Shape-changing robotic materials using variable stiffness elements and distributed control.

by

M. A. McEvoy

M.S., University of Colorado at Boulder, 2014

M.S., Texas A&M University, 2005

B.S., Texas A&M University, 2004

A thesis proposal submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2017

This thesis entitled:
Shape-changing robotic materials using variable stiffness elements and distributed control.
written by M. A. McEvoy
has been approved for the Department of Computer Science

Prof. Nikolaus Correll

Prof. Dirk Grunwald

Prof. Rick Han

Prof. Chris Heckman

Prof. Kurt Maute

Date ____________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

McEvoy, M. A. (Ph.D., Computer Science)

Shape-changing robotic materials using variable stiffness elements and distributed control.

Thesis directed by Prof. Nikolaus Correll

Tightly integrating sensing, actuation and computation into materials enables a new generation of smart systems that can change their appearance and shape autonomously. Applications for such materials include airfoils that change their aerodynamic profile, vehicles with camouflage abilities, bridges that detect and repair damage, or robots and prosthetics with a rich sense of touch. While integrating sensors and actuators into composites is becoming more common, the opportunities afforded by embedding computation have only been marginally explored.

I present a composite material that embeds sensing, actuation, computation and communication and can perform shape changes by temporarily varying its stiffness and applying an external moment. I describe the composite structure, the principles behind shape change using variable stiffness and the forward and inverse kinematics of the system. Experimental results use a 5-element beam that can assume different global conformations using two simple actuators.

A distributed algorithm that calculates inverse kinematic solutions for shape-changing beams with integrated sensing, actuation, computation and communication is presented for beams consisting of n segments that can change their curvature and twist, perform computation and communicate with their local neighbors. The presented method distributes the computation among the n segments by sequentially applying the damped least squares method to m-segment neighborhoods, reducing the computational complexity of each individual update to $\mathcal{O}(n)$. The resulting solution does not require any external computation and can autonomously calculate a curvature profile to reach a desired end-pose. Results show that the proposed distributed approach performs as well as the centralized approach and grows linearly with the number of element in the beam.

Dedication

To Jen, Aiden, William, mom, and dad. Thanks for your love and support. This adventure would have been impossible without it.

Acknowledgements

I would like to thank all my mentors during my time time here. First to my advisor, Dr. Nikolaus Correll, thank you for your guidance these last few years. With your help and support I've been able to do more that I ever thought possible. I leave here more confident in my abilities and what I am capable of because of your constant pushing to always do my best. To Dr. Rick Han, thank you for your support throughout the entrepreneurship classes. Your support to develop and chase my own ideas will never be forgotten. Dr. Chris Heckman, thank you for all of your support and encouragement along the way.

I'd also like to thank my colleagues in Dr. Nikolaus Correll's lab. Thanks for making the last few years so enjoyable! Dr. Erik Komendera, thank you for helping me integrate back into student life when I first entered the lab. Dr. Dave Coleman, thank you for your mentorship in ROS and C++ and leading our team in the Amazon Picking Challenge.

Contents

Chapter

Figures

Figure

Chapter 1

Introduction

1.1 Shape Changing robotic materials

Advancements in material science, manufacturing processes and the continual miniaturization of electronic components have enabled a class of multi-functional materials that tightly integrate sensing, actuation, communication and computation. We refer to such materials as “robotic materials”, analogous to the field of robotics which combines mechanisms with sensing and control. Unlike conventional stimuli-response materials that change one or two physical properties in response to an external stimulus, robotic materials allow the relationship between sensing and actuation to be described algorithmically. Some of the multi-functional materials that serve as inspirational robotic materials are shown in Figure 1.1.

Artificial skin promises to equip prosthetic and robotic hands with tactile sensing that comes close to that of human performance. Existing systems do not yet provide the resolution, bandwidth, and dynamic range of the human skin [17]. Here, integrating computation into the skin can alleviate the bandwidth requirements of high-resolution, high dynamic range sensing with pre-processing and help to discern task-relevant information from background noise.

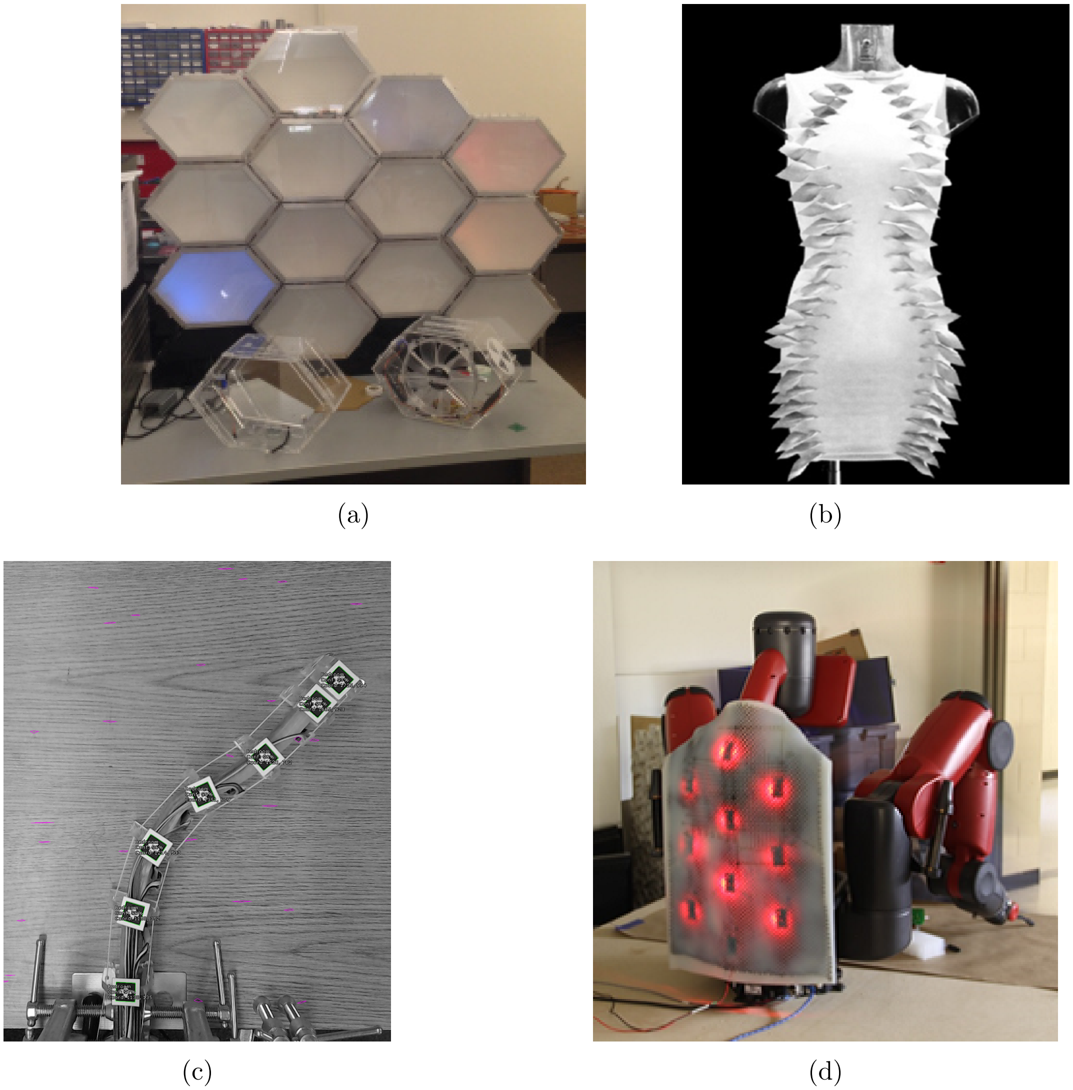
Various artificial mechanisms ranging from optical metamaterials [52,81] to smart composites [69] have been proposed that would change their appearance in response to the environment. While these mechanisms have the potential to induce appearance change, very few works have attempted the system-level integration of sensing, pattern recognition and distributed control into a composite material that can actually respond to the environment.

Figure 1.1: Our inspiration comes from biological systems that tightly integrate sensing, actuation and controls and engineering applications that could benefit from this approach. Top: Biological materials that exhibit this tight integration such as the cuttlefish (camouflage), an eagle's wing (shape change), the banyan tree (adaptive load bearing) and human skin (tactile sensing). Bottom: The engineering applications that can possibly take advantage of similar principles, motivating novel materials that tightly integrate sensing, actuation, computation and communication.

In an engineering context, non-destructive evaluation devices embedded into wings, bridges, and other safety critical systems should make it possible to detect potential problems before they appear, reducing inspection and maintenance costs [2]. Materials with embedded sensors could monitor changing structural loads and self diagnose, allowing for materials that self repair and automatically adapt to changing conditions. Materials could self-repair by releasing chemical agents in the material [10], or locally change their stiffness to re-distribute loads.

Morphing aerodynamic surfaces could improve efficiency during different flight regimes, reduce noise and save fuel. Early designs used mechanical actuators in series that would distort the shape of the wing [5, 89, 92, 96]. However, these concepts do not scale; every additional actuator increases the required load carrying capacity of all actuators in the chain. This leads to increased weight, which again requires stronger (and heavier) actuators. robotic materials might alleviate this problem through a tighter integration of sensing, actuation and control, for example by combining variable stiffness with bending actuation.

Creating robotic materials that address the above applications with seamlessly integrated, mass-produced products will require advances in material science and manufacturing. However, macroscopic robotic materials with useful functionality have already been realized with existing materials and processes. Examples shown in Figure 1.2 include an amorphous façade that recognizes a user's input gestures and responds with changes in its opacity and color [21], a dress that can localize sound sources and indicate their direction using vibro-tactile feedback [77], a shape-changing variable stiffness beam [60], and a robotic skin that senses touch and texture [37].



(a)

(b)

(c)

(d)

Figure 1.2: Examples of robotic materials that combine sensing, actuation, computation and communication. (a) An amorphous façade that recognizes gestures and changes its opacity and color [21], (b) a dress that can localize sound sources and indicate their direction using vibro-tactile feedback [77], (c) a shape-changing variable stiffness beam [60], and (d) a robotic skin that senses touch and texture [37].

For my thesis, I concentrate on the shape-changing robotic materials shown in Figure 4.6. Unlike the rest of the robotic materials presented in Figure 1.2, shape-changing robotic materials require embedded actuation that changes the physical properties of the underlying material. Variable stiffness materials embedded with sensing, actuation, computation and communication will enable shape changing materials with unprecedented ranges of motion and controllability.

High-value applications such as airfoils, prosthetics and camouflage might be among the first robotic materials that are able to find favorable trade-offs between added functionality and increased cost, weight, and inferior structural properties of embedded sensing, actuation and computation. In the long run, however, the success of robotic materials lies in the systems integration and manufacturing challenges that are common to all robotic materials. The integration of sensing, actuation, communication and control is a very general problem. Focusing on shape changing materials I will advance the field of robotic materials by providing a better understanding of how discrete sensors and actuators can interact with, influence, and control a continuous material.

1.2 Contributions

This work shows that sensing, actuation, communication and computation allows for advanced materials that can actively react to their environments. Traditionally the constituent parts of robotic materials have been isolated in their various domains. While significant advances have been made there has not been much emphasis on how to combine each of these components. The integration of these components into a single material requires the cooperation of these disparate disciplines to understand the fundamental challenges of creating such systems. Once these view points are combined, it will be natural for designers to trade the additional functionality that can be gained through the use of computation vs the structural properties lost by embedding the required electronics.

Emphasis of this work is not on the individual components and specific solutions to implement variable stiffness, distributed sensing, computation and actuation, but on how these components interact to create an autonomous system and the fundamental challenges toward creating tightly integrated robotic materials that mimic the complexity of biology systems.

1.3 Outline

This dissertation is the synthesis all of my published works. Chapter 2 describes the principle components of robotic materials [63]. Chapter 3 discusses current methods of creating variable stiffness elements and the variable stiffness elements that I use in the presented robotic material [61]. In Chapter 4 I discuss the mechanics of shape change and how the variable stiffness elements are assembled into a robotic material [60]. Chapter 5 discusses the distributed control algorithms that makes shape change in the robotic material possible [62]. Finally Chapter 6 discusses the results, challenges and future work required to push shape changing robotic materials forward.

Chapter 2

Related Work

2.1 Background of Robotic Materials

An early vision of smart materials with embedded, networked computation are networks of microelectromechanical systems (MEMS) [8]. MEMS allowed for the manufacturing of micro-scale structures with the same processes that are used for making conventional analog and digital semi-conductor circuits, allowing their tight integration. An example of a mainstream MEMS device is an accelerometer that consists of a cantilevered beam with a small mass and circuitry to measure its displacement in response to acceleration loads, and can easily be mass-produced. Tiny MEMS sensing devices could be deployed in large numbers and carried away by the wind, a vision that [8] describes as "smart dust". This vision is extended by [27] to millimeter-scale units that can locomote by themselves, allowing the MEMS devices to reconfigure and form "programmable matter."

In addition to the material science challenges, such a vision poses a series of deep challenges in networking and computation, which has inspired two active fields, namely sensor networks and amorphous computing. Amorphous computing [1] has laid the foundation for computation in large-scale distributed systems in which individual computing elements can be unreliable and do not need to be manufactured in a precise geometrical arrangements. Hardware demonstrations that came out of this movement include "paintable computing" [14], a distributed system of locally communicating nodes that used gradient information to display lines and simple characters, pattern formation in bacteria colonies that are receptive to chemical gradients and can be designed to act as simple high, low and band-pass filters [6], and a modular robotic [100] system that can adapt its shape to the environment via local sensing [102]. At the same time, the sensor network community has begun to explore the foundations of networking and routing in these systems [53], although focusing almost exclusively on geo-spatial sensing applications rather than integrating sensor networks into materials.

The vision of materials that can change their physical properties has also been explored in the context of designing new interactions between computers and people. “Tangible bits” [41] or “radical atoms” [40] promote the idea of presenting information in physical form, not limited to pixels. This concept has found physical implementation in “pushpin computing” [55], which aims at adding additional layers of information to everyday objects such as push pins and floor tiles, and a series of works that consider materials that change their physical properties such as stiffness [74], physical extension [51], or weight [72]. As such, these works explore a series of applications as well as their enabling principles, but leave their implementation into systems or products to science and engineering.

Distributed MEMS, the related concepts it helped spawn, and modular robotics emphasize the system-level integration of sensing, actuation, computation and communication, but fall short in addressing the structural properties of the resulting systems. The structural properties of a composite are an integral part of “multi-functional materials”, a field that traditionally aims at optimizing design by addressing both structural (e.g. strength and stiffness) and non-structural (e.g. sensing and actuation, self-healing, energy harvesting) requirements of a system [26], but largely ignores the opportunities of integrated computation that have been articulated by [1, 8, 46]. Multi-functionality at the nano- and micro-scale has also been studied in physics in the context of metamaterials. Metamaterials are “macroscopic composites having a man-made, three-dimensional, periodic cellular architecture designed to produce an optimized combination, not available in nature, of two or more responses to specific excitation” [93]. Metamaterials traditionally exploit the frequency properties of structures to deflect optical waves in non-natural ways, but the above definition allows a broader interpretation, both in terms of the constituents of individual cells and their scale, making it applicable to some of the computational systems discussed here.

The physical properties of the material itself do not just affect sensing and actuation, but also computation. Indeed, material dynamics allows classes of computation to be shifted into the material itself by tuning the geometry and material properties of a structure. For example, feedback control by exploiting thermal or chemical deformations to regulate a process, compensating for motion parallax as in an insect's eye [22], or the transformation of a signal into the frequency domain as in the cochlea in the inner ear. This effect is known as "morphological computation" [76], and has become an important aspect of the design of robotic systems.

2.2 Constituent Parts of Robotic Materials

Robotic materials consists of sensors, actuators, computing and communication elements. While these terms are very broad the integration of these elements into composites has the potential to enable robotic materials with novel, unprecedented functionality.

2.2.1 Sensing

Classical stimuli-response materials "sense" their environment by changing some of their properties in response to one or more external stimuli, including acoustic, electromagnetic, optical, thermal, and mechanical. Robotic materials integrate dedicated sensors that, in combination with appropriate signal processing, let the composite identify and respond to environmental patterns of arbitrary complexity, only limited by available sensors and computation. An example of the complex signal processing that has been accomplished in a robotic material is the sensing and localization of textures that touch an artificial skin [37] (Figure 1.2d). This artificial skin is made by distributing nodes throughout a silicon based material. Each node is equipped with a microphone and can analyze the high frequency sound signal generated by a texture rubbing the skin. Local communication between nodes allows the position of the touch to be triangulated. Once triangulated, the closest node to the source analyzes the material and classifies it. With this approach the nodes sample and process high-bandwidth information locally and then route high-level information back to a central computer only when important events occur. This example, using embedded MEMS microphones, lends itself to many related material-centric applications such as sound localization [77,83], vibration analysis [25,43], or — when combined with piezo actuators — structural health monitoring [2,38,103,104].

Similarly, accelerometers can detect impacts [2] or determine orientation of a robotic material with respect to gravity. Capacitive touch sensors [17] can be embedded into the surface of a robotic material as input devices. Optical sensors such as full-color light sensors, Infrared (IR) sensors and photoresistors can measure ambient light levels for camouflage applications. Thermistors would allow robotic materials to measure temperature of either the environment or the material itself at high resolution [61,85]. Mechanical sensors that measure applied force [75,88], strain [99,105], or deflection [57] can monitor the flow over an aerodynamic surface and monitor its shape change as it morphs into an optimal configuration.

Most of the sensors discussed above have been developed for, or are at least suitable for, operation embedded in a material. However, deploying such sensors in large numbers and at high densities requires solving problems in system integration, which can partly be alleviated by co-locating those sensors with computing elements to preprocess and network information, as discussed below.

2.2.2 Local Computation

Although it is possible to route actuation signals and sensing information to and from a central processor, this approach becomes increasingly difficult with scaling of both the required bandwidth and the number of sensors/actuators to be embedded. A system such as the sensing skin [37] illustrates this difficulty with respect to sensing, a shape changing material such as [60] with respect to actuation, and the smart façade with respect to simultaneous sensing and actuation. Routing vibration signals sampled at 1kHz becomes increasingly difficult when the number of sensors increases. Instead, when computing information locally, only selected information needs to be transferred outside of the material. In the shape changing material [60] that controls local stiffness by melting, temperature readings are only used locally for feedback control and are not needed outside of the material. Therefore, the desired stiffness profile needs only to be sent once and can then be controlled locally. Finally, a façade whose transparency and color can be adjusted by a user does not need to disseminate sensed gestures through the system, but only the resulting actuation command that a user intends.

Algorithms that run on a robotic material must have the following properties: first, they must scale as the material grows in size; second, they must be able to run with the limited computation and memory resources provided in each node; and third, they have to be robust with respect to the failure of individual nodes. Limiting information exchange to local communication strongly promotes scalability. Algorithms that run in constant time, independent of the size of the network are known as local algorithms. An overview of such algorithms is presented in [20, 86] in the context of wireless sensor networks. These local algorithms are used to determine conflict-free sets of activities, such as simultaneous data transmissions, by using matching, independent set, and coloring algorithms, which are important primitives in higher level distributed algorithms. One major limitation of the algorithms discussed in [86] is that they assume synchronous communication, which creates additional overhead, see, e.g., [97].

From a computational perspective, robotic materials can be viewed as an amorphous [1] or spatial computer [7], which attempts to formalize a distributed computation model for systems that are limited to local communication and limited computational resources at each node. A key challenge in amorphous computing is how to design local interactions so that a desired global behavior can emerge. One approach to address this problem is using programming languages that provide abstractions that allow to describe desired global behaviors and then automatically compile the corresponding local rules. What programming paradigm, i.e. procedural or functional, is most conducive to program large numbers of distributed computing elements remains an open question, and [7] provides a comprehensive survey to the field.

Designing distributed algorithms and solving the global-to-local challenge are hard problems. Their solution is not on the critical path for large scale deployment of computing infrastructure into robotic materials, which might benefit from enhanced signal processing, local control and networking, all of which are established fields.

2.2.3 Local Communication

Robotic materials require communication not only to transport sensing and control information, but also for more complex spatial dynamics to emerge. The key challenge for transporting data is that point-to-point connections from sensing locations to a central processing unit become quickly infeasible due to large number of cable crossings, the effect of embedded wiring on the material's structural properties, or radio frequency challenges. The local computation in robotic materials not only offers local pre-processing of sensing information, but also offers the routing of information through a computer network, i.e., a shared communication channel that is used by all participants of the network, a problem that has been widely studied in sensor networks [3, 90].

Local computation becomes particularly interesting when individual processing nodes can access information from neighboring nodes via local communication. Some example robotic materials that take extensive advantage of this are distributed gesture recognition in an amorphous façade [21], where local communication is used to pass tactile sensing events along the physical path they occur, texture identification in a robotic skin [37], where local communication allows triplets of nodes to triangulate the location of a vibration event by comparing local measurements, and distributed sensor-based control of a rolling robot [16], where local communication is used to infer the overall orientation of the material with respect to the ground.

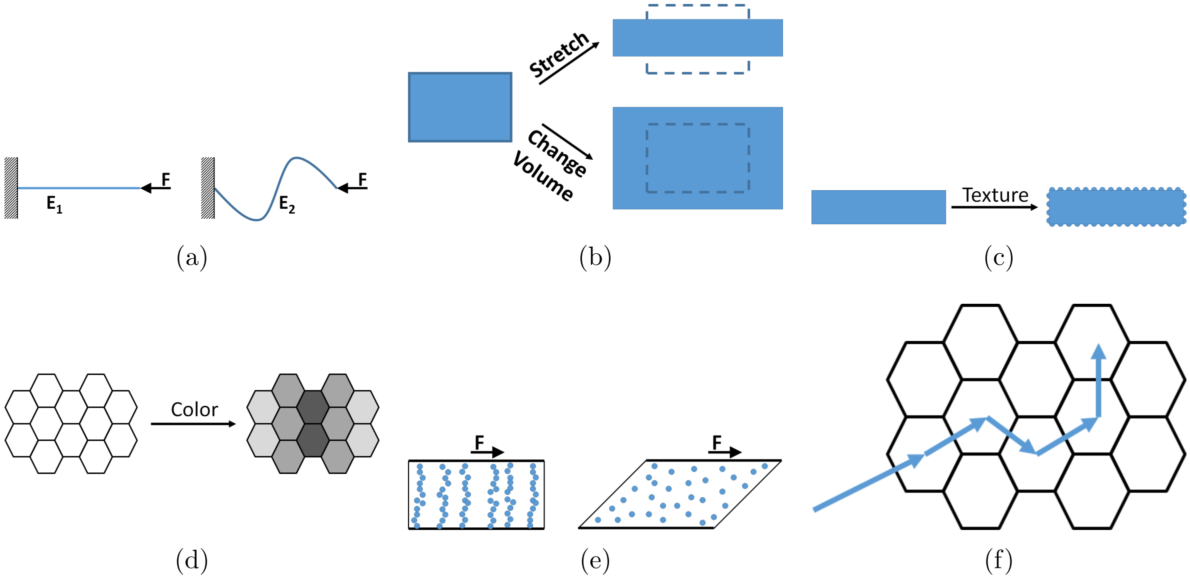
The speed of communication through a robotic material has significant impact on the performance of the robotic material. For example, a robotic skin that touches a hot surface needs to process and route that event quickly through the material, and might forgo the processing and forwarding of high-bandwidth texture information. In addition to actual bandwidth, communication speed is also highly dependent on the network topology [78] and node density [39], which leads to important design considerations in robotic materials as density and topology not only affect the computational properties of the system, but also its structural properties. Finally, tighter integration of future robotic materials consisting of possibly millions of tiny computing elements might require a departure from traditional networking and routing algorithms, requiring solutions that trade-off performance with memory [32] or computational [35, 56] requirements.

There are only few works that address hardware implementations of wired communication infrastructure embedded into materials. Various robotic skins use hierarchical industry bus-systems, which, however, scale poorly both with respect to bandwidth as well as the total number of nodes that the system can support [17]. A distributed optical sensor network built into multi-functional materials that allows the distribution of both power and information for structural health monitoring applications is described in [12]. Here, using optical wave guides that can transport both power and information bears the potential to minimize impact on structural properties, but is limited in the power density that it can achieve. In practice, combinations of peer-to-peer wired communication and long range, high-bandwidth backbones using wired buses or wireless links might allow a robotic material to maintain both scalability and overall throughput.

2.2.4 Actuation

In a robotic material, actuation refers to changing the material properties of the underlying base material. Some possible actuations are expanding, contracting, changing stiffness, changing surface texture or changing color (Figure 2.1), while possible actuators include heat, electricity, light, magnetism, or the release of chemicals.

Variable stiffness actuators have received attention as the basis for morphing airfoils and active vibration control, resulting in a large number of actuators that are potentially suitable for use in robotic materials. One common approach to variable stiffness is sandwiching a thermoplastic between two metal plates [24,70] and then exploiting the thermoplastic's change in stiffness with increasing temperature. When the thermoplastic is at a low temperature, the metal plates are tightly coupled together, acting as a single stiff composite. At higher temperatures, the thermoplastic has much less resistance to shear and the plates act as if they were uncoupled from each other, creating a composite with a much lower stiffness. A similar approach is shown in [64,65] which segments the



(a) (b) (c)

(d) (e) (f)

Figure 2.1: The in-situ actuators of a robotic material work to change the material properties of the base material. Changes in (a) stiffness and (b) volume could enable a shape changing robotic materials. Robotic skins could utilize changes in (c) appearance and (d) surface texture. While self-healing and self-regenerating robotic materials could use venous systems enabled by (e) variable viscosity fluids or (f) the rerouting of the healing compounds through the material.

rigid layers and uses a shape memory polymer [66] as the sandwich layer. Alternatively, the friction between plates can also be altered pneumatically. In [74], a number of sheets are inserted into a vacuum bag which remains flexible until a vacuum is applied and the deformed shape is locked in place. Similarly, particle jamming [11] is a technique where a granular material is encased in a flexible material. When pressed against an object the granular material conforms to the object's shape. Evacuating the case causes the material to contract and harden, pinching the object.

Simply changing the stiffness of a material, however, will not result in a shape changing material; additional actuators must be embedded to initiate the shape change. Recent advances in the development of artificial muscles might make their large-scale integration into robotic materials feasible. An artificial muscle made from fishing line or conductive sewing thread is described in [31]. The artificial muscles are created by twisting the threads until they start to coil up on themselves. The stroke and actuation force can be tuned by changing the weight used when coiling the thread, using multiple coils, or by coiling around a mandrel. Shape memory alloys have been used in many artificial muscle applications [71]. Typically nickel-titanium or copper-aluminum-nickel alloys, shape memory alloys are able to change from a deformed shape back to their parent shape when heated above their transition temperature. Examples are articulated joints in an artificial bat wing [23] and in origami inspired robots [73], which demonstrate how artificial muscles could be embedded into a robotic material. McKibben actuators are pneumatic artificial muscles that are light weight, flexible and able to achieve large displacements [19, 49, 87] these McKibben actuators place an inflatable bladder inside of a woven mesh; when the bladder is inflated, the diameter of the woven mesh expands while the length contracts. Efforts made to miniaturize these devices are reviewed in [19] while [87] presents a McKibben actuator that makes use of shape memory polymer to maintain the actuators displacement without continuous control, demonstrating how a robotic material could use both variable stiffness materials and artificial muscles to achieve shape change.

Pneumatic and hydraulic systems that create volumetric changes have been extensively used in soft robots and could be implemented in a robotic material to create distributed volumetric changes for shape changing and morphing applications. Chambers embedded into a soft elastomer can be filled with fluid or air, causing the elastomer to expand and change its shape. This effect has been used for locomotion in [16, 47, 58, 84] where soft robots crawl, roll, swim, and bend into an arbitrary 2D configuration, respectively. A challenge of pneumatic and hydraulic robotic materials is not only pressure distribution, but also the requirement for possibly large numbers of miniature valves. A miniature electro-rheological fluid based valve [101] or a miniature latchable microvalve based on low-melting point metals [80] could be embedded into such robotic materials and enable the control of fluidic channels in a self-healing composite [10] or the control of embedded fluidic channels for camouflage and display in soft robots [69]. Here, the soft robots are designed with microfluidic networks that can be filled with colored, temperature controlled fluid to change their appearance in both the visible and infrared spectrum.

Volumetric change can also be influenced by the construction of the base material itself. In a cellular material, for example, changing the geometry allows for designs with different Poisson ratios [29]. This also allows large changes in a material's area or volume, for example cellular geometries that are allowed to buckle in local regions, drastically reducing their surface area, are described in [33].

Similar to sensors for robotic materials, the actuators discussed here lend themselves to implementation in large numbers and parallel operation. Furthermore, computation might overcome integration challenges by reducing communication requirements due to local control.

2.3 Control of Robotic Materials

Robotic materials require control at two different levels. First, the local control of each actuator using feedback from an appropriate sensor and/or state information from neighboring controllers. Second, global control that implements a desired spatio-temporal pattern across the material, either in a distributed or centralized manner. For example, to achieve shape change in [61], the composite embeds a thermistor, power electronics, and a small microcontroller co-located with each heating element to implement feedback control of a precise temperature across a bar to vary its stiffness by melting. In [60], a global controller solves the inverse kinematics of a beam with many such variable stiffness elements in series to achieve a desired shape, and disseminates appropriate stiffness values into the robotic material where they are controlled by local feedback control. An example of local control that requires neighborhood information is the rolling belt from [16], where a state transition from deflated to inflated to induce rolling motion is a function not only of the local sensor, but also of those to the left and right of each controller.

These types of controllers pose two fundamental challenges: first, designing controllers requires a fundamental understanding of the material dynamics: how they heat, deform, or change appearance as a function of provided energy and time, e.g., and second, understanding how large numbers of distributed controllers interact. Both of these problems are further complicated by the fact that the dynamics of the underlying physics are continuous whereas the computational aspects of the system are discrete. This is illustrated in Figure 2.2.

There are two popular approaches to make these systems analytically tractable: discretizing the material by describing it as a lumped element model or maintaining its continuous properties

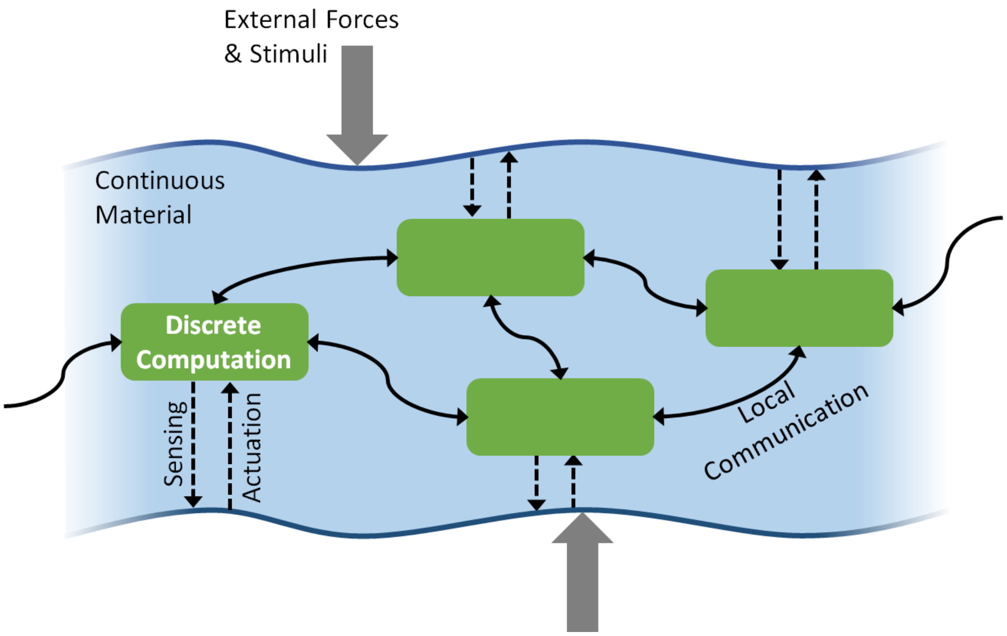


Figure 2.2: Relationship between the cyber and physical components of a robotic material. Continuous material properties can be sensed, processed in a computing element, and actuated upon. Whereas sensors, actuators, and computing elements are at discrete locations and can communicate locally, the material itself provides continuous coupling between sensors and actuators at different locations.

by modeling it as a distributed parameter system. Lumped element models of mechanical systems can be solved relatively easily, e.g. by using variational integrators [44, 59], whereas distributed parameter systems require solving partial differential equations (PDE). Assuming that the distribution of the computing elements is quasi-continuous — consistent with the amorphous computing paradigm [1] — allows part of this burden to be moved into the material itself and allows for the individual computing elements to each solve parts of the relevant PDEs [91].

While there is a large body of work on the control of large-scale distributed systems, much of which is relevant to the control of robotic materials [18, 50], only a few of these approaches have been explored experimentally due to the absence of real systems with thousands of sensors and actuators. In addition to providing the ability to implement distributed control inside the material, robotic materials also offer the possibility of calculating their own dynamics, which is an important capability in a distributed model-predictive control framework [79].

2.4 Summary

Robotic materials are a class of multi-functional materials enabled by recent advances in material science, electronics, distributed computation and manufacturing. While composites now include the ability to sense damage or self repair, for example, state-of-the-art composites fully integrate sensing, actuation, computation, and communication.

Of the applications highlighted, many would significantly benefit from integrated distributed computation. In general, decentralized computation is critical when the required sensing bandwidth is high or the material requires high-speed feedback control. In both cases, routing of information to a central processing system quickly becomes infeasible. These problems are common to seemingly unrelated applications such as camouflage or morphing airplane wings, which are currently being investigated by disjoint communities.

Although a number of manufacturing processes for robotic materials exist, ranging from deposition to folding, robotic materials will require vertical integration of a number of these processes. Additional challenges include programming techniques that synthesize low-level code from a high-level, emergent behavior provided by the designer, and creating interfaces between disciplines that allow experts from currently disjoint disciplines to address common system challenges. If these challenges can be overcome, robotic materials will lead to robotic systems with unprecedented sensitivity and adaptivity.

Chapter 3

Variable Stiffness Materials

The octopus arm, the elephant trunk, or mammalian tongues (Figure 3.1) are capable of dramatic shape changes in the absence of skeletal structures [48]. Instead, these systems employ variable stiffness, which allows muscles to generate motion while also acting as supporting structure. Even when severed from the body, the octopus arm is able to perform motions completely autonomously [28]. The octopus achieves this by tight integration of sensing and neural computation, with studies showing that the arms host two-thirds of the animal's neurons [34]. These neurons somehow solve inverse kinematics of a soft appendage with a large number of degree of freedoms, which is known to be a computationally hard problem that is equally vexing to biologists [30] and roboticists [15, 45]. I believe the tight integration of sensing and neural computation is what makes the octopus arm intelligent and that designing materials that utilize distributed sensing, actuation and computation allows us to make more intelligent robots.

While we are far off from creating high fidelity replications of biological materials, such as the variable stiffness arms of hydrostats or the color changing skin of a cuttlefish, we can break these systems down into their functional components and study the integrated systems that result from combining sensing, actuation, computation and communication in a material-like fashion.

Inspired by how the octopus arm combines variable stiffness, distributed sensing and distributed computation to respond to external stimuli and forces, I present a shape-changing robotic material that changes shape by selectively varying its stiffness. This chapter presents the variable stiffness actuators that will comprise the robotic material. In the next chapter I will show how

Figure 3.1: Muscular hydrostats are biological examples of structures that are not supported by a skeleton. (Left) The octopus uses its tentacles to grab prey and bring food to their mouths. (Middle) Elephants use their trunks to also grab food and bring it to their mouths. (Right) A cow uses its tongue to pick its nose.

these elements are assembled into the robotic material and how it is able to change its shape.

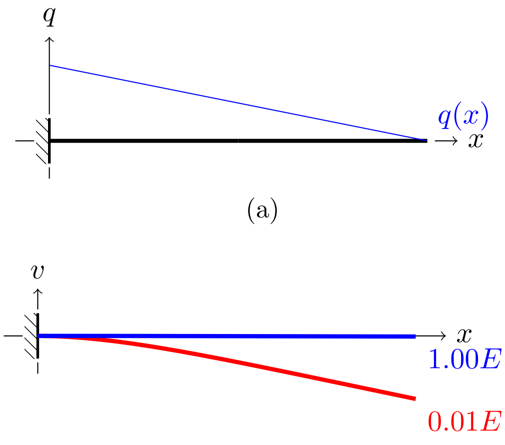
3.1 Principle of Operation

The concept of variable stiffness is best exemplified through examination of the classical beam equation. The displacement of a beam, $v(x)$, is governed by the applied moments M, the stiffness E and the cross-sectional inertia I, all of which may vary along the length of the beam as shown in Equation 3.1. Typically though, stiffness and cross-sectional inertia are fixed at the time of manufacture. Variable stiffness materials allow the stiffness of the beam to be changed on demand, which results in very different displacement profiles for a particular load as demonstrated in Figure 3.2.

$$\frac{v''(x)}{(1+v'(x)^2)^{\frac{3}{2}}} = \frac{M(x)}{EI} \tag{3.1}$$

3.2 Material

Of the many variable stiffness materials outlined in Chapter 2 we need one that offers a large range in stiffness and is easy to work with. For these reasons we choose to use the thermoplastic Polycaprolactone, a low melting-point polymer whose properties have been extensively studied [94,



(a)

(b)

Figure 3.2: Varying the Young's modulus of a beam will cause the beam to behave differently under the same loading. The more we are able to vary the Young's modulus of the beam, the greater the change in behavior. At 100% stiffness (blue line), the beam is able to resist the applied load $q(x)$ and only deflects a small amount. At 1% stiffness, the deformation is much more severe (red line). Changing a material's stiffness is the underlying principle of our variable stiffness actuators.

98]. Thermoplastics are inexpensive, easily manufactured and formed, and allow us to change their stiffness over multiple orders of magnitude simply through heating and cooling of the thermoplastic.

Using Joule heating and temperature control, we can take advantage of the stiffness changes that occur as the thermoplastics' temperature rises. Polycaprolactone has a glass transition temperature of -50 C and is in its rubber state at room temperature. The Young's Modulus of Polycaprolactone is approximately 190 MPa at room temperature and drops to nearly 2 MPa when molten (60 C). The displacements shown in Figure 3.2b correspond to beams with stiffnesses of 200 MPa (blue) to 2 MPa (red) and show the range of motion possible using Polycaprolactone as the base material for our variable stiffness elements. Further material properties for Polycaprolactone are described in [4].

3.3 Construction

This section outlines the construction techniques that I've used to manufacture the variable stiffness elements. The process has evolved over repeated attempts and the lessons learned are

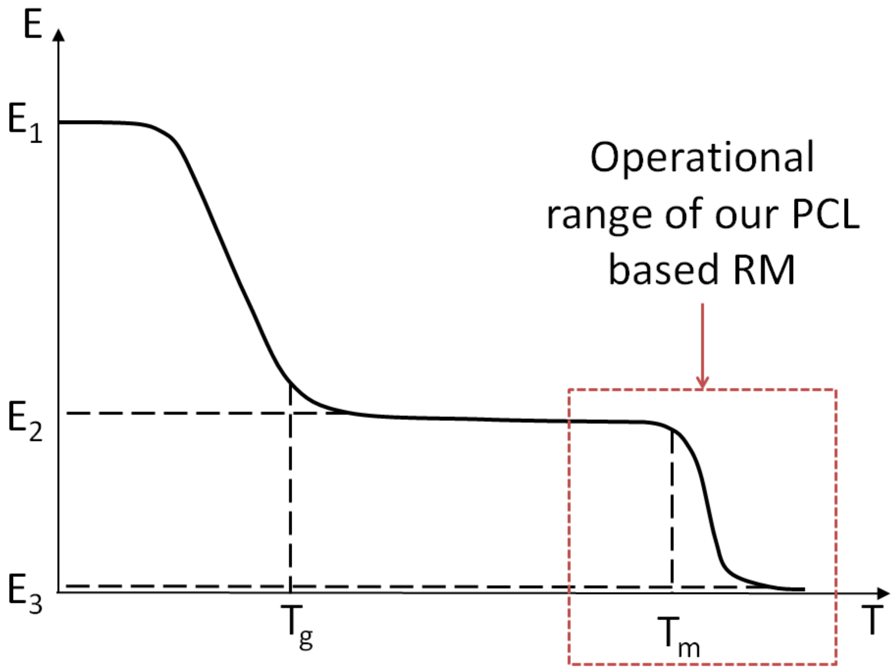


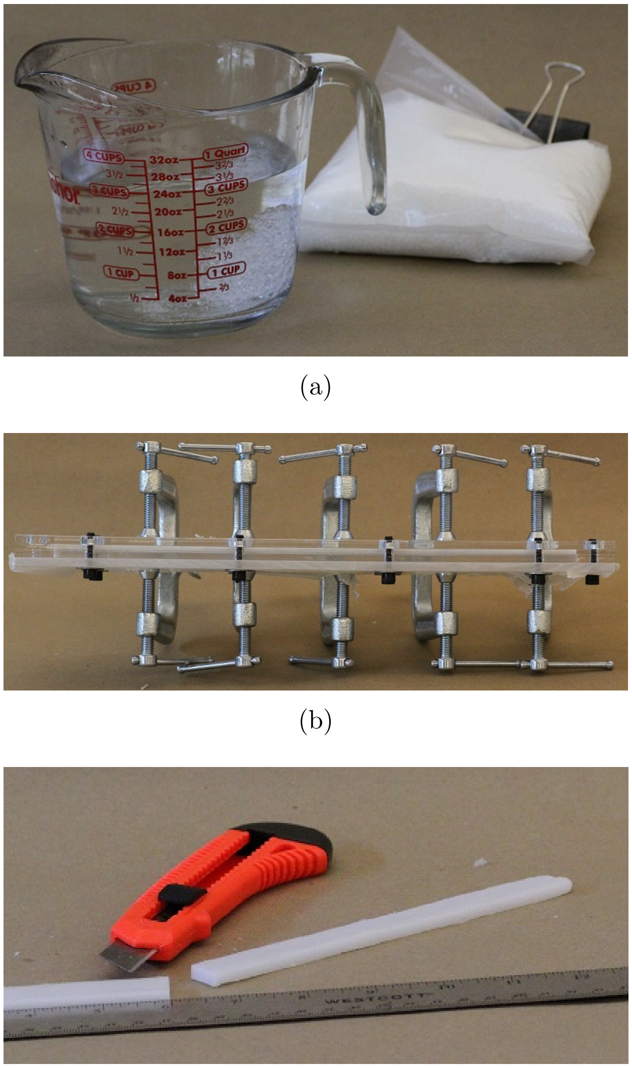
Figure 3.3: For a typical thermoplastic, the Young's modulus drops when approaching the glass transition temperature and then drops again when approaching the melting temperature. It is this variation of Young's modulus with temperature change that we exploit in this particular RM. Our choice of Polycaprolactone limits us to the region highlighted in red. Choosing a different method of heating or a thermoplastic could allow for a much larger range in stiffnesses to be achieved.

distilled here. I break the construction process down into three parts: forming the Polycaprolactone beams, embedding the heater and thermistor and maintaining cross-sectional shape.

3.3.1 Forming Polycaprolactone Beams

Polycaprolactone is available from a large number of suppliers under various trade names. We order our Polycaprolactone from SparkFun Electronics which sells Polycaprolactone under the trade name **Polymorph** (P/N TOL-10951). SparkFun Electronics does not offer a material certification for this product and, at the time of writing this, is only sold in pellet form so must be molded into bars manually.

The process to make the bars is shown in Figure 3.4. First the pellets are heated above their melting point of 60 C, to a working temperature between 80 C and 100 C. This is accomplished with either a bath of boiling water or by heating them directly in an oven. Polycaprolactone is extremely sticky when heated and the hot water bath allows for heating of the pellets without having them stick to the container or mold walls.



(a)

(b)

(c)

Figure 3.4: The general procedure for molding Polycaprolactone beams. (a) First the Polycaprolactone pellets are melted in a bath of boiling water. The ideal working temperature is between 80 C and 100 C. (b) The melted glob of Polycaprolactone is kneaded to remove any air bubbles and pressed into a mold until it has cooled back down to room temperature. Excess material is allowed to escape through a spillway at one end of the mold. (c) The cooled Polycaprolactone is cut to length with a razor blade.

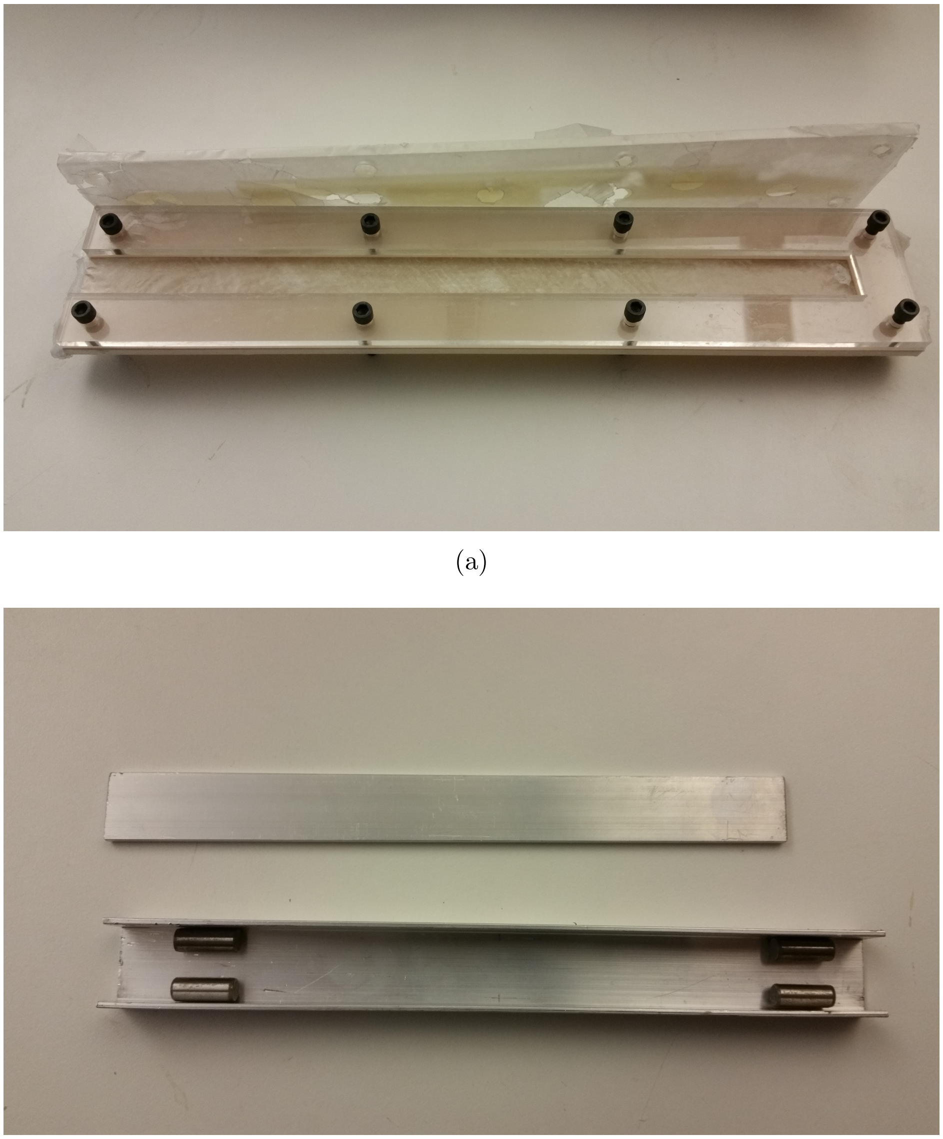
In Figure 3.4b the molten glob of pellets is pressed into the mold to form the bars. The figure shows an acrylic mold, however metal molds have also been used with great success. Figure 3.5 shows a cross section of each type of mold. The advantage of the metal mold is that it can be heated directly in the oven, but you cannot then knead the glob of Polycaprolactone to ensure that there are no air bubbles in the bar. While the Polycaprolactone does not stick to the metal mold at room temperature, it does stick when heated to 80 C. The last step is to cut the beams to the desired length. The Polycaprolactone is a rather soft thermoplastic, so using a razor blade works well for this process.

3.3.2 Embedding Heaters and Thermistors

Heating the Polycaprolactone to the desired temperature is accomplished through Joule heating using nickel chromium wire. The nickel chromium wire is 60% nickel, 16% chromium and 25% iron. Two main quantities are considered when sizing the nickel chromium wire, what length of the Polycaprolactone beam needs to be heated and what voltage power supply is available. This directly impacts the time required to change the stiffness of an element and is easily evaluated using commercial off the shelf finite element tools such as SolidWorks.

Embedding the Polycaprolactone bar with the nickel chromium wire has been done in two different ways (Figure 3.6). The first is to directly wrap the beam with the nickel chromium. The drawback of this method is that large deformations of the beam at high temperatures can cause the nickel chromium wire to shift, causing uneven heating on subsequent trials or, in the worst case, a short. Another drawback is that the bar loses its cross-sectional shape rather easily. After a few operations the bottom of the beam is noticeably thicker than the top as the thermoplastic slowly flows down due to gravity. With out rigid supports to constrain the heating element and thermistor, this method is not suitable for experiments with large deformations or where actions must be repeated many times.

The other method is to use a jig like the one shown in Figure 3.6b. With this method, a lattice hinge is laser cut into the acrylic, giving it a very low stiffness on the whole, but allowing for locally rigid places to anchor the nickel chromium wire. Notches cut into the edges of the jig



(a)

(b)

Figure 3.5: Two different molds used for producing Polycaprolactone bars. (a) An acrylic mold lined with wax paper to help release the Polycaprolactone. (b) An aluminum mold where dowel pins are used to control thickness and a C-channel is used to control width. The tops of both molds are clamped down to provide the pressure needed to mold the Polycaprolactone into bars.

to keep the nickel chromium wire in a consistent position when the bar is deformed. Placeholders for thermistors and other components can also be cut into the acrylic, making their positioning more reliable. The main drawback here is that the jig needs to be placed between two sheets of Polycaprolactone during the molding process. This is accomplished using the aluminum mold shown in Figure 3.5b. Stacking the layers inside the C-channel (Figure 3.8), a light pressure is applied to the mold and heated to 60 C. This allows the Polycaprolactone to bond to the acrylic creating a laminate bar.

To monitor the temperature of the variable stiffness elements Negative Temperature Coefficient (NTC) thermistors are embedded into the bar. Temperature measurements are made from the thermistor using the Steinhart-Hart equation, a third order approximation of the relationship between the resistance and temperature. If the thermistor is not placed in the center of the beam, a model of the temperature propagation through the material must be used to estimate the temperature of the beam given the current reading of the thermistor (Section 3.4). The two different ways of embedding of the thermistor are also shown in Figure 3.6.

3.3.3 Maintaining Shape

In order to get the widest possible range of stiffness, the Polycaprolactone beam must be heated close to its melting point. At this temperature the molten beam can be easily deformed, and over time, gravity causes the Polycaprolactone to settle and flow downward. To counteract this I encase the beam in a thin layer of silicone. This flexible layer provides a gentle pressure to the molten Polycaprolactone, allowing the bar to maintain its general cross-sectional shape as it undergoes deformations.

A completed bar is shown in Figure 3.7 while an overview of the variable stiffness beam construction is shown in Figure 3.8.

3.4 Temperature Control

Each variable stiffness element is wrapped with a length of nickel chromium with a resistance of 62 Ω. Using a 12 V power supply each element consumes 2.3 W of energy. To accurately model the time required to reach a given temperature within the beam, I use SolidWorks to create a finite element model of the bar.

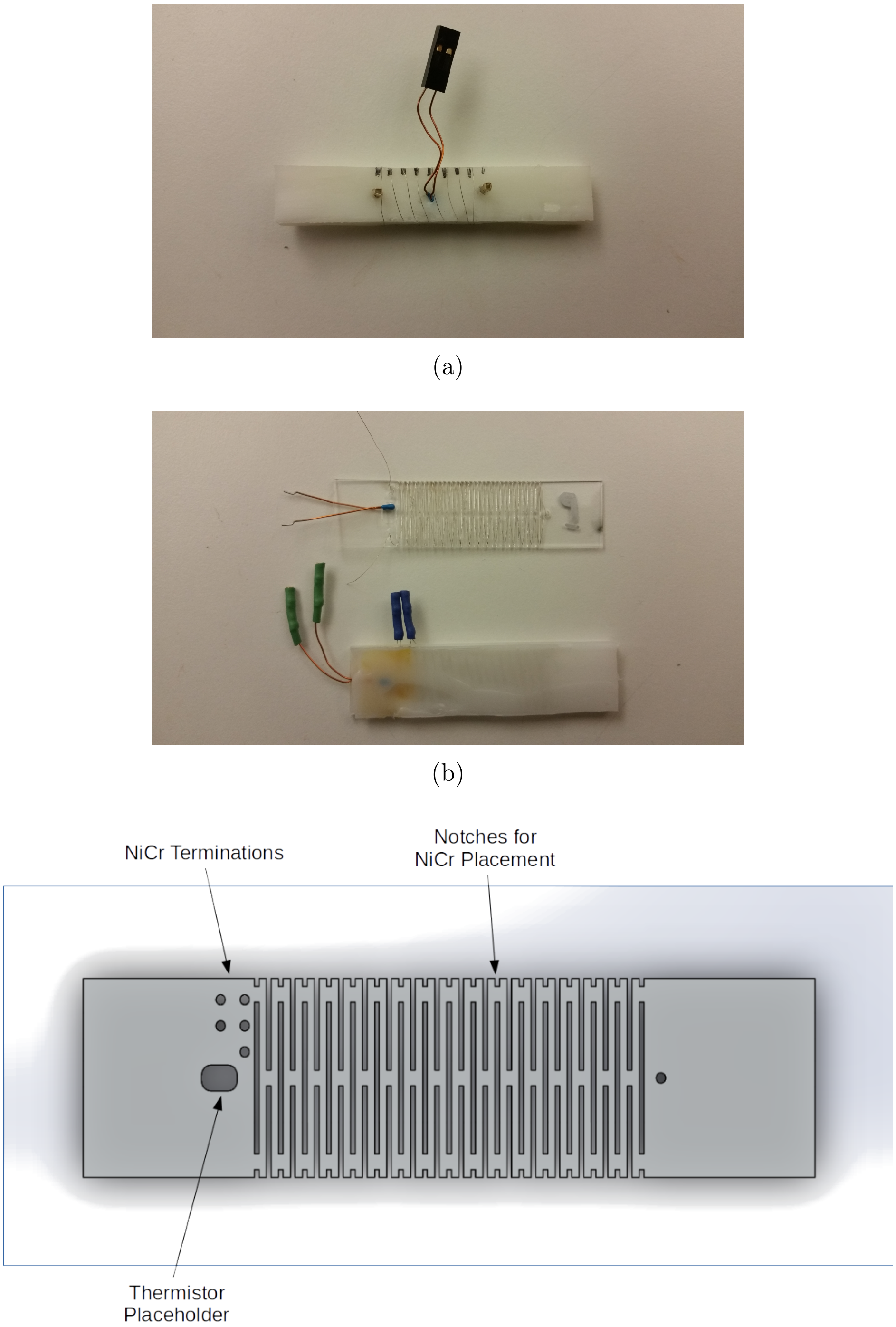
The finite element model is shown in Figure 3.9a and models the energy input from the nickel chromium wire as well as free convection on the exterior of the silicone rubber casing. The initial temperature is set as room temperature (22 C) and the simulation is run 700 seconds. I compare this model to five different variable stiffness elements as they are heated to 25 degrees above room temperature (Figure 3.9c). For our finite element model, we model Polycaprolactone with a specific heat of $350\frac{J}{kg \cdot K}$, a thermal conductivity of $0.1\frac{W}{m \cdot K}$, and a density of $1145\frac{kg}{m^3}$. Convection at the surface of the silicone casing is modeled as $13.5\frac{W}{m^2 \cdot K}$ A more detailed method for developing the temperature relationship analytically is found in [82].

3.5 Shape Locking

The variable stiffness element can also be locked into a new shape. Instead of using actuators to maintain a desired shape, the material could be locked into a new shape until a change is required, allowing the actuators to be powered down for possibly significant amounts of time [87]. With the thermoplastic base material this can be done by raising the temperature above the melting point and then letting the element cool. A drawback would be that in the molten state, the element can not support much load so the loading must be compensated for by other elements in the material.

3.6 Summary

In this section I discussed the variable stiffness elements that are used in the shape-changing robotic material. Unlike a conventional material, a variable stiffness element allows a varying response to loads placed on the element is able to vary its response to a given load by changing its stiffness. For the variable stiffness elements I have chosen to use Polycaprolactone, a low-melting point thermoplastic that is commercially available and safe to work with. The variable stiffness elements are formed using molds to shape the Polycaprolactone into bars. An acrylic jig provides support for nickel chromium wire and a thermistor while a silicone rubber encasing provides support to maintain the elements cross-section. Stiffness is changed by heating the element with the nickel chromium wire and monitoring the temperature using the thermistor. From this data I am able to determine how long it will take to heat and cool the elements in the material and discuss in Chapter 5 how this can be leveraged to set the shape of multiple elements at once, saving a significant amount of time over sequentially setting the curvatures of the elements. The relationship between temperature and curvature of the element is discussed in the next chapter as the relationship depends on the construction and application of the material. Lastly, I note that the shape of the variable stiffness elements can be permanently changed by elevating the temperature of the thermoplastic to its melting temperature and then letting it cool while the desired shape is maintained. This is also discussed in more detail in Chapter 2.3.



(a)

(b)

(c)

Figure 3.6: Two methods of embedding nickel chromium wire and thermistors into the Polycaprolactone beam. (a) The nickel chromium wire is wrapped directly around the Polycaprolactone with a thermistor embedded directly into the center of the bar. With out rigid support, these elements lose their positioning after a few actuations. (b) The top shows an acrylic jig used to wrap the nickel chromium wire and place the thermistor while the lower portion shows the jig fully embedded into Polycaprolactone. This rigid support of the elements allows for accurate positioning of the elements even after many cycles. (c) A close up of the lattice hinge showing the notches and placement features for the thermistor and nickel chromium terminals.

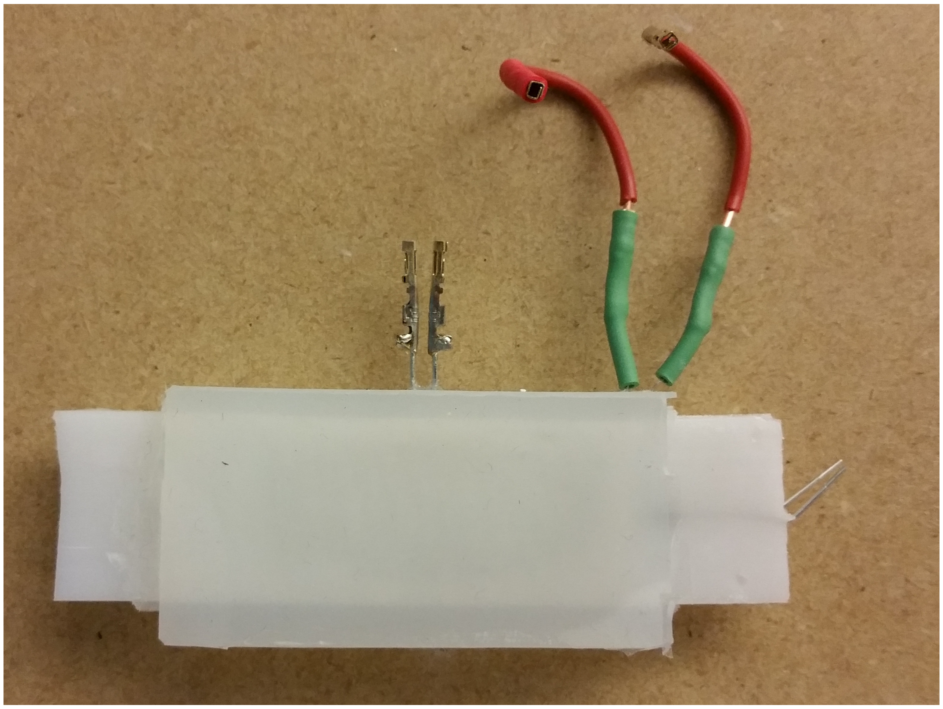
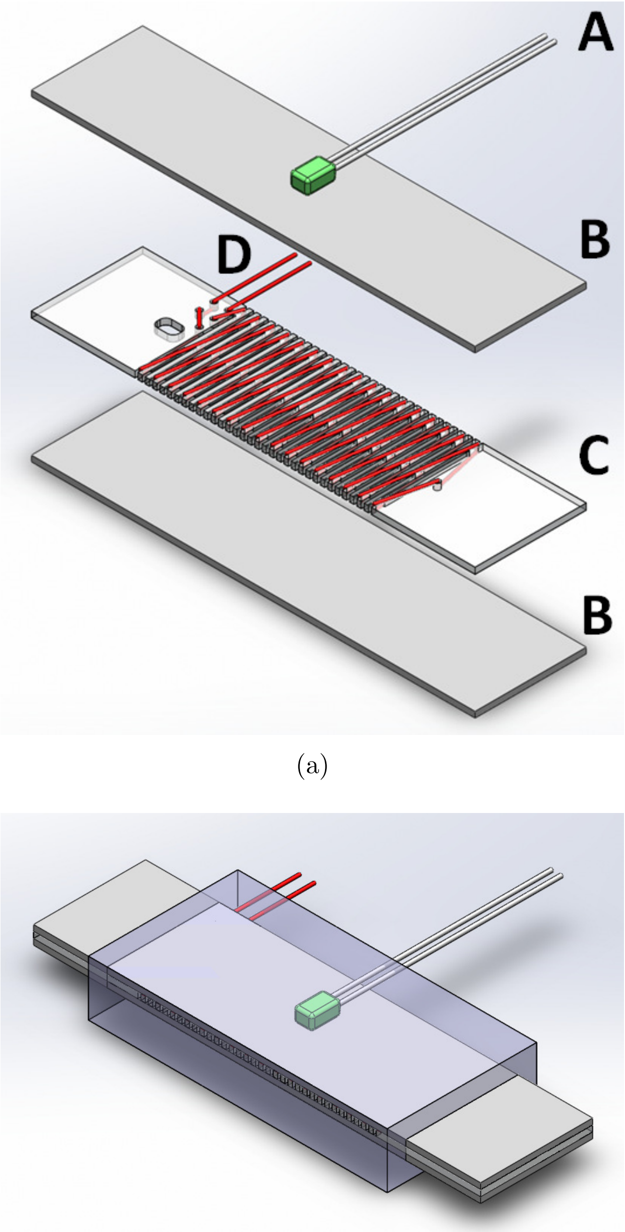


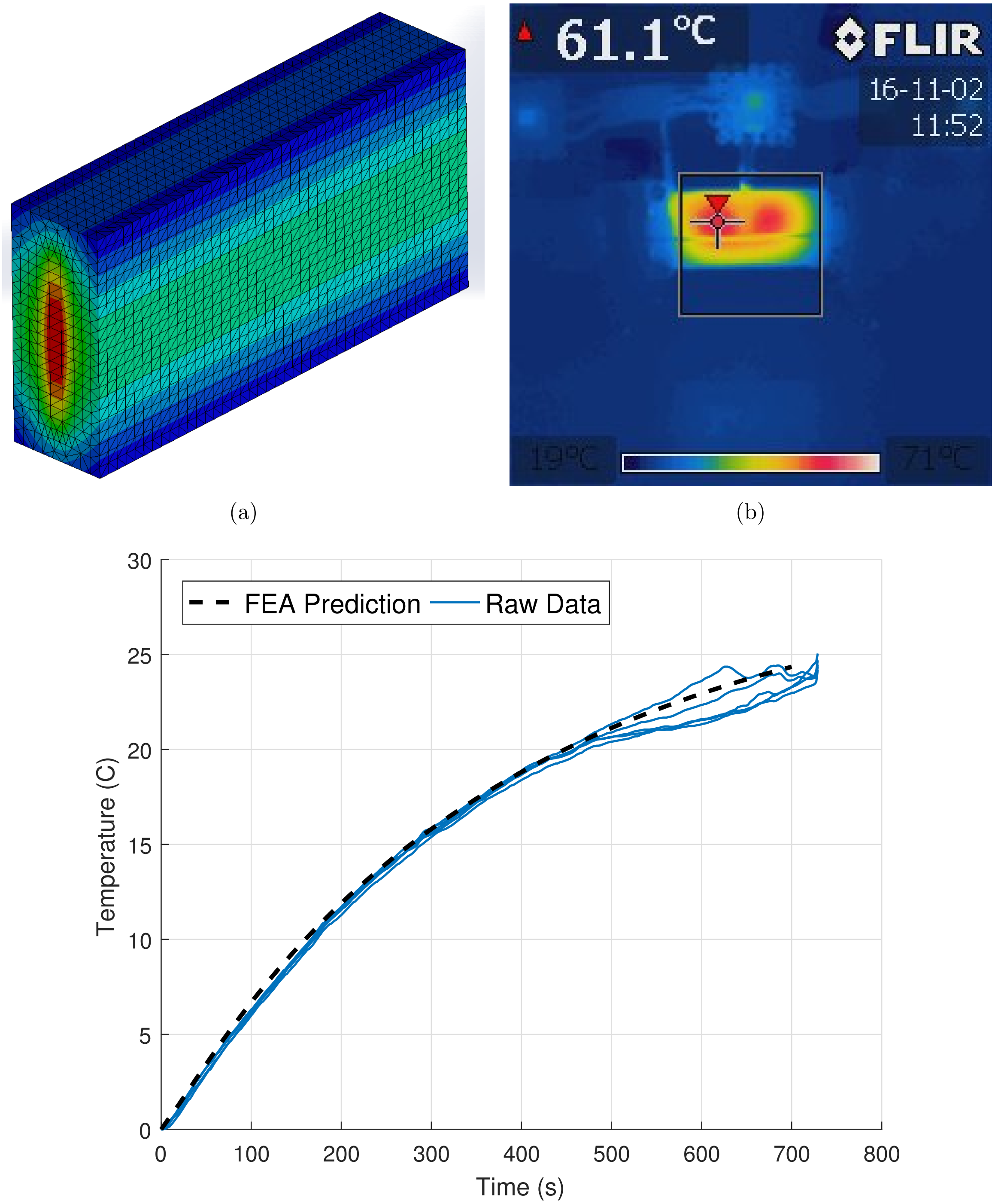
Figure 3.7: A Polycaprolactone based variable stiffness element. The element is encased in silicone rubber to help maintain the cross-section when the beam is at elevated temperatures and deformed.



(a)

(b)

Figure 3.8: This figure shows how the variable stiffness element is constructed. (a) The thermistor (A) monitors the temperature of the Polycaprolactone (B). The acrylic frame (C) is laser cut with a lattice hinge and provides a flexible support to the nickel chromium heating element (D). To create the laminate bar, the Polycaprolactone and acrylic layers are sandwiched together under a light compressive load and elevated temperature. (b) The bar is encased in a layer of silicon rubber to help maintain cross-section at elevated temperatures.



(a)

(b)

(c)

Figure 3.9: (a) The finite element model of the variable stiffness element created in SolidWorks. The acrylic jig, layers of Polycaprolactone and silicon casing are modeled with the energy from the Joule heater being applied at the interface of the acrylic and Polycaprolactone layers and free convection on the external faces of the silicon casing. (b) Thermal imaging taken during a heat up test of the element. (c) A comparison of the finite element model to the recorded data.

Chapter 4

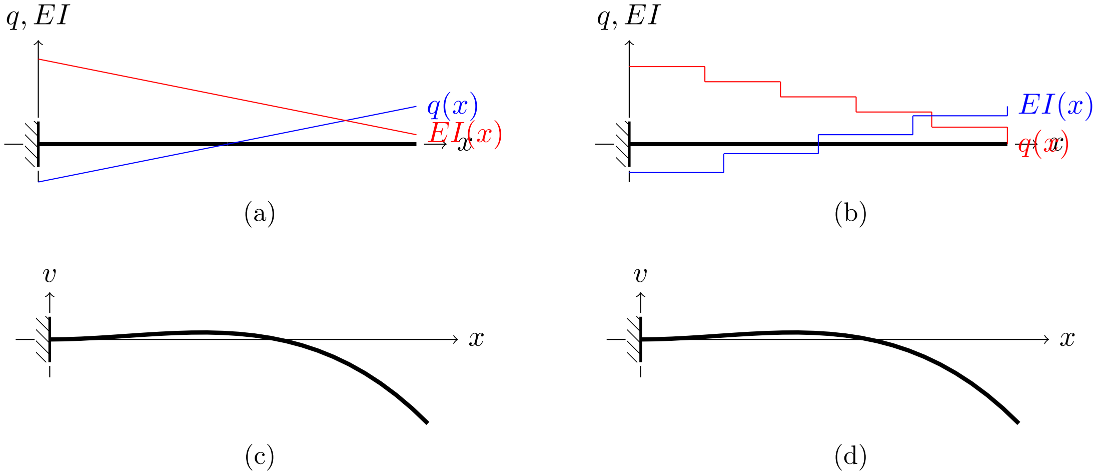
Shape Change Through Variable Stiffness

This chapter details how individual variable stiffness elements are manufactured into a robotic material. The principle of using variable stiffness for shape change is discussed followed by the details of how external loads are applied to the material. An overview of the embedded electronics and the fabrication of they system is presented and the chapter finishes with an overview of the experimental setup used to validate the system.

4.1 Principle of Operation

Referring back to the classical beam equation (Equation 3.1), a continuously varying stiffness and loading profile can be approximated by discretizing the beam's stiffness EI profile to achieve a desired deformation v from a given load q. This is the basis for our shape-changing material and is shown in Figure 4.1. Arbitrary shapes can be produced by controlling the stiffness of each variable stiffness element and the load that is applied to the beam as a whole.Constructing variable stiffness elements into a robotic material allows the designer to customize materials for specific applications through trade-offs between actuator sizes and the individual element's range of variable stiffness. The addition of computation also moves some of the burden of material development to algorithm development.

I reduce an intelligent material that changes its shape to a set of distributed computational elements which are connected to a small group of curvature sensors and variable stiffness actuators. External stimuli and disturbances are detected by the sensors and an appropriate response is



(a)

(b)

(c)

(d)

Figure 4.1: An illustration of using variable stiffness elements to create shape changing Robotic Materials. (a) A cantilevered beam with a continuously varying load $q(x)$ and stiffness $EI(x)$. (b) A cantilevered beam that has been discretized into five segments. (c) and (d) the resulting loading conditions for each case, respectively.

computed and applied by the actuators. Each computational element is able to communicate locally with neighboring elements to share sensor data and the computational burden of more complex data analysis tasks. We refer to this grouping of computation, sensing and actuation as a node of the material and the region over which the node has influence as an element of the robotic material.

In general, robotic materials could consist of one, two, or three dimensional arrangements of elements and be laid out in either a grid like fashion or an amorphous manner. In the simplest form, the forward kinematics can be calculated as communication flows through the material from one end to the other, each element updating their neighbors with transformation data. The inverse kinematics can be solved in an iterative fashion as communication flows up and down the material, local neighborhoods of elements collaborating to compute updates to their degrees of freedom then passing that information along to elements outside of the neighborhood where the process is repeated (Section 5.2).

For our robotic material we consider a beam with n elements laid out along the length of the beam with equal spacing. Each beam has two degrees of freedom, curvature and twist, allowing the beam to move through a three dimensional workspace. The following sections outline the forward and inverse kinematics for the shape changing beam.

4.2 Application of Loads

To apply the external loads needed to change the beam's shape we use two mechanical actuators as in [54] that pull tendons running along each side of the bar. The cables are held in place by supports that are placed at increments along the length of the beam, leading to a number of discrete sections. Tension applied to a tendon produces a moment M across each section and allows us to set either a positive or negative curvature across all of the segments at once. A schematic of this setup is shown in Figure 4.2.

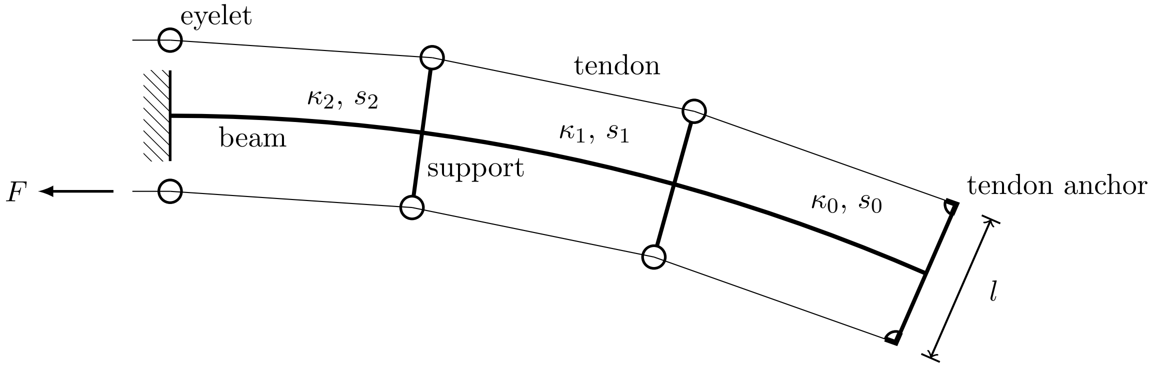


Figure 4.2: A schematic of how external loads are applied to the shape-changing robotic material. Tendons placed on either side of the beam are routed through eyelets in the supports. Tension on these tendons produces a distributed load over the entire length of the beam.

4.3 Printed Circuit Boards

Each variable stiffness element is outfitted with a printed circuit board that allows local communication, sharing of power, and computation of curvature updates. The board utilizes an ATXMega128a4u microcontroller and allows for up to four connections to neighboring elements. An over view of the board and the main components is shown in Figure 4.3.

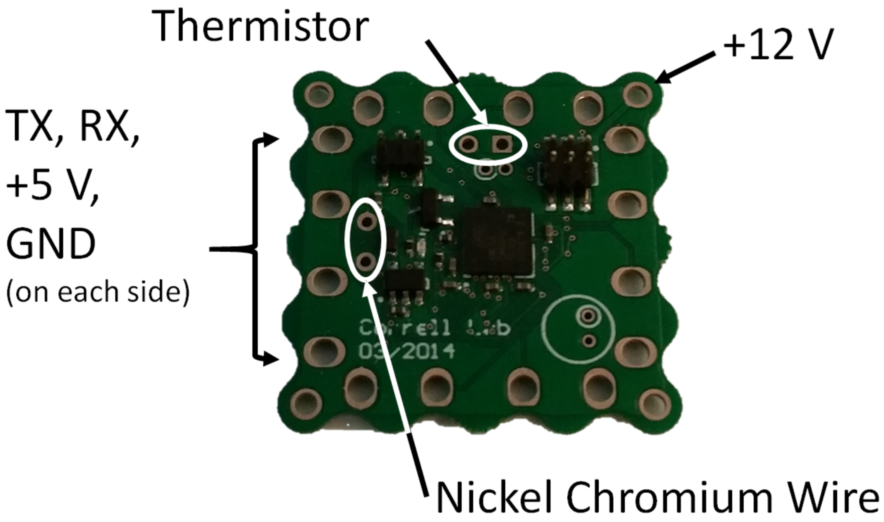


Figure 4.3: The printed circuit board used in each variable stiffness element. Each board is able to communicate and share power with up to four neighbors and has connections for the thermistors and nickel chromium heater.

The software is written such that the elements are in one of two modes, either calculating the curvature and twist updates or executing the update. When in update calculation mode, the elements are either in an master, support, or idle state. In the master state, the element communicates with neighboring elements to get their state information and calculates their curvature and twist updates. In the support state the elements are sending needed information to the master element and awaiting for their updated curvature and twist values. In the idle state, the element is outside of the neighborhood and will not be receiving an update to its curvature or twist.

Once the curvature and twist values have been calculated to reach a desired pose, the elements switch over to the execute mode where they are either in the execute or idle state. In the execute state the elements are heating up and changing their curvatures and twists. In the idle state they are waiting for the signal to start executing their updates.

4.4 Fabrication

The variable stiffness elements are connected together into a beam using ribs that connect two variable stiffness elements. The ribs act as thermal isolation between elements as well as supports for routing the tendons of the external actuators along the beams length. To move smoothly on the table top, the ribs are fitted with ball bearings on their feet. Printed circuit boards are attached to each element and connected to neighboring elements. Figure 4.4 shows the internal components of the beam before they are embedded into the structural material.

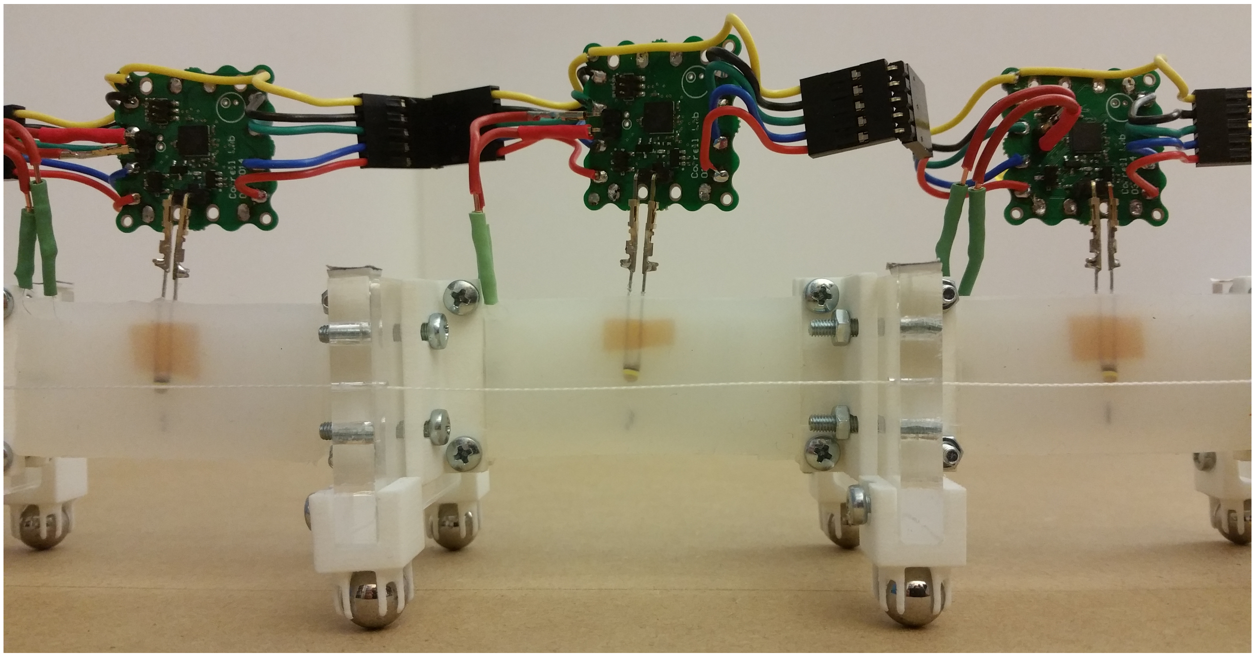


Figure 4.4: The variable stiffness elements are assembled into a beam using acrylic ribs. Small printed circuit boards are attached to each element and are connected to neighboring elements.

The final step in the creation of the robotic material is to embed these components into a structural foam. A laser cutter is used to remove material from the foam sheets (Figure 4.5) so that the components can be embedded directly into the foam. I have chosen to cut lattice hinge elements into the foam, favoring a more flexible design at the expense of structural rigidity. This

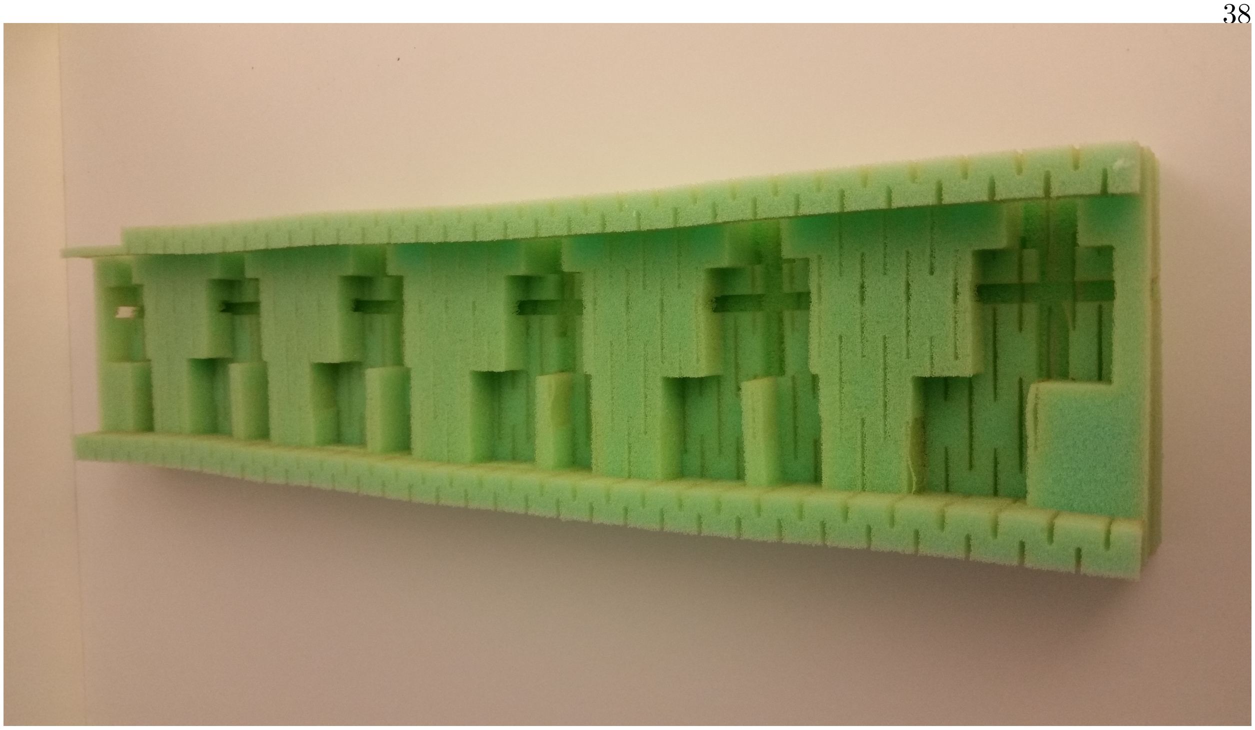


Figure 4.5: The last step in creating the robotic material is to laser cut the structural foam into which the variable stiffness elements will be embedded. For this construction I have included lattice hinge cuts into the foam to maximize the flexibility of the beam. I note that this is an application specific trade-off, mobility vs. strength.

is another area where design can be customized for a particular application. Figure 4.6 shows the completed shape-changing robotic material.

4.5 Experimental Setup

I conducted experiments on a five element robotic material beam. The beam is mounted to the table using a fixed rib section. The tension is applied to 65 pound test spectra fishing line using two Dynamixel RX-64 servo motors outfitted with pulleys. Fiducials placed on the rib segments are used to track the position of the beam and the curvatures through each section. And lastly, a fan is used to speed up the time required to cool the beam segments back to room temperature. The setup is shown in Figure 4.7.

The joints that connect the variable stiffness elements together (Figure 4.4) add a length d to each end of the variable stiffness elements that will remain straight and not change in curvature.

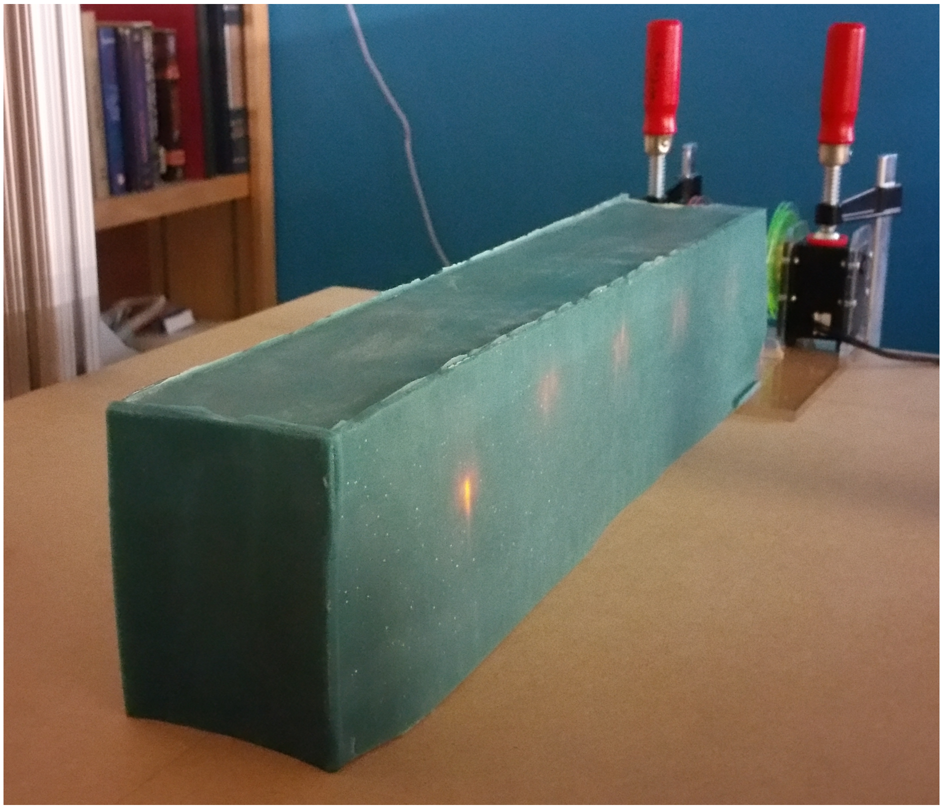


Figure 4.6: The completed shape-changing robotic material. The variable stiffness elements and computational nodes have been embedded into structural foam, resulting in a single material.

With the curvature being estimated through fiducials mounted on the rib, a correction to the curvature must be made as the elements do not take this straight portion into account. Examining the geometry in Figure 4.8 the relationship between the two curvature measures is easily calculated and shown in Equation 4.3.

$$a = \frac{d}{\tan\theta} \tag{4.1}$$

$$b = \frac{d}{\sin\theta} \tag{4.2}$$

$$r_2 = r + a + b = r + d\left(\frac{\cos(\theta) + 1}{\sin(\theta)}\right) \tag{4.3}$$

This relation is also useful in the control of the beam. The curvature is directly related to the tendon length in each segment. For a single segment, its change in length is related to curvature through Equation 4.4 and is shown pictorially in Figure 4.9. Setting elements sequentially, tracking

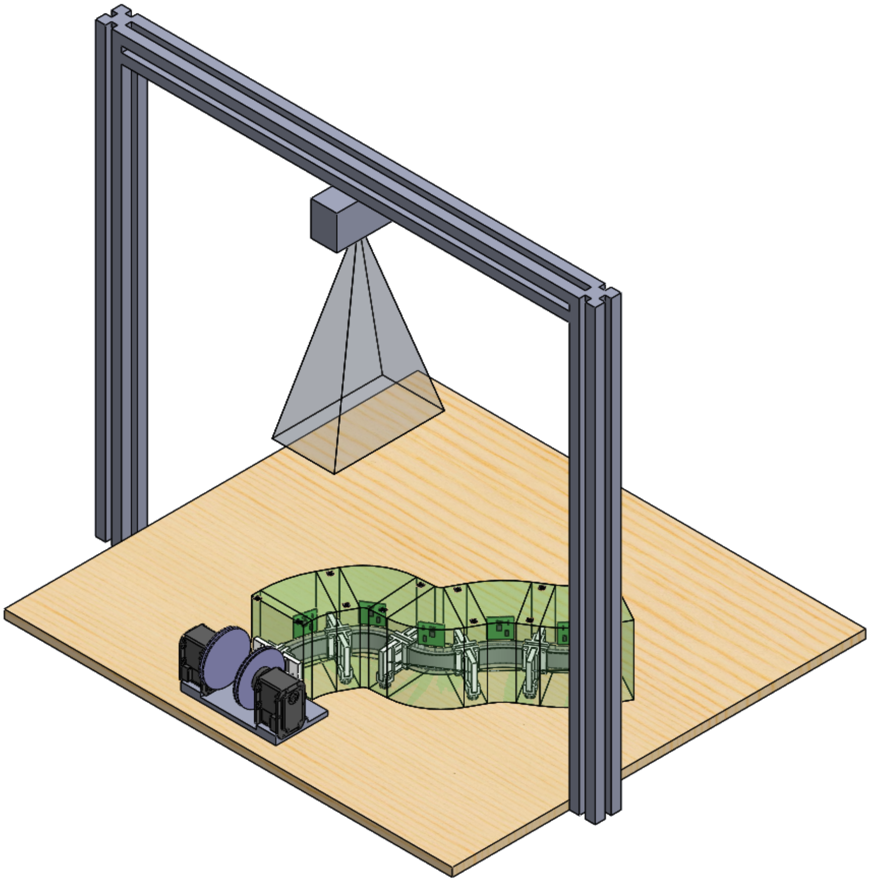


Figure 4.7: The experimental setup used to validate the shape-changing beam. The beam is mounted to a base plate that also houses the two Dynamixel actuators. A camera is placed above the beam to record the beam's shape and the segment curvatures during the trials.

tendon length could be used as an open loop control for the curvature of each element.

$$c_{inner} = 2(r_2 - d)\sin(\frac{\theta}{2})c_{ouiter} = 2(r_2 + d)\sin(\frac{\theta}{2}) \tag{4.4}$$

Distributing the computation and sensing throughout the material allows our system to be robust to manufacturing defects and interference from the environment. Manufacturing defects and differences between variable stiffness elements are overcome by testing the beam under a constant load. The friction in our table top setup is an external force that is implementation specific and is also easily accounted for in the distributed nature of the material. The elements at the tip of the beam experience a greater curvature than the elements at the base for a given load, in this case a

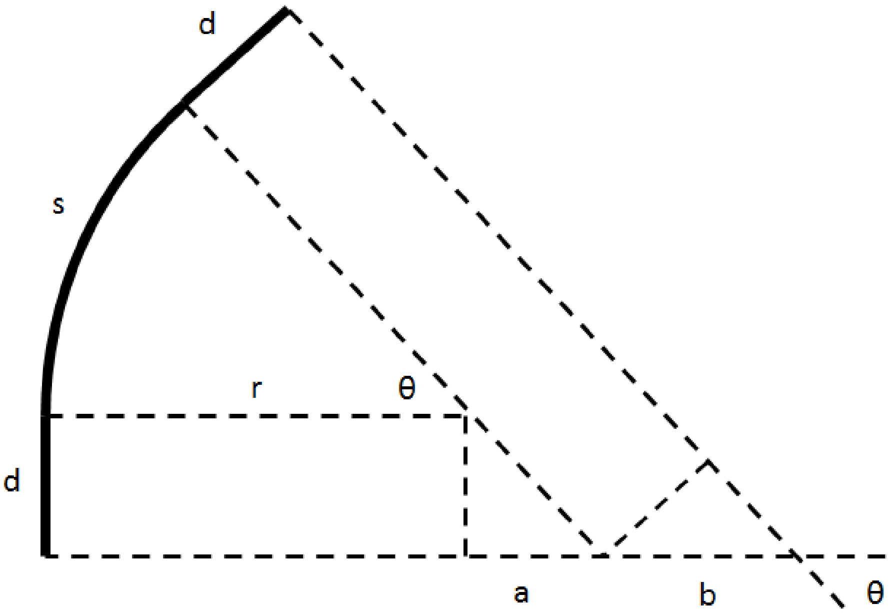


Figure 4.8: The ribs that connect the variable stiffness elements add a small straight section to the element profile. Because the fiducials are mounted to the ribs, this straight section must be accounted for and a relationship between the measured curvature and the curvature through the variable stiffness element must be established. This figure shows that the relationship is established with simple geometric relationships.

500 g weight hung from one of the tendons. Figure 4.10a shows the curvature versus temperature profile of the beam and, equivalently, the range of motion of the beam. From these plots we see that elements three and four most likely suffer from a manufacturing defects as they experience much larger curvatures when pulled in one direction than the other.

4.6 Summary

I've shown the principle of operation for our shape-changing beam. Being able to control the stiffness of the material and the loading of the beam allows us to induce any desired shape into the beam. To apply the external moments we use two tendon based actuators and leverage the results from the field of continuum robotics. The embedded electronics allow for local communication and power sharing with up to four neighbors and the simple state machines used during each phase of the beam's operation are described.

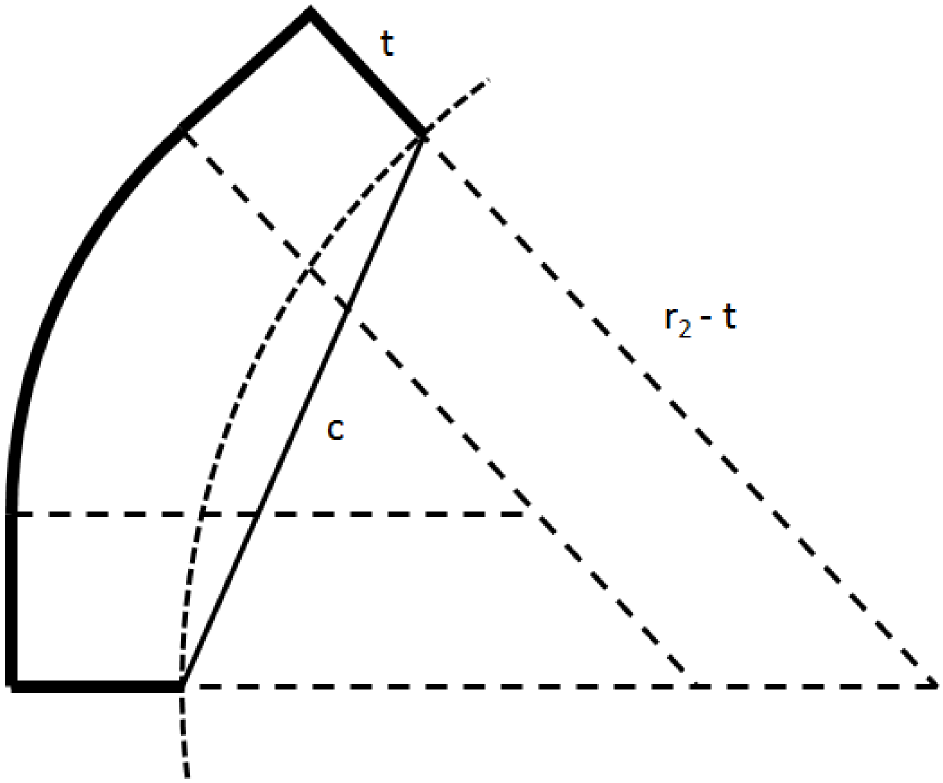
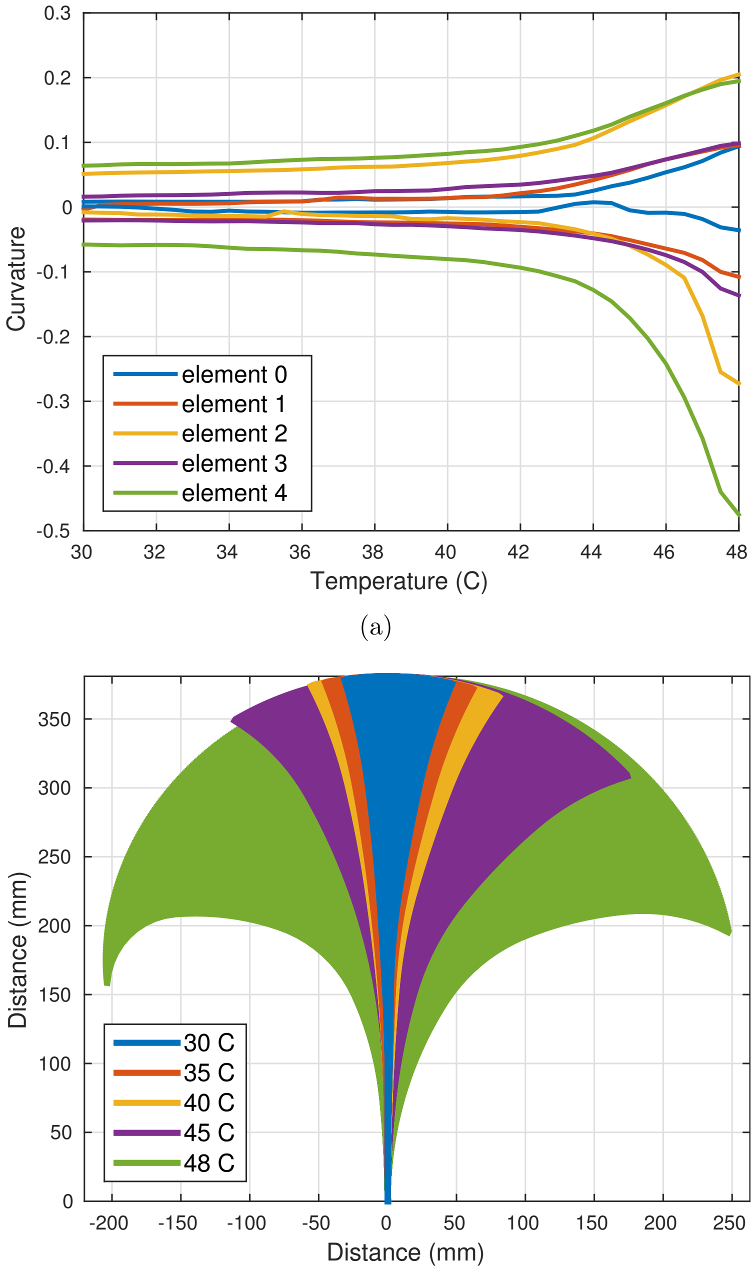


Figure 4.9: For a single element, the change in length of the tendon can be directly related to the curvature of the element. For multiple elements the tendon length can be tracked and used for open loop control of the elements curvatures.

The shape-changing beam is constructed in a modular fashion where each variable stiffness element is connected together with a rigid rib. The experimental demonstration is a five element beam that is limited to changing curvatures in each section. Finally, the variable stiffness elements, printed circuit boards and interconnects are embedded in structural foam to create a composite material. The last section of this chapter describes the table top experimental setup that we use to validate the shape-changing robotic material and the tests conducted to characterize the robotic material.



(a)

(b)

Figure 4.10: To evaluate the curvature that each element can achieve under load, we heat the entire beam while applying a constant load to one of the tendons. A) Average curvature versus temperature profiles for static load tests using a 500g weight. B) Shown in an alternate form, the workspace of our shape changing beam under the constant load.

Chapter 5

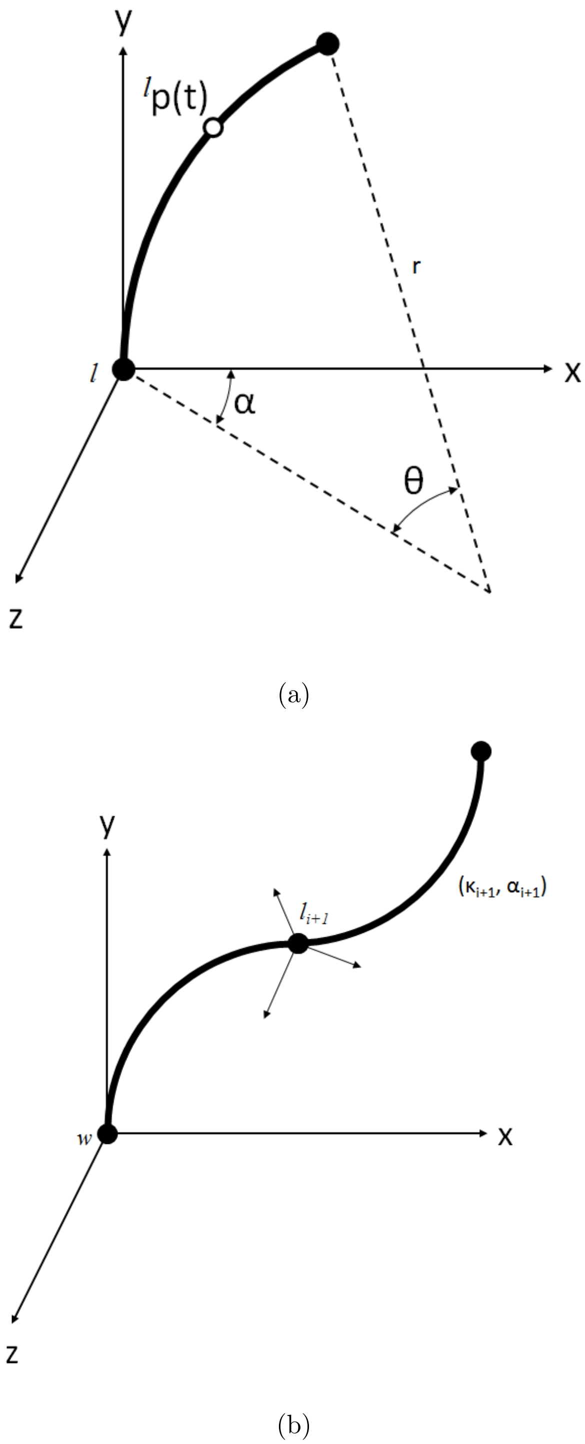
Distributed Control

This chapter described the distributed algorithm used for solving the inverse kinematics problem and how the variable stiffness elements can be used to lock shapes into place. The distributed inverse kinematics algorithm can be executed on a network of n micro-controllers — one for each variable stiffness element — distributed across the beam's length. The proposed algorithm is not bioinspired, but is derived from the continuous curvature approach commonly used in continuum robotics. Reduced to a set of feedback controllers that could possibly be represented by an artificial neural network, it is a biologically plausible architecture that leads us to a scalable shape-changing robotic material.

5.1 Forward Kinematics

In order leverage the results and continuing research in the field of continuum robotics [58,95] we will apply the piecewise constant curvature model to the beam and dispense with the piecewise differential equations described in Chapter 3. We assume that each element of the robotic material can control its curvature $\kappa = r^{-1}$ and twist α as shown in Figure 5.1a. The piecewise constant curvature model assumes that the curvature and twist are constant through each element. Using this assumption we show the forward and inverse kinematics and derive a method for distributing the computation among the elements of the robotic material.

The forward kinematics can be computed by a series of rotations and translations taking into account each element's curvature, twist and length. Looking first at only the curvature, the position



(a)

(b)

Figure 5.1: a) The local coordinate system for each element in the robotic material. r is the radius of curvature and α is the angle of twist. $\kappa = \frac{1}{r}$ and the segment length $s = \frac{\theta}{\kappa}$. b) Calculating the transform from the element's base to the element's tip allows the forward kinematics to be calculated as a string of rotations.

of a point ${}^{l}\mathbf{p}_i(t)$ along the length of any element i described in the element's local coordinate frame l is given by Equation 5.1. Where $\kappa = r^{-1}$ is the element's curvature, s is the element's length and $0 \leq t \leq 1$ is the position along the element's length. Then the end point of the element, $p_i(t = 1) = p_{i+1}(t = 0)$, has an orientation of $\theta_i = \kappa_i s_i$.

$$ {}^{l}\mathbf{p}_i(t) = \begin{bmatrix} {}^{l}x_i \\ {}^{l}y_i \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \kappa_i^{-1} \left(1 - \cos\left(\kappa_i s_i t\right)\right) \\ \kappa_i^{-1} \sin\left(\kappa_i s_i t\right) \\ 0 \\ 1 \end{bmatrix} \tag{5.1} $$

To describe the end point of this element in the world coordinate frame w, the point must be rotated by the element's twist, α, as in Equation 5.2. If we consider a second element added to the robotic material, as in Figure 5.1b, we must construct a homogeneous transformation between the second element's local frame and the world frame. Such a transformation between the world frame and element i's local coordinate frame is shown in Equation 5.3. The location of the robotic material's end point is then the successive transformations through each element in the material (Equation 5.4).

$$ {}^{w}\mathbf{p}_i(t = 1) = \begin{bmatrix} R_y(\alpha) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} {}^{l}\mathbf{p}_i(t = 1) \tag{5.2} $$

$$ \mathcal{H}_i^w = \begin{bmatrix} & {}^{w}x_i \\ R_z(\theta) \cdot R_y(\alpha) & {}^{w}y_i \\ & {}^{w}z_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{5.3} $$

$$ {}^{w}\mathbf{p}_n(t = 1) = \mathcal{H}_0^w \cdot \mathcal{H}_1^0 \cdots \mathcal{H}_n^{n-1} \cdot {}^{l}\mathbf{p}_n(t = 1) \tag{5.4} $$

The forward kinematics can be distributed throughout the material where each element computes their individual transformation matrix and the position of the end can be computed through

$\mathcal{O}(n)$ communications and $\mathcal{O}(n)$ computations to propagate and compute the homogeneous transformations. Additionally, each element could store $\mathcal{H}_{i-1}^{w}$ and $\mathcal{H}_{n}^{i+1}$ so that the forward kinematics can be calculated locally, updating neighbors only when a change to curvature or twist is made.

5.2 Inverse Kinematics

The inverse kinematics problem is to find the curvature for each segment such that the end of the robotic material reaches some goal pose $\mathbf{g} = [x, y, z, \theta, \phi, \psi]$, where $[x, y, z, [\theta, \phi, \psi]$ is the six degree of freedom pose of the tip in three dimensional space. Solving the forward kinematics equations directly becomes infeasible for more than a handful of elements. For larger problems, iterative methods to approximate the solution are commonly used, for example the pseudoinverse or damped least squares methods [13]. Done in a naive centralized manner, taking the pseudo inverse of the Jacobian is $\mathcal{O}(n^3)$ time using Gauss-Jordan elimination and can be reduced to $\mathcal{O}(n^{2.373})$ time using optimal methods. Distributing this computation has been shown to lower the time to $\mathcal{O}(\log n)$, however $\mathcal{O}(n^4)$ computers are required [9].

Neither of these methods are suitable for scalable embedded systems with limited processing capabilities, so instead we use the damped least squares method [13] which allows for computing the update in $\mathcal{O}(n)$ time. Let $\mathbf{s}(\kappa, \alpha)$ be the current pose of the robotic material's tip as a function of the element curvatures and twists, then an update to these values is calculated using Equation 5.5 and solved repeatedly until the delta between the tip pose and the goal pose is below some minimum specified threshold or another exit criteria is reached.

$$\Delta(\kappa, \alpha) = J^T \left(JJ^T + \lambda^2 I\right)^{-1} (\mathbf{g} - \mathbf{s}(\kappa, \alpha)) \tag{5.5}$$

J is the $6 \times 2n$ Jacobian matrix, λ is the scalar damping constant and $\mathbf{s}(\kappa, \alpha))$ is the current pose of the beam's tip. With this formulation, the inverse is always a 6×6 matrix, so the time complexity of this operation is $\mathcal{O}(n)$ based on the computation for the matrix multiplications.

This operation, however, must be done repeatedly until a solution is found and numerical computation of the Jacobian is not necessarily linear. The Jacobian is shown in Equation 5.6. To compute this numerically, one of the degrees of freedom must be changed and the change in pose calculated through repeated calls to the forward kinematics function. Multiplication of every element's transformation matrix is an $\mathcal{O}(n)$ operation and it must be computed for every degree of freedom, $2n$ in this case since each element can control its curvature and twist, resulting in a time complexity of $\mathcal{O}(n^2)$. This can be reduced by precomputing and storing the various transformations, but the problem becomes one of storage and a trade off between communication, storage and computation in the distributed system must be made [36].

$$J = \frac{\partial \mathbf{s}(\kappa)}{\partial \kappa} = \begin{bmatrix} \frac{\partial x}{\partial \kappa_0} & \cdots & \frac{\partial x}{\partial \kappa_n} & \frac{\partial x}{\partial \alpha_0} & \cdots & \frac{\partial x}{\partial \alpha_n} \\ \frac{\partial y}{\partial \kappa_0} & \cdots & \frac{\partial y}{\partial \kappa_n} & \frac{\partial y}{\partial \alpha_0} & \cdots & \frac{\partial y}{\partial \alpha_n} \\ \frac{\partial z}{\partial \kappa_0} & \cdots & \frac{\partial z}{\partial \kappa_n} & \frac{\partial z}{\partial \alpha_0} & \cdots & \frac{\partial z}{\partial \alpha_n} \\ \frac{\partial \theta}{\partial \kappa_0} & \cdots & \frac{\partial \theta}{\partial \kappa_n} & \frac{\partial \theta}{\partial \alpha_0} & \cdots & \frac{\partial \theta}{\partial \alpha_n} \\ \frac{\partial \phi}{\partial \kappa_0} & \cdots & \frac{\partial \phi}{\partial \kappa_n} & \frac{\partial \phi}{\partial \alpha_0} & \cdots & \frac{\partial \phi}{\partial \alpha_n} \\ \frac{\partial \psi}{\partial \kappa_0} & \cdots & \frac{\partial \psi}{\partial \kappa_n} & \frac{\partial \psi}{\partial \alpha_0} & \cdots & \frac{\partial \psi}{\partial \alpha_n} \end{bmatrix} \tag{5.6}$$

We distribute the inverse kinematics problem to the robotic material by reducing the number of elements that are allowed to change their curvature and twist simultaneously. The $2n$ degree of freedom system is reduced to a $2m$ degree of freedom system where $1 \geq m \leq\leq n$ and the $n - m$ elements are held rigid during the update. In this scheme, an m element neighborhood identifies an element to compute the groups' curvature and twist updates using Equation 5.5. Once the updates are computed, the m-element neighborhood moves down the length of the beam and the process is repeated (Figure 5.2). Evaluation of Equation 5.5 is still linear, but now numerically computing the Jacobian is reduced to linear time as well since the forward kinematics function is called only m times. Further speed ups are possible, but again communication, storage and computation in the distributed system must be evaluated. Results comparing this method to traditional centralized approach are presented in a later section.

Figure 5.3 shows the communication model for a 3-element neighborhood. For a neighbor-

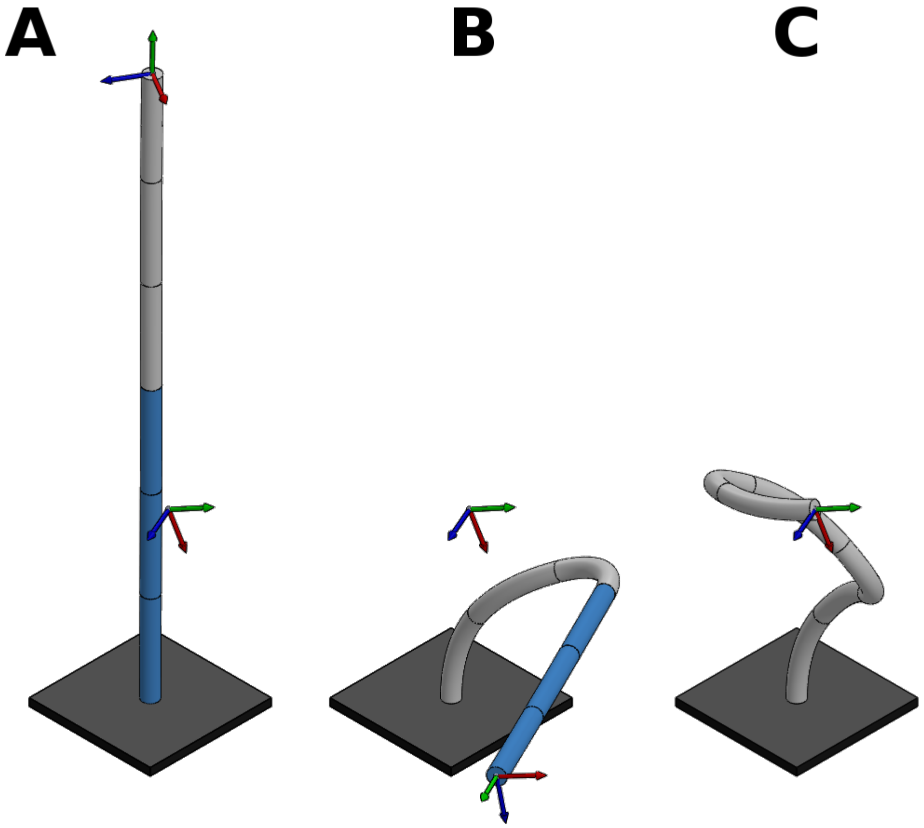


Figure 5.2: The distributed inverse kinematics method used in our robotic material beam. Instead of solving the n-link inverse kinematics, we reduce the problem to sets of m-link neighborhoods (shaded blue). Shown here with $n = 6$ and $m = 3$. As communication flows down the beam, each m-link neighborhood computes and executes updates to its degrees of freedom then propagates this information along the length of the beam. Elements outside of the neighborhood (shaded gray) are considered rigid and their degrees of freedom are not updated. The coordinate systems shown are the end effector and goal pose. A) The 3-link neighborhood at the base computes and executes updates for its degrees of freedom. B) The communication flows down the length of the beam to the next neighborhood. C) This process continues until the goal, or some other exit criteria, is reached.

hood of m-elements, $2(m-1)$ communications are needed to request and receive the curvature, twist and transformation data from each element in the neighborhood. Once the updates have been completed, $m-1$ communications are needed to update the curvatures and twists in the neighborhood and $n-1$ communications are needed to update the transformation matrices of each element. Since we assume $n >> m$ for the embedded system, the communication complexity is on the order of $\mathcal{O}(n)$.

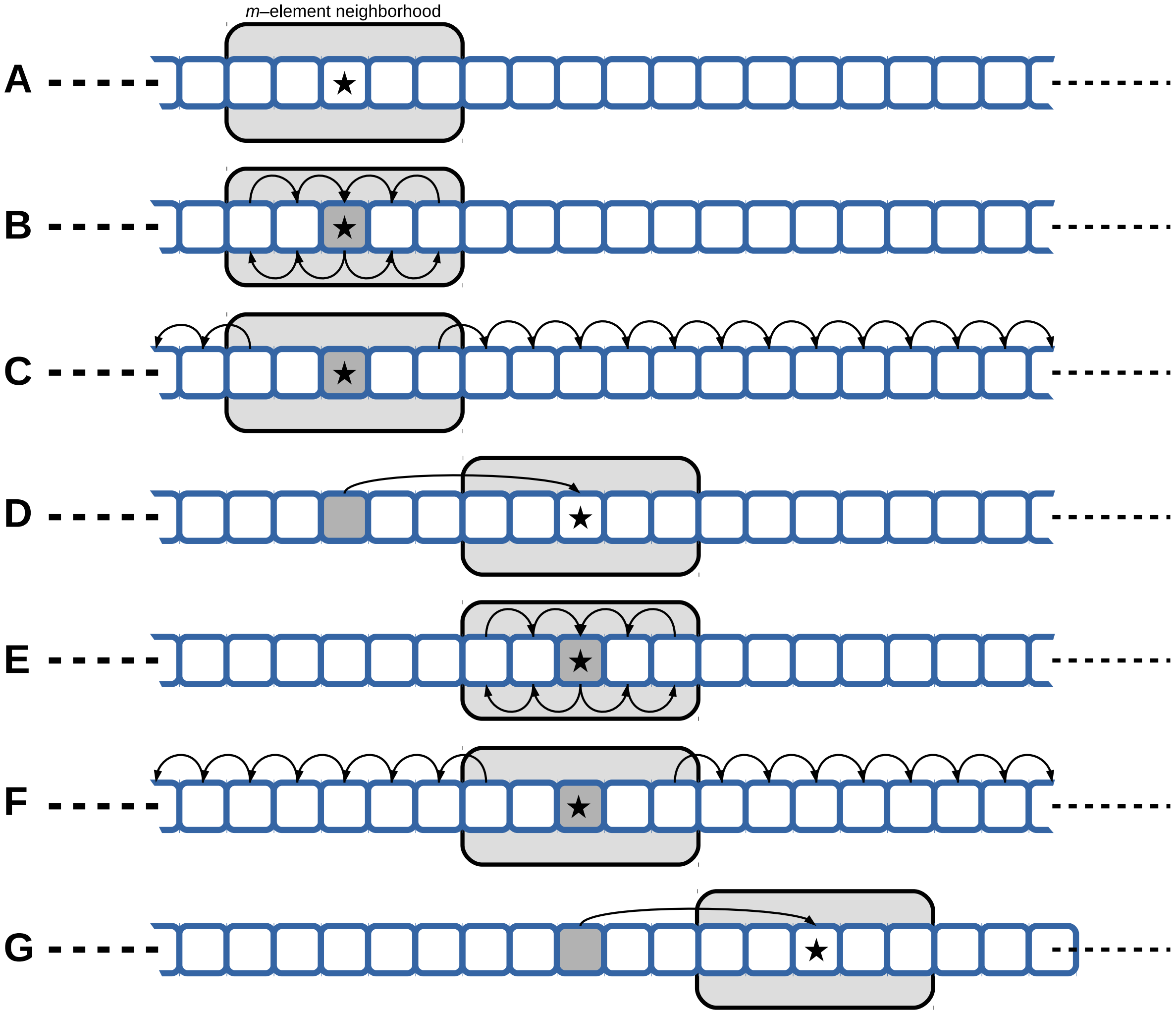


Figure 5.3: The communication model used in this robotic material. A) an m-element neighborhood is established and the central element is identified (labeled with a star). B) The central element collects state information from the neighborhood, computes updates to these degrees of freedom and sends this information back to the elements in the neighborhood. C) The information is propagated along the length of the beam. D) Communication flows down the length of the beam and a new m-element neighborhood is established. E - G) The process continues along the length of the beam.

5.3 Evaluation

First I examine the distributed computation of the shape-changing beam. Figure 5.4 shows the advantages of distributing the inverse kinematics problem through out the material. With our method we see linear growth when computing the update expressed in Equation 5.5, compared to the exponential growth seen in the centralized method. For each trial the beam starts from a straight profile and tries to align with a reachable, random goal. For these trials the goal is the position and orientation of the beam's tip. Trials were terminated when the goal was met within a given tolerance, a local minimum was reached or the trial exceeded a maximum number of iterations.

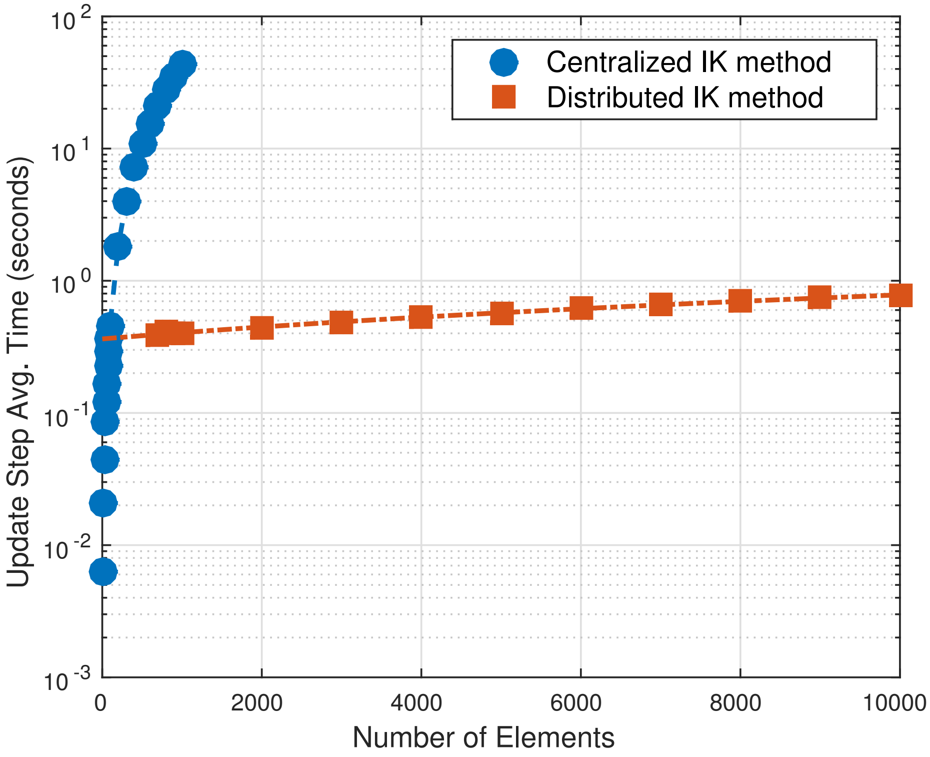


Figure 5.4: This plot compares the average run time using a centralized approach for the inverse kinematics (blue) and the distributed method (red) described above. The distributed method grows with $\mathcal{O}(n)$ compared to the $\mathcal{O}(n^2)$ growth of the centralized method.

In the above experiment, neighborhood size was held constant at $m = 101$ elements. Increasing the size of the neighborhood with the number of elements will increase the time complexity to $\mathcal{O}(n^2)$ as shown in Figure 5.5. This represents another trade-off between computation and communication which will depend on the system architecture and communication protocol chosen for they system.

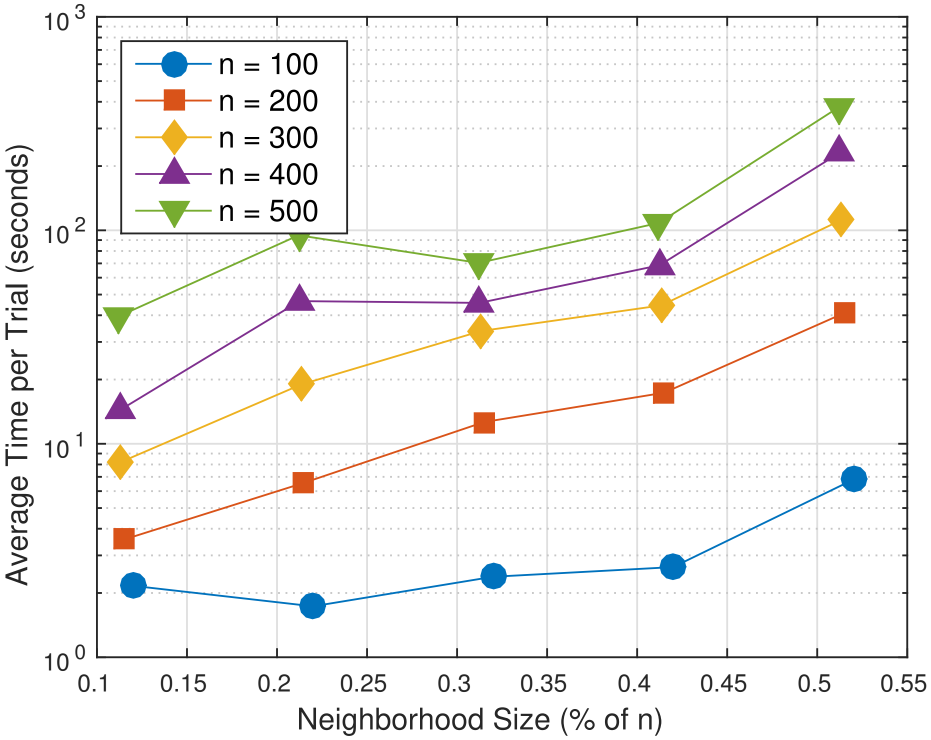


Figure 5.5: Changing neighborhood size as the number of elements in the robotic material grows pushes the time complexity toward exponential growth.

The last aspect of the distributed computation that I examine in the amount of overlap in the neighborhoods. Increasing the overlap of the neighborhoods will also make the time complexity move toward exponential. Again there is a trade-off as a higher overlap was observed to create smoother profiles in the final solution of the beam.

To compare the centralized solution to the distributed solution I chose ten beam configurations and used the tip positions as goals for each method. Figure 5.7a shows the ten different beam configurations used in this comparison. For each configuration I ran the centralized method and distributed method 100 times and compared how each did at finding a solution for the given goal pose. Configurations a through f represent configurations where curvature and twist are constant or vary continuously over the length of the beam whereas configurations g through j represent configurations where the curvature and twist are discontinuous. Configurations d and e represent the extremes of the beam's workspace. Figure 5.7b shows that the solutions provided by the dis-

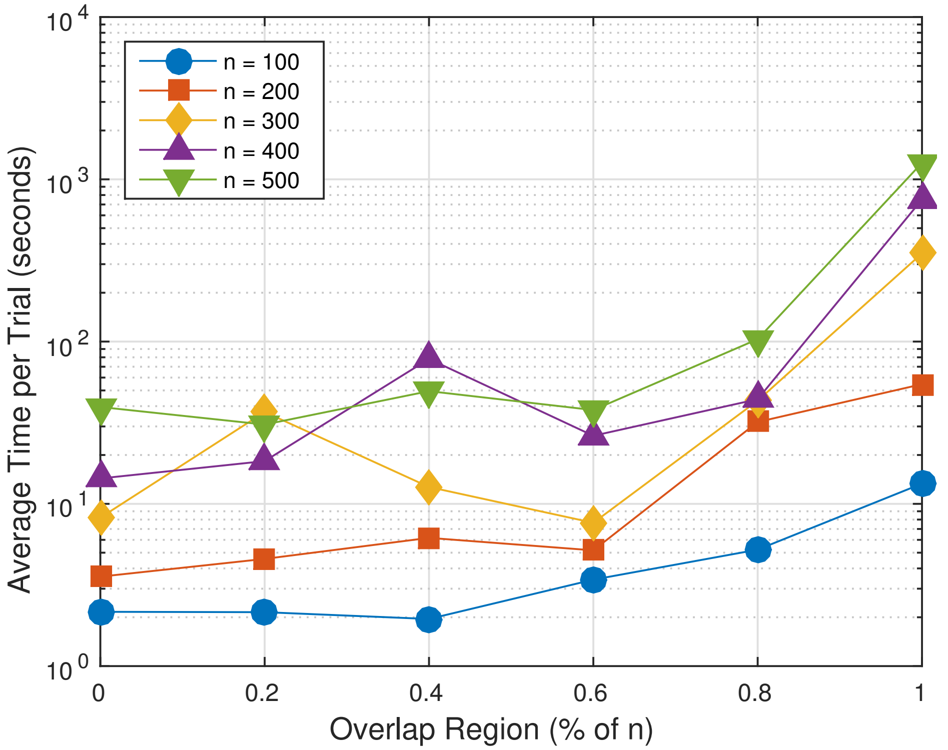
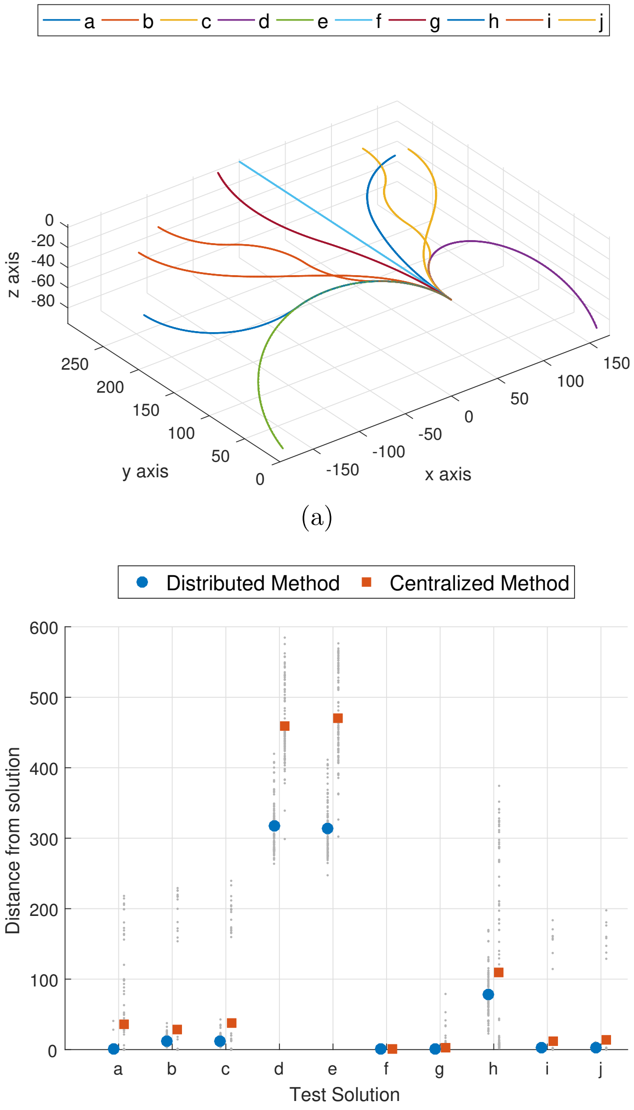


Figure 5.6: One alteration to the distributed algorithm presented is to allow the neighborhoods to overlap. This seems to provide smoother solutions, much like those found in the centralized case, but also makes the time complexity tend toward an exponential.

tributed algorithm are as good, or slightly better in a few cases, than the centralized version of the algorithm.

Figure 5.8 shows the solutions found for case *g*. The solutions for the distributed case all fall on straight line while the solutions from the centralized case are clustered is a tight grouping near the goal. The solutions from the distributed case also cover a larger space of the beam's self motion while the centralized results seem to tend toward they same solution. When a solution is not found, the failure cases from the centralized version are much more erratic and farther away from the goal pose. The failures from the distributed case are still in line with the solution and remain closer to the goal pose.



(a)

(b)

Figure 5.7: (a) The ten configurations used to evaluate the effectiveness of the distributed inverse kinematics algorithm compared to the centralized version. Configurations a through f have constant or continuous variations of curvature and twist while the remaining configurations have discontinuous variations in curvature and twist. (b) A comparison of the solutions found for each case. The light gray dots show the individual results of each trial while the large dots show the average. In general the distributed algorithm does as well or slightly better than the centralized version.

5.4 Shape Locking

In some cases it might be desirable to lock the robotic material into a new shape. One advantage of this ability would be the actuators maintaining this shape may be powered off, saving

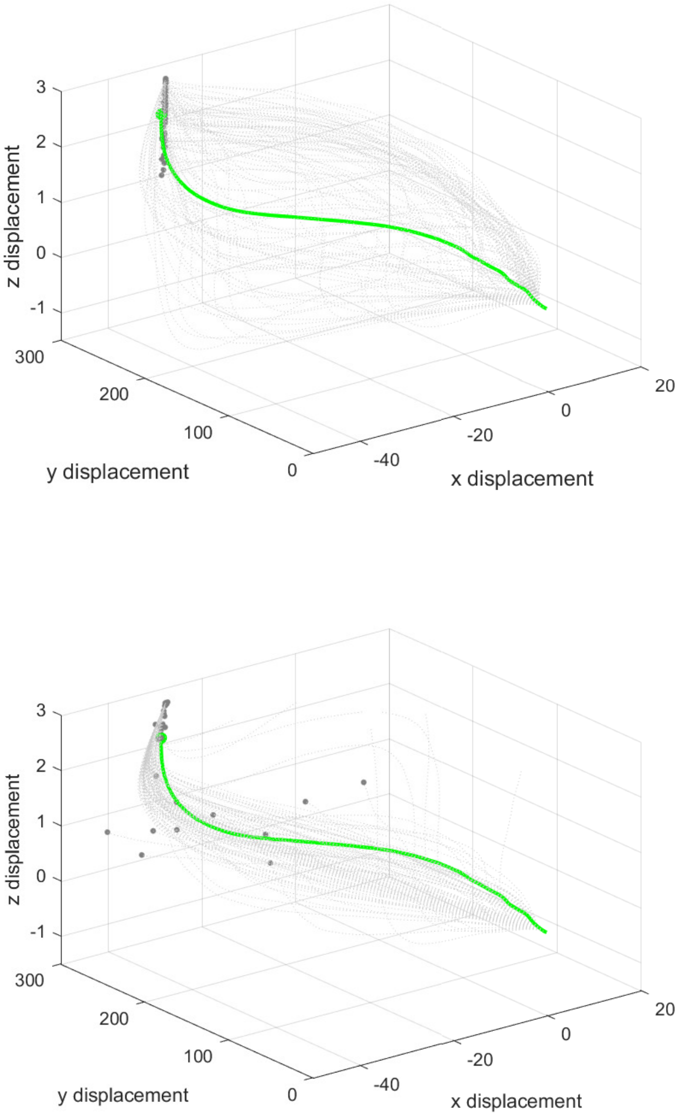


Figure 5.8: A comparison of the solutions found using the distributed algorithm and the centralized algorithm for configuration g, shown in green. The solutions (light grey lines) from the distributed algorithm all fall in a line while the solutions from the centralized algorithm all cluster around the same general area. The final pose of each position is indicated by the grey dots.

energy. For example, [87] uses a McKibben actuator paired with a shape memory polymer to maintain the actuators displacement without continuous control, demonstrating how a robotic material could use both variable stiffness materials and artificial muscles to achieve shape change. In our case, we wish to lock the shape so that we can create complex shapes with only one or two actuators.

In the current configuration the beam has one actuator for inducing positive curvatures and one for negative curvatures. With this setup, the procedure to induce a shape change in a thermoplastic beam is outlined in Figure 5.9. Setting the curvatures sequentially is the most straight forward option. The first element would be heated to a temperature above its melting point (Figure 5.9A). While the temperature is elevated, the appropriate actuator provides tension to the tendon to induce the desired curvature (Figure 5.9B). The final step to changing an elements curvature is to let it cool so that the new shape is retained (Figure 5.9C). This process is repeated for the next elements along the length (Figure 5.9D-I) until the entire beam's shape is set (Figure 5.9J).

Finally we test the beam's ability to reach and maintain a desired curvature profile obtained through the distributed inverse kinematics algorithm presented earlier. Figure 5.10 shows the final curvatures achieved by the beam over one of the trials. For this experiment, the error is not fed forward from one element to the next. More accurate results can be obtained by feeding the element error forward and calculating a correction before the next element's curvature is set.

5.5 Summary

In this chapter I show the distributed inverse kinematics algorithm developed for shape-changing beams. I first described the forward kinematics of the beam and show that the position of the beam can be found after a communication message has flowed from the base of the beam to the tip. I then show how the inverse kinematics problem is reduced from one where updates to all n elements are considered simultaneously to one in which updates are considered in neighborhoods of m elements sequentially. The distributed algorithm is then shown to scale linearly with the number of elements in the robotic material and deliver results that compare to, and in some cases surpass, the centralized algorithm. In order to develop efficient algorithms for robotic materials a careful

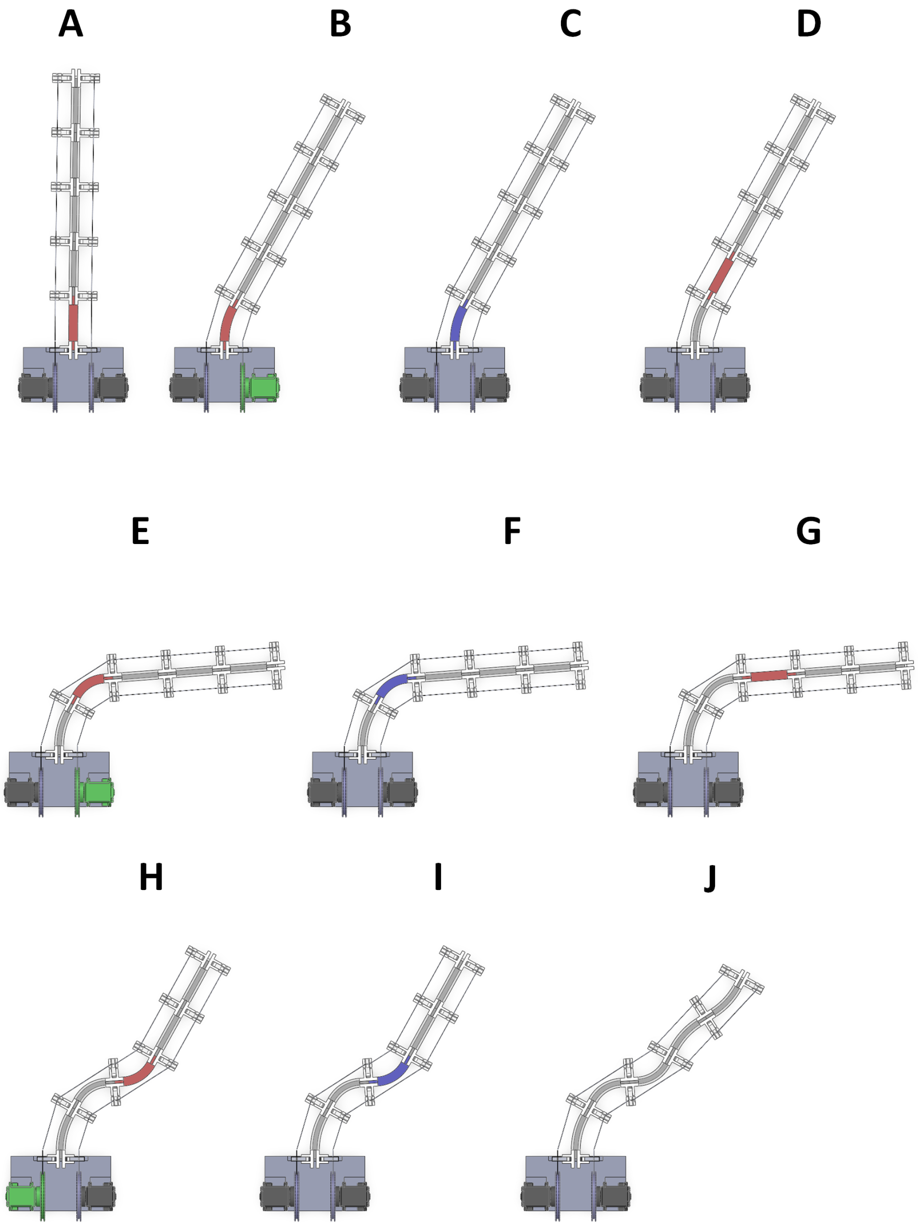


Figure 5.9: Sequentially setting the curvatures in the 2D experimental setup. A) The first element is heated to melting. B) The desired shape is set. C) The element is allowed to cool. D-F and G-I show this process repeating down the length of the beam. J) the final configuration of the beam.

trade off between communication, storage of information, and computation must be conducted. The final section of this chapter shows how the variable stiffness elements can be locked into a desired shape by sequentially setting the curvature of each element.

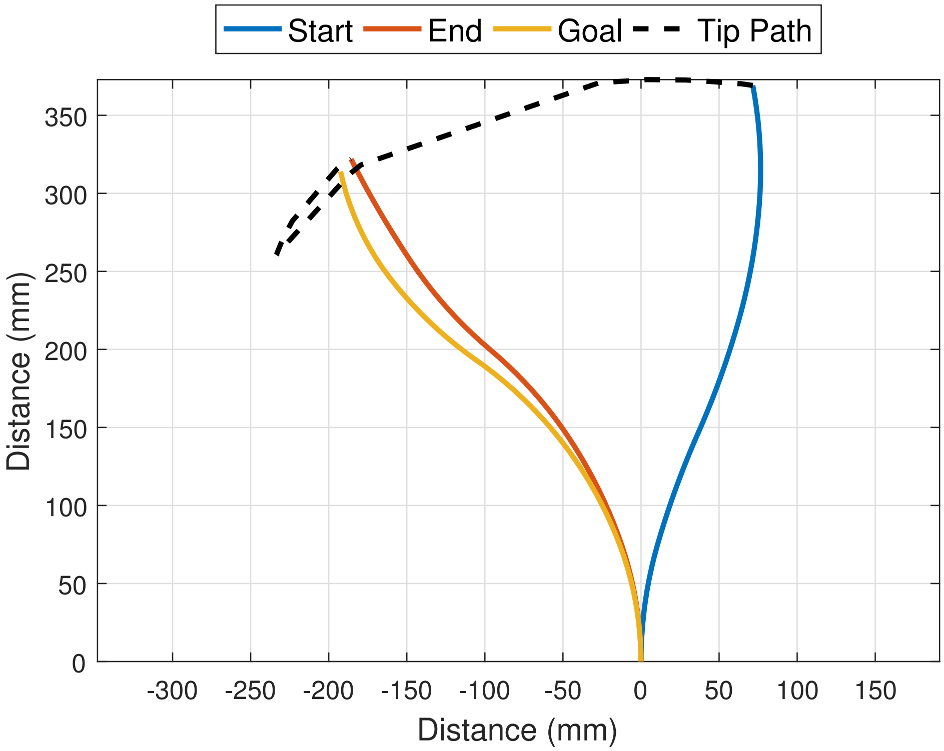


Figure 5.10: A trial from the table top experiment. The beam was started in the blue configuration and given a goal to reach. The inverse kinematics solution found is shown in orange and the final configuration of the beam is shown in red. For this experiment, the error in each segment is not fed forward to the next element.

Chapter 6

Discussion and Conclusion

The constituent parts of robotic materials are typically studied in isolation, each a field of scientific inquiry in their own right. This work shows that materials that couple sensing, actuation, communication and control offer a way to achieve advanced material functionality when considered together as a whole. This allows the designer an extended trade space in which to work. With the realization that computation can be leveraged for advanced functionality, fundamental advancements must be made to understand the impacts that each of these components has on the other when integrated into a complete system.

The first of these fundamental challenges is seen in the choices made for our variable stiffness element. By itself, thermoplastic is a fairly simple material. Combined with a thermistor, Joule heater and some simple computation I am able create a variable stiffness material whose stiffness is controlled through a feedback control loop. One drawback of our thermoplastic variable stiffness element is time required to set each element and that positive and negative curvatures must be set in separate steps. A trade of more complex computation or the use of a more complex material could be made to minimize the time required to set each element's curvature.

With a sufficient material model this burden could be minimized through more advanced control whereby multiple elements are heating and changing their curvatures at the same time. All elements with either positive or negative curvature could begin their heating process at the same time. As the load is applied and elements reach their desired curvatures, their heating could be turned off while the other elements continue to heat up and set their curvatures. This scheme would require significantly more communication overhead as elements would need to communicate curvature rates and contain the material model in order to predict their final curvature. On the other end of the trade more advanced materials could be used. Fast acting variable stiffness elements could be developed, or the actuators could be embedded directly into the material. For example, through the use of shape memory alloys or shape memory polymers. This way each element could change their shape at the same time.

The algorithm presented scaled well with the addition of new elements to the material. The neighborhood setup presented was strictly sequential, but with more communication overhead the neighborhoods could be randomly or periodically distributed over the length of the beam. In this way updates wouldn't be confined to one location in the material, but spread over a larger area at each step.

Another tradeoff that bears consideration is that of distributing the computation. While the application may dictate whether a central computer is possible or not, there is a threshold that must be passed before a distributed solution is viable. A centralized solution might work well for a material with only a few hundred elements, while a distributed system might not be preferable if the allowed computational elements are not sufficiently powerful. And while this will be application specific, the solution will require a blending of the various disciplines. The larger the computational element the more structural properties are likely to suffer, but maybe the extra computational power allows for overcoming the diminished structural properties.

Along with the challenges of integrating separate disciplines, advancements in manufacturing processes must also be made if robotic materials are to succeed. The range of motion of our robotic material is limited by the construction of the various composite parts. While Shape Deposition Manufacturing [68] could offer many benefits, embedding rigid components into a soft material is still a challenge. Using substrates with gradually varying stiffness is a possible way to embed rigid components into soft materials [67] without severe stress concentrations. Embedding of the interconnects between each node could be solved using a technique where the base polymers are functionalized with a coating of copper [42] resulting in a part that could be populated in a modified circuit board assembly machine.

Future work for the shape-changing beam would be to distribute path planning and collision avoidance into the material. Proximity sensors could be embedded to detect the distance of nearby objects and a reactionary response could be formed from elements that interpret nearby objects as a danger. The beam and the distributed algorithms could also be extended into two dimensional and three dimensional shape-changing materials. For materials that lock their shape, investigating the stress distribution and rerouting of load paths through the material while the variable stiffness elements set will also be a challenge. Power distribution through the material and the impacts of that distribution on the number of available actuators is also of great importance and again represents a trade in complexity of control algorithms and more powerful power supplies.

6.1 Conclusion

Robotic materials [63] have the promise to enable a new class of multi-functional materials. The shape changing Robotic Material I created is the first robotic material where the embedded actuation influences the material properties to achieve a desired result. My research forms a foundation that allows others to study different aspects of the challenges that face robotic materials, namely integrating discrete components in to a continuous material, controlling the continuous material with discrete actuators and sensors, and manufacturing the robotic material elements in a reliable manner.

I have presented a shape-changing beam that uses variable stiffness elements and distributed control. The proposed approach was shown to scale linearly with the number of elements in the material using the presented distributed inverse kinematics algorithm. In creating the material there are many trade offs that can leverage computation and communication. The presented approach showed that the distributed algorithms, and an understanding of the composite variable stiffness material, result in a functioning shape-changing beam.

Robotic materials are a seamless integration of sensing, actuation, computation and communication into a composite material that can sense and respond to the environment. Sharing sensor data and computational information using local communication allows the designer to program robotic materials to exhibit arbitrary spatio-temporal behavior. Creating such materials though, requires collaboration between disparate fields of study with the result that few works examine the system level integration of these components. Robotic materials leverage computation to simplify the design of the components and, consequently, the system, giving robotic materials the opportunity to be seen as auxiliary components, added to the system as needed. In the long run robotic materials will be able to perform complex computations and autonomously adapt to their environment.

Bibliography

[1] Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F Knight Jr, Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss. Amorphous computing. Communications of the ACM, 43(5):74–82, 2000.

[2] Douglas Adams. Health monitoring of structural materials and components: methods with applications. John Wiley & Sons, 2007.

[3] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks: a survey. Wireless communications, IEEE, 11(6):6–28, 2004.

[4] L Averous, L Moro, P Dole, and C Fringant. Properties of thermoplastic blends: starch–polycaprolactone. Polymer, 41(11):4157–4167, 2000.

[5] Silvestro Barbarino, Onur Bilgen, Rafic M Ajaj, Michael I Friswell, and Daniel J Inman. A review of morphing aircraft. Journal of Intelligent Material Systems and Structures, 22(9):823–877, 2011.

[6] Subhayu Basu, Yoram Gerchman, Cynthia H Collins, Frances H Arnold, and Ron Weiss. A synthetic multicellular system for programmed pattern formation. Nature, 434(7037):1130–1134, 2005.

[7] J Beal, SO Dulman, K Usbeck, M Viroli, N Correll, and Marjan Mernik. Formal and practical aspects of domain-specific languages: Recent developments. IGI Global, pages 436–501, 2012.

[8] Andrew A Berlin and Kaigham J Gabriel. Distributed mems: New challenges for computation. Computational Science & Engineering, IEEE, 4(1):12–16, 1997.

[9] Dimitri P Bertsekas and John N Tsitsiklis. Parallel and distributed computation: numerical methods, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[10] BJ Blaiszik, SLB Kramer, SC Olugebefola, Jeffrey S Moore, Nancy R Sottos, and Scott R White. Self-healing polymers and composites. Annual Review of Materials Research, 40:179–211, 2010.

[11] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. Proceedings of the National Academy of Sciences, 107(44):18809–18814, 2010.

[12] Christoph Budelmann and Bernd Krieg-Brückner. From sensorial to smart materials: Intelligent optical sensor network for embedded applications. Journal of Intelligent Material Systems and Structures, 2012.

[13] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. IEEE Journal of Robotics and Automation, 17(1-19):16, 2004.

[14] William Butera. Text display and graphics control on a paintable computer. In Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on, pages 45–54. IEEE, 2007.

[15] Gregory S Chirikjian. Hyper-redundant manipulator dynamics: a continuum approximation. Advanced Robotics, 9(3):217–243, 1994.

[16] Nikolaus Correll, Cagdas D Onal, Haiyi Liang, Erik Schoenfeld, and Daniela Rus. Soft autonomous materials—using active elasticity and embedded distributed computation. In 12th International Symposium on Experimental Robotics, Springer Tracts in Advanced Robotics Vol 79, pages 227–240, 2014.

[17] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. Robotics, IEEE Transactions on, 26(1):1–20, 2010.

[18] Raffaello D'Andrea and Geir E Dullerud. Distributed control design for spatially interconnected systems. Automatic Control, IEEE Transactions on, 48(9):1478–1495, 2003.

[19] Michaël De Volder and Dominiek Reynaerts. Pneumatic and hydraulic microactuators: a review. Journal of Micromechanics and Microengineering, 20(4):043001, 2010.

[20] Matt Duckham. Decentralized spatial computing: foundations of geosensor networks. Springer, 2012.

[21] Nicholas Farrow, Naren Sivagnanadasan, and Nikolaus Correll. Gesture based distributed user interaction system for a reconfigurable self-organizing smart wall. In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, pages 245–246. ACM, 2014.

[22] Nicolas Franceschini, Jean-Marc Pichon, Christian Blanes, and JM Brady. From insect vision to robot vision [and discussion]. Philosophical Transactions of The Royal Society Of London. Series B: Biological Sciences, 337(1281):283–294, 1992.

[23] Stephen J Furst, George Bunget, and Stefan Seelecke. Design and fabrication of a bat-inspired flapping-flight platform using shape memory alloy muscles and joints. Smart Materials and Structures, 22(1):014011, 2013.

[24] Farhan Gandhi and Sang-Guk Kang. Beams with controllable flexural stiffness. Smart Materials and Structures, 16(4):1179–1184, 2007.

[25] Mehrdad N Ghasemi-Nejhad, Richard Russ, and Saeid Pourjalali. Manufacturing and testing of active composite panels with embedded piezoelectric sensors and actuators. Journal of intelligent material systems and structures, 16(4):319–333, 2005.

[26] Ronald F Gibson. A review of recent research on mechanics of multifunctional composite materials and structures. Composite structures, 92(12):2793–2810, 2010.

[27] Seth Copen Goldstein, Jason D Campbell, and Todd C Mowry. Programmable matter. Computer, 38(6):99–101, 2005.

[28] P Graziadei. Muscle receptors in cephalopods. Proceedings of the Royal Society of London B: Biological Sciences, 161(984):392–402, 1965.

[29] George Neville Greaves, AL Greer, RS Lakes, and T Rouxel. Poisson's ratio and modern materials. Nature Materials, 10(11):823–837, 2011.

[30] Yoram Gutfreund, Henry Matzner, Tamar Flash, and Binyamin Hochner. Patterns of motor activity in the isolated nerve cord of the octopus arm. The Biological Bulletin, 211(3):212–222, 2006.

[31] Carter S Haines, Márcio D Lima, Na Li, Geoffrey M Spinks, Javad Foroughi, John DW Madden, Shi Hyeong Kim, Shaoli Fang, Mônica Jung de Andrade, Fatma Göktepe, et al. Artificial muscles from fishing line and sewing thread. Science, 343(6173):868–872, 2014.

[32] Tian He, John A Stankovic, Chenyang Lu, and Tarek Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on, pages 46–55. IEEE, 2003.

[33] C. Henry and G. McKnight. Cellular variable stiffness materials for ultra-large reversible deformations in reconfigurable structures. In Proc. SPIE, Smart Structures and Materials 2006: Active Materials: Behavior and Mechanics, volume 6170, page 12, 2006.

[34] Binyamin Hochner. How nervous systems evolve in relation to their embodiment: what we can learn from octopuses and other molluscs. Brain, behavior and evolution, 82(1):19–30, 2013.

[35] Homa Hosseinmardi, Nikolaus Correll, and Richard Han. Bloom filter-based ad hoc multicast communication in cyber-physical systems and computational materials. In Wireless Algorithms, Systems, and Applications, pages 595–606. Springer, 2012.

[36] Homa Hosseinmardi, Akshay Mysore, Nicholas Farrow, Nikolaus Correll, and Richard Han. Distributed spatiotemporal gesture recognition in sensor arrays. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 10(3):17, 2015.

[37] D. Hughes and N. Correll. A soft, amorphous skin that can sense and localize texture. In IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014.

[38] Jeong-Beom Ihn and Fu-Kuo Chang. Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: I. Diagnostics. Smart Materials and Structures, 13(3):609, 2004.

[39] Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Impact of network density on data aggregation in wireless sensor networks. In Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on, pages 457–458. IEEE, 2002.

[40] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. Radical atoms: beyond tangible bits, toward transformable materials. Interactions, 19(1):38–51, 2012.

[41] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, pages 234–241. ACM, 1997.

[42] A Islam, Hans Nørgaard Hansen, Peter Torben Tang, and Jie Sun. Process chains for the manufacturing of molded interconnect devices. The International Journal of Advanced Manufacturing Technology, 42(9-10):831–841, 2009.

[43] Shinae Jang, Hongki Jo, Soojin Cho, Kirill Mechitov, Jennifer A Rice, Sung-Han Sim, Hyung-Jo Jung, Chung-Bang Yun, Billie F Spencer Jr, and Gul Agha. Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation. Smart Structures and Systems, 6(5-6):439–459, 2010.

[44] Elliot R Johnson and Todd D Murphey. Scalable variational integrators for constrained mechanical systems in generalized coordinates. Robotics, IEEE Transactions on, 25(6):1249–1261, 2009.

[45] Bryan A Jones and Ian D Walker. Practical kinematics for real-time implementation of continuum robots. IEEE Transactions on Robotics, 22(6):1087–1099, 2006.

[46] Joseph M Kahn, Randy H Katz, and Kristofer SJ Pister. Next century challenges: mobile networking for "smart dust". In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pages 271–278. ACM, 1999.

[47] R. K. Katzschmann, A. D. Marchese, and Daniela Rus. Hydraulic autonomous soft robotic fish for 3D swimming. In International Symposium on Experimental Robotics (ISER), Marrakech, Morocco, 2014.

[48] William M Kier and Kathleen K Smith. Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats. Zoological Journal of the Linnean Society, 83(4):307–324, 1985.

[49] Glenn K Klute, Joseph M Czerniecki, and Blake Hannaford. Mckibben artificial muscles: pneumatic actuators with biomechanical intelligence. In Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 221–226. IEEE, 1999.

[50] Cédric Langbort, Ramu Sharat Chandra, and Raffaello D'Andrea. Distributed control design for systems interconnected over an arbitrary graph. Automatic Control, IEEE Transactions on, 49(9):1502–1519, 2004.

[51] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration. In Proceedings of the 27th annual ACM symposium on User interface software and technology, pages 461–470. ACM, 2014.

[52] Ulf Leonhardt. Metamaterials: Towards invisibility in the visible. Nature Materials, 8(7):537–538, 2009.

[53] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. In Ambient intelligence, pages 115–148. Springer, 2005.

[54] Changquing Li and Christopher D Rahn. Design of continuous backbone, cable-driven robots. Journal of Mechanical Design, 124(2):265–271, 2002.

[55] Joshua Lifton, Deva Seetharam, Michael Broxton, and Joseph Paradiso. Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks. In Pervasive Computing, pages 139–151. Springer, 2002.

[56] Shang Ma, Homa Hosseinmardi, Nicholas Farrow, Richard Han, and Nikolaus Correll. Establishing multi-cast groups in computational robotic materials. In IEEE Int. Conf. on Cyber, Physical and Social Computing CPSCon, pages 311–316. IEEE, 2012.

[57] C Majidi, R Kramer, and RJ Wood. A non-differential elastomer curvature sensor for softer-than-skin electronics. Smart Materials and Structures, 20(10):105017, 2011.

[58] Andrew D. Marchese, Komorowski Konrad, Cagdas D. Onal, and Daniela Rus. Design, curvature control, and autonomous positioning of a soft and highly compliant 2D robotic manipulator. In 2014 IEEE Int. Conf. Robotics and Automation. IEEE, 2014.

[59] Jerrold E Marsden, George W Patrick, and Steve Shkoller. Multisymplectic geometry, variational integrators, and nonlinear PDEs. Communications in Mathematical Physics, 199(2):351–395, 1998.

[60] M. A. McEvoy and N. Correll. Shape change through programmable stiffness. In International Symposium on Experimental Robotics (ISER), Marrakech, Morocco, 2014.

[61] M. A. McEvoy and N. Correll. Thermoplastic variable stiffness composites with embedded, networked sensing, actuation, and control. Journal of Composite Materials, 2014.

[62] M. A. McEvoy and N. Correll. Distributed inverse kinematics for shape-changing robotic materials. In 3rd International Conference on System-integrated Intelligence: New Challenges for Product and Production Engineering, Paderborn, Germany, 2016.

[63] MA McEvoy and N Correll. Materials that couple sensing, actuation, computation, and communication. Science, 347(6228):1261689, 2015.

[64] Geoff McKnight and Chris Henry. Variable stiffness materials for reconfigurable surface applications. Smart Structures and Materials, pages 119–126, 2005.

[65] Geoffrey Mcknight, Robert Doty, Andrew Keefe, Guillermo Herrera, and Chris Henry. Segmented reinforcement variable stiffness materials for reconfigurable surfaces. Journal of Intelligent Material Systems and Structures, 21(17):1783–1793, 2010.

[66] Qinghao Meng and Jinlian Hu. A review of shape memory polymer composites and blends. Composites Part A: Applied Science and Manufacturing, 40(11):1661–1672, 2009.

[67] Y. Menguc, Y.-L. Park, H. Pei, D. Vogt, P. Aubin, E. Winchell, L. Fluke, L. Stirling, R.J. Wood, and C.J. Walsh. Wearable soft sensing suit for human gait measurement. Int. J. of Robotics Research, November 2014. doi:10.1177/0278364914543793.

[68] Robert Merz, FB Prinz, K Ramaswami, M Terk, and L Weiss. Shape deposition manufacturing. Engineering Design Research Center, Carnegie Mellon Univ., 1994.

[69] Stephen A Morin, Robert F Shepherd, Sen Wai Kwok, Adam A Stokes, Alex Nemiroski, and George M Whitesides. Camouflage and display for soft machines. Science, 337(6096):828–832, 2012.

[70] Gabriel Murray and Farhan Gandhi. Multi-layered controllable stiffness beams for morphing: energy, actuation force, and material strain considerations. Smart Materials and Structures, 19(4):11, 2010.

[71] Adelaide Nespoli, Stefano Besseghini, Simone Pittaccio, Elena Villa, and Stefano Viscuso. The high potential of shape memory alloys in developing miniature mechanical devices: A review on shape memory alloy mini-actuators. Sensors and Actuators A: Physical, 158(1):149–160, 2010.

[72] Ryuma Niiyama, Lining Yao, and Hiroshi Ishii. Weight and volume changing device with liquid metal transfer. In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, pages 49–52. ACM, 2014.

[73] Cagdas Denizel Onal, Robert J Wood, and Daniela Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 4608–4613. IEEE, 2011.

[74] Jifei Ou, Lining Yao, Daniel Tauber, Jürgen Steimle, Ryuma Niiyama, and Hiroshi Ishii. jamSheets: thin interfaces with tunable stiffness enabled by layer jamming. In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, pages 65–72. ACM, 2014.

[75] Yong-Lae Park, Carmel Majidi, Rebecca Kramer, Phillipe Bérard, and Robert J Wood. Hyperelastic pressure sensing with a liquid-embedded elastomer. Journal of Micromechanics and Microengineering, 20(12):125029, 2010.

[76] Rolf Pfeifer and Fumiya Iida. Morphological computation: Connecting body, brain and environment. Japanese Scientific Monthly, 58(2):48–54, 2005.

[77] H Profita, N Farrow, and N Correll. Flutter. Adj. Proc. of ISWC, 2012.

[78] Paolo Santi. Topology control in wireless ad hoc and sensor networks. ACM computing surveys (CSUR), 37(2):164–194, 2005.

[79] Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control–a review. Journal of Process Control, 19(5):723–731, 2009.

[80] Kashan A Shaikh, Shifeng Li, and Chang Liu. Development of a latchable microvalve employing a low-melting-temperature metal alloy. Microelectromechanical Systems, Journal of, 17(5):1195–1203, 2008.

[81] Vladimir M Shalaev, Wenshan Cai, Uday K Chettiar, Hsiao-Kuan Yuan, Andrey K Sarychev, Vladimir P Drachev, and Alexander V Kildishev. Negative index of refraction in optical metamaterials. Optics Letters, 30(24):3356–3358, 2005.

[82] WL Shan, T Lu, ZH Wang, and C Majidi. Thermal analysis and design of a multi-layered rigidity tunable composite. International Journal of Heat and Mass Transfer, 66:271–278, 2013.

[83] Xiaohong Sheng and Yu-Hen Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. Signal Processing, IEEE Transactions on, 53(1):44–53, 2005.

[84] Robert F Shepherd, Filip Ilievski, Wonjae Choi, Stephen A Morin, Adam A Stokes, Aaron D Mazzeo, Xin Chen, Michael Wang, and George M Whitesides. Multigait soft robot. Proceedings of the National Academy of Sciences, 108(51):20400–20403, 2011.

[85] Takao Someya, Yusaku Kato, Tsuyoshi Sekitani, Shingo Iba, Yoshiaki Noguchi, Yousuke Murase, Hiroshi Kawaguchi, and Takayasu Sakurai. Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes. Proceedings of the National Academy of Sciences of the United States of America, 102(35):12321–12325, 2005.

[86] Jukka Suomela. Survey of local algorithms. ACM Computing Surveys (CSUR), 45(2):24, 2013.

[87] Kazuto Takashima, Jonathan Rossiter, and Toshiharu Mukai. Mckibben artificial muscle using shape-memory polymer. Sensors and Actuators A: Physical, 164(1):116–124, 2010.

[88] Hong Z Tan, Lynne A Slivovsky, and Alex Pentland. A sensing chair using pressure distribution sensors. Mechatronics, IEEE/ASME Transactions on, 6(3):261–268, 2001.

[89] C Thill, J Etches, I Bond, K Potter, and P Weaver. Morphing skins. The Aeronautical Journal, 112(1129):117–139, 2008.

[90] Sameer Tilak, Nael B Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless microsensor network models. ACM SIGMOBILE Mobile Computing and Communications Review, 6(2):28–36, 2002.

[91] Tommaso Toffoli and Norman Margolus. Programmable matter: concepts and realization. Physica D: Nonlinear Phenomena, 47(1):263–272, 1991.

[92] Srinivas Vasista, Liyong Tong, and KC Wong. Realization of morphing wings: A multidisciplinary challenge. Journal of Aircraft, 49(1):11–28, 2012.

[93] Rodger M Walser. Electromagnetic metamaterials. In Proc. SPIE 4467, Complex Mediums II: Beyond Linear Isotropic Dielectrics, pages 1–15, San Diego, CA, July 2001.

[94] Yaming Wang, Miguel A Rodriguez-Perez, Rui L Reis, and Joao F Mano. Thermal and thermomechanical behaviour of polycaprolactone and starch/polycaprolactone blends for biomedical applications. Macromolecular Materials and Engineering, 290(8):792–801, 2005.

[95] Robert J Webster and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. The International Journal of Robotics Research, 29(13):1661–1683, 2010.

[96] Terrence A Weisshaar. Morphing aircraft systems: Historical perspectives and future challenges. Journal of Aircraft, 50(2):1–17, 2013.

[97] Geoffrey Werner-Allen, Geetika Tewari, Ankit Patel, Matt Welsh, and Radhika Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In Proceedings of the 3rd international conference on Embedded networked sensor systems, pages 142–153. ACM, 2005.

[98] Maria Ann Woodruff and Dietmar Werner Hutmacher. The return of a forgotten polymer—polycaprolactone in the 21st century. Progress in Polymer Science, 35(10):1217–1256, 2010.

[99] Takeo Yamada, Yuhei Hayamizu, Yuki Yamamoto, Yoshiki Yomogida, Ali Izadi-Najafabadi, Don N Futaba, and Kenji Hata. A stretchable carbon nanotube strain sensor for human-motion detection. Nature Nanotechnology, 6(5):296–301, 2011.

[100] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. Robotics & Automation Magazine, IEEE, 14(1):43–52, 2007.

[101] Kazuhiro Yoshida, Kazuhito Kamiyama, Joon-wan Kim, and Shinichi Yokota. An intelligent microactuator robust against disturbance using electro-rheological fluid. Sensors and Actuators A: Physical, 175:101–107, 2012.

[102] Chih-Han Yu, F-X Willems, Donald Ingber, and Radhika Nagpal. Self-organization of environmentally-adaptive shapes on a modular robot. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pages 2353–2360. IEEE, 2007.

[103] Xiaoliang Zhao, Huidong Gao, Guangfan Zhang, Bulent Ayhan, Fei Yan, Chiman Kwan, and Joseph L Rose. Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. defect detection, localization and growth monitoring. Smart materials and structures, 16(4):1208, 2007.

[104] Xiaoliang Zhao, Tao Qian, Gang Mei, Chiman Kwan, Regan Zane, Christi Walsh, Thurein Paing, and Zoya Popovic. Active health monitoring of an aircraft wing with an embedded piezoelectric sensor/actuator network: Ii. wireless approaches. Smart materials and structures, 16(4):1218, 2007.

[105] Jun Zhou, Yudong Gu, Peng Fei, Wenjie Mai, Yifan Gao, Rusen Yang, Gang Bao, and Zhong Lin Wang. Flexible piezotronic strain sensor. Nano Letters, 8(9):3035–3040, 2008.