh A, Paik J (2017) Soft actuation and sensing towards robot-assisted facial rehabilitation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 306–313
179. Seo S, Park W, Lee D, Bae J (2021) Origami-structured actuating modules for upper limb support. *IEEE Robot Autom Lett* 6(3):5239–5246
180. Ze Q, Wu S, Nishikawa J, Dai J, Sun Y, Leanza S, Zemelka C, Novelino LS, Paulino GH, Zhao RR (2022) Soft robotic origami crawler. *Sci Adv* 8(13):eabm7834
181. Miyashita S, Guitron S, Yoshida K, Li S, Damian DD, Rus D (2016) Ingestible, controllable, and degradable origami robot for patching stomach wounds. In: IEEE international conference on robotics and automation (ICRA), pp 909–916
182. Edmondson BJ, Bowen LA, Grames CL, Magleby SP, Howell LL, Bateman TC (2013) Ori-clops: origami-inspired forceps. In: Smart materials, adaptive structures and intelligent systems, vol 56031, p V001T01A027
183. Bassik N, Brafman A, Zarafshar AM, Jamal M, Luvsanjav D, Selaru FM, Gracias DH (2010) Enzymatically triggered actuation of miniaturized tools. *J Am Chem Soc* 132(46):16314–16317

Index

A

Adaptation, 60
Agility, 60
Algorithm design, 62
Animal whisker, 73
Augmented reality, 253

B

Bar-and-hinge model, 325
Behavioural lenses, 75
Behavioural lensing, xiv, 45, 56, 64
Bioinspiration, 9
Bio-inspired artificial intelligence, 60
Bio-inspired deformable mobile robot, 61
Biological organisms, 60
Bistability, 328

C

Capacitive sensor, 212
Classification of different textures, 77
Cluttered settings, 60
Cochlea, 72
Compensatory behaviours, 61
Compliance, 63, 141, 282
Construction, 282
Contact, 141
Continuum, 282
Continuum hose robots, 10
Contraction theory, 49
Cosserat rod model, 266
Coupled kinematic system, 41

D

Deformable, 9, 141
Deformable mobile robot, 56
Deformable soft robotics, 60
DeforMoBot, 61
Deployable, 333
Design ideation, 57
Design iteration, 58
Dexterity, 10
Dielectric elastomer actuator, 186
Digital twins, 57
Distal phalange bone, 77
Distance geometry, 244
Dynamic equation, 51
Dynamics, 41, 157

E

Ecoflex-0030, 77
Eigenvalue, 53
Elasticity, 69, 148
Elastomers, 5, 186, 207
Electromechanical sheet, 330
Electromyography, 54
Embedded, 36
Embedded computing, 41
Embeddedness, 36
Embodied, 36
Embodied artificial intelligence, 60
Embodied computation, 36
Embodied intelligence, 9, 36, 41, 61, 198
Embodiment, 36, 64
Endoscopy capsule, 68, 72

Entropy reduction, 220
 Environmental interactions, 60
 Evolution, 60

F

Field robots, 62
 Fingernail, 77
 Fingertip, 77
 Finite element analysis, 191
 Finite element method, 141
 Force fields, 40
 Forward kinematics, 39, 142, 251
 Frequency separation, 73
 Frobenius distance, 77

G

Gaussian process regression, 272
 Generic modeling, 10
 Geometric constraints, 60, 193
 Ghost circuits, xiv, 45, 64
 Graphical format, 59
 Graphical sequences, 59
 Grasping, 180

H

Hall effect device, 218
 Hall effect sensors, 77
 Hall sensor, 73
 Heatmaps, 79
 Hertzian contact, 180
 Heuristic prototypes, 58
 Human-robot interaction, 254, 335
 Hydraulic actuators, 5
 Hydrogels, 5, 332
 Hyperelasticity, 148

I

Impedance control, 2
 Incipient slip, 207
 Inflatable membrane, 186
 Integrated electromechanical devices, 330
 Internal energy, 193
 Inverse kinematics, 142, 252

J

Jacobian matrix, 51, 168
 Jamming, 184

K

Kinematic modeling, 9
 Kinematics, 39, 141
 Kinematic tuning, xiv, 45, 50, 64
 Kirigami, 322

Kirigami designs, 322
 Kresling and Yoshimura patterns, 321

L

Lagrange multiplier, 198
 Lamé's constants, 191
 Laminate, 329
 Locomotion, 334

M

Magnetic, 332
 Magnetic sensor, 218
 Malleable link, 228
 Malleable robots, 10, 228
 Manipulation, 179, 334
 Mechanical Logic, 328
 Mechanism design, 61, 322
 Memory primitives, 59
 Mobile manipulation, 63
 Mobile robots, 62
 Modularity, 227
 Morphological computation, 60
 Morphological feature, 58
 Morphological features of a fingertip, 77
 Motor programs, 41
 Muscle co-contraction, 52
 Mutual information, 38

N

Navigation, 62
 Navigation among movable obstacles, 64
 Negative Poisson ratio, 69
 Nodal displacement vector, 191
 Nodal points, 191
 Nonlinear dynamic, 56, 141

O

Origami
 action, 323
 curved, 321
 degree-4, 320
 programmable compliance, 326
 rigid, 318

tessellation, 320, 326
Origami curved, 336
Origami pattern, 319
Origami robotics, 10, 60, 316
Origami structure, 68, 326
Origami vertex, 320

P

Passive dynamics, xiv
Perception, 64
Permittivity, 213
PETG, 78
Physical contact, xiii
PLA, 78
Pneumatic actuators, 3
Pneumatically driven, 4, 332
PneuNet actuator, 184, 196
Poisson's ratio, 180, 196
Pouch motor, 332
Proprioception, 63
Proximity, 215
Pseudo-rigid-body model, 324
Purposeful structural response, 3

R

Random variables, 43
Reactive and sensor-based planning, 63
Real-time computation, 62
Realtime computing, 41
Reconfigurable robot, 227, 333
Reconstructing, 72
Releasing, 180
Reservoir computing, 72
Resistive sensor, 208
Reward, 58
Robot design, 61, 322, 336
Robot kinematic tuning, 75
Robot morphologies, 60
Roughness estimation, 73

S

Save energy, 72
Second largest eigenvalue, 53
Self-folding, 331
Sensor fusion, 62
Sensorimotor computation, 36
Shannon entropy, 37

Shape-changing robots, 60
Shape-memory alloy, 331
Shape-memory polymer, 331
Softness, 1
Soft robotic hands, 10, 180
Soft robots, xiii, 3, 60
Soft sensors, 6, 207
State transition probability matrices, 54
Statistical, 58
Stiffness linkage, 69
Stiffness matrix, 192
Strain potential energy, 188

T

Tactile perception, 206
Tapered whisker, 72
Tapered Whisker-based Reservoir Computing (TWRC) system, 73
Tendons, xix, 4
Terrain classification, 73
Terrain identification, 72
Tissue dynamics, 6
TPU sensor, 6

U

Unstructured environments, 61, 63

V

Variable stiffness, 3, 230
Variational principle in statics, 193
Voltage divider circuit, 211
Volume resistivity, 209

W

Wheatstone bridge circuit, 211
Whisker, 72
Whisker-based navigation, 61
Whisker-based reservoir computing system, 73

Work, 192

Workspace, 247

Y

Yin and Yang, 1
Young's modulus, 180, 188, 196, 209