

**Soft Robotic Testbeds for Physics Simulator Validation, Closed Loop Control, and Comparative Biomechanics**

Submitted in partial fulfillment of the requirements for  
the degree of  
Doctor of Philosophy  
in  
Mechanical Engineering

Zach J. Patterson

B.S., Mechanical Engineering, University of Pittsburgh

Carnegie Mellon University  
Pittsburgh, PA

August 2022

© Zach J. Patterson, 2022

All Rights Reserved

## Acknowledgements

I would first like to thank Professor Carmel Majidi for his mentorship as I have progressed through the work contained in this thesis. You have taught me how to be a successful researcher, not only in terms of doing quality science and successfully communicating it, but also in navigating the stressful and sometimes Kafkaesque world of academic bureaucracy. You struck a perfect balance between pushing me when needed while also granting me the broad autonomy and trust to grow at my own pace and to take the research where I thought it should go. Most importantly, you have also served as a model of how to balance a successful academic career with a rich personal life.

I would also like to thank my committee, Professors Henry Astley, Sarah Bergbreiter, and Rob Wood. Your guidance and collaboration have contributed to a significant portion of the research contained in this document, and you've provided me advice that has served me well both in preparing to defend this dissertation and in preparing to progress my career beyond my PhD.

Thank you to my wonderful lab group. It's hard to keep track of the large and rotating cast of characters who have made up the Soft Machines Lab for the past four years. I'd like to first thank Sean, Tess, Bugra, Stuart, Chengfeng, and the many other senior members of the lab when I joined SML. Through our many conversations, your diligent efforts to train and aid me, and through your excellent examples, you helped me to get my feet under me as a researcher. I'd like to highlight Sean especially. The works we've published together that form the first chapter of this thesis are a small testament to the magnitude of your mentorship. More importantly, you all made me feel instantly at home and I will always cherish your warmth and friendship.

I'd next like to thank those other PhD students in my cohort, especially Yun, Pratik, Jiahe, Kiyn. Through all the (ongoing) ups and downs of our PhD student lives together, it was a

constant comfort to have you all alongside me. You have each been wonderful colleagues and, wherever life takes us next, lifelong friends. Also, shoutout to the Friday afternoon Oishii crew; I'll always cherish the memories of those lunches, both for the conversation and the exquisite bulgogi. Next is Drew. Drew, I knew you would be a valuable addition to the lab and I knew we would collaborate in our work. I couldn't have anticipated just how much your mentorship contributed to my development. Through your time here in Pittsburgh, you devoted so much time to help me be the best researcher I could be and our constant brainstorming sessions have shaped my thinking in a profound way about robotics and control. You were always a constant bright spot and source of positivity, and a great friend. Thank you. Finally, to all SML members, old and new, thank you. You've all made the last few years of sometimes gruelling work tolerable and, in many cases, extremely enjoyable.

Thank you to the many friends who have made life outside of work unbelievably fulfilling. I could not have completed these last four years without you all being there for me. Late nights, road trips, dinners out, movies, walks in the park, and pickup games were the fuel that enabled me to keep going. Thank you to my family, especially my many cousins who have shaped me through their friendship, example, and occasional bullying. Thank you to my aunts and uncles. I've been lucky enough that all of you are in the Pittsburgh area and were always a phone call away, whether it was for an emergency, a ride to the airport, or just a home-cooked meal, you were all there for me.

Thank you to my grandparents. To my grandpa, George Patterson, who was an EE at CMU while it was still Carnegie Tech, you've been the foremost inspiration in my life driving me towards engineering. I wouldn't be here without you. To my grandma, Lorraine Patterson, your delicious cooking, constant prayers, and unconditional love have given me nourishment and strength. To my late nonno, Charlie Costanza, you died before I started this program, but thinking of tying one on with you still makes me cry to this day. I miss you and I wish you could be here to see this. And to my nonna, Rosella Costanza, you died long ago but your influence

and love have shaped my family and my life.

Most importantly, thank you to my parents and siblings. Your support and love are the single most important factors that have brought me to this point.

This work was supported by the Office of Naval Research (Award N00014-17-2063), and National Oceanic Partnership Program (Award N00014-18-12843).

## **Abstract**

While robots made of flexible and deformable materials, commonly called soft robots, have shown potential to function more autonomously in unstructured environments, most such robots remain confined to the lab and tethered to a benchtop power supply. To develop more autonomous mobile soft robots that can function outside of the lab, we require more sophisticated modeling, simulation, and control tools as well as soft robotic testbeds that are unthethered and of sufficient maturity and complexity to develop and validate these tools. Autonomous soft robot development can also be aided by better understanding how highly flexible animals move and the connections between physics, morphology, and control. This thesis presents several mobile, untethered soft robotics testbeds that are used to validate physics simulators, to develop novel controllers, and to study the interplay between morphology and locomotion for underwater walking organisms and robots. These tools, and the robots themselves, enable the design, development, and deployment of more autonomous soft robots in highly uncertain environments.

# Contents

- 1 Introduction** **1**
  
- 2 Validation of Discrete Elastic Rods for Soft Robots** **7**
  - 2.1 Motivation . . . . . 8
  
  - 2.2 Star-shaped robot for underwater DER . . . . . 8
    - 2.2.1 Design . . . . . 8
  
    - 2.2.2 Application . . . . . 9
  
    - 2.2.3 Testing . . . . . 11
  
    - 2.2.4 Results & Discussion . . . . . 11
  
  - 2.3 Closed Loop Motion Planning with Discrete Elastic Rods for Swimming Robots . 12
    - 2.3.1 Experimental Testbed . . . . . 12
  
    - 2.3.2 DER Simulation Tool . . . . . 15
  
    - 2.3.3 Online Planner for Path Following . . . . . 16

2.3.4	Discussion . . . . .	21
2.4	Conclusion . . . . .	23
<b>3</b>	<b>PATRICK - A Brittle Star Inspired Unthethered Soft Robot</b>	<b>25</b>
3.1	Motivation . . . . .	25
3.2	Methods . . . . .	26
3.2.1	Robot Design . . . . .	26
3.2.2	Fabrication . . . . .	29
3.2.3	Testbed . . . . .	30
3.2.4	Motion Planning . . . . .	32
3.3	Results . . . . .	35
3.4	Discussion . . . . .	36
3.5	Conclusion . . . . .	38
<b>4</b>	<b>Manufacturing Fieldable, Miniaturized Soft Robots with 3D Printing</b>	<b>39</b>
4.1	Motivation . . . . .	39
4.2	Materials and Methods . . . . .	43
4.2.1	3D printing and robot fabrication . . . . .	43
4.2.2	Robot Experimental Setup . . . . .	45

4.3	Results . . . . .	48
4.3.1	Actuator Characterization . . . . .	48
4.3.2	Robot Function . . . . .	49
4.4	Discussion . . . . .	51
4.5	Conclusion . . . . .	55
<b>5</b>	<b>Controlling Brittle Star Robot Limbs</b>	<b>56</b>
5.1	Motivation . . . . .	56
5.2	Manipulator Design and System Model . . . . .	58
5.2.1	Hardware and Manufacturing . . . . .	59
5.2.2	System and Input Models . . . . .	59
5.3	Control System . . . . .	61
5.3.1	Singular Value Decomposition (SVD) Compensator . . . . .	61
5.3.2	PI Controller . . . . .	62
5.3.3	Anti-Windup . . . . .	62
5.3.4	Overall Controller Structure and Implementation . . . . .	64
5.4	Controller Stability . . . . .	64
5.4.1	Robust Stability with a Saturation Nonlinearity . . . . .	65
5.4.2	Stability Verification: Nominal System . . . . .	67

5.4.3	Stability Verification: Dynamic Uncertainty . . . . .	68
5.5	Controller Performance on Hardware . . . . .	71
5.6	Discussion . . . . .	72
5.7	Conclusion . . . . .	74
<b>6</b>	<b>Understanding the Influence of Mass Distribution on Brittle Star Locomotion with PATRICK</b>	<b>75</b>
6.1	Motivation . . . . .	76
6.2	Modeling and Simulation . . . . .	78
6.2.1	Quasistatic Analytic Model . . . . .	79
6.2.2	Dynamic Computational Model . . . . .	81
6.3	Brittle Star Weight Distribution . . . . .	87
6.4	Brittle Star Robot Experiments . . . . .	88
6.5	Discussion . . . . .	91
<b>7</b>	<b>Conclusions and Future Work</b>	<b>94</b>
7.1	Integrating Sensors and Controllers into PATRICK Platform . . . . .	95
7.2	Examining Other Echinoderms and Organisms . . . . .	97
7.3	Better Actuators, Better Robots . . . . .	98
7.4	Data Availability . . . . .	100



# List of Figures

1.0.1 Survey of untethered soft robots . . . . .	3
2.2.1 Dynamic SMA actuator layup . . . . .	10
2.2.2 Star-shaped robot testbed and simulated model . . . . .	10
2.2.3 Test setup for star-shaped robot . . . . .	11
2.2.4 Star-shaped robot simulation validation . . . . .	13
2.3.1 Frog-like robot overview . . . . .	14
2.3.2 Discretization, coordinates, and dimensions of frog-like robot for simulation. . . . .	15
2.3.3 Closed loop path planning results . . . . .	21
3.1.1 The flexible underwater untethered brittle star robot, PATRICK. . . . .	26
3.2.1 Brittle stars . . . . .	27
3.2.2 Design and electronics of brittle star robot . . . . .	30
3.2.3 Brittle star robot system architecture . . . . .	31

3.2.4 Brittle star robot SMA activation sequence . . . . .	34
3.3.1 Brittle star robot representation and motion primitives . . . . .	36
3.3.2 Brittle star robot locomotion results . . . . .	37
4.1.1 3D printed brittle star robot overview . . . . .	42
4.1.2 Examples of 3D printed soft robots . . . . .	42
4.2.1 3D printed robot gait . . . . .	47
4.3.1 Characterization of 3D printed actuators . . . . .	50
4.3.2 3D printed robot closed loop demonstration . . . . .	51
4.3.3 3D printed robot operating in a lake . . . . .	52
5.1.1 Soft manipulator state space . . . . .	58
5.3.1 Nominal SVD controller . . . . .	62
5.3.2 Block diagram of closed loop control system . . . . .	65
5.4.1 Block diagrams for robust stability analysis . . . . .	66
5.4.2 Step response and trajectory tracking results . . . . .	69
5.4.3 Disturbance rejection results . . . . .	70
6.1.1 Brittle stars and brittle star robot . . . . .	78
6.2.1 Free body diagrams of brittle star robot . . . . .	80

6.2.2	Snapshots from dynamic simulations of brittle star locomotion . . . . .	83
6.2.3	Optimal Control Results . . . . .	84
6.4.1	Velocity for brittle star robot with various distributions of mass . . . . .	90
7.1.1	Sensorized PATRICK robot . . . . .	96

# List of Tables

5.1	Error during trajectory trials for various gains and speeds. . . . .	72
6.1	Brittle star mass and density distribution metrics . . . . .	88
6.2	Regression Results: Water weight and center of mass versus average velocity . .	91
6.3	Regression Results: Water weight and center of buoyancy versus average velocity	91

# Chapter 1

## Introduction

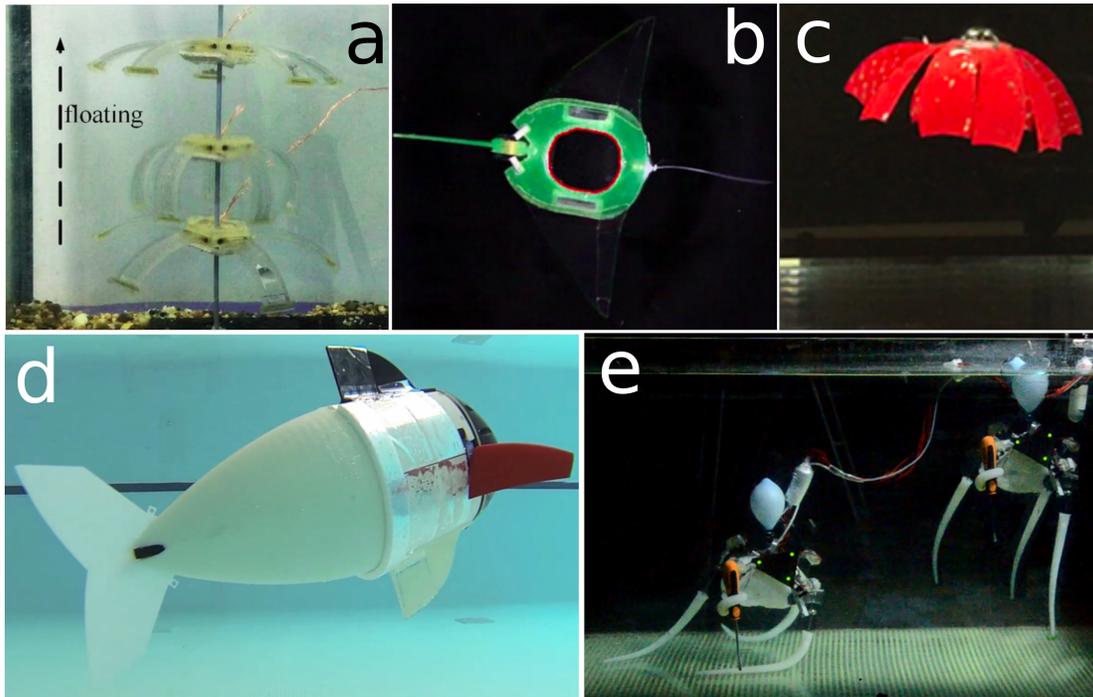
Soft robots have been hailed as a coming revolution in robotics [1]. Because they take advantage of inherently soft, stretchable, or flexible materials, soft roboticists often claim that their robots can potentially perform better in an uncertain world with less central control by offloading the “computation” to deformable structures [2], a concept sometimes referred to as morphological intelligence [3, 4]. However, if soft robots are to ultimately accomplish useful tasks, they must still be subject to control at some level and they must also integrate sensors and system level state estimation. This presents a challenge because the mechanical deformability of soft robots makes them more difficult to model, predict, and control [5]. Additionally, even ignoring the control problem, sufficient modeling is still required to properly design morphological intelligence.

Underwater environments are particularly interesting for mobile soft robot development because the buoyancy force reduces the need for structural support. We can see the results of this in the animal kingdom: many more varieties and sizes of soft and flexible bodied organisms - such as octopuses, jellyfish, echinoderms, etc - reside in aquatic environments, while soft bodied organisms on land are constrained to small creatures like worms. Soft roboticists have therefore developed many robots that function in underwater environments [6]. However, most of these

aquatic soft robots, like most soft robots in general, are subject to two drawbacks. First, they are almost always tethered, which is a problem because it reduces the ability for the robot to move freely throughout the world. More specifically, the tether can interfere with the mechanics of the robot, especially in the buoyant aquatic environment where even small forces can produce significant deviation from physical models and simulations. Second, most mobile soft robots in the literature are limited one or two simple (but often mechanically impressive) behaviors [7]. Even when mobile soft robots are more complex, they are often controlled in the open loop with bang-bang control [8]. This makes sense since much of the focus of the field has been on the creation of novel actuators, sensors, materials, and other machine components and so many existing robots are used to showcase those individual components. Unfortunately, these problems reduce the potential use cases of the robots and reduce their value as platforms for developing controls/planning strategies, developing physics models and simulations, or for examining the mechanics and locomotion of natural organisms [9, 10].

For soft robots to ultimately achieve their potential, they must successfully perform complex tasks, ideally ones that would cause a conventional robot to fail. To reach this point, we need robots that can capably serve as testbeds to refine controls, planning, and learning approaches. Significant progress has been made on this front for stationary continuum manipulators (see [11][12]). Existing mobile soft robots, as discussed previously, usually only perform very simple behavior with limited potential for accomplishing a diversity of tasks. There are, of course notable exceptions [13, 14]. OCTOPUS [15], SUPERBALL [16], the cable driven quadruped by Bern et. al. [17], and Harvard's recent echinoderm robots [18, 19] are a non-exhaustive list of recent soft robots that are complex and high dimensional, in principle allowing exploration of interesting controls, planning, and learning concepts. However, most existing mobile soft robots are tethered, which restricts autonomy and maneuverability [13, 20–24]. Figure 1.0.1 shows several underwater soft robots from other research groups. In the context of the previous discussion, Fig.1.0.1a and 1.0.1e are both tethered, while Fig.1.0.1b and 1.0.1c are untethered

but use relatively simple actuators with a low controllable state space. Fig.1.0.1d is the closest to addressing all of our concerns, and has even previously demonstrated autonomous functionality [13]. Although Fig.1.0.1e is tethered, it is also a valuable testbed for addressing problems at the cutting edge of soft robot control and planning.



**Figure 1.0.1:** a) A soft tethered SMA powered robot that can walk terrestrially or swim vertically with a rowing gait [25]. b) A ray inspired untethered DEA powered soft robot [26]. c) A jellyfish inspired untethered DEA powered soft robot [27]. d) An untethered soft robotic fish capable of autonomous maneuvers [28]. e) A tethered octopus robot that can walk along the sea bed, swim, and manipulate objects [29].

As soft robots gradually become more mature and complex, the challenges of modeling become magnified. Presently, the design and control methodologies are as often ad-hoc, more of an art than a science, as they are guided by principled modeling. While in the past this has largely been due to the lack of such tools, in recent years there has been a proliferation of options, each with different strengths and weaknesses [30]. Generalizing broadly, the range of options can typically be sorted into two classes: those that are based on more traditional analytic mechanics and robotics models, and those that utilize the finite element method or similar methods to capture the complex elastic mechanics of deformable structures.

The first group includes standard rigid robotics models [31], the piecewise constant curvature models that extend the rigid body formulation [32], and more general rod based methods [33, 34]. The chief advantage of these methods is that standard robotics controls, planning, learning, and design tools are readily used. Additionally, such models tend to be quite fast, and in the best case are the same speed as a standard robotics simulator.

The second group, the more complex FEM-based methods, includes the SOFA simulator [35], DiffPD [36], and Incremental Point Contact [37]. While these simulators are highly accurate in simulating complex mechanical behavior, they are also slow. Although they have been used to design robots and gaits using traditional trajectory optimization [38], Dubied et. al. notes the difficulty of using such tools for that purpose [39], and they inevitably rely on model order reduction techniques [40].

Within this landscape, Discrete Differential Geometry-based methods such as Discrete Elastic Rods (DER) serve as a potential middle ground, enabling fast simulation of highly deformable structures such as hair [41]. DER has been shown to be highly accurate in modeling terrestrial soft robots and is capable of doing so speeds faster than real-time on a mainstream CPU thread [42, 43]. However, DER had not previously been adapted to simulate aquatic soft robots and the extension to accommodate complex fluid-soft structure interactions is non-trivial. Finally, while DER has shown to generate accurate simulations, it has yet to be applied to robot control or planning tasks.

When deploying complex soft robots, control is also an open challenge [44]. This is in no small part due to the modeling challenges discussed previously, but it is compounded by the extra complexity of the control task for real-life soft robotics systems. For example, the control theorist, when dealing with traditional robots, can model the actuators (motors) as generalized torques within the natural configuration space. This works because the low level torque PID controllers on modern motors are highly reliable. Some soft robotic actuation systems can rely on

similar assumptions; for example, Katzschmann and Della Santina got a lot of mileage through the use of modeling independently controlled pneumatic bending segments as constant curvature structures controlled by generalized torques [32, 45, 46]. However, such an approach is limited to situations where one can avoid dealing explicitly with actuator dynamics (as is the case with some pneumatic actuators) or else where the actuator dynamics are sufficiently modeled and controlled to produce a generalized torque. In situations where actuator dynamics are complex, such as shape memory alloys (SMA), such approaches are more difficult to implement. More subtly, these approaches are not explicitly adapted to be robust to contact and therefore are potentially unreliable on contact rich robots.

In light of the problems discussed in this chapter, my thesis objectives are as follows:

#### THESIS OBJECTIVES:

1. Use untethered soft robots to validate the DER physics engine for aquatic soft robot simulation and online motion planning.
2. Design a complex, high dimensional, bioinspired, and untethered soft robot research platform.
3. Develop and implement controllers able to maintain stability for such a robot in uncertain, contact-rich environments.
4. Use untethered soft robots to explore the connections between morphology and locomotion in underwater walking.

In this document, I will present my work on developing untethered soft robotics testbeds that are used to validate soft robotics simulators, to develop soft robotics controllers, and to elucidate the connections between morphology and locomotion for a class of fast-moving echinoderms, the brittle stars. The second chapter will cover my work on untethered soft robots for validating the

application of Discrete Elastic Rods simulators to soft robotics. I also show that these simulators can be used for soft robotics control and motion planning tasks. This work improves the DER simulation tool and its applications to soft robots, allowing more principled design and control for future soft robots.

In the third chapter, I discuss the untethered underwater bioinspired robot, PATRICK. I show that PATRICK is capable of performing relatively advanced tasks with minimal control. PATRICK's large configuration space and high amount of actuators makes it a versatile platform for mobile soft robot experimentation. In Chapter 4, I present a scaled-down version of PATRICK produced using a new 3D-printing workflow and I deploy the robot in a natural environment.

In Chapter 5, I present a novel, robust controller for shape memory alloy (SMA) driven soft manipulators, such as those found on PATRICK. I prove stability and show that the controllers are robust to unmodeled contact and other disturbances, making them appropriate for contact-rich locomotion strategies. Finally, in Chapter 6, the PATRICK robot is used to investigate the interplay between mass distribution and underwater walking locomotion, utilizing optimal control to explain the physical interactions at play. Together, the works collected in this thesis represent progress towards development of fully autonomous aquatic soft robots. The contributions include the robots themselves, but also the modeling and control tools presented that can be utilized on other robots in the future. These contributions, while seemingly diverse, constitute a cohesive research agenda in which the connections between mechanics, control, and morphology are explored and exploited. In the concluding chapter, I elaborate on these connections, showing the common thread running through this work that inspires a great deal of future work that can build upon it.

## Chapter 2

# Validation of Discrete Elastic Rods for Soft Robots

*Publications and manuscripts:*

X. Huang\*, W. Huang\*, Z.J. Patterson\* (\* co-first author), Z. Ren, M. K. Jawed, C. Majidi, “Numerical Simulation of an Untethered Omni-Directional Star-Shaped Swimming Robot.” *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

X. Huang\*, Z.J. Patterson\* (\* co-first author), A.P. Sabelhaus, K. Chin, W. Huang, K. Jawed, C. Majidi, ”Simulating Dynamic Locomotion of Untethered Soft Robots for Design and Motion Planning.” Under Review *Advanced Intelligent Systems*

*Contributions:* I contributed to robot design, fabrication, experiments, and algorithm design and implementation.

## 2.1 Motivation

As discussed in the Introduction, aquatic applications are a valuable use case for mobile soft robots due to the buoyancy force reducing the need for rigid skeletal support structures as well as the inherent safety of soft structures when dealing with delicate ecosystems. Many soft roboticists have developed underwater robots, but few (if any) have created untethered platforms, which are important for the long term goal of achieving true autonomy and the shorter term goal of developing appropriate physical models, control and planning systems, and simulations. This section will discuss the design of two soft untethered aquatic robots and their use to validate Discrete Elastic Rods (DER) simulations and to develop a planning algorithm based on those simulations.

The DER approach to physics simulation discretizes a slender body as a series of point masses, or nodes. These nodes are connected by "rods" which can stretch, bend, or twist. The forces from these deformation modes are calculated based on the relative positions between the nodes [41]. Since only adjacent nodes directly effect each other, the Jacobian is banded, giving the dynamics a high degree of structure that can be efficiently exploited using implicit integration methods [47]. Due to the representation of the nodes, which is essentially maximal coordinates, DER also readily incorporates contact constraints [48]. I will cover the design of the robots, present applications of the robot for DER validation and planner development, and discuss lessons learned that led to future robot designs.

## 2.2 Star-shaped robot for underwater DER

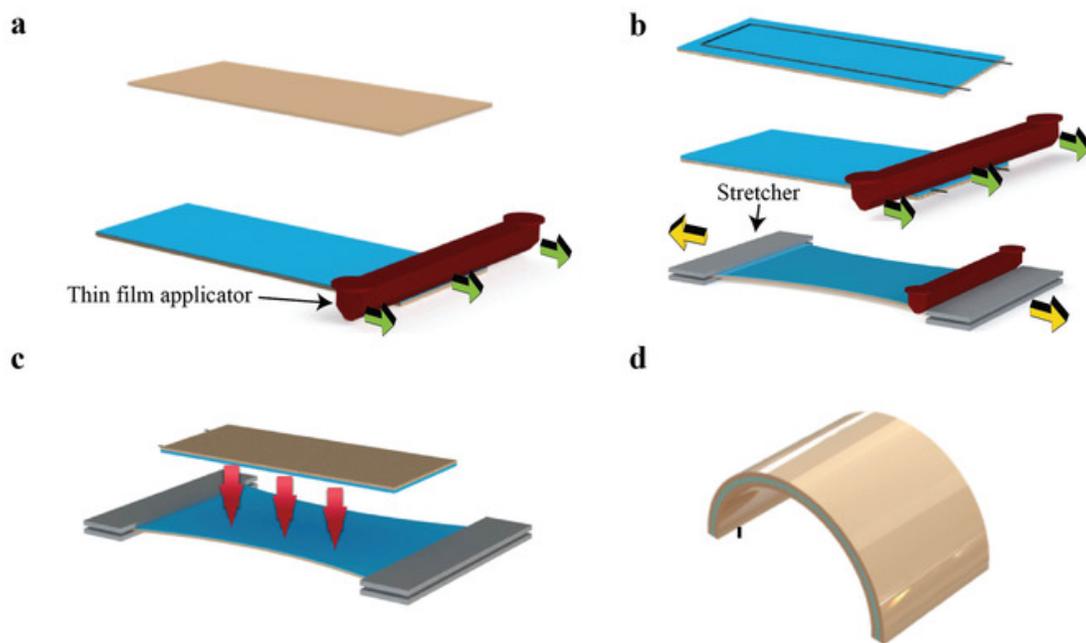
### 2.2.1 Design

We use the actuator design from Huang et. al. [49]. Briefly, a shape memory alloy wire loop is attached using Eco-Flex 30 to a layer of thermally conductive silicone (material). Then, an-

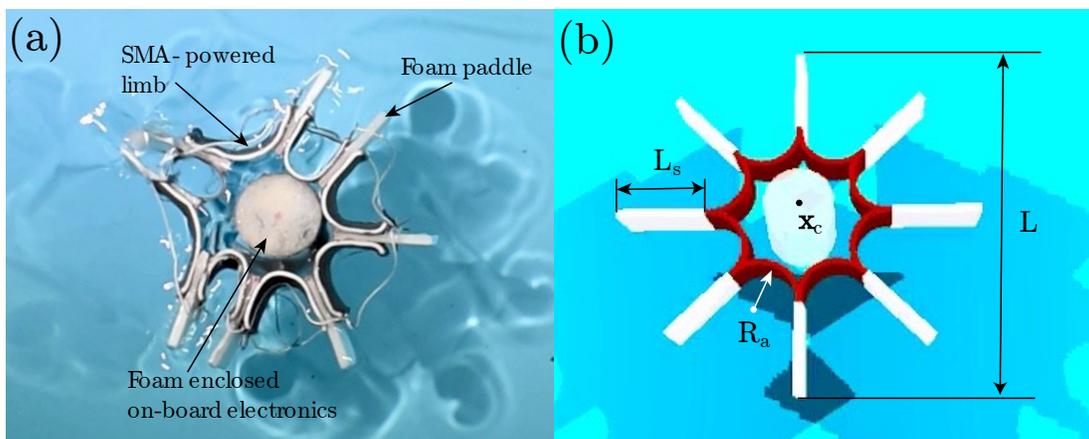
other layer of thermally conductive silicone is pre-stretched before being attached as well. This structure is then naturally curved while unactivated, but when activated with joule heating, the SMA wire causes the structure to straighten out rapidly. See Huang et. al. for a full description and characterization of the actuator. Eight of these actuators are radially connected, resulting in the star shaped structure shown in Fig. 2.2.2a. Each actuator is controlled with a transistor (AO3416, Alpha & Omega Semiconductor Inc.). The robot utilizes a Laird BL652 SoC which contains a Bluetooth enabled nRF52832 microcontroller. The robot is powered by a drone battery (300mAh, 45/75C, BETAFPV). The microcontroller uses Bluetooth Low Energy (BLE) to communicate with an offboard microcontroller, which relays communications via UART from an ubuntu controller running the Python program that serves as the control script to provide instructions. The core challenges in the design is ensuring that the electronics are waterproof and that the robot is neutrally buoyant. To make the robot waterproof, all electronic components are encased in silicone (DragonSkin 10 and SilPoxy). Neutral buoyancy is more difficult since the actuators are dense and the battery is obviously relatively heavy. For the actuators, we attach rectangular pieces of foam to the tips of the star structure so that the ring floats by itself. The electronics and battery are encased in a foam cylinder (Soma Foama 15, Smooth-On) with enough volume to ensure that the buoyancy force cancels out the weight of the battery. The robot's final diameter is 104mm and the mass is 91g.

### **2.2.2 Application**

Because of the complexity of soft robot mechanics, simulation is a difficult task. Recently, Huang et al have shown that DER can be useful for efficiently simulating their behavior, achieving faster than real-time computation [50]. We determined that the star shaped swimming robot is a good platform to empirically validate an extended DER simulation that takes hydrodynamic forces into account. Simulation results were gathered by my collaborator, Weichung Huang, while I worked with Xianoan Huang to gather experimental data.



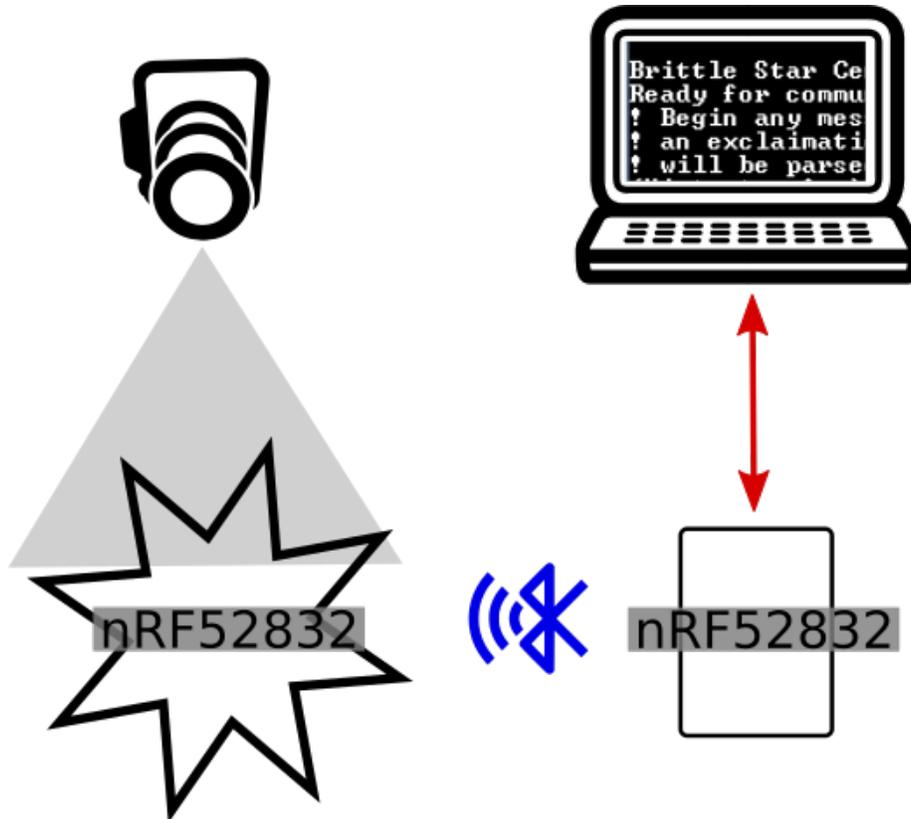
**Figure 2.2.1:** Schematic showing fabrication process and layout for a dynamic SMA-based rubber actuator.



**Figure 2.2.2:** (a) Components of the untethered star-shaped soft robot used as the experimental testbed. (b) Geometry of the untethered star-shaped soft robot as modeled in the computational simulation.

### 2.2.3 Testing

The robot testbed is powered on and placed in a pool. An overhead camera records for later analysis. The robot microcontroller connects via Bluetooth to its off-board counterpart. The robot operator determines the gait that is used and sets the Python program to use that gait. One of five gaits is used, with depictions of the gaits shown in Figure 2.2.4.



**Figure 2.2.3:** Test setup for the untethered floating robot. The setup includes a computer for issuing gait instructions, a microcontroller connected to the computer for relaying that information via Bluetooth to the robot, and an overhead camera for recording robot motion for offline analysis.

### 2.2.4 Results & Discussion

For each gait, the results for the star shaped robot are a competition between a drag force and a thrust (jetting) force. Whichever force dominates determines the direction of motion. In the cases A1 and A1 + A2, drag dominates and the robot moves in the “forward” direction. In cases

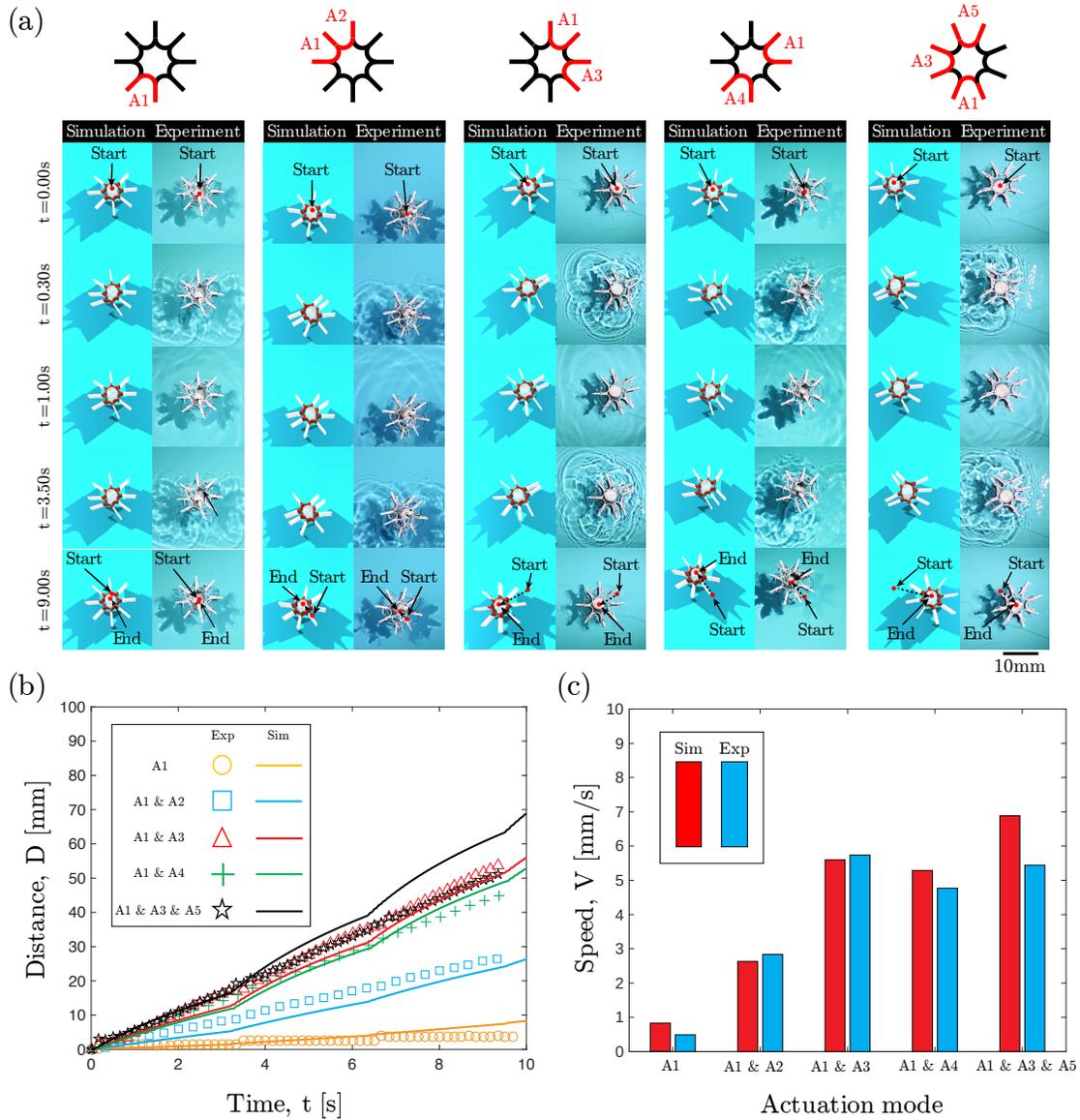
A1 + A3, A1 + A4, and A1 + A3 + A5, the thrust force dominates, causing the robot to move in the “reverse” direction. Case A1 is the minimum speed, with the drag and thrust forces nearly canceling each other out. A1 + A3 + A5 is the peak speed; multiple limbs actuating causes a great deal of fluid to jet out of the openings between them, accelerating the robot and causing those thrust forces to dominate. Figure 2.2.4 shows the experimental results vs the simulation. Good agreement is achieved for all gaits except for A1 + A3 + A5, where the simulation overestimates the speed. We hypothesize that this is due to non planar motion of the experimental robot caused by mass distribution asymmetry and/or fluid turbulence.

## **2.3 Closed Loop Motion Planning with Discrete Elastic Rods for Swimming Robots**

In the previous work, I showed the design of a star-shaped robot used for validating in-water DER simulations. In this work, my collaborators and I built upon that foundation by using DER to specify design and gait parameters for a frog-inspired robot. I then used a DER-based pipeline to develop an online planner capable of following arbitrary trajectories.

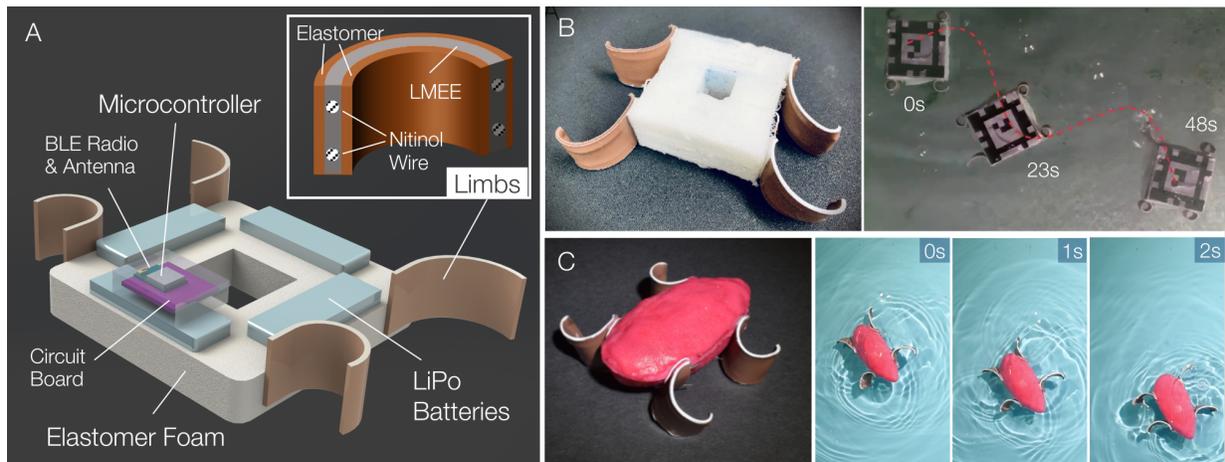
### **2.3.1 Experimental Testbed**

The untethered frog-inspired soft robot is composed of four SMA-driven soft actuators, four LiPo batteries, and an on-board PCB. (Fig. 2.3.1a). The actuators are composed of stretched and unstretched layers of thermally conductive silicone that are bonded around an interface layer of LM-elastomer composite containing an SMA wire loop. The LM-elastomer is composed of microscale droplets of eutectic gallium-indium (EGaIn) LM alloy that serve to enhance the thermal conductivity[51] of the interface layer. The design is based on [52]. The amount of EGaIn used in the interface layer decreases the cooling time, increases actuator speed, and increases the mass. The power and control electronics are housed inside of the foam bodies and sealed from



**Figure 2.2.4:** (a) Snapshots of the star-shape robot in five different swimming gaits with actuation of (i) actuator A1, (ii) actuators A1 & A2, (iii) actuators A1 & A3, (iv) actuators A1 & A4, and (v) actuators A1 & A3 & A5. (b) The displacement of the robot centroid as a function of time for 3 cycles from both experiments (dashed lines) and simulations (solid lines). (c) The absolute speed comparison between simulations and experiments in five different swimming gaits

water using silicone. Instructions are relayed to the robot's microcontroller via Bluetooth Low Energy from a remote computer and microcontroller. The body of the robot has a hollow square shape, which is selected so that the body of the robot can be taken as four elastic rods with high stiffness connected together (Fig. 2.3.1b). The robot limbs are attached to the body with velcro, resulting in a modular robot design so that each limb can be replaced within seconds. This enables the exploration of limbs with various actuation bandwidth without changing the robot design.

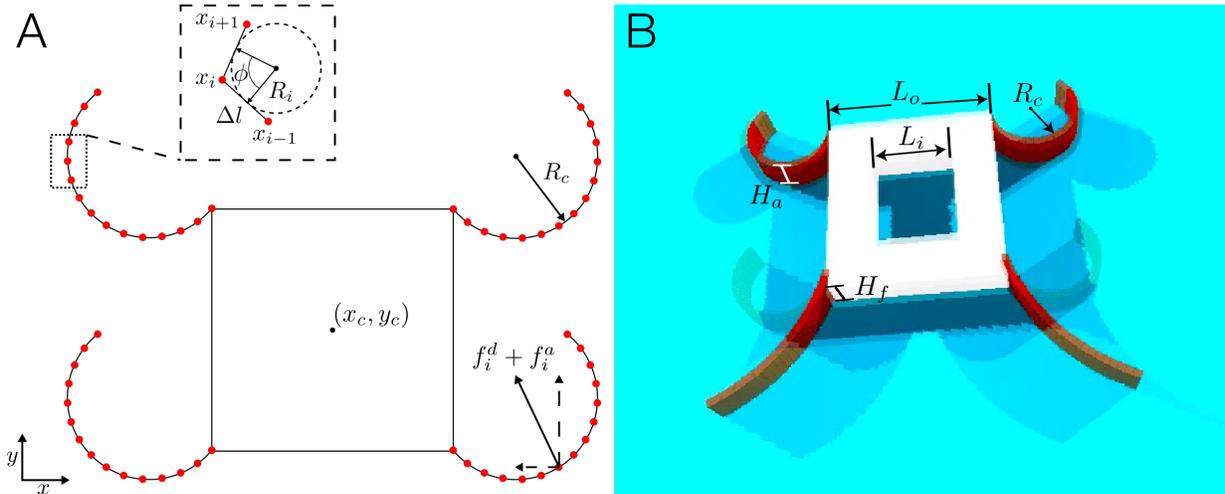


**Figure 2.3.1:** Design and geometry of the untethered frog-inspired soft robot. **a** Components of robotic system. **b** Square-shaped robot for examining motion along a curvilinear path. **c** Stream-lined motion for  $>1$  blps swimming along a straight line.

To demonstrate the versatility of this design, we also implement a more streamlined version of the robot that is capable of swimming at  $>1$  blps speed (Fig. 2.3.1c). Narrowing the central body of the untethered robot improves its hydrodynamics and allows for less drag during forward swimming. Although this allows for faster forward swimming, the narrow profile reduces the ability of the robot to make in-place turns. This is due to the reduced moment arm and in-plane torque that can be induced during differential limb actuation. For this reason, the remainder of this study will largely focus on the more maneuverable square-shaped robot. Nonetheless, the streamlined robot is presented here to demonstrate the potential capability for faster swimming speed that is possible within this design framework.

### 2.3.2 DER Simulation Tool

Although I did not develop the DER simulation, I will present an overview for reader understanding. To simulate this robot, we utilized the DER algorithm for modeling structures with slender elastic elements [41]. Starting from the discrete representation of elastic energies, we formulate equations of motion at each node and solve in a backward Euler approach to update the configuration of the robot (i.e. position of the nodes) in time.



**Figure 2.3.2:** Robot simulation. **a** Discretization of the robot with geometric properties and forces.  $R_c$  is the radius of curvature of the limb,  $(x_c, y_c)$  is the robot's center of mass position, and  $f_j^d$  and  $f_j^a$  are the drag and virtual mass forces on the  $j$ th node respectively. In the inset are properties of the discretized rod geometry, notably the turning angle,  $\phi$ , and the rod length,  $\Delta l$  which are later used to calculate bending energy.  $x_{i-1}$ ,  $x_i$ , and  $x_{i+1}$  are the labels of successive nodes. **b** Rendering in simulation and geometry of the frog-inspired soft robot.  $L_o = 90\text{mm}$ ,  $L_i = 35\text{mm}$ ,  $H_f = 35\text{mm}$ ,  $H_a = 24\text{mm}$ .

Similar to the DDG-based method presented in [42, 53], this numerical framework starts with a discrete representation of the robot. The soft robot is treated as a collection of discrete elastic rods [41], shown schematically in Fig. 2.3.2a. The rods are comprised of  $N$  nodes, located at  $\mathbf{x}_i = [x_i, y_i]^T$  (with  $i = 0, \dots, N - 1$ ), along the centerline. The limbs and the body of the robot are represented by rods with different density. In the discrete setting in **Fig. 2a**, the robot is represented by a lumped mass at each node and associated elastic stretching and bending energies – reminiscent of a mass-spring system. Since the motion of the robot remains in 2d, we do not include a twisting energy of the rod, although this can be readily integrated into our framework.

The rod segment between two consecutive nodes is an edge that can stretch as the robot deforms – analogous to a linear spring. The turning angle  $\phi_i$  (Fig. 2.3.2a) at node  $\mathbf{x}_i$  between two consecutive edges can change – similar to a torsional spring. The elastic energy from the strains in the robot can be represented by the linear sum of two components: (1) stretching energy of each edge  $E_i^s$  and (2) bending energy  $E_i^b$  associated with variation in the turning angle  $\phi_i$  at the nodes. The elastic stretching and bending forces acting on a node  $\mathbf{x}_i$  can be obtained from the negative gradient of the elastic energies. The external forces acting on a node  $\mathbf{x}_i$  are  $\mathbf{f}_i^{ext} = \mathbf{f}_i^d + \mathbf{f}_i^a$ , where  $\mathbf{f}_i^d$  is the damping force from fluid experienced by the soft limbs and body and  $\mathbf{f}_i^a$  is the added-mass force from periodically accelerating the surrounding fluid (Fig. 2.3.2a). These forces are then used to formulate equations of motion, which are implicitly solved for the next time step. After validating the simulation and selecting the robot design and gait parameters, we develop a motion planning framework that enables closed loop path following along arbitrary shapes in 2D space. While DER is capable of faster than real-time simulations [42], the dynamics implementation is not fast enough to run inside an online optimization loop. Instead, we leverage the capability to run many DER simulations in a relatively short period of time to collect a large set of data capturing the robot’s dynamics resulting from a set of 9 actions (or motion primitives). This data library can then be used as the basis for a closed loop motion planner - similar in form to explicit model predictive control - to choose the best available action according to some cost function associated with the projected future state.

### 2.3.3 Online Planner for Path Following

The first step in preparing our motion planning framework is collecting our data library. The data library consists of a large number of transition models that give the future state of the robot for some initial state and some action. To produce such a library in a way that allows fast operation at run-time, we reduce the space of needed samples by constraining the number of actions that the robot can perform and the number of initial states of the simulations. We

manually choose nine actions, i.e. SMA actuation sequences, corresponding to behaviors like “go forward,” “turn and go forward”, “turn in place,” and “no action.” Next, we take advantage of the invariance of the robot’s dynamics to its position and orientation and only sample over distributions of initial linear and angular velocity states. The resulting set of simulations consists of 8,503 initial velocity states (giving 76,527 total transitions). These simulations are performed offline and the data is collected and processed into a large matrix such that the transitions can be applied at run-time by efficient indexing operations.

In order to use the data library in a path following feedback control policy, we implemented a search-based planning algorithm over a tree generated by branching over primitives, allowing the robot to choose the best available action given its current state. The implemented algorithm is a receding horizon planner using the nearest neighbor in the data library to form predictions about future states. Paths are specified by generating a series of waypoints from a parametric curve in world-space. To track the path, the current state is first captured from a camera that tracks the robot using an Apriltag [54, 55]. Then, this state is transformed from the world coordinate frame into the body frame to compare it to the states in the data library. We find the closest state in the library and apply the projected transitions for each of our 9 actions to the current state and transform back to world coordinates to get our projected next state. We recursively repeat the above until we reach the specified depth of our motion planning tree. Finally, we compare the costs associated with our projected state and choose the action leading to the lowest cost branch on the tree.

The desired path waypoints are determined by parametrizing 2D curves by a single parameter,  $s$ , which we discretize as 100 evenly spaced points in  $[0, 1]$ . The straight line path is

$$\mathbf{p}_{line} = [\ell s, 0]^T,$$

where  $\ell$  is a constant that sets the path length. The sinusoidal path is

$$\mathbf{p}_{sin} = \left[ \ell s, \frac{\ell}{A} \sin(fs) \right]^T,$$

where  $f$  is the frequency and  $A$  is an arbitrary constant representing the height of the sin curve relative to the length,  $\ell$ . The ellipse path is

$$\mathbf{p}_{ellipse} = \left[ \frac{\ell}{2} (1 - \cos(2\pi \frac{s}{s_f})), \frac{\ell}{2A} \sin(2\pi \frac{s}{s_f}) \right],$$

where  $A$  is again an arbitrary constant representing the ratio of height to length of the ellipse and  $s_f$  is the final value of the parameter  $s$ .

The algorithm for controlling the robot to follow a specified path is shown in Algorithm 1.

When predicting the next state of the robot based on a given transition, the following update is applied:

$$\begin{aligned} \begin{bmatrix} x(t+T) \\ y(t+T) \end{bmatrix} &= \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + R(\theta(t)) \begin{bmatrix} x_{next} \\ y_{next} \end{bmatrix} \\ \theta(t+T) &= \theta(t) + \theta_{next} \\ \begin{bmatrix} \dot{x}(t+T) \\ \dot{y}(t+T) \end{bmatrix} &= R(\theta(t)) \begin{bmatrix} \dot{x}_{next} \\ \dot{y}_{next} \end{bmatrix} \\ \dot{\theta}(t+T) &= \dot{\theta}_{next} \\ \mathbf{q}(t+T) &= [x(t+T), y(t+T), \theta(t+T), \dot{x}(t+T), \dot{y}(t+T), \dot{\theta}(t+T)]^T. \end{aligned} \tag{2.1}$$

The cost function used to evaluate the best action has 3 components (as in [56]) - distance from the nearest waypoint, difference between the robot angle and the angle of the tangent of the path at the nearest waypoint, and the progression along the trajectory associated with the

---

**Algorithm 1: Robot Planner**

---

```
1  $C \leftarrow \infty$ ;  
2  $i \leftarrow 1$ ;  
3 Get state  $\mathbf{q}(t) = [x(t), y(t), \theta(t), \dot{x}(t), \dot{y}(t), \dot{\theta}(t)]^T$  from camera;  
4 while  $i > H$  do  
5   Transform from world to body coordinates by rotating  $[\dot{x}(t), \dot{y}(t)]^T$  by  $-\theta$  using the  
   standard 2D rotation matrix/;  
6   For states in library  $\{q_k\}, k = 0, \dots, K-1$  calculate distance  $d_k$  and stack into vector  
   d:  
       
$$d_k = \left\| \left[ (\dot{x}(t) - \dot{x}_k) / \dot{x}(t), (\dot{y}(t) - \dot{y}_k) / \dot{y}(t), (\dot{\theta}(t) - \dot{\theta}_k) / \dot{\theta}(t) \right] \right\|_2.$$
  
       
$$\mathbf{d} = [d_0, d_1, \dots, d_{K-1}]$$
  
7    $k_{min} = \arg \min_k \mathbf{d}$   
8   for  $a_j \in \mathcal{A}$  (where  $a_j$  is an action and  $\mathcal{A}$  is the set of all actions) do  
9     Obtain transition prediction of system state after primitive execution period  $T$ ,  
      $q(t+T) = \mathbf{q}_{next} = F(\mathbf{q}_{k_{min}}, a_j)$  from Eq. 10.  
10    Rotate the transitions to the world frame by  $\theta$  using the standard rotation matrix  
      $R(\theta)$  and apply to the current state by applying Eq. 11.  
11     $\mathbf{q}(t+T) \leftarrow \mathbf{q}(t)$ .  
12     $i = i + 1$   
13  end  
14  Calculate cost function from Eq. 12.  
15   $a^* = \arg \min_j C$   
16 end  
17 Execute  $a^*$ 
```

---

nearest waypoint:  $cost = w_a * dist + w_b * ang + w_c * prog$ , where the  $w$ 's represent weights for each element of the cost. This cost function induces the robot to stay near the path, maintain the correct heading, and continue forward along the path.

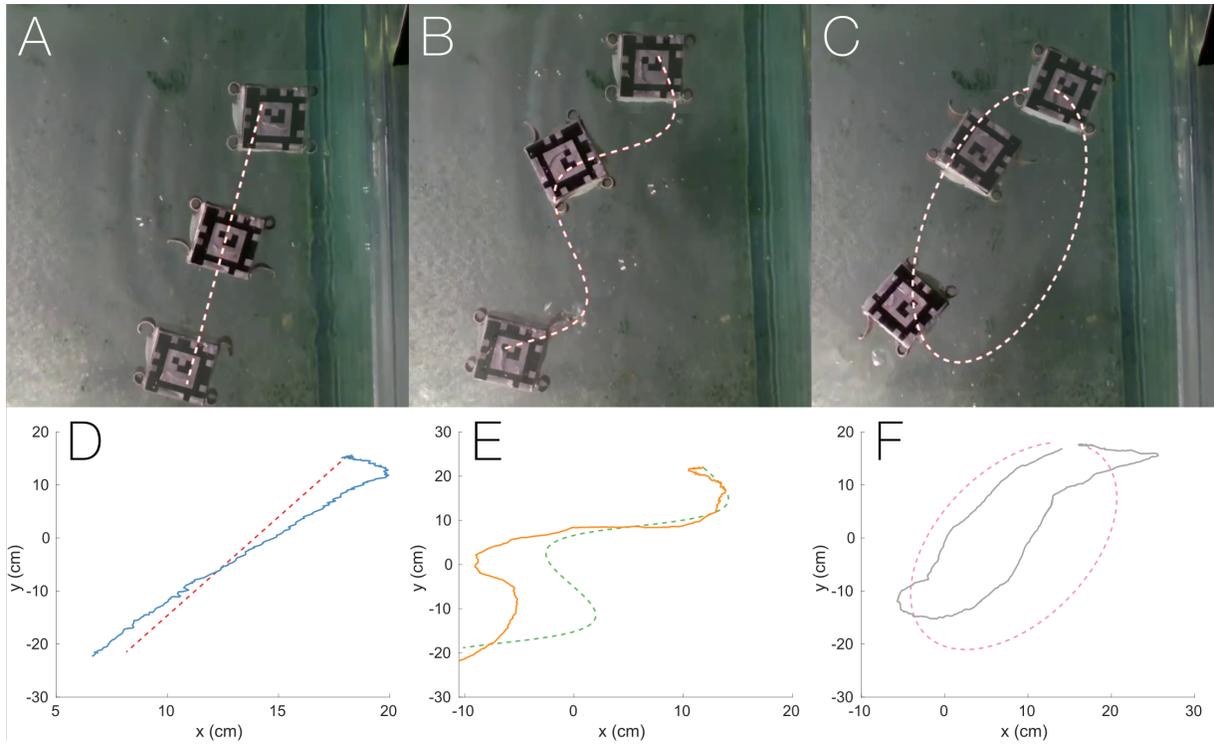
The cost function used to determine the best action is

$$C = w_a * \min(\|\mathbf{x}_c(t) - \mathbf{p}(s_c)\|) + w_b * \frac{\mathbf{x}_c(t) \cdot \mathbf{p}(s_c)}{\|\mathbf{x}_c(t)\| \|\mathbf{p}(s_c)\|} + w_c * (1 - s_c), \quad (2.2)$$

where  $s_c = \arg \min_s(\|\mathbf{x}_c(t) - \mathbf{p}(s)\|)$ .

The cost function is intended to incentivize proper positioning on the path, proper angular heading and forward progression. The weights used in the experiments presented are  $w_a = 500$ ,  $w_b = 50$ , and  $w_c = 300$  for positioning, angular heading, and forward progression components respectively.

We tested the motion planning approach on several different types of paths including a straight line, a sinusoidal curve, and an ellipse. Fig. 2.3.3a, b, & c shows snapshots of the robots along each trajectories and Fig. 2.3.3d, e, & f shows plots of the x-y position of the robot compared to the waypoints of each of the paths. Fig. 2.3.3g shows the distance of the robot to the nearest point on the discretized trajectory for each case. From these results, we can see that the robot can successfully follow paths of varying complexity at fairly high speed (2.2-3.2 cm/s). While tracking is not perfect due to the high dimensional representation and low control bandwidth, the robot recovers robustly from deviations and is able to maintain the trajectory qualitatively towards the goal. In practice, we found that a depth of one (i.e. only considering the next step) for the motion planner resulted in the best performance.



**Figure 2.3.3:** Stills of the robot at the beginning, middle, and end of a **a** straight line path, **b** sinusoidal path, and **c** ellipsoid path. 2D plots showing  $(x,y)$  position of the robot and the paths for **d** straight line, **e** sinusoid, **f** ellipsoid. **g** Distance from the nearest point on the path for each trial color coded to correspond with the plots in **d-f**

### 2.3.4 Discussion

After demonstrating that DER could predict system dynamics with reasonable accuracy across varying design and gait parameters, we performed a parameter sweep to identify the fastest and most efficient sets of design parameters (EGaIn volume fraction) and gait parameters (phase, frequency). We showed that there is good agreement between the predicted speeds based on simulation and the resulting robots' speeds. Thus, DER is shown to be an effective tool for soft robot design. While the parameter space examined here is narrow, since the simulation is fast and easily automated it would be easy to sweep across a larger parameter space with more candidate robots and gaits.

We also demonstrate, for the first time, an online planning framework leveraging DER. Our resulting implementation compares favorably with recent simulation-driven soft robot trajectory

optimization and open loop control schemes [38], enabling high level tasks to be performed at relatively fast robot speeds (3.2 cm/s versus 0.75 cm/s) and control frequencies. Our planning approach, where we precompute a large library of trajectories based on motion primitives, is popular across many areas of robotics [57] including flying robots [58], grasping robots [59], autonomous vehicles [60], and other systems with real-time execution requirements and constraints on run-time execution. At run-time, we can then plan ahead efficiently by finding the nearest neighbor within our library to the current state and projecting forward based on the pre-simulated transition model using a receding horizon plan [61]. We experimented with interpolation but nearest neighbor proved more performant. This library-based, tabular method has pros and cons versus solving the full dynamics. By representing the dynamics as essentially a large array of transitions, our function evaluations are far more efficient than solving differential equations and we can plan deeper trees. We could also straightforwardly implement the planner on low level hardware to enable cheap mobile autonomy and efficient collision avoidance [62]. On the other hand, we are constrained to a pre-defined set of motion primitives and cannot optimize at run-time over the full space of possible actuations. This produces trajectories that are predictably sub-optimal. Also, although we implement a receding horizon planner capable of fast performance at large depth, in practice we find that the best performance occurs for a tree depth of one. One likely explanation is a distribution mismatch between the library and the states produced by performing sequential action on the hardware. This would mean that transition prediction error is propagated at each level and degrades performance at larger search depth. This problem could be remedied by sampling more of the state space during library generation. To assess these issues, in future work we will compare the performance across various sizes of the data library and compare against a planner that queries the DER simulation itself for transitions in the loop. Regardless, the effectiveness of our approach on hardware demonstrates that DER can be readily adapted into standard robot motion planning frameworks.

While our approaches to design and control were effective for the class of robot introduced,

the effectiveness is uncertain for more complex soft robots [19, 63]. While it has been previously shown that such robots can operate with even simpler high level planning over a small set of motion primitives [64], it would be useful to use DER to compare performance across design parameters, to optimize gaits offline, and to enable more complex sequences of actions. Finally, while DER can efficiently and accurately include contact, we do not include it in our simulations. In future work, we will optimize performance during contact based locomotion.

This work introduced a new class of frog-like soft robots that can achieve fast locomotion speeds. We also show multiple new functionalities of the Discrete Elastic Rod (DER) framework that make it an effective modeling tool for simulating and controlling robots with deformable components. In utilizing DER, we first calibrate the simulation and show that it provides realistic results across a range of parameters, including robot design and gait. Then, we performed a parameter sweep to find the fastest and most efficient sets in the space. Next, we use the simulation to generate an online planner that can be used for trajectory tracking with one of our selected robot designs. Lastly, we implement this motion planning scheme on an experimental testbed and demonstrate the ability for a fully untethered, frog-like soft robot to swim along various pre-defined paths within a water tank.

## 2.4 Conclusion

This Chapter covered my work on creating 2D floating robots for validation of Discrete Elastic Rods simulators for fluid-soft-structure interactions. Critically, it was shown that these simulators can serve as high throughput tools for robot design, gait design, and online planning. In the future, such simulators can systematize the design of the next generation of soft robots, replacing the ad-hoc design processes that characterized the robots of this chapter. The robots themselves are also an interesting contribution in their own right, and the frog robot in particular is very fast and agile. After working on these projects, I determined that, although the simplicity of the actuators can be considered a virtue, it diminishes their usefulness for sophisticated control or

planning. Specifically, the actuators only have one controllable degree of freedom. What's more, the high speed and dynamic nature of the actuators compounds the problem. These properties are excellent if we want to build a simple robot that can move rapidly with open loop control, but if we want to experiment with more sophisticated control or motion planning, we are deeply constrained.

A more effective SMA based actuator for these purposes would have the following features:

- Multiple controllable degrees of freedom for a larger active state space
- 3 dimensional task space
- Larger deformation capability
- Slower actuator dynamics

## Chapter 3

# PATRICK - A Brittle Star Inspired Unthethered Soft Robot

*This work appears in:*

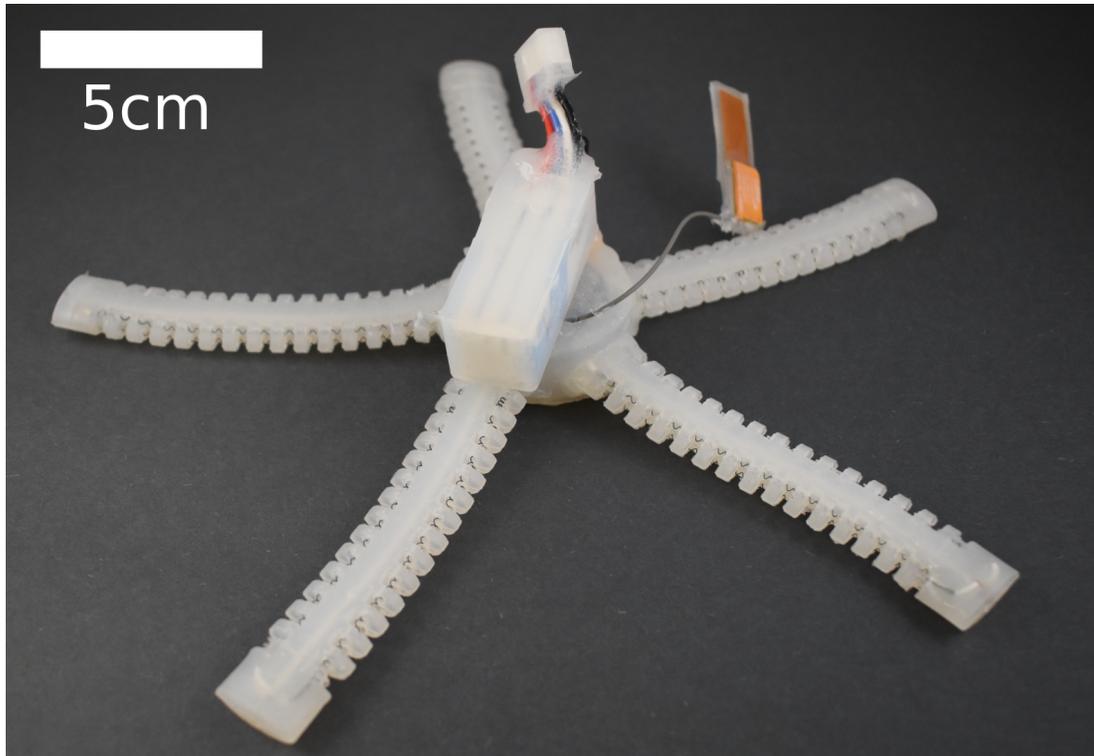
Z.J. Patterson, A.P. Sabelhaus, K. Chin, T. Hellebrekers, C. Majidi, “An Untethered Brittle Star Robot for Closed-Loop Underwater Locomotion.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020.

*Contributions:* I contributed the robot design, fabrication, experiments, and helped with developing motion planning algorithms and embedded code.

### 3.1 Motivation

Based on the work presented in Chapter 2, I concluded that we would need a more complex untethered platform to investigate problems of control and planning, with the ultimate goal of a highly versatile soft robot research platform. Additionally, I wanted to use biology to inform the design of the robot to enable the platform’s use in an experimental comparative biomechanics

framework. I chose the brittle star as a model organism because a) it is an underwater organism, b) the omnidirectional movement of the organism is an interesting capability to use for a robot, and c) relatively few robots have been used to investigate underwater walking and I felt I could create a robot that more effectively mimics the biomechanics than existing underwater walking robots. This chapter will discuss the design of this robot, called PATRICK, and its testbed, along with initial high-level motion planning experiments to validate the usefulness of the platform.



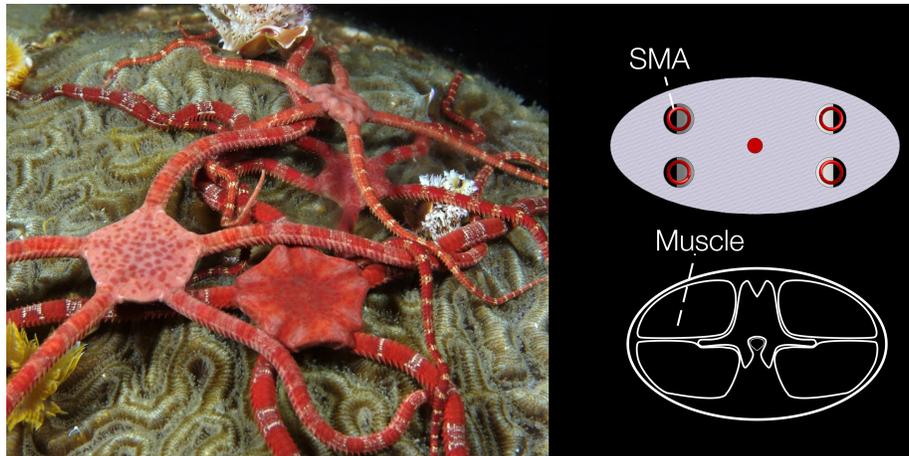
**Figure 3.1.1:** The flexible underwater untethered brittle star robot, PATRICK.

## 3.2 Methods

### 3.2.1 Robot Design

Ophiuroids, more commonly referred to as brittle stars, are echinoderms of the class Ophiuroidea and are closely related to sea stars. A highly successful class, brittle stars live in most regions of the ocean [65] and move by using their flexible arms to crawl along the sea floor. There

are many individual species with an immense amount of variation in morphology, but relatively common features. They can vary in diameter from a few millimeters to up to 60 cm [65]. Most brittle stars have five limbs arranged in a pentaradially symmetric configuration. These limbs have a calcium carbonate skeleton formed by structures called vertebral ossicles, which are actuated by longitudinal muscles running through the arms. See Figure 3.2.1 for a cross section of the arm morphology.



**Figure 3.2.1:** Left: image of several brittle stars. Right: Cross section of robot arm compared to brittle star arm.

In order to capture a credible analog of brittle star morphology, I first designed actuator-embedded appendages for locomotion, which I will call limbs. The limbs should be conceptualized as lengthy continuum beams with artificial muscles embedded in the structure to create bending deformations. See Figure 3.2.1 for a cross section comparison of my design concept vs the morphology of a brittle star arm. This design was also inspired by the work of Walters and McGoran from Bristol Robotics [66], who demonstrated a simple actuator concept utilizing wedge shaped notches around the artificial muscle to improve flexibility. I took this concept and expanded it from two actuators to the four actuator structure specified by the brittle star morphology.

The next design decision was the choice of actuator to serve as artificial muscle. There are many options for soft or flexible actuators, each with its own tradeoffs. I chose SMA coils (Dy-

nalloy, 0.008in diameter wire) for the following reasons. SMA coils can recover very high strains ( $> 200\%$ ) while maintaining reasonably high pull force ( $385.5mN$ ). They are easily integrated both electronically and mechanically, with few additional components required because they can be activated by simple joule heating. Although they are very inefficient (usually  $< 1\%$  depending on the application [67]) and require high electric current for rapid response ( $> 1A$ ), they are easily powered by high discharge rate LiPo batteries, such as those used for drones. Finally, I had already successfully designed untethered robots powered by SMA as presented in Chapter 2, and so I could take advantage of my existing knowledge.

Given these traits of the fundamental actuator technology, the robot developed has the following properties. The robot is relatively simple to design, with simple electronics. Since SMA is so compact, the robot can be made compact as well. There will be a dichotomy between the robot dynamics and the actuator dynamics, with the actuator dynamics (incorporating material stress/strain, temperature, and electric charge) occurring rapidly while the robot dynamics occur slowly. Because the actuators are controlled by joule heating, the cost of transport will be high (due to the inherent inefficiency of joule heating). Also, in a convective environment, this style of activation along with the use of flexible structures results in a highly dissipative system (in an energetic sense - informally, the rate of energy supplied is always greater than the rate of energy stored by the system), and therefore a theoretically highly stable system. Because SMA function is highly variable due to slight changes in material composition and sensitivity to environmental conditions, performance will be effectively non-stationary. By designing these actuators, we therefore manifest a tradeoff: we sacrifice energy efficiency and highly repeatable function for a compact robot that has decent control properties in the configuration space. Ultimately, the goal is a platform that allows us to bypass the low level problems at the actuator level and begin experimenting with control and planning at the level of robot geometry.

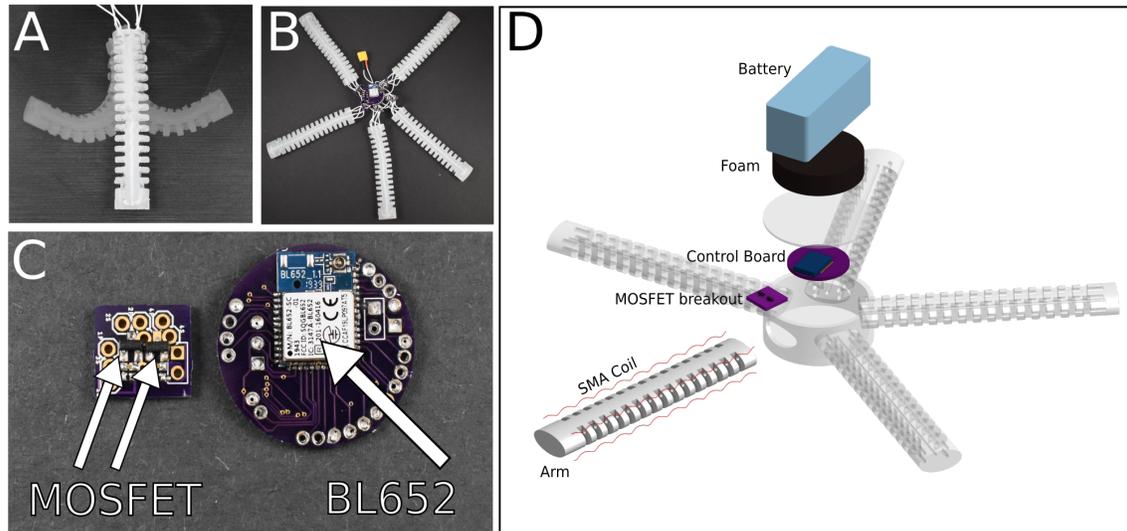
### 3.2.2 Fabrication

With the full limb concept specified, an individual limb is fabricated as follows. First, the rod-like silicone (Smooth-On DragonSkin 10) limbs are molded over a central wire, with inserts placed in the mold to provide the channels for the SMA coils. After demolding, the four SMA coils are fed through the channels and attached to the central wire at the distal end. An end cap is then overmolded at the distal end, along with a tab at the proximal end to eventually serve as the mating part with the body of the robot. The limb is then operated by applying electric current to the wires that correspond to a desired deformation. Because of the asymmetry in the configuration of the notches, the limb bends more readily along the major axis of the ellipse than the minor axis - just like a brittle star limb. The robot has a total of five limbs, in line with the majority of brittle stars. Therefore, the robot has 20 SMA actuators, resulting in a potentially very large controllable state space.

The electronics are very similar to that of the robots in Chapter 2. I again utilize the Laird BL652 SoC, which incorporates the nRF52832. The controller cannot directly power the SMA coils; instead, I use SQ23 (40V) surface mount MOSFETs with the controller GPIOs controlling the gates. To actuate, the MOSFET pulls the end of the coil low, causing a voltage difference which results in current flow and joule heating. The robot is powered by an off the shelf BETA FPV 11.1V, 300mAh, 45/75C drone battery. A design challenge that arises from operating in an aquatic environment is the interface of the battery with the robot. In early stages of the design, the battery was plugged directly into the robot using the battery's default XT30 connectors. Unfortunately, these are not waterproof, and so the interface had to be sealed in silicone each time the battery was replaced, resulting in a roughly 20 minute turnaround time. To solve this problem, I replaced the XT30 connectors with JST WPJ series waterproof connectors. These allow the battery to be rapidly replaced while also maintaining a watertight interface.

The custom PCB containing the controller and MOSFETs is attached to the limbs of the robot

before being sealed into the silicone body to prevent water from entering. The body is designed with slots for the limbs; the limbs have tabs at the proximal end that pull through the slots and keep them securely in place. The completed robot is 25 cm in diameter and 140 grams.



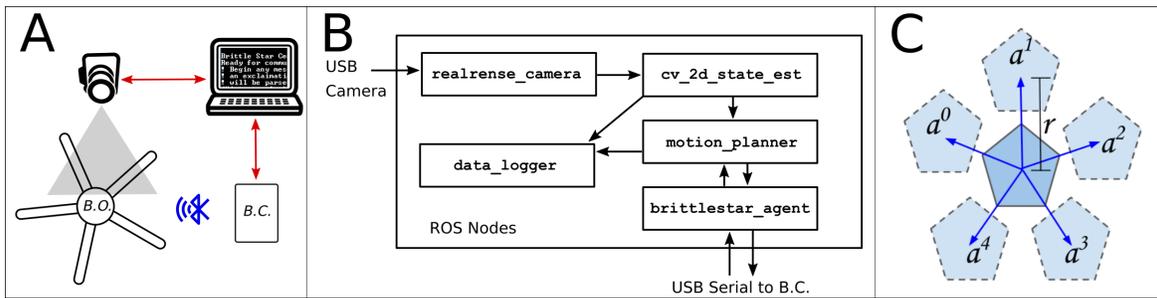
**Figure 3.2.2:** (A) Brittle star robot limb with SMA actuators. (B) Robot electronics and limbs before sealing into the central silicone hub. (C) Onboard system electronics including the microcontroller PCB and a MOSFET breakout board (5X). (D) System subcomponents from top to bottom: battery, foam for neutral buoyancy, control PCB, MOSFET breakout PCB, SMA spring, silicone arm.

### 3.2.3 Testbed

Control instructions for the testbed are generated in algorithms implemented in Python scripts or manually entered into a Python script by the user. For an untethered underwater robot, this leads to a design challenge of wireless communication underwater. Standard RF communication protocols do not work well in this setting because the signal is quickly degraded by reflection, refraction, and other optical effects of the water. Acoustic communication is therefore the typical choice. Unfortunately, there are no easily implementable off the shelf acoustic communications solutions available. So, I chose to use Bluetooth for communication, accepting the drawbacks of using RF. This was primarily to avoid getting bogged down in designing and implementing a workable acoustics solution since that is far outside the scope of my objectives or expertise. The

nRF52832 microcontroller has a built in Bluetooth antenna, which we use to communicate with a separate, off-robot nRF52832 controller. Because the robot crawls along the bottom of the water, placing the off-robot controller under the raised tank ensures minimal travel through the water and the signals are therefore received with little to no degradation. This solution only works in the lab; for any future experiments or applications outside of the lab, a new communication strategy would likely be necessary.

The off-board controller, which we call `brittlestar_central`, is connected to the USB port on a computer running Ubuntu, relaying communications to and from the computer via UART. The final piece of hardware in the testbed is an Intel RealSense camera that can be used for feedback control or ground truth position tracking.



**Figure 3.2.3:** (A) System architecture for PATRICK and its testbed. A camera, pointed at the robot, tracks various markers via OpenCV running on a connected computer. Embedded software on the robot (“Brittlestar Onboard”) communicates over Bluetooth Low Energy to similar software running on a separate board (“Brittlestar Central”) for communication over USB (red) to various ROS nodes on the computer. (B) Architecture for the various nodes used in the Robotic Operating System (ROS) package for the PATRICK robot. The Intel RealSense camera package supplies frames, then four separate nodes perform state estimation from markers on the robot, motion planning, data logging, and communication with the robot itself. Commands are sent to and from the hardware robot via the *brittlestar\_central* microcontroller attached over a USB serial port. (C) The idealized state transition model used by the planner.

Initially, we used ROS to implement the software, with several nodes handling the asynchronous functionality. More recently, we’ve replaced ROS with custom Python scripts for greater control over the code and better access to mature libraries. In either case, separate nodes or modules are used to handle the different functions of the system including camera frame capture, state estimation using computer vision, motion planning, and serial communication. The camera frame capture module is supplied by Intel and is implemented in its own ROS node for

the ROS program or as an API for the Python program. The state estimation module is built using OpenCV to track either colored dots or april tag fiducials. The motion planning module is a Python or Cython script running the motion planning algorithm. Finally, the serial communication module sends instructions to brittlestar\_central, which then relays them to the robot. A command library was implemented to operate each SMA coil as a finite state machine. By varying the time of activation, the motion planning system can control deformation of the limbs. Safety timers are implemented to prevent SMAs from burning out if an off command is never received.

### 3.2.4 Motion Planning

To generate directed locomotion for PATRICK, we could consider a number of motion strategies. For this work, we decided to use a very simple strategy based on discrete motion primitives as a proof of concept. Ongoing and future work will expand upon it. This motion planning framework is implemented as a search over a set of manually tuned motion primitives. The following section will cover it in detail.

As discussed, PATRICK’s actuators exhibit complex dynamics, including phenomena such as large strain deformation and thermal dynamics, that are highly nonlinear and often coupled. A full approach that takes into account actuator dynamics would therefore be difficult to practically implement, especially given that SMA properties vary from batch to batch due to sensitivity to small variations in composition. So, for this work we choose to take a low dimensional representation for the robot and its inputs. The Intel camera allows us to track the position and orientation of PATRICK in the XY plane. Because PATRICK does not currently swim, this is sufficient for tracking planar crawling at the bottom of the tank. The full robot’s state is then specified as

$$\mathbf{x}_t = [x, y, \theta], \quad \mathbf{u}_t \in \{0, 1\}^{20},$$

where  $u_t$  represents each of the 20 actuators as activated or deactivated. Finally, the goal state is specified as  $\bar{\mathbf{x}} = [\bar{x}, \bar{y}]$ . For motion planning, each *motion primitive*, or equivalently *action*, is specified as a sequence of inputs  $u_t$  over time,

$$\mathbf{a}_t = [\mathbf{u}_t, \mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+T}].$$

The transition model between states is taken with respect to these motion primitives instead of with respect to time and is represented as an unknown function

$$\mathbf{x}_{t+1} = \mathbf{x}_t + F(\mathbf{x}_t, \mathbf{a}_t), \quad F(\mathbf{x}, \mathbf{a}^i) = [\Delta x^i, \Delta y^i, 0],$$

where each  $\Delta x^i, \Delta y^i$  are a translation due to action  $i$ . These state transitions are assumed to translate the robot without changing its orientation, an assumption based on brittle star locomotion. Typically, when a brittle star moves, it chooses a “leading limb” to serve as the “front.” When making large changes of direction, instead of turning its body so that this leading limb remains in front, the brittle star simply selects a new leading limb corresponding to the desired direction. Although in practice PATRICK’s orientation often changes during the transition, the algorithm is robust to large deviations because orientation is resampled by the state estimator at each step and the next decision is made only with respect to the current step.

I will now discuss the motion primitives themselves. They are designed to mimic the “rowing” gait of the brittle star as presented by Astley /cite. Because each requires a leading limb, this implies five primitives, or one for each limb. Because of this, and the radial structure of the robot, the primitives are most easily specified in polar coordinates. For action  $i$ ,

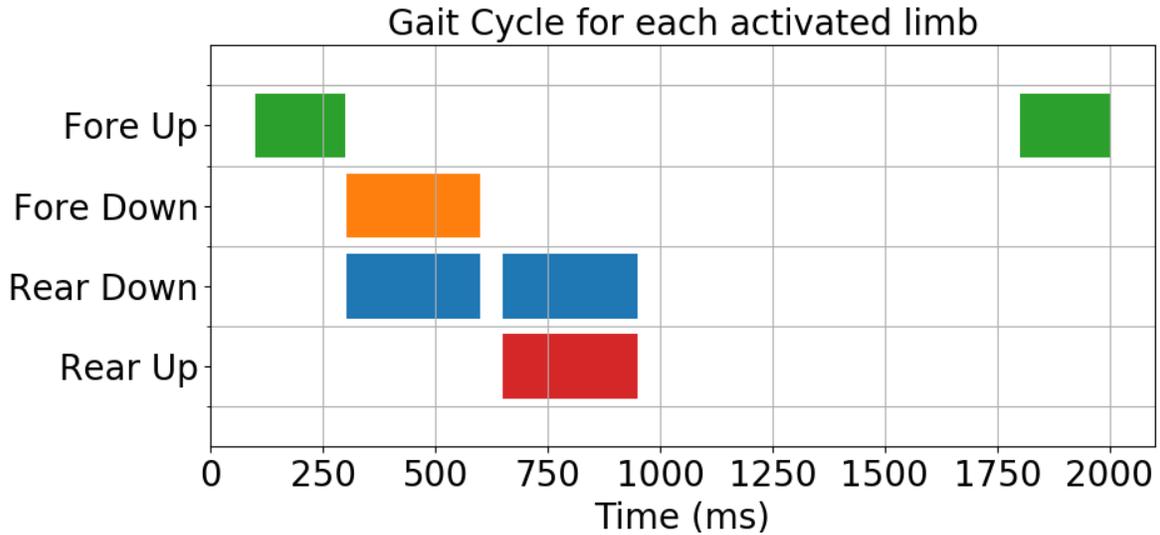
$$[\Delta x, \Delta y] = [r^i \cos(\theta_t + \phi^i), r^i \sin(\theta_t + \phi^i)],$$

which results in a displacement in the direction of the corresponding leading limb. Designating leg 0 to point in the  $+Y$  axis at  $90^\circ$ ,

$$\begin{bmatrix} r^i \\ \phi^i \end{bmatrix} = \begin{bmatrix} r^0 & r^1 & r^2 & r^3 & r^4 \\ 90^\circ & 18^\circ & 306^\circ & 234^\circ & 162^\circ \end{bmatrix}$$

where  $r^i$  represents a constant displacement magnitude and  $\phi^i$  represents the direction of the leading limb relative to the robot frame.

The precise sequence of SMA activations for the primitives was determined through a manual, trial and error process. Qualitatively, this rowing gait is as follows. The two limbs on either side of the leading limb swing forward and off of the ground, then down into the ground to push the robot up before finally swinging backwards to propel the robot forward.



**Figure 3.2.4:** The activation sequence of the SMA coils for the motion primitives

With the hard coded actions (motion primitives) and transitions, a search-based motion planning algorithm was implemented to follow a greedy policy to the goal. This greedy policy is a sequential optimization over the set of motion primitives, minimizing the cost function which is

the Euclidean distance,

$$c(\mathbf{x}_t) = \|\mathbf{x}_t - \bar{\mathbf{x}}\|_2,$$

where the previously specified transition model is used to predict the next state for each possible current action. This policy, shown in Algorithm 1, uses a single step horizon, which prevents error from accumulating.

---

**Algorithm 2:** A greedy model-based policy

---

**Input** : Robot state  $\mathbf{x}_t$ , goal position  $\bar{\mathbf{x}}$ , set of primitives  $\mathbf{a} \in \mathcal{A} = \{\mathbf{a}^0, \mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3, \mathbf{a}^4\}$ , transition function  $\mathbf{x}_{t+1} = \mathbf{x}_t + F(\mathbf{x}_t, \mathbf{a}_t)$ , tolerance  $d$  for distance to goal

**Output:** Closed loop trajectory of motion primitives  $\mathbf{a}_t$  to get to the goal

```

1 while  $\|\mathbf{x}_t - \bar{\mathbf{x}}\|_2 > d$  do
2   for  $\mathbf{a}^i \in \mathcal{A}$  do
3      $\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t + F(\mathbf{x}_t, \mathbf{a}^i)$ 
4      $c(\hat{\mathbf{x}}_{t+1}) = \|\hat{\mathbf{x}}_{t+1} - \bar{\mathbf{x}}\|_2$ 
5   end
6    $\mathbf{a}^t = \arg \min_{\mathbf{a}^i} c(\hat{\mathbf{x}}_{t+1})$ 
7    $\mathbf{x}_{t+1} \leftarrow \text{Robot.execute}(\mathbf{a}^t)$ 
8 end

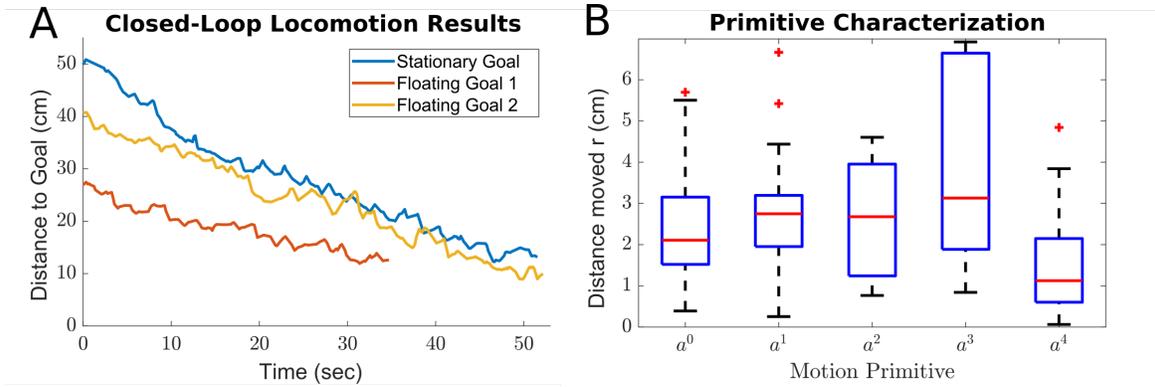
```

---

### 3.3 Results

We characterized the execution characteristics of the primitives, illustrated in Fig. 3.3.1B. With the symmetric gait presented in this work, the mean distance the robot covers per iteration is 2.31 cm, and the mean execution time was 2.52 seconds. This works out to approximately 1 centimeter per second ( $\sim 0.04$  body-lengths/s).

Results from our goal finding experiments are presented in Fig. 3.3.1A and an image of the experiment called "Stationary Goal" is shown in Fig. 3.3.2. The robot executes its greedy policy and moves towards the goal with a near monotonic decrease in cost. It is important to note that in all cases the distance to goal never reaches 0 because the robot runs into the goal and its arms - 10 centimeters long - prevent it from getting any closer.

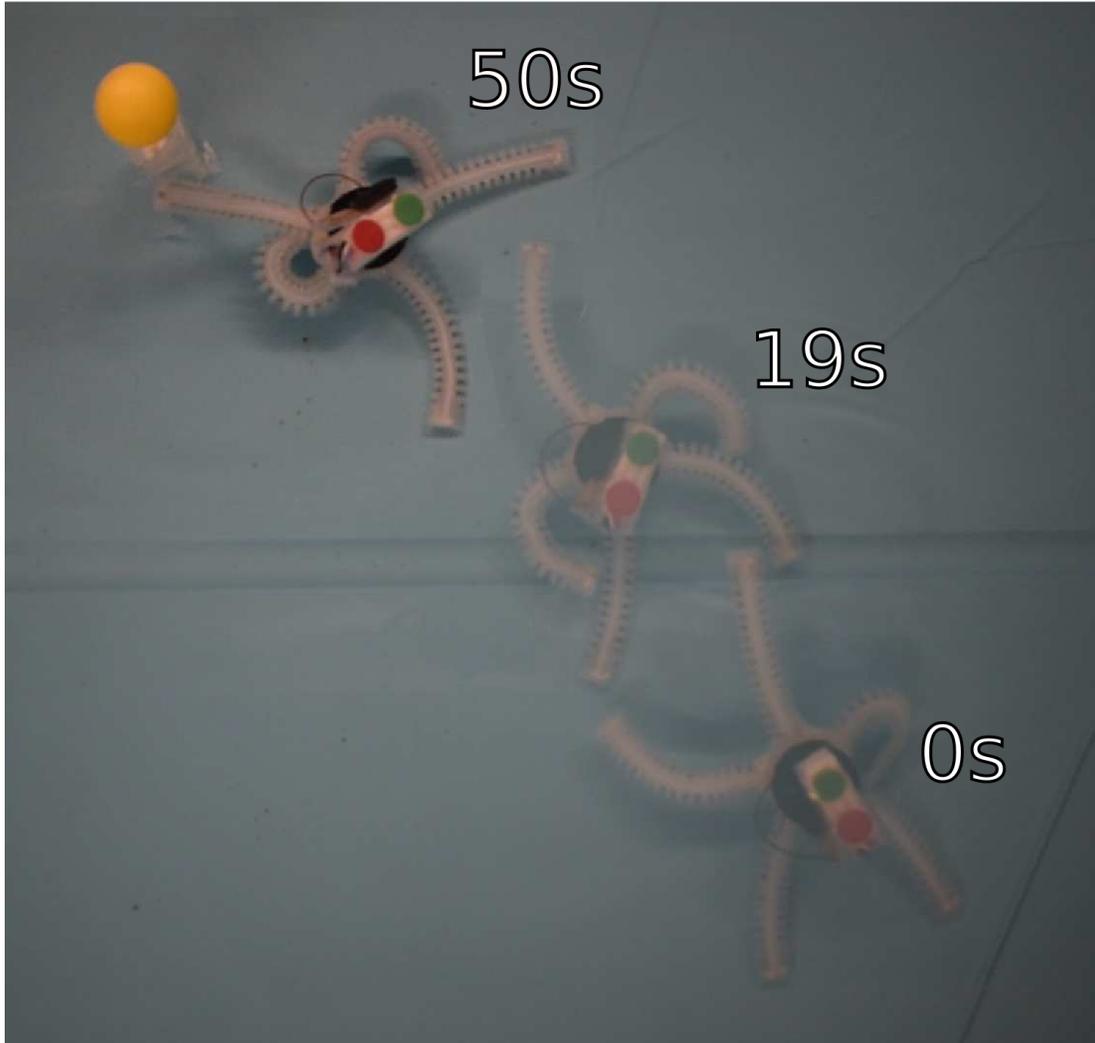


**Figure 3.3.1:** (A) Movement of the robot during three representative hardware tests of the goal-seeking policy, as tracked by the computer vision system. In the “floating goal” tests, the target moves over time, and the closed-loop planner adjusts the robot’s course. The greedy policy gives an almost-monotonic decrease in cost, even in hardware. (B) Characterization of the actual state transitions induced by executing the primitives. In other words, once a leading limb is chosen - corresponding one to one with the set of primitives - this is the magnitude of the net displacement in the direction of that leading limb.

### 3.4 Discussion

While PATRICK was not optimized for speed, its average velocity of 1 cm/s is comparable to the range of speeds of biological brittle stars (roughly 0.5 - 2 cm/s) [68]. It achieves this speed using the limited, unparameterized motion primitives characterized in Fig 3.3.1B. It is notable that the distance varies significantly both within each primitive and between them. The inter-limb variance is due to uncharacterized limb force output and displacement which are sensitive to differences in manufacturing. Additionally, because of the complex nature of the SMA actuators coupled with the rubber beam which comprises each limb, every actuator is itself a nonlinear dynamical system with a large amount of hysteresis. While it is tractable to model such a system with a deterministic dynamical systems model [69] [70], with DER [71][72], or within the broad framework of geometric mechanics [33], the open-loop performance of the system is likely still not sufficient for useful, repeatable behavior.

Given this variability in limb functionality and primitive execution, it is notable that PATRICK is still able to reach the goal. This means that the strategy of closed-loop planning over high level behaviors enables a mobile robot with deformable elements to deal with a great degree of uncer-



**Figure 3.3.2:** Stationary-goal locomotion test (yellow ball is fixed to the terrain), with frames of the robot along its path to the goal and timestamps for each image. The robot moves roughly 40 cm over the course of 50 seconds to reach the goal state, stopping before collision.

tainty, in line with findings of various other soft robotics groups [73–75] and robotics groups in general [76–79]. For PATRICK, as for others, this expressiveness is possible even in the context of the system’s noisy and complicated dynamics due to the deformable structure’s robustness to disturbance and the empirical, high-level approach to control. The ”morphological intelligence” of these structures obviates many of the potential problems caused by variance in the execution of primitives. For rigid systems, small noise in actuation space can cause large, potentially unsafe variations in the behavior of the physical system. The continuum nature of the limbs of the soft system both reduces discontinuities in the task space evaluation of nominal primitives, as well as increases the safety of executing imprecise primitives. This reinforces the idea that we can produce robots that leverage complex dynamics to successfully navigate in an uncertain world without sacrificing the ability to perform useful high level tasks.

### 3.5 Conclusion

In this chapter I presented work that led to development of the brittle star robot, PATRICK. This robot is untethered, has a high-dimensional configuration space, and has many actuators. It also moves at relatively fast, biologically relevant speeds that are rarely achieved by untethered soft robots (thus separating it from its pneumatic cousin developed by Bell et. al. [19]). This combination of features is unique in the mobile soft robot world. More importantly, this combination of features makes PATRICK more than an interesting robot; it is a research *platform*, a tool that can be used to generate and test hypotheses about robot (and animal) manufacturing, locomotion, autonomy, and control. The following chapters focus on various questions that are generated and answered by the PATRICK platform. The next chapter covers manufacturing. The original PATRICK is produced by molding silicone, which is advantageous for mass manufacturing but potentially limiting for research purposes. A more streamlined, 3D printing-based workflow for SMA-driven soft robots can allow more rapid prototyping, deployment, and iteration to facilitate rapid development.

# Chapter 4

## Manufacturing Fieldable, Miniaturized Soft Robots with 3D Printing

*This work appears in:*

Z.J. Patterson\*, D.K. Patel\* (\* co-first author), S. Bergbreiter, L. Yao, C.Majidi, “A Method for 3D Printing and Rapid Prototyping of Fieldable Untethered Soft Robots.” *Soft Robotics*, Accepted.

*Contributions:* I contributed the design, assembly, characterization, experiments, algorithms, and embedded code.

### 4.1 Motivation

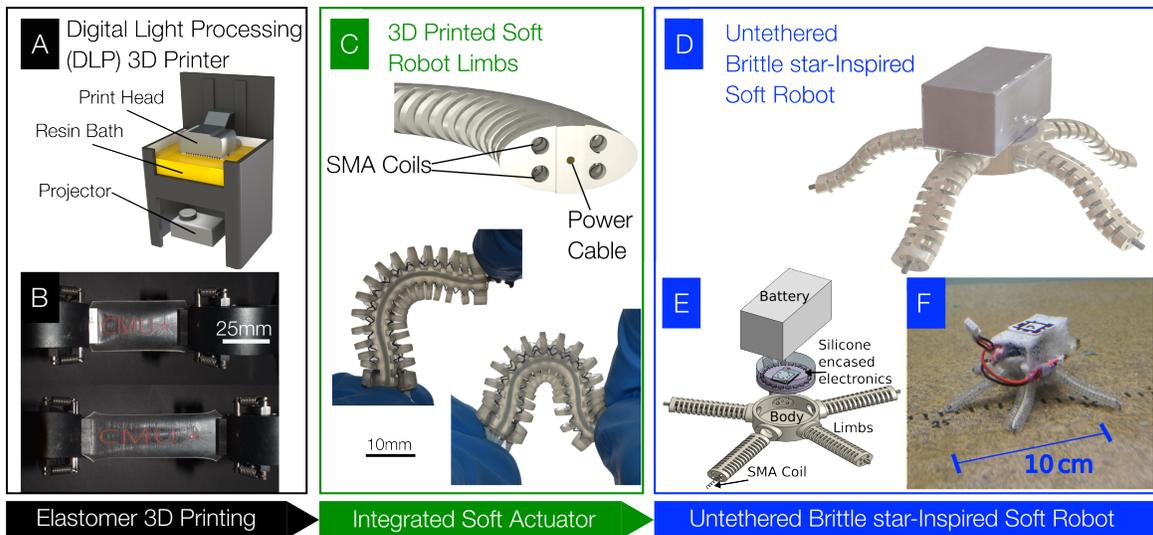
After developing PATRICK, we determined that the manufacturing could possibly be more automated and streamlined to provide a rapid prototyping alternative to injection molding. While molding has the potential to scale well for industrial scale manufacturing and are compatible with popular materials like silicone, they have several distinct disadvantages. First, design versatility

is limited to moldable geometries: stark overhangs, thin walls, and complex internal geometries present significant problem[80]. Practically, the mold design task presents significant design and engineering overhead. Additionally, iteration upon an early-stage design requires producing a new mold for each prototype, a process that can be time consuming and expensive. Together, these issues impair the ability of many practitioners in the field to create soft robots with complex 3D structures and rapidly iterate through design.

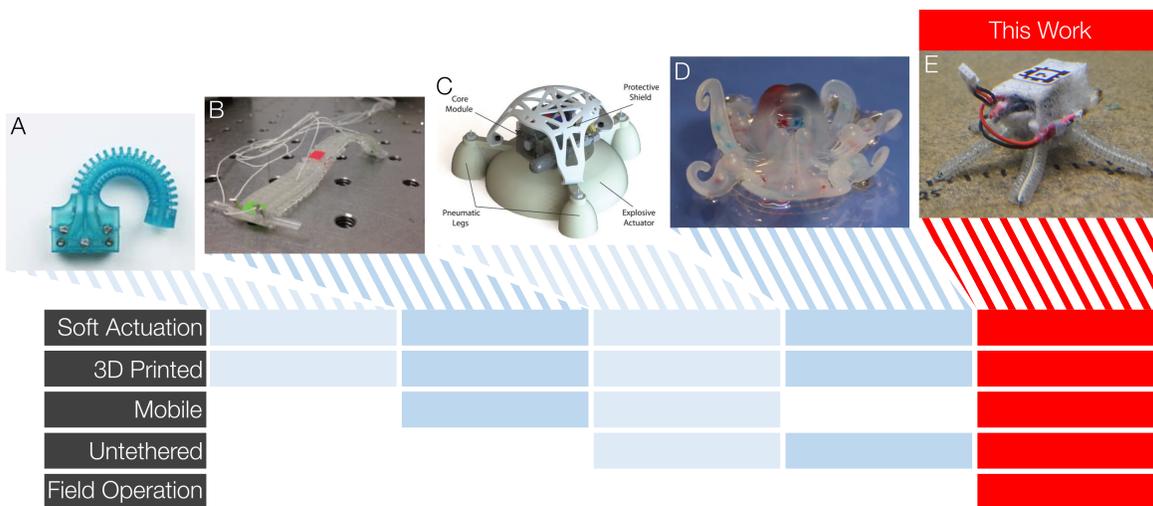
Additive manufacturing (AM) methods like 3D printing have the potential to alleviate the above issues through direct write fabrication processes, which can save time and labor and reduce the number of steps for fabrication [81]. However, despite its tremendous promise and potential, 3D printing has not yet been widely adopted in the fabrication of soft robotic systems [82]. Many factors may play a role, but one important reason lies in the limited mechanical properties of the soft materials that are printable with commercial 3D printing systems. To overcome this, researchers have used direct ink write (DIW) 3D printing technology for printing silicones. However, these need hours to cure and the lateral resolution is determined by the size of printing nozzle [83]. Another disadvantage of this technique is that it requires a support bath for printing overhang structures, which limits the use of these technique for fabricating structures with high complexity [84]. There are other printing platforms that can be used for printing elastomers or soft materials. These include digital light processing (DLP), stereolithography (SLA), polyjet, and binder jetting. Compared to other techniques, DLP-based 3D printing is relatively low cost and high throughput. Printing takes place in a liquid environment comprising of monomers, cross-linker and photo-initiators. It can generate highly complex structures with overhangs at multiple scales with sub-micrometer resolution. In recent years, researchers have made advancement in DLP based 3D printing such as continuous liquid interface production (CLIP) enabling 100 times faster printing [85], projection micro-stereolithography (p- $\mu$ SLA) providing micro to sub-micrometer printing resolution [86], and large-area projection micro-stereolithography creating 3D features spanning from nanometers to centimeters [87].

Here, we contribute a pipeline for rapid prototyping of fieldable untethered soft robots with 3D printed elastomer. This work leverages recent methods developed by Patel et al for printing a stretchable and UV-curable elastomeric resin with tunable mechanical properties and elongation at break from 240-1100 % [88], which is yet to be leveraged to develop autonomous soft robots. Referring to Fig. 4.1.1A-B, we use DLP to print the resin and create elastomeric structures that have silicone-like compliance and elasticity. Leveraging the high printing resolution (100  $\mu\text{m}$  along Z and 27  $\mu\text{m}$  along X-Y), we design and print a dedicated structure to embody and fix shape memory alloy (SMA) coils. The 3D printed structures with embedded SMA function as soft robot actuators that have low stiffness and are highly deformable (Fig. 4.1.1C). With this manufacturing approach, we are able to create untethered soft robots, such as the legged robot shown in Fig. 4.1.1D-F inspired by the brittle star.

This work advances the fields of soft robotics and additive manufacturing by introducing a workflow that uniquely combines DLP-based 3D printing, soft and elastic rubber, untethered soft robot functionality, and bio-inspired mobility. Previous efforts at 3D printed soft robots have largely focused on implementations that are tethered [83], i.e., the robot's power source and/or control electronics are located away from the robot. While this makes sense in many applied contexts – e.g., pipe exploration, hospitals, nuclear facilities – it is disadvantageous for a mobile and autonomous robot meant to traverse unstructured terrain [89]. While there are examples of 3D printed systems that are untethered [90–94], these are not mobile robots that are capable of performing controlled, goal-oriented locomotion tasks [95–98]. Additionally, in this work we demonstrate preliminary results of robot performing in a natural environment. Referring to Fig. 4.1.2, the manufacturing workflow presented in this paper is capable of producing bio-inspired robots that captures all aspects of soft actuation, mobility, untethered functionality, and 3D printed elastic elements.



**Figure 4.1.1:** Fabrication process for 3D printed robot. (A) Rendered depiction of the printer with the critical components labeled. (B) Rubber-like 3D printed elastomer at 0% and 100% extension. (C) Top: Rendered model of a cut section of the actuator to highlight the location of several important features including the SMA, power wire (connecting to the positive terminal of the battery), and notches. Bottom: Photos of deformed actuators to showcase flexibility. (D) Robot CAD Rendering. (E) Explosion of robot features showing the critical components of the robot. (F) Photo of the robot sitting in a tank.



**Figure 4.1.2:** Progression of soft robotics engineering. (A) Single 3D printed soft actuator [66]. (B) Tethered 3D printed soft robot [99]. (C) Untethered robot with rigid and flexible components [94]. (D) Fully untethered and autonomous 3D printed soft robot without mobility [90]. (E) This work builds off previous work by combining untethered mobility and feedback to enable autonomous goal-oriented behavior and performance in unstructured environments outside the lab.

## 4.2 Materials and Methods

### 4.2.1 3D printing and robot fabrication

The soft robot is composed of elastomeric limbs that are embedded with coils of nickel-titanium (NiTi) SMA. When electrical current is supplied to the SMA, the coils contract and cause the limb to bend in a prescribed direction. As shown in Fig. 4.1.1C, each limb actuator contains four SMA coils, which allow for bi-directional motion in and out of the plane. This approach to creating SMA-powered soft robot limbs is based on design principles previously reported in [64, 100].

Referring to Fig. 4.1.1C, the soft robot limbs are 4.5 centimeters long 3D elastomeric structures with ellipsoid cross sections (12x6 centimeters) composed of outer notches and inner channels (in which the SMA coils are inserted). The notches reduce the actuator's mechanical resistance to bending by reducing the amount of material that must be deformed for a given actuator bend angle. Fig. 4.1.1C (top) shows a rendering of the design, highlighting the critical features. The actuators are 3D printed using a DLP-based 3D printer (PicoHD@27, Asiga) using a stretchable and UV curable (SUV) elastomeric resin. The elastomeric resin used for 3D printing consists of epoxy aliphatic acrylate (EAA, Ebecryl 113, Allnex USA) and aliphatic urethane acrylate (AUD, Ebecryl 8413, Allnex, USA) in ratio 1:1 by wt. 2% TPO (Diphenyl(2,4,6-trimethylbenzoyl)phosphine oxide, Genocure TPO, RAHN USA Corp.) of total weight of the polymeric/elastomeric resin was added as the photo-initiator. The printer is a top-down DLP system with a digital mirror device (DMD) and a UV-LED light source operating at 385 nm. The printer was maintained at 42 °C during printing and each layer was irradiated for 1 s and layer thickness was 100  $\mu\text{m}$ . The printed structures were sonicated with isopropyl alcohol (IPA) for 3 min to remove uncured resin followed by a 5 min post curing in a UV oven (UVP CL-1000 UV Oven). The particular material was chosen because of its high similarity to conventional

silicone rubber, especially in its hyperelastic material response. The material is closer to silicone rubber than commercial 3D printed rubbers such as TPU.

After printing the actuator form factor, SMA coils are stretched and cut to size ( 40 mm). They are inserted into their respective holes in the printed part. At the distal end of the actuator, they are crimped together with the central wire. At the proximal end, they are each crimped to an individual wire. A dab of fast cure one-part super glue (Loctite Gel) is placed at each crimp to secure it to the actuator.

The manufacturing method introduced here is used to create a soft robot inspired by the brittle star, a mobile species of sea star that uses its flexible arms to pull itself along the ocean floor. In contrast to the previous implementation, which we produced using conventional elastomer molding methods, the 3D printing approach developed here enabled us to implement a much smaller design (that would be difficult or infeasible with casting techniques) inspired by smaller brittle star species, such as *Ophioderma appressum* [101]. Despite its small size, the robot is fully untethered – i.e. contains all necessary power and control electronics onboard. After all actuators are fully fabricated, five of them are connected to a custom PCB. The PCB contains the microcontroller (BL652 SoC with nRF52832), transistors for control of the SMA, and voltage regulation to step the 7.4 V battery down to 3.3V for the microcontroller. The PCB is then sealed in silicone (Smooth-On DragonSkin 10) to prevent water damage. The actuators and sealed PCB are then inserted into the 3D printed robot body. An off the shelf drone battery (Venom Fly 30C 2S 300mAh) is waterproofed with a liquid rubber sealant (Flex Seal) before being cast in foam (Smooth-On Soma Foama 15) to reduce the effective weight in the water. Finally, An AprilTag fiducial30 is placed at the top of the robot for visual tracking (Fig. 4.1.1F). Overall, the robot is 12.4 cm in diameter, has a mass of 65 grams, and has an overall specific gravity of 1.15. To activate an actuator, the microcontroller sends a signal to the transistor gate, which pulls one end of the SMA to ground while the other is held at 7.4V by the battery. The amount of time that the gate is held high on the resistor, which we call activation time, is proportional to the current

that is provided to the SMA. Because the SMA has a given resistance, the current produces Joule heating, which causes the SMA to undergo a phase change from the compliant detwinned martensite phase to the stiff austenite phase. This phase change is accompanied by a macroscopic contraction. Therefore, the activation time is proportional to the force and the strain produced by the actuator.

## 4.2.2 Robot Experimental Setup

The robot functions using similar gaits to the brittle star. These gaits are based on the pentaradial symmetry of the brittle star morphology in which each limb is equally equipped to be the “front” or the “leading limb” [102]. After the identity of the leading limb is determined, the two limbs adjacent to that limb are moved in a “rowing” motion, pulling the robot or organism along the substrate. The three remaining limbs remain passive. Fig. 4.2.1A shows a diagram of a single gait cycle under this locomotion scheme. The arms swing forward and then down to push the robot off the surface before swinging back to push the robot forward. To accomplish this gait, a set of SMA activation times, the control input in this work, is determined by trial and error (Fig. 4.2.1B and 4.2.1C).

Using Ohm’s law and given the specifications of the battery, the measured resistance of the SMA circuit ( $3\Omega$ ), and the timing characteristics of the gait, power consumption and battery life of the robot can be estimated. Based on this, the power for a single actuation is found to be 18.25W while the average power over a gait cycle is 6.85W. With the used battery, operation time for a single charge at this gait is about 20 minutes.

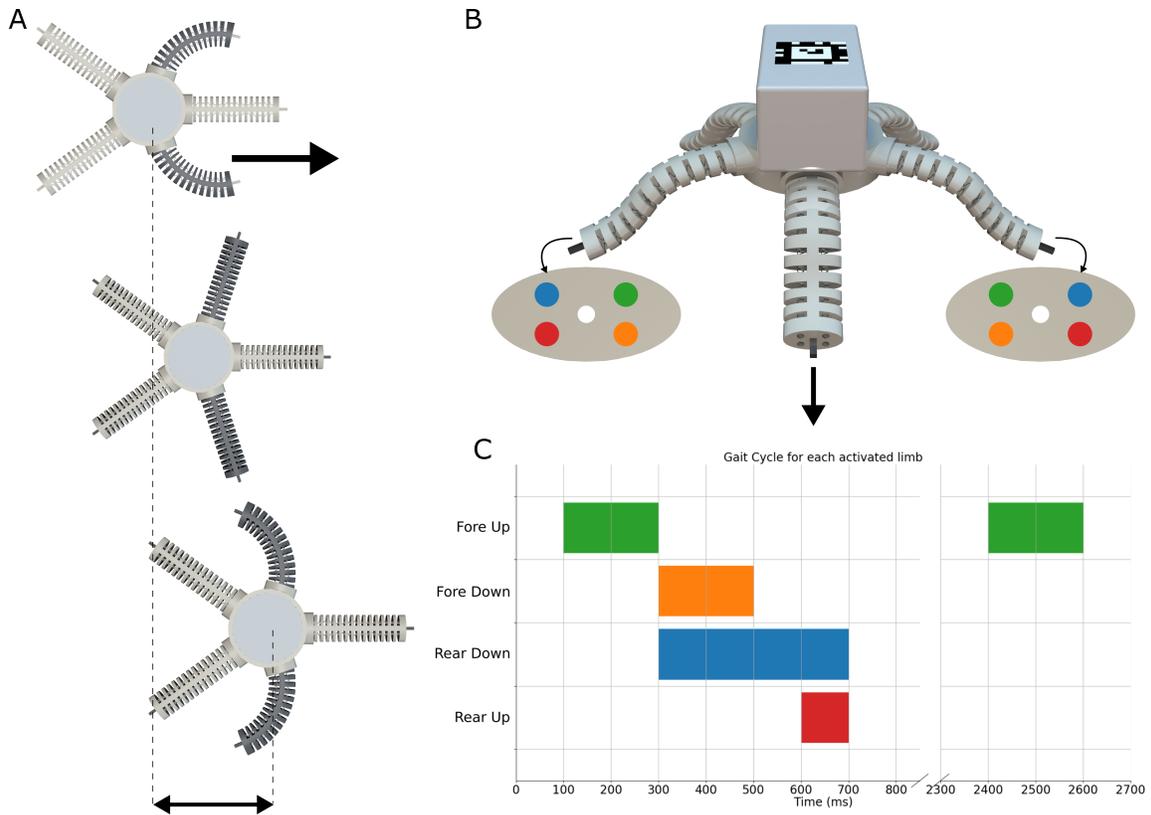
The feedback control algorithm is described as follows. A target position is selected by the operator by clicking on a display of the environment. The planner considers five potential actions, or motion primitives, one corresponding to the direction of each limb. The camera finds the position and orientation of the Apriltag fiducial. A transition model is then used to find the

action expected to minimize the distance to the target. This transition model essentially states that the motion primitive will move the robot 2 centimeters in the direction of the limb corresponding to that primitive, without changing direction. This results in a greedy policy where the planner only considers the effect of the action on the next state.

The robot runs a custom microcontroller program that controls the limbs. It receives instructions for input identity and magnitude via Bluetooth Low Energy (BLE) from a remote microcontroller (also nRF52832). This remote microcontroller receives these instructions via the serial port from custom Python scripts on an Ubuntu computer.

The robot can move in the open loop in any underwater environment. In our test setup, we included an overhead camera (Intel Realsense D435) to enable feedback. The camera is placed approximately 54.3 centimeters above the container that is used for the water tank. A piece of Rock-on-a-Roll from Aquatica Water Gardens covers the bottom of the tank to provide a more frictional surface than the bare plastic. A remote computer running the software stack is nearby and is connected to the remote microcontroller. Of course, Bluetooth transmission is woefully short-range through water. We observe that it rapidly degrades for this robot over a distance of 30 cm in water. Therefore, for extended operation at realistic depths, the gait must be hard coded into the robot. A second camera is placed in front of the container on a tripod to capture high resolution footage.

To conduct the field tests of the robot, the robot was transported to Panther Hollow Lake in Schenley Park in Pittsburgh, USA. The robot was placed in the water and instructed to move using one of its motion primitives. A GoPro Hero 2 Camera attached to a tripod was used to capture the video underwater.



**Figure 4.2.1:** Robot Gait. (A) The robot moves forward with a bioinspired rowing gait during which the arms to either side of the “leading limb” are swung forward and planted in the ground before pulling the body forward. (B) Sequence of inputs to the SMA actuators for each of the active limbs during a single gait cycle. (C) Shows the position of the SMAs for the presented gait cycle.

## 4.3 Results

### 4.3.1 Actuator Characterization

To gauge actuator performance, the elastomer material and the actuators were put through a battery of tests. First, a blocking force test was performed on the fabricated actuator to get an understanding of the actuator force output. The blocking force is relevant because the resulting force is the upper bound of force output that can be commanded during locomotion. The actuators are actuated while up against a force plate to prevent motion. The actuator is clamped to an acrylic fixture which is held in place by the Instron clamp. An acrylic plate is affixed to a universal testing machine (Instron 5969, 10 N load cell). The actuator is then activated for various levels of the input and the data is collected by the Instron DAQ. Matlab was used for post-processing. Here, we use activation time as our control input to keep the results consistent with the control input of the robot, which is presented in the following subsections. We vary the activation time from 50 to 300 milliseconds. Results from the blocking force test for 11 actuators are shown in Fig. 4.3.1A, while a schematic of the test setup is displayed as an inset. It was observed that as the activation time is increased, the blocking force also increases with a roughly linear dependency. With activation times of 200 and 300 milliseconds, the observed blocking force is around  $0.45 \pm 0.09$  N and  $0.5 \pm 0.14$  N, respectively.

Using the same blocking force test, a longer-duration test was performed on a single actuator to determine any degradation of performance after multiple cycles (Fig. 4.3.1B). The actuator was periodically activated with a 70 millisecond pulse every 10 seconds for over 2500 cycles. These activation parameters were set to ensure that the actuator can fully cool before an additional actuation to prevent thermal buildup. It was observed that beyond 2600 cycles, the device was functional and producing similar forces.

Next the forces of the actuators are characterized in a more realistic scenario where the ac-

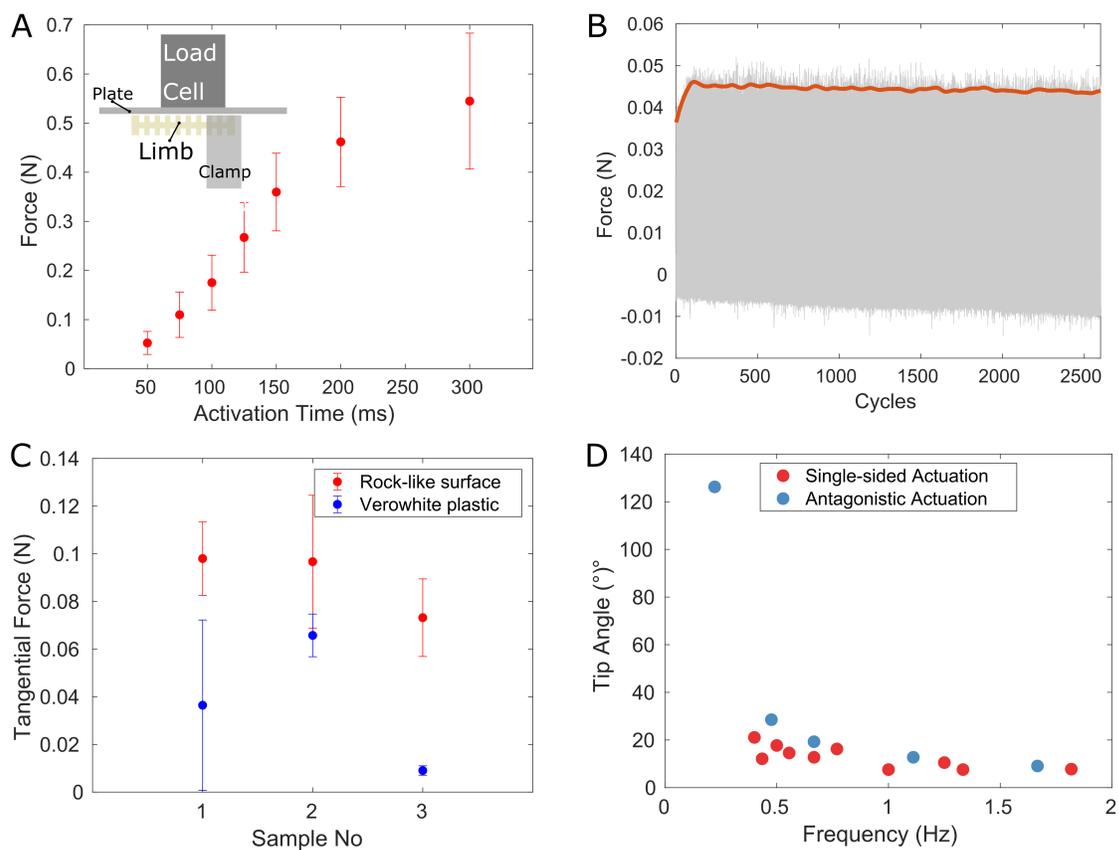
tuator is activated tangentially to the ground. Ideally, the actuator sticks due to frictional forces. To collect force data, an ATI Nano17 6 axis transducer is attached to a custom 3D printed plastic (VeroWhite) plate and bolted to an optical table. The actuator is held parallel to the plate and is lowered until just touching. We then program a similar gait as described for the robot and record forces from the plate. This test is performed on the high friction Rock-on-a-Roll surface and on smooth VeroWhite, with results plotted in Fig. 4.3.1C.

The actuator's angular displacement is then characterized at different frequencies to show actuation-recovery cycles under both antagonistic and non-antagonistic (unforced recovery) actuation scenarios (Fig. 4.3.1D).

### **4.3.2 Robot Function**

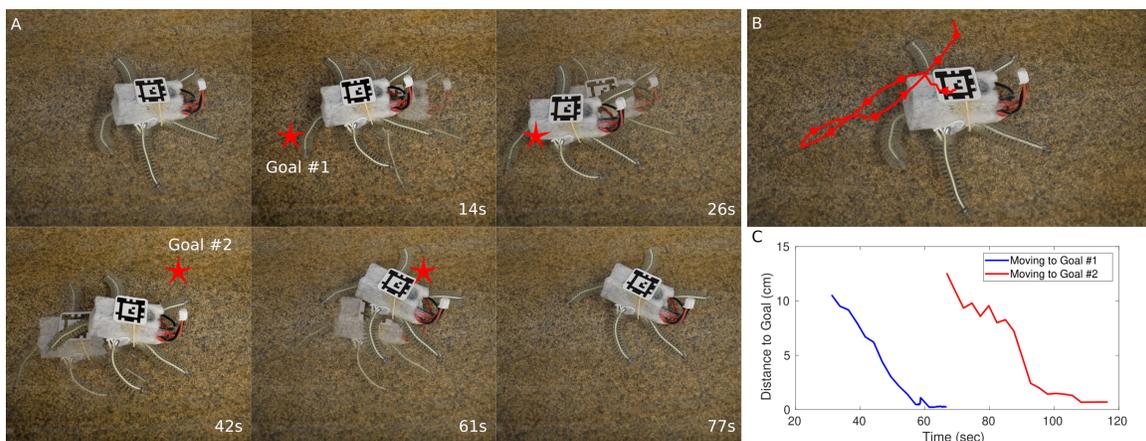
The robot's core functionality is to reach a desired location within an underwater environment. The feedback used to accomplish this task is produced by an overhead camera. Using the gait from the previous section as a motion primitive, the robot functions as follows. First, a goal position is chosen by the operator on a remote computer by clicking on the screen. The limb that is most directly pointing towards the goal is then selected as the leading limb and the gait cycle is executed. After this gait cycle, the camera is again used to determine the leading limb. This process repeats until the robot has reached the goal (within an arbitrary threshold). Fig. 4.3.2 shows a demonstration of this functionality. The robot moves with an average velocity of 0.7204 cm/s, or 0.06 BL/s. This speed is in line with speeds of brittle stars observed in Astley [102].

After validating the performance of a robot in the lab, we built a new one and brought it to a local lake to demonstrate that it can perform in unstructured terrain outside of the lab. We note that the color of the fielded robot (shown in Fig. 4.3.3) is red. This is a cosmetic change to increase visibility that is induced by adding dye to the 3D printable resin. It has no effect on material properties. The robot was placed in the lake and remotely instructed to perform its



**Figure 4.3.1:** Characterization. (A) Blocking force test results for 11 samples with schematic of test inset. Error bars represent standard deviations (B) Long term blocking force test results with a single sample activated for 2500 cycles. The red line represents a moving average. (C) Friction force tests showing the tangential force from single actuators performing the robot gait for a rough and smooth surface. (D) Frequency versus angular displacement experiments. Antagonistic actuation refers to trials in which opposing pairs of SMA coils are actuated whereas single-sided actuation refers to trials in which the opposing pair of coils is not actuated, meaning that all recovery is exclusively due to the passive elasticity of the printed material.

previously specified gait in the open loop (see Fig. 4.3.3A for setup). The robot moved itself over the flora-covered bottom of the lake, leveraging its flexibility and untethered architecture to avoid being tangled or destroyed in the presence of unmodeled interactions (Fig. 4.3.3B). It was closely approached by some local wildlife as well, including the fish shown in Fig. 4.3.3B and several frogs.

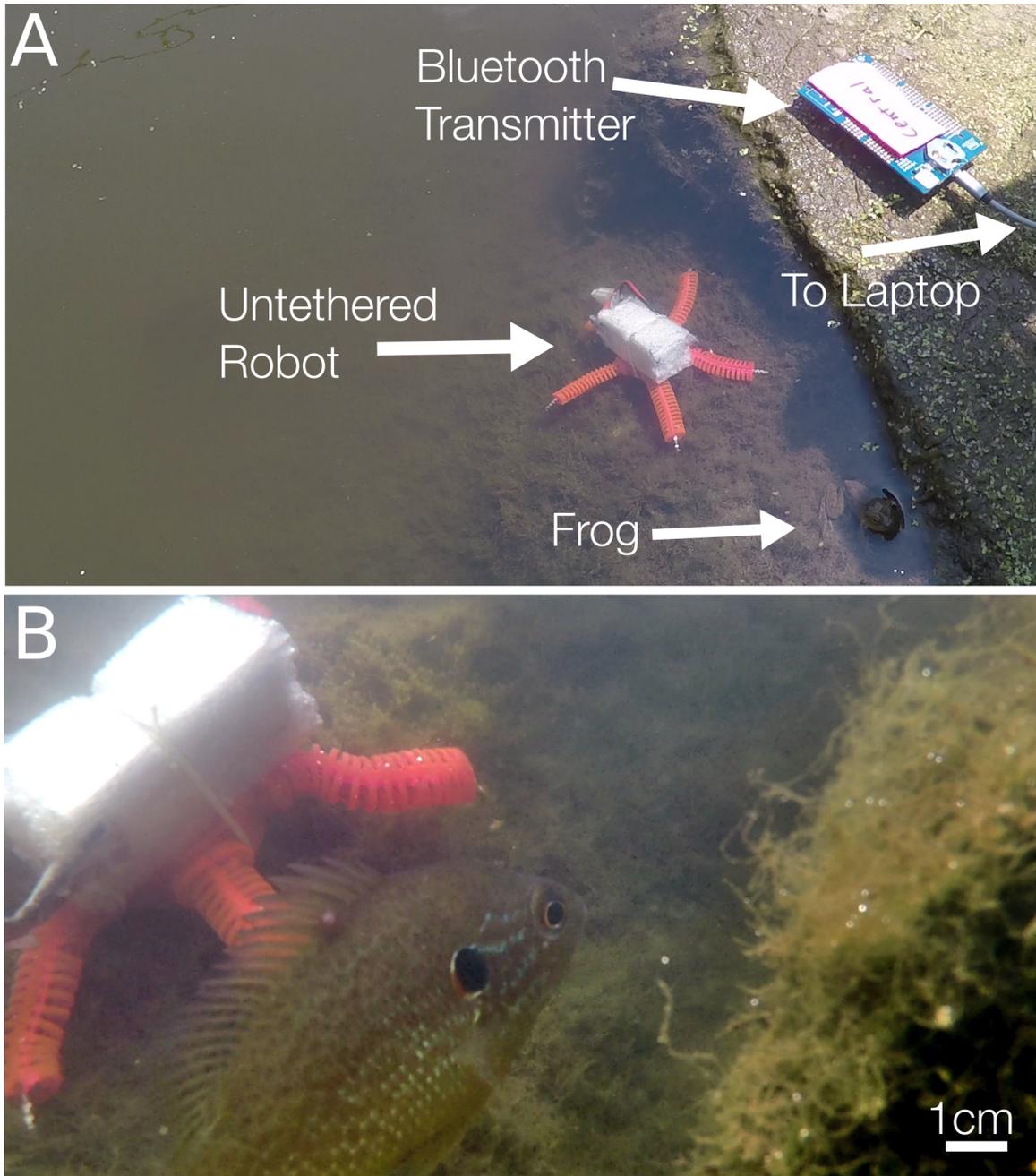


**Figure 4.3.2:** Robot Controlled Demonstration. (A) Images of the robot navigating between desired goals, which are simply positions in 2D space commanded by the teleoperator. In top 3 panels, the robot moves to the position represented by the red star marked Goal 1. After reaching Goal 1, the robot waits for new instructions. The user inputs a new goal position, Goal 2, and the robot navigates to this goal in the bottom panels. (B) Path followed by the robot. (C) Distance to the goal position plotted for the depicted trial.

## 4.4 Discussion

In this work, we used a recently developed 3D printable elastomer to produce an untethered robot that can autonomously navigate to a user specified location. In Fig. 4.1.2 we show a visualization of how this work is contextualized and built upon previous work within the field. The combination of highly flexible elastomeric material, untethered design, and high-level feedback driven motion planning represents a step forward towards 3D printing fully autonomous bioinspired soft robots capable of operation in the real world.

We chose SMA as the actuator because it is relatively compact, has high work density [103], and is easily integrated into a fast moving untethered robot [42]. However, SMA has several lim-



**Figure 4.3.3:** Operation in a natural aquatic environment. (A) Field test setup. (B) Robot as it moves along the bed of the lake.

itations that diminish its effectiveness as an actuator for this application, especially low power efficiency and high shape change variability due to environmental conditions, loading, and manufacturing variance [104]. While we tried to reduce or eliminate these limitations, some are unavoidable. For example, it is likely that small differences in the amount of preload on the actuator during assembly can account for a significant portion of the high variance in force output and angular displacement that we observed across samples, as shown in Fig. 4.3.1. This limitation could be mitigated by incorporating onboard sensing to measure states that are relevant to locomotion, such as the curvature of the limbs or the contact forces with the ground [100]. Incorporating such sensors would enable the robot to operate with far more autonomy and would allow controllers to compensate for differences in performance of the actuator. Another possible improvement is to incorporate thermoelectric materials for more precise control of heating and cooling. This could leverage some of the soft thermoelectric material architectures recently introduced in Zadan et. al [105, 106].

Even with such drawbacks, our robot was able to function in the field in an aquatic environment. It is also notable that the robot was approached by multiple animals during our experiments, including a frog that used the robot as a hiding spot and a fish that approached within a few centimeters. It has previously been noted that soft robots may be valuable as ecological survey tools because of their relatively limited impact on the environment [107–109]. Our robot seemed to not disturb the wildlife at all, possibly because of its low operating sound and smooth, stable motion that avoided the creation of turbulence or turbidity. Further systematic study is necessary to determine what sorts of soft and rigid robots can closely interact with aquatic wildlife, but our result suggests that such a robot may be useful for underwater surveys, particularly in benthic regions with lots of fragile wildlife. Due to limited access to the lake with available filming equipment, the robot’s operation was limited to where we could readily film with a GoPro. Future field tests can use remotely operated tracking and filming technology to explore more remote areas of the lake.

One obvious criticism of the tests is that they are performed in relatively shallow water. During this work, we did not have access to substantially deeper water or a pressurized test setup to test, for example, deep ocean conditions. However, we can see from the existing literature that soft robots perform well at substantial depths [110]. The decrease in performance or outright failure of our robot as we increase depth of operation come from two main factors. First, as the temperature drops, the amount of activation time and therefore current necessary to achieve substantial actuation rises. This is easily remedied by using a design that isolates the SMA from the water. Second, as the pressure increases the electronics may fail. When exactly this will occur is hard to say without FEA or experiment, but Li et al. provide us with estimates [111]. According to their paper, the density of electronics and potting in silicone are the critical factors determining whether the electronics will fail. Here, distances between components of greater than 2.4 mm are recommended. We are close to this threshold. Also, our circuit is potted in silicone for pressure tolerance.

Another area of improvement is the drop in autonomy in the field. This is due to the use of the benchtop camera for feedback of the robot's position and the use of Bluetooth communication to get that information to the robot. Since the Bluetooth signal attenuates in water, it is unreliable in realistic conditions. One approach from the literature is to instead use acoustic transmission, but this is far too large a payload for a robot at our scale and, furthermore, is very power-hungry. There are two possible approaches that come to mind instead. One is to operate in tandem with a traditional submersible vehicle that can drop our robot into the field, monitor its progress from afar, and get in close to communicate when necessary. However, turbidity in the water and occlusions would be issues with this approach. The other option, one that may be more favorable, is to instead add an IMU to the robot and estimate the robot's state onboard as best as possible, eliminating the necessity of wireless communication for closed loop navigation.

In closing, we have shown that we can rapidly fabricate and deploy untethered and autonomous soft robots. This is enabled through the use of DLP-based 3D printing to create rubber-

like actuators with complex miniaturized features not possible with 2.5D elastomer molding. As in [90] and other soft robot implementations that involve additive manufacturing, we note that not all elements of the robot were produced with 3D printing. Only the elastomeric limbs and the main body (to carry electronics) are 3D printed with DLP and the SMA coils, circuit board, battery, and microelectronics are all incorporated later in the manufacturing process. As soft actuator and printed electronics and battery technologies continue to advance, along with advances in soft matter computation and power transmission [112], it could eventually be possible to create untethered, mobile soft robots that are composed entirely of 3D printed materials.

## 4.5 Conclusion

Our experiments with PATRICK demonstrate that the platform is flexible in its design and applications and is theoretically able to play host to a variety of controllers and planners. However, we only scratched the surface of this robot’s potential as a research platform. The fact that we are able to get desired behavior with such little controls overhead or physics modeling is indeed a virtue. However, I opened this thesis with a discussion about how existing soft robots largely are not capable of *doing* anything useful. I put forward as an explanation (or at least a part of one) the fact that most mobile soft robots are not designed to be highly expressive; their structure may be deformable and therefore complex, but without control of those structures, we risk being unable to take full advantage of it. PATRICK and other robots like it (my mind goes to the Laschi group’s octopus robots) address this problem by utilizing highly deformable structures that also have enough controllable degrees of freedom to create a nontrivial set of potential actions. Within this context, therefore, the behavior we demonstrated on PATRICK was still simplistic. It makes no difference how expressive the legs can be in theory because in practice they were used essentially the same way as a simple, bang-bang 1 DOF actuator. To go further, we therefore require a way to operate the legs in a more complex way.

# Chapter 5

## Controlling Brittle Star Robot Limbs

*This work appears in:*

Z.J. Patterson, A.P. Sabelhaus, C.Majidi, “Robust Control of a Multi-Axis Shape Memory Alloy-Driven Soft Manipulator.” *Robotics and Automation Letters (RAL)*, Jan. 2022.

*Contributions:* I contributed all of the technical work including controller design and analysis and hardware fabrication and experiments.

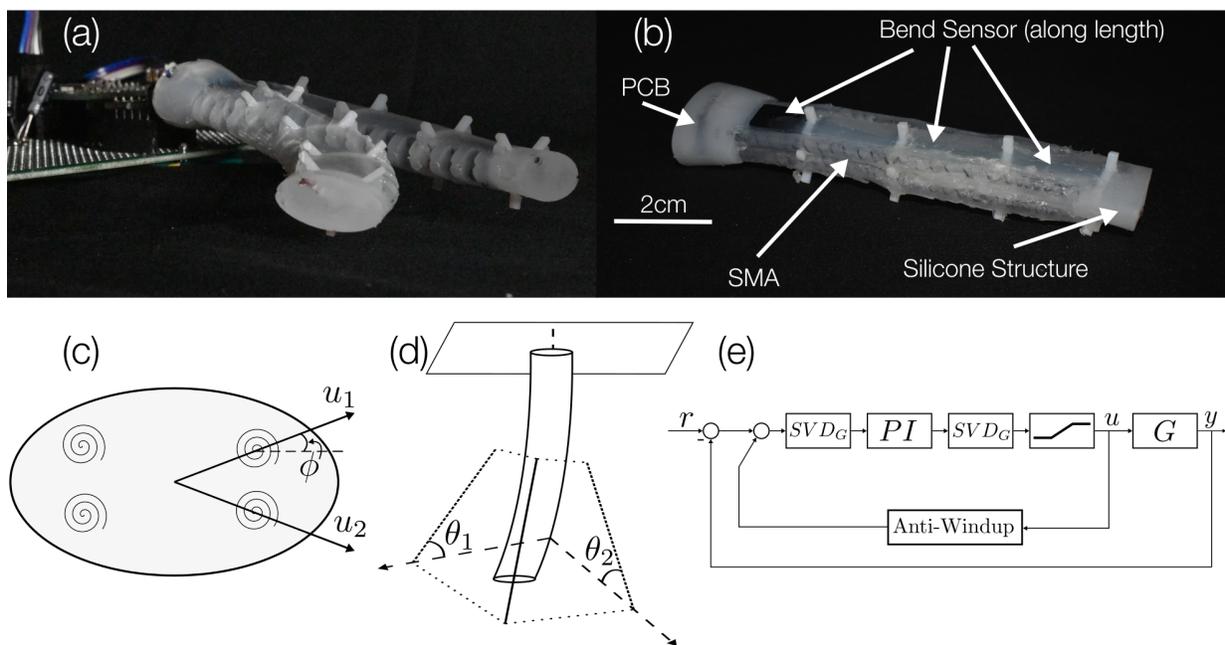
### 5.1 Motivation

Following the work in the previously presented chapters, I needed a way to more precisely control PATRICK limbs. These limbs require something that is robust to unmodeled dynamics, that addresses the problem of SMA input saturation, and that ideally is provably stable. It also needs to work for 3D cases. State-of-the-art soft robot control, with state feedback, is usually limited to planar motions. With pneumatic actuation, model-based state feedback with verified stability has been performed for single-axis bending [46, 113]. Approaches without stability verification have been used in 3D [32], including learned models [34, 114–118] and open-loop

motions [119]. While machine learning approaches often exhibit good performance, model-based state feedback does not require extensive data collection and is more generalizable beyond a given prototype, design, or actuator choice [44]. With smart and thermal actuators, controllers for the planar case have been based on system identification [120]. Finally, some past work has shown three-dimensional feedback for SMA-powered soft manipulators [121], but these do not deal with actuator limits or manipulator mechanics.

In this article, we give the first example of provably stable and robust control of a multi-axis soft manipulator with coupled (linear) dynamics and actuator limits. Our approach to this problem is verified in hardware and demonstrates trajectory tracking with high accuracy in line with the literature. Our model-based controller performs feedback on the tip position of the soft manipulator (Fig. 5.1.1(a)). Two modeling assumptions are made, then a feedback controller accounts for model mismatch. First, the dynamics are approximated with a static beam bending model which gives a linear time-invariant (LTI) transfer function for the body pose. Second, we simplify the input model of our pulse-width-modulated (PWM) control signal to the SMA wires as a force applied directly to the beam tip.

The tools of robust control theory can then be used to account for model mismatch in place of body dynamics, hysteresis, and internal unmodeled actuator states [122], and in this form, can address the two more salient phenomena of actuator saturation [123] and MIMO state feedback [124]. This approach does not require the computational complexity of e.g. model-predictive control in soft and flexible robots [114, 125]. Since constitutive models of beams alongside smart and thermal actuators typically require internal states and dynamics [89, 104], it is reasonable to expect that our approach is generalizable to other soft actuators without the need for application-specific changes.



**Figure 5.1.1:** (a) Our soft, SMA-powered robot manipulator can track an arbitrary reference bend angle. (b) Critical components of the soft robotic manipulator include the SMA coils, bending sensor, and PCB control board embedded in a silicone structure. (c) Our inputs,  $u_i$ , are oriented along the axes of diagonal SMA pairs. (d) In comparison, the output variables  $\theta_i$  are bend angles in the yaw and pitch directions, and are not geometrically aligned with the actuators. (e) Our control structure at a high level includes a singular value decomposition block to account for input vs. output geometry, a core feedback controller (here chosen arbitrarily as PI feedback), and a saturation constraint. Our beam and actuator model is the plant  $G$ .

## 5.2 Manipulator Design and System Model

The soft robot manipulator considered in this study is composed of a silicone elastomer body embedded with two antagonistic pairs of SMA coils and a dielectric elastomer bend sensor (Fig. 5.1.1(b)). The actuator design is based on a soft robot limb design previously introduced by the authors for an untethered brittlestar-inspired robot [64]. The four SMA springs are arranged in a rectangular configuration and the silicone rubber has an elliptical cross section (Fig. 5.1.1(c)). A soft capacitive 2-axis angular displacement sensor (Bendlabs, Inc.) is embedded in the center of the manipulator, measuring the two angles  $\theta_1$ ,  $\theta_2$  for 3D body pose (Fig. 5.1.1(d)). A custom PCB containing a connector and MOSFETs for driving the SMAs is embedded in the proximal end. The approach in this article is generalizable to any actuator configuration with two inputs

and two-axis bending motion.

### 5.2.1 Hardware and Manufacturing

This SMA-powered manipulator is manufactured by assembling the electronics, actuators, and sensors, then overmolding with a silicone body. First, the SMA springs are attached to the circuitboard by inserting their tips through vias in the PCB, crimping these tips to a ferrule on the opposite side, then soldering the ferrule to the via pad. The bend sensor is then soldered to the PCB. This assembly is fed through a series of 3D printed "ossicles" (inspired by the vertebrate-like structures in brittle star limbs [126]) that hold the sensor and SMA coils in place. Silicone food tubing (3mm ID and 4mm OD) is then threaded onto each SMA and sealed with SilPoxy. Finally, the distal ends of the SMA are crimped together with a power wire. The completed assembly is then placed in a 3D printed injection mold and DragonSkin 10 Medium is injected.

The limb's circuitboard is then connected to a larger circuit that contains a Laird BL652 SoC with a Bluetooth enabled nrF52832 microcontroller on board. This microcontroller receives commands via Bluetooth from a nrF52 development kit connected to a computer. The microcontroller drives the gates of MOSFETs connected to each SMA with a pulse width modulated voltage signal (nominal 10.3V).

### 5.2.2 System and Input Models

Our state space consists of two bending angles  $\mathbf{x} = [\theta_1, \theta_2]^\top$ , and our four SMA wires' PWM signals are grouped into two pairs of antagonistic inputs. Since thermal actuators cannot be cooled via Joule heating, this configuration provides two bidirectional inputs  $\mathbf{u} = [u_1, u_2]^\top$  where  $u_i < 0$  is mapped to the corresponding antagonistic coil.

For the soft body mechanics, we use a simplified static model in which the manipulator is a cantilevered Euler-Bernoulli beam subject to a constant moment,  $M = F * d$ , where  $d$  is the

distance from an SMA to the center of the beam and  $F$  is the force applied by that actuator. Our second approximation considers  $F$  to be our commanded PWM signal. The equation for the bend angle along axis  $i$  in response to the combined actuator force in 3D is

$$\theta_i(u_1, u_2) = \frac{F_i(u_1, u_2)Ld}{EI}, \quad (5.1)$$

where  $L$  is the length of the manipulator,  $E$  is the Young's Modulus, and  $I$  is the area moment of inertia. Prior work has justified Young's Modulus as approximating the stress-strain relationship of elastomers under small strain [127].

To complete our model, we note that the applied force in each direction is simply a projection along the angle  $\phi$  between our SMAs' axes and the two principle axis (Fig. 5.1.1(c)), and therefore the two angles are

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{Ld_x \cos(\phi)}{EI_y} & \frac{Ld_x \cos(\phi)}{EI_y} \\ \frac{Ld_y \sin(\phi)}{EI_x} & -\frac{Ld_y \sin(\phi)}{EI_x} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \Rightarrow \mathbf{x} = \mathbf{B}\mathbf{u}, \quad (5.2)$$

where  $d_i$  and  $I_i$  are the moment arms and area moments of inertia for each axis.

Eqn. (5.2) is a stateless, MIMO, linear system, since  $\mathbf{B}$  is a constant matrix. For the control analysis below, we write the transfer function  $G(s)$  as simply  $G(s) = \mathbf{B}$ . Though  $G$  is constant, we retain the argument  $s$  for notation's sake.

Finally, to approximate  $E$  in eqn. (5.2), we average estimates of the Young's Modulus of DragonSkin 10 and the 35 ShA rubber tubing (0.19MPa [128] and 1.4MPa [129] respectively). The cross section of the actuator is approximated as a rectangle with dimensions 16.4×8 mm for the purposes of calculating  $I$ .

## 5.3 Control System

Our feedback controller seeks to track a changing reference tip angle using concepts from robust MIMO control system design. The system has three critical elements (Fig. 5.1.1(e)). First, a Singular Value Decomposition (SVD) pre- and post-compensator shapes the plant to decouple the MIMO system. Second, a diagonal PI controller performs tracking. Third, an anti windup feedback known as a Hanus conditioner addresses actuator saturation.

### 5.3.1 Singular Value Decomposition (SVD) Compensator

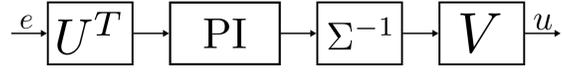
One approach to the control of MIMO dynamical systems is to apply *compensators* to re-shape the plant  $G$  and then perform SISO feedback along each dimension[124]. An SVD is a special class of compensator design in which the plant is shaped by pre- and post-compensators from the singular value decomposition of the plant at a frequency of interest. For the plant at frequency  $\omega$ , we can make the approximation  $G(j\omega) = U\Sigma V^\top$ , as per the definition of the singular value decomposition. A diagonalized version of the plant is then  $U^\top G V$ . Assuming we will design a diagonal controller  $K_s(s)$  for the diagonalized plant, the corresponding controller for the unshaped  $G$  is

$$K(s) = V K_s U^\top \quad (5.3)$$

A good choice of the diagonal controller  $K_s$ , according to [124], is:

$$K_s = \ell(s) \Sigma^{-1}, \quad (5.4)$$

where we choose nominal controller  $\ell(s)$  to be a scalar and so can be tuned using SISO techniques.



**Figure 5.3.1:** Nominal controller for our system including a PI block and SVD pre- and post-compensators.

This design controls the plant along the axes of its singular values instead of along some other, usually naive, choice of axes. Hovd et al. found that this structure is optimal for plants consisting of symmetrically interconnected subsystems [130], as is our plant.

### 5.3.2 PI Controller

The SVD compensator design allows flexibility when specifying the SISO feedback block  $\ell(s)$ . For this work, we choose PI controllers for  $\ell(s)$  as the standard first approach for any SISO system:

$$\ell(s) = K_P \left( 1 + \frac{K_I}{s} \right) \quad (5.5)$$

where  $K_P$  is the proportional gain and  $K_I$  is the integral gain. In principle, one could choose different  $K_P$  and  $K_I$  for each dimension, since eqn. (5.4) could equivalently have a diagonal matrix of multiple different  $\ell(s)$ . However, we chose the same gains for both SISO PI loops because of the symmetry of the problem. Our nominal controller therefore has the form shown in Fig. 5.3.1.

### 5.3.3 Anti-Windup

For many practical soft robot designs, the above control scheme may fail due to actuator saturation. Saturation occurs when the controller commands a value of the system input,  $u_c$ , that is beyond the limits of the real actuator, and the maximum possible value is applied instead. Assuming this maximum to be  $u = 1$  without loss of generality,

$$u_i = \text{sat}(u_{c,i}) = \begin{cases} u_{c,i} & |u_{c,i}| \leq 1 \\ \text{sign}(u_{c,i}) & |u_{c,i}| > 1. \end{cases} \quad (5.6)$$

For example, with our pulse-width modulation input as a duty cycle  $\in (0, 1)$  for one actuator, our input constraints are  $u_i \in [-1, 1]$  for the  $i = 1, 2$  antagonistic pairs.

In response, integral terms in the controller accumulate error, referred to as windup. In the SISO case, *anti*-windup can be performed by an additional scalar feedback term [131]; however, MIMO anti-windup is less clear due to coupling.

There are a variety of MIMO anti-windup approaches [132], but a general solution was developed by Hanus et al [133]. The *Hanus conditioner* starts with a nominal state space controller and augments it with a new term  $H$  that feeds back the difference between commanded vs. applied input:

$$\dot{v} = Av + Be + H(u - u_c) \quad (5.7)$$

$$u_c = Cv + De. \quad (5.8)$$

Substituting (5.8) into (5.7) yields

$$\dot{v} = (A - HC)v + (B - HD)e + Hu \quad (5.9)$$

$$u_c = Cv + De. \quad (5.10)$$

Thus, choosing  $H = BD^{-1}$  eliminates windup [133].

The final challenge for MIMO windup is ill-conditioned feedback control. If the inputs saturate and the direction of control changes from the controller's intent, we risk instability [134]. We employ an approach from the literature where, after performing anti-windup, the applied input  $u$  is adjusted by the largest value of  $u_c$ :

$$u = Nu = \begin{cases} u_c & \|u_c\|_\infty \leq 1 \\ \frac{u_c}{\|u_c\|_\infty} & \|u_c\|_\infty > 1. \end{cases} \quad (5.11)$$

### 5.3.4 Overall Controller Structure and Implementation

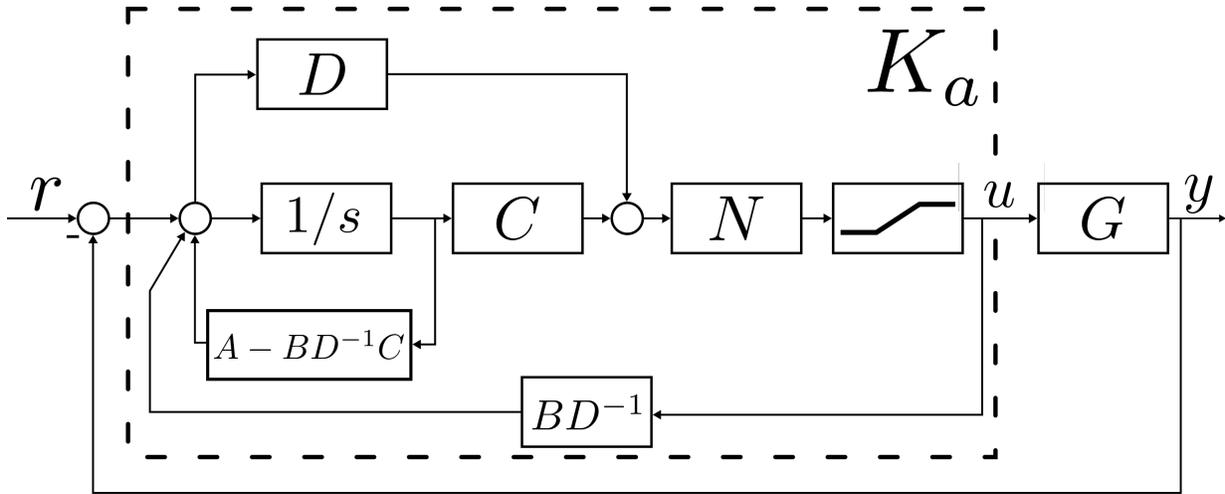
Incorporating each of the above results in the control structure in Fig. 5.3.2. First, for the SVD block, combining the conditioner with PI controller yields the transfer function

$$K(s) = V \begin{pmatrix} K_P(1 + \frac{K_I}{s}) & 0 \\ 0 & K_P(1 + \frac{K_I}{s}) \end{pmatrix} \Sigma^{-1} U. \quad (5.12)$$

We then convert  $K(s)$  into state space and use eqns. (5.9)-(5.10) for the anti-windup conditioner. This forms the full  $K_a(s)$  controller (Fig. 5.3.2).

## 5.4 Controller Stability

Analyzing the stability of the system  $K_a(s)G(s)$  could be performed a number of ways, since it is nonlinear. However, we note from Fig. 5.3.2 that our system has only a single nonlinearity: the saturation function in  $K_a$  representing our actuator constraint of eqn. (5.6). Our setup, with an LTI system cascaded to a saturation nonlinearity, addressed using the Hanus anti-windup



**Figure 5.3.2:** Block Diagram of the full closed loop system.  $A, B, C, D$  refer to the state space matrices of the nominal controller,  $BD^{-1}$  is the anti-windup block,  $N$  is the adjustment to maintain the direction of the commanded input, and the saturation block constrains the inputs delivered to the plant to  $u \in [-1, 1]$ .

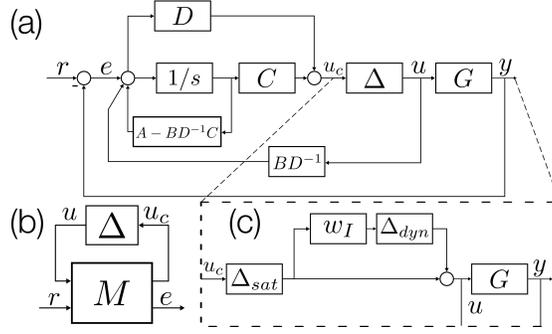
conditioner and the directionality compensation in eqn. (5.11), was studied in [135] using robust control theory. We adapt the stability verification technique from that article.

### 5.4.1 Robust Stability with a Saturation Nonlinearity

The authors of [135] note that the nonlinearity described by (5.6) can be bounded by a cone, allowing the use of the “M- $\Delta$ ” approach of robust control. To do so, we rearrange the block diagram of Fig. (5.3.2) into the feedback interconnection shown in Fig. 5.4.1(a), where the  $\Delta$  block is in the place of our nonlinearity (the saturation function). After the block diagram algebra is performed [135], the matrix  $M(s)$  in Fig. 5.4.1(b) is an LTI transfer function, of which the most important block,  $M_{11}$ , takes the form:

$$M_{11}(s) = [2I + R + KG]^{-1}[R - KP] \quad (5.13)$$

where  $K$  is the controller from eqn. (5.12),  $G$  is our plant, and  $R = KD^{-1} - I$  is the saturation



**Figure 5.4.1:** (a) Modified block diagram, replacing the nonlinearities with a cone-bounded uncertainty,  $\Delta$ . (b) Standard  $M\Delta$  interconnection for stability analysis. The block diagram in (a) is put into this form for our analysis. (c) Inclusion of an additional multiplicative  $\Delta$  block for unmodeled dynamics.

compensation arising from the Hanus conditioner after some algebra.

Then, we seek to *bound* the nonlinearity by some other expression for  $\Delta$  for which stability is verifiable using standard robust control theory. A cone bound can be represented by a choice of  $\Delta \in \mathbf{\Delta}$ , with

$$\mathbf{\Delta} = \{\Delta \mid \Delta = \text{diag}(\Delta_1, \dots, \Delta_n, \Delta_{n+1}, \dots, \Delta_m)\} \quad (5.14)$$

where  $\Delta_1, \dots, \Delta_n$  are cone-bounded nonlinear operators and  $\Delta_{n+1}, \dots, \Delta_m$  are LTI operators with  $\bar{\sigma}(\Delta_i) \leq 1 \forall i = n+1, \dots, m$ . For a nonlinearity of a diagonal saturation operator (one saturation function per input), the article [135] proposes  $\Delta_{1, \dots, n} = \alpha_i I$  where  $\alpha_i$  are any scalars: i.e., the bounds are scaled identity matrices. The following then holds.

**Theorem 1** *From Campo et al. [135]. The system in Fig. 5.4.1(b) is stable  $\forall \Delta \in \mathbf{\Delta}$  if:*

- 1)  $M(s)$  is stable.
  - 2)  $\exists \beta < 1$  s.t.  $\inf_{W \in \mathcal{W}'} \|WM_{11}(s)W^{-1}\|_{\infty} \leq \beta$ ,
- (5.15)

where

$$\mathcal{W}' \equiv \{W | W \in \mathcal{W} \text{ and } W \in \mathcal{C}^{n \times n}\}, \quad (5.16)$$

$$\mathcal{W} \equiv \{W | W \Delta W^{-1} \in \mathbf{\Delta} \quad \forall \Delta \in \mathbf{\Delta}\}. \quad \square \quad (5.17)$$

The authors of [136] show that infimum calculation in condition (2) of the theorem is equivalent to solving the following linear matrix inequality (LMI) for the matrix  $W$ :

$$\begin{bmatrix} A^T Q + QA & QB - C^T W \\ B^T Q - WC & \delta I - 2W - WD - D^T W \end{bmatrix} \leq 0, \quad (5.18)$$

$$Q > 0,$$

$$\delta > 0,$$

where  $A$ ,  $B$ ,  $C$ , and  $D$  are the state space matrices of  $M_{11}(s)$  from eqn. (5.13),  $Q$  is a real symmetric matrix, and  $\delta \in \mathbb{R}$  is a scalar. This LMI can be readily solved using e.g. MATLAB's LMI Toolbox, since it is a convex optimization problem.

#### 5.4.2 Stability Verification: Nominal System

Our system's closed loop properties, and thus existence of  $\beta < 1$ , depends on our PI controller. To verify the stability of our controller, we performed a sweep of controller gains (Alg. 3). The integral gain was set at  $K_I = 1.5$  (chosen experimentally). We found that the maximum which satisfied the theorem was  $K_P^{max} = 2.0$  (at  $\beta = 0.99$ ). Stability was therefore verified for the nominal system.

---

**Algorithm 3:** Maximum Stable Controller Gain

---

```
1  $K_p \leftarrow K_p^{MIN}; \beta = 0.0;$ 
2 while  $\beta < 1.0$  do
3    $M(s) \leftarrow$  eqn. (5.13) and [135], substituting in  $K_p$ ;
4   Confirm  $M(s)$  stable;
5    $W \leftarrow$  soln. eqn. (5.18) (MATLAB LMI Toolbox);
6    $\beta \leftarrow$  eqn. (5.15);  $K_p \leftarrow K_p +$  increment
7 end
8 return  $K_p, \beta$ 
```

---

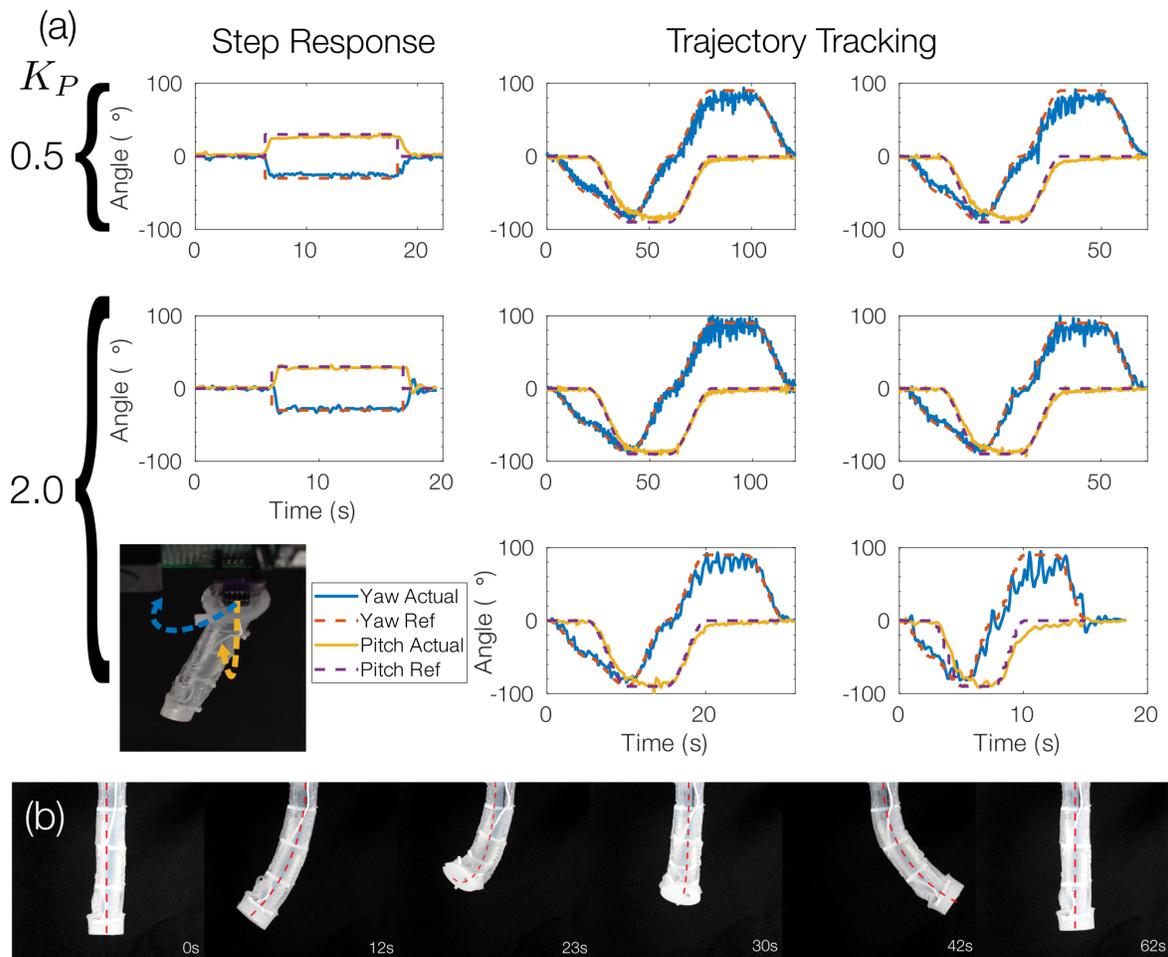
### 5.4.3 Stability Verification: Dynamic Uncertainty

Our nominal plant model only considers the static deflection of our soft robot arm. So, we next extend our stability verification to include plant dynamics, which we hypothesize can take into account many unmodeled hysteretic behaviors. We incorporate unmodeled dynamics as a multiplicative uncertainty as in [124] with a weight of the form

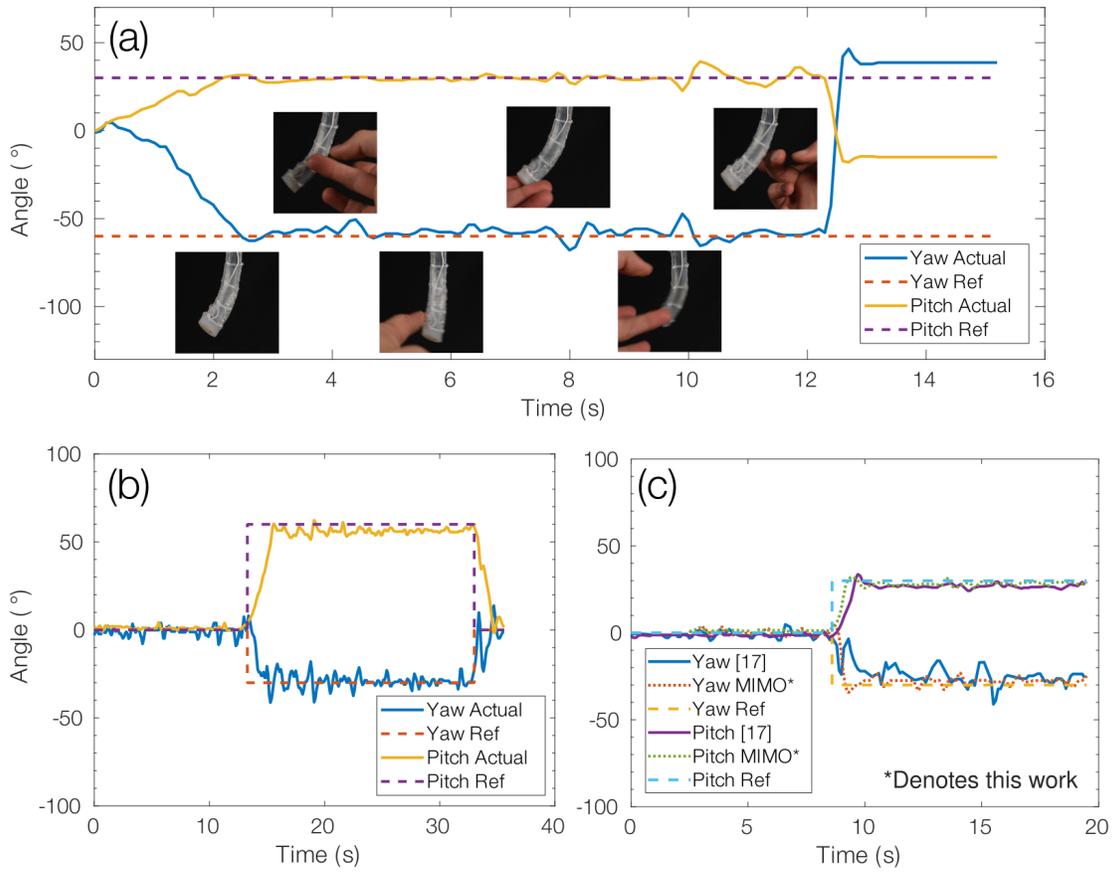
$$w_{dyn}(s) = \frac{\tau s + r_0}{(\tau/r_\infty)s + 1}, \quad (5.19)$$

where  $r_0$  is the relative uncertainty at steady state,  $1/\tau$  is the frequency at which the relative uncertainty reaches 100%, and  $r_\infty$  is the high frequency magnitude of the weight. Here, we choose  $r_0 = 0.1$ ,  $r_\infty = 1.5$ , and  $\tau = 0.1$ .

This  $w_{dyn}$  and an additional uncertainty block  $\Delta_{dyn}$  are added to the system model as in Fig. 5.4.1(c). After re-deriving  $M(s)$ , and now with  $\Delta = \text{diag}(\Delta_{dyn}, \Delta_{sat})$ , we re-run Alg. (3). With these dynamics,  $K_p^{max} = 0.5$ . We have therefore proven stability even with uncertain dynamics (including possible hysteresis).



**Figure 5.4.2:** Control results. (a) Step responses to a reference of  $30^\circ$  and trajectory tracking for gains  $K_P = 2.0$  (rise time of 0.5s) and  $K_P = 0.5$  (rise time of 0.8s). In every case, the shape of the trajectory is identical but the time over which it is executed is varied. (b) Snapshots of the manipulator during a one minute trajectory trial. Dashed red lines indicate the current desired bending shape.



**Figure 5.4.3:** (a) Controller performance when the manipulator is subjected to repeated disturbances. (b) Step response when the manipulator is placed perpendicular to gravity (as in Fig. 5.1.1(a)). (c) Comparison of a step response between the presented work and the PID controller from Yang et al. [121].

## 5.5 Controller Performance on Hardware

In this section, we present demonstrations of the controller from the previous sections on the SMA-actuated soft robot manipulator. We choose the gains discussed in the Stability Proof section, namely,  $K_I = 1.5$  and  $K_P = 2.0$  or  $K_P = 0.5$ .

The test setup is described as follows. The manipulator is attached to a custom perfboard containing the microcontroller and power management electronics. This assembly is attached to a simple 80/20 assembly with the manipulator hanging down. A remote computer sends desired bend angles over bluetooth. The controller itself is implemented in its state space form on the microcontroller. The actuators are powered by a power supply set to 10.3V and maximum current output is limited to 2.5A.

First, we show simple step response plots in Fig. 5.4.2(a) for a step of 30 degrees in both pitch and yaw for each of the two proportional gains. Of particular interest is the steady state error for both cases. For  $K_P = 2.0$ , the average yaw error is  $2.19^\circ$  and average pitch error is  $1.94^\circ$  while for  $K_P = 0.5$ , the average yaw error is  $4.65^\circ$  and average pitch error is  $3.74^\circ$ .

After testing the step responses, we characterized the trajectory tracking for each of the two cases for various speeds. Fig. 5.4.2(a) also shows these results and Fig. 5.4.2(b) shows images of a trial. The shape of the trajectory is the same for each case, but we test over successively shorter durations. Gain  $K_P = 0.5$  is tested with two minute and one minute trajectories, while gain  $K_P = 2.0$  is additionally tested over 30 seconds and 15 seconds. It can be observed that the higher gain achieves better performance performance degrades as we increase the speed of the trajectory (see Table I).

We also performed experiments with the manipulator subjected to unmodeled loads. Fig. 5.4.3(a) shows a trial in which the manipulator is repeatedly poked, flicked, and grabbed. The controller maintains the desired bend angles until the manipulator is disconnected from the power

$K_P$	Duration (s)	Yaw Error ( $^\circ$ )	Pitch Error ( $^\circ$ )
0.5	120	8.44	4.84
0.5	60	9.37	5.52
2.0	120	5.18	2.93
2.0	60	6.43	3.84
2.0	30	7.35	4.69
2.0	15	12.02	6.68

**Table 5.1:** Error during trajectory trials for various gains and speeds.

wiring from the force of a flick. Fig. 5.4.3(b) shows the manipulator step response under a gravitational load. Finally, we compared the proposed controller with the multi-axial controller demonstrated by Yang et al. [121]. Briefly, this controller uses the kinematics of the robot to calculate the desired length of each SMA coil, using standard PID control is used to control those lengths to match the desired bending angle. Fig. 5.4.3(c) shows the result. For the same step response trials as in Fig. 5.4.2(a), we find average errors of  $6.1640^\circ$  in yaw and  $4.9671^\circ$  in pitch.

## 5.6 Discussion

Our results exhibit good performance, consistently tracking signals within about  $5^\circ$  to  $10^\circ$  error at most - even for fast trajectories and large deformations. The controller can also compensate for unmodeled disturbances, as shown in Fig. 5.4.3. In addition, this performance is in line with the state of the art in the literature for both SMA-driven flexible manipulators as well as soft manipulators in general [120, 137–139]. We especially highlight our favorable comparison (shown in Fig. 5.4.3(c)) with the controller from [121], which represents the most similar alternative for multi-axial soft robot control from the literature. The errors for this method were 2-3X larger than for the proposed control, due in part to this work’s explicit treatment of the windup problem and the coupling inherent to multi-axial control.

Performance gradually degrades as trajectory speed increases; in the most extreme case, we observe large errors in the yaw as we attempt to move from  $-90$  to  $+90^\circ$  in 5 seconds ( $36^\circ/s$ ). We

think the primary cause is the greater structural stiffness in the yaw direction which increases the actuator effort necessary for accurate tracking.

3D bending control is achieved with a simple system model, avoiding explicit treatment of the electro-thermo-mechanical dynamics of the SMA (especially hysteresis), hyperelasticity and viscoelasticity of the silicone rubber structure, and large deformations of the beam away from the modeled equilibrium. Our approach demonstrates that simple LTI models with robust control elements to account for unmodeled disturbances can achieve results similar to complicated models with simple control. This is likely enabled in part by the high force output of SMA actuators, allowing us to ignore many of the other forces at play.

While the controller is developed for and demonstrated on an SMA driven system, the approach is applicable to many soft robotic systems as long as bending loads for a given actuator type can be approximately calculated at the tip of the manipulator. For an illustration of such loading calculations, see [140] (pneumatic) and [141] (cable). Additionally, the approach should be easily generalized to multi-segment soft manipulators. Demonstration of this generality is left to future work.

We highlight two particular areas of improvement over these results. First, better analysis may be possible with our current framework. Future work will use a less conservative stability proof that incorporates more knowledge about the structure of actuator saturation [136].

Second, improvements to the model within our framework may address some of the errors observed in tracking. From Fig. 5.4.2, the oscillations are larger when the manipulator is further from its equilibrium. One likely cause is sensor noise observed by the authors, which can be reduced with better filtering. Additionally, our Euler Bernoulli beam model assumes small deflections, so future work will examine if a linearized large deflection beam model (e.g. Timoshenko [142]) will provide better performance for larger regions of the state space at high speeds.

## 5.7 Conclusion

This article contributed a robust control approach to multi-axis soft robotic manipulators. The results demonstrate, for the first time, provably stable and robust control of a multi-axis soft manipulator with coupled dynamics and actuator limits. The controller introduces several feedback elements - including singular value decomposition control and MIMO anti-windup - for use in soft robotics. This approach provides an out-of-the-box option for control of a wide variety of soft manipulators with novel actuators, without the need for machine learning or system identification particular to a specific robot. In doing so, these results provide a pathway for state feedback of more complicated SMA-powered soft robots, including multiple links in a serial chain, with future application to untethered legged soft robots.

# Chapter 6

## Understanding the Influence of Mass Distribution on Brittle Star Locomotion with PATRICK

*A manuscript is in preparation for this work.*

*Contributions:* I contributed all technical work including robot design and experiments, analytic models, and simulation and trajectory optimization results. Dissections were performed in collaboration with Prof. Astley at the University of Akron.

While the previous work focuses on the control of the PATRICK robot limbs, there are also insights to be had regarding the fundamental physics of brittle star-like locomotion and how those physics shape the optimal morphologies and gaits for brittle stars and brittle star-like robots. Because PATRICK crawls along the bottom of the seabed in a similar way to the brittle star, we can use the robot as a research tool to ask and answer questions about locomotion, morphology, evolution, etc, that would be difficult or impossible to test using live animals. Here, I examine

the interplay between brittle star mass and locomotion performance.

## 6.1 Motivation

*Ophiuroidea*, commonly known as the brittle star, is an ancient class of echinoderms, with the earliest fossils preserved from the early Ordovician [143]. Superficially confused with their cousins, the sea stars (*Asteroidea*), brittle stars are quite different, especially in their locomotion. While sea stars slowly move by using their tube feet [144], brittle stars utilize their flexible arms to climb, grab objects, and push themselves more rapidly along the ocean floor, resulting in a type of underwater walking. In rare cases they have even been reported to swim [145]. According to Stohr et. al., brittle stars are an ideal model organism for study of macro-ecology and macro-evolutionary trends due to their diversity and the abundance of both living animals and fossilized skeletal parts [146].

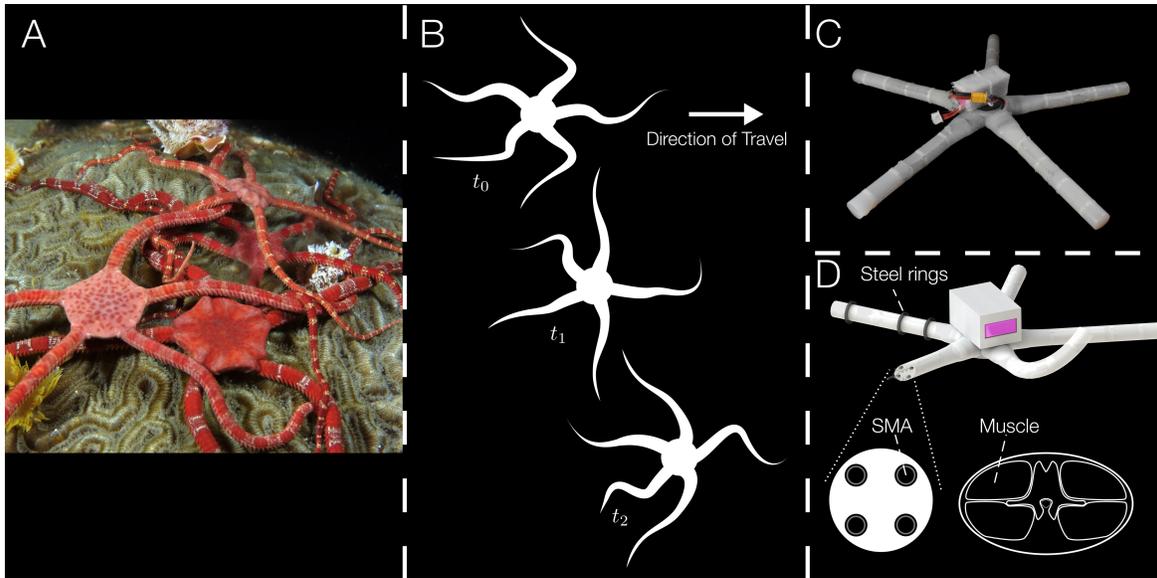
Scientists have been fascinated with legged locomotion for much of human history [147]. In recent decades, with the advent of robotic technology [148] and powerful analytical tools to understand legged systems, such as SLIP [149], fascination has generated capable machines [150] and unearthed principles of animal behavior [151], ecology [152], and evolution [153].

While terrestrial legged locomotion is fairly well tread ground, relatively less attention has been paid to brittle star-like walking or crawling. There are works examining various aquatic organisms such as octopuses [154], hippos [155], newts [156], and crustaceans [157, 158], but no unifying model exists and these existing terrestrial and underwater walking works do not apply to the brittle star.

From the perspective of the roboticist and biomechanician, brittle stars are interesting to study because of their unique morphology and locomotion strategy [159], which, like all animals, are shaped by the interplay of evolution and mechanics. Most brittle stars have five limbs (sometimes referred to as arms or tentacles), and exhibit pentaradial symmetry (Fig. 6.1.1A). When it

comes time to move, however, brittle stars choose a direction of travel and adopt a bilaterally symmetric, coordinated gait called rowing. Briefly, the animal chooses the limb pointing in the desired direction of travel to be the "leading limb." Then, the two adjacent limbs perform a rowing motion, swinging forward, pushing down on the substrate, and then rowing back to push the animal forward. As the brittle star executes this motion, sometimes one of the (momentarily) rear limbs will also "kick" against the substrate for marginal added propulsion. Figure 6.1.1B shows a diagram of the gait. See Astley [102] for a full analysis of the kinematics. While this locomotion strategy has similarities to lobster [158] or octopus [160] walking, there are key differences. The flattened form of the brittle star likely reduces the importance of hydrodynamic drag and the assumption of massless limbs does not hold for the brittle star, as we will show. These differences make the empirical and modeling approaches described in the cited works difficult to apply to this organism.

Instead, realizing that a critical difference is the relatively high mass of the legs, we hypothesized that this fact is actually advantageous for brittle star-style walking. To investigate this hypothesis, several complementary approaches are used. First, measurements are collected from brittle stars to determine mass and density of different body parts. Next a biomimetic robot is built and a set of experiments with various mass/density configurations are performed in order to systematically test effects of mass distribution on performance (Figures 6.1.1C & D show the robot). Finally, two theoretical models are used to investigate the nature of the experimental observations. The first is a simple static model, which illustrates that mass distribution in brittle star morphology significantly effects stick-slip behavior. The second is a more complex dynamic simulation. Leveraging the deep connections between optimal control theory, mechanics, and morphology [161–164], we use iLQR [165] to validate our hypothesis and again to investigate root causes.



**Figure 6.1.1:** A. Several brittle stars. B. Example of a brittle star rowing gait. C. Image of the brittle star robot. D. Rendering of the brittle star robot showing a section view of an arm. To vary the mass distribution, we add mass to the arms by placing steel rings around either distal, medial, or proximal sections; to the central disc by adding steel cylinders under the battery (not pictured); and we add buoyancy by adding foam under the battery (not pictured). The arrangement of the muscles in the continuum manipulator (bottom left) is meant to mimic the arrangement of muscles in the brittle star arm cross section (bottom right).

## 6.2 Modeling and Simulation

To begin, it is not necessarily intuitive why the distribution of mass would change performance of brittle stars or analogous robots during locomotion. We might first predict that a heavier central disc will result in more frequent contact between the disc and the substrate, resulting in additional friction opposing locomotion. As we will demonstrate, this is true. However, if the mass is distributed out to the limbs, then the inactive limbs will drag just as the disc would, eliminating much of the possible benefits. Conversely, we predict that friction plays an additional, more subtle role in aiding the locomotion. Specifically, the arm tips must stick to the substrate during the power stroke in order to efficiently push the rest of the body forward; any slipping results in less effective forward motion. Therefore, stick-slip behavior is critical. Here, we discuss two modeling approaches that examine this stick-slip mechanism of the underwater locomotion problem.

## 6.2.1 Quasistatic Analytic Model

The first model examines this behavior and its relationship to weight distribution on a drastically simplified basis. It is a static model in which the brittle star is represented as a body and four limbs, all point masses. We examine two extreme cases, one in which all weight is concentrated in the body and the other in which all weight is concentrated at the tips of the limbs. In either case, the forces we are ultimately interested in are shown in Fig. 6.2.1A. The actuator in the limb exerts a moment,  $M$ , on the tip of the arm which must be resisted by friction if slipping is to be avoided. This tells us that

$$\mu = \frac{M}{\ell_1 N_a}, \quad (6.1)$$

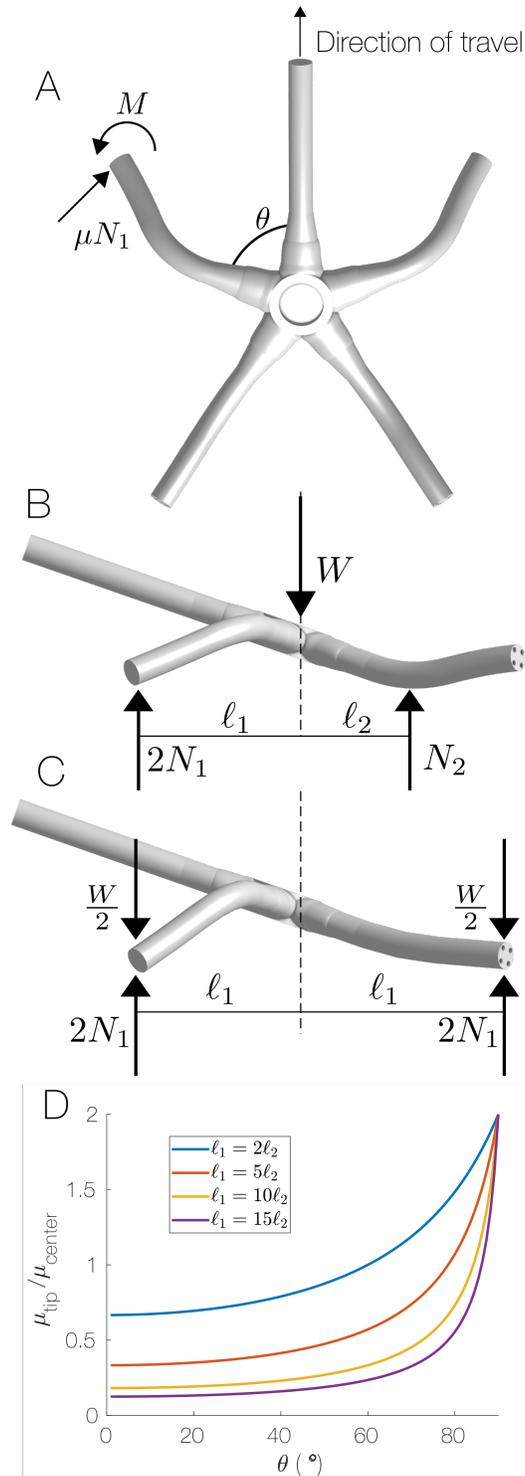
where  $\ell_1$  is the length of the limb and  $N_a$  is the normal force on the arm tip. Next, looking at the FBD for the case where weight is concentrated in the body (Fig. 6.2.1B), we make the assumption that the body rests on the substrate as the power stroke is beginning. Under this assumption, keeping in mind that only two limbs are used for a power stroke, we resolve the statics to find that the normal force  $N_a$  for this case is

$$N_{a,1} = \frac{W \ell_2}{2(\ell_2 + \ell_1 \cos(\theta))}, \quad (6.2)$$

where  $W$  is the weight,  $\ell_2$  is the distance between the center of mass and the point of contact of the body, and  $\theta$  is the angle of the arm with respect to the direction of travel. Note that this equation assumes that  $\ell_1 \gg \ell_2$ , which is the case for brittle stars.

Now looking at the second case, where the weight is concentrated at the tips of the arms (Fig. 6.2.1C), we again resolve the statics to find the normal force

$$N_{a,2} = \frac{W}{4}. \quad (6.3)$$



**Figure 6.2.1:** Static model. A. Free body diagram depicting the torque from the limbs that must be resisted by friction to avoid slipping. B. Free body diagram depicting a case in which all mass is concentrated in the central disc. C. Free body diagram depicting a case in which all mass is concentrated in the tips of the limbs. D. Ratio of the coefficients of friction necessary to sustain the actuator torque for the case depicted in C to that depicted in D plotted against the geometric parameters of the model. A ratio lower than 1 means that case C is less likely to slip.

Finally, plugging these normal forces each into Equation 6.1, we can find a ratio between the two cases of the friction coefficient necessary to support a given load,

$$\frac{\mu_{tip}}{\mu_{center}} = \frac{2\ell_2}{\ell_2 + \ell_1 \cos(\theta)}. \quad (6.4)$$

This ratio is plotted in Fig. 6.2.1D for various arm length ratios and thetas. In most of the operating area, cases with distally concentrated mass require a lower friction coefficient to sustain contact and are therefore less likely to slip.

## 6.2.2 Dynamic Computational Model

To validate that this phenomenon holds for dynamic situations, we used a state of the art physics simulator, Dojo, to simulate robots/brittle stars with various distributions of mass. The chief advantages of Dojo for this work are its combination of physically real contact model and differentiability, allowing rapid simulation and optimization of contact rich problems like the one at hand [166]. We first compare performance across different distributions for given sequences of inputs. Then, to validate findings in more realistic scenarios, we compare performance across distributions of optimal trajectories. The algorithm used for optimal gait design is iLQR, which exploits the DDP structure of the trajectory optimization problem to rapidly find solutions. Fig. 6.2.2 and Fig. 6.2.3A show snapshots of various gaits.

Briefly, the robot model is specified in simulation as follows. First, a branching structure is defined in which there is a central disc and five evenly-spaced limbs. Each of these limbs is composed of two rigid links connected by a spherical joint, and an identical spherical joint connects the proximal link of each limb to the central disc. The joints have both stiffness and damping to more appropriately (if crudely) simulate elastomeric structures or biological soft tissue (as in [31]). The robot is represented in minimal coordinates, with the state vector consisting of the position  $\mathbf{r}_b$ , orientation  $\phi_b$ , linear  $\mathbf{v}_b$ , and angular velocity  $\boldsymbol{\omega}_b$  of the central disc in Cartesian

space and the set of joint angles and velocities in minimal coordinates  $[\theta_i^x, \theta_i^y, \dot{\theta}_i^x, \dot{\theta}_i^y]$  for joint  $i$ . The full state is then a fifty-two element vector:

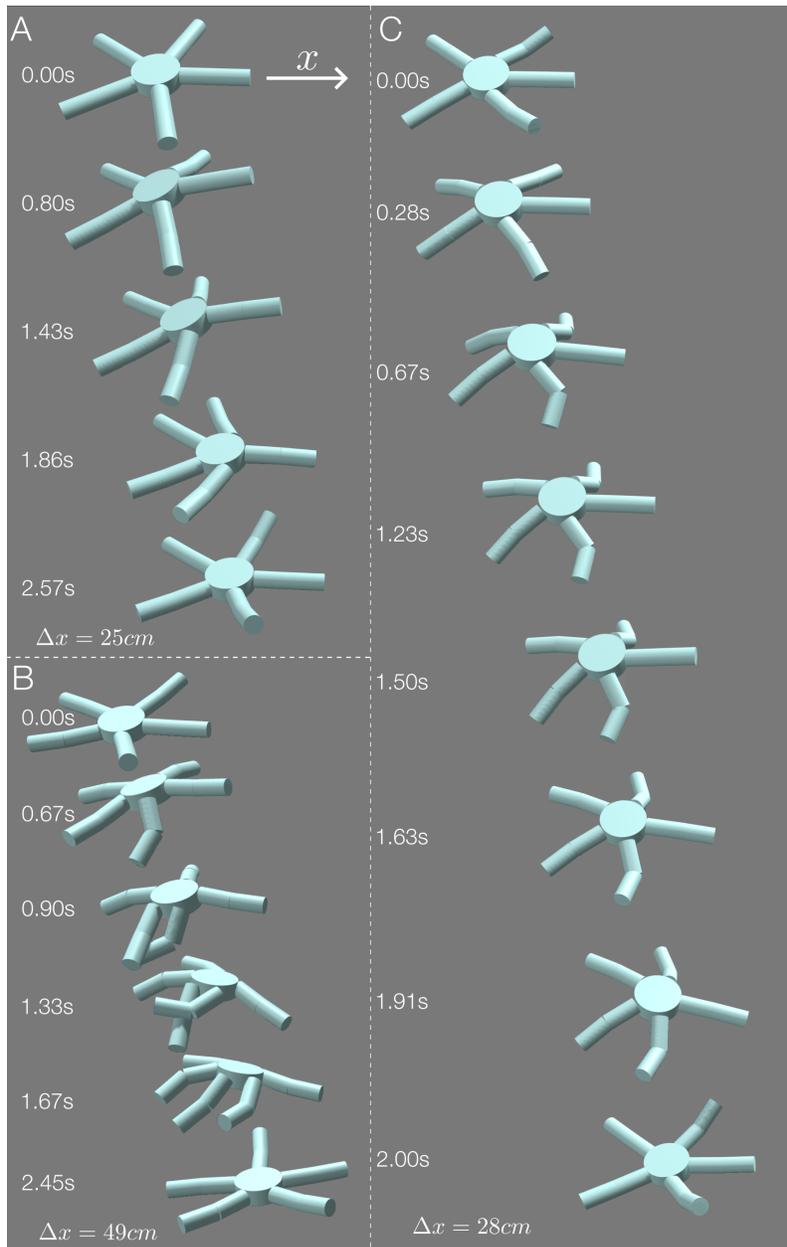
$$\mathbf{x} = [\mathbf{r}_b, \boldsymbol{\phi}_b, \mathbf{v}_b, \boldsymbol{\omega}_b, \theta_1^x, \theta_1^y, \dot{\theta}_1^x, \dot{\theta}_1^y, \theta_2^x, \theta_2^y, \dot{\theta}_2^x, \dot{\theta}_2^y, \dots]^\top. \quad (6.5)$$

One note on the representation; while this is the state that the user interfaces with in this work, under the hood Dojo uses maximal coordinates to represent rigid bodies and quaternions to represent rotations [166]. Finally, the inputs in this model is the vector generalized torques,  $\mathbf{u}$ , acting on the joints.

To compare different mass distributions, we specify four cases in which we simply vary the percentage of mass in the limbs versus in the central disc and perform two analyses on this set. First, a simple, hand tuned gait is created to perform open loop rollouts in simulation. Snapshots from this gait is shown in Fig. 6.2.2A. For each case, we can look at the distance traveled for an idea of relative performance. From Fig. 6.2.3B, it is clear that more mass in the arms relative to the body results in better overall performance.

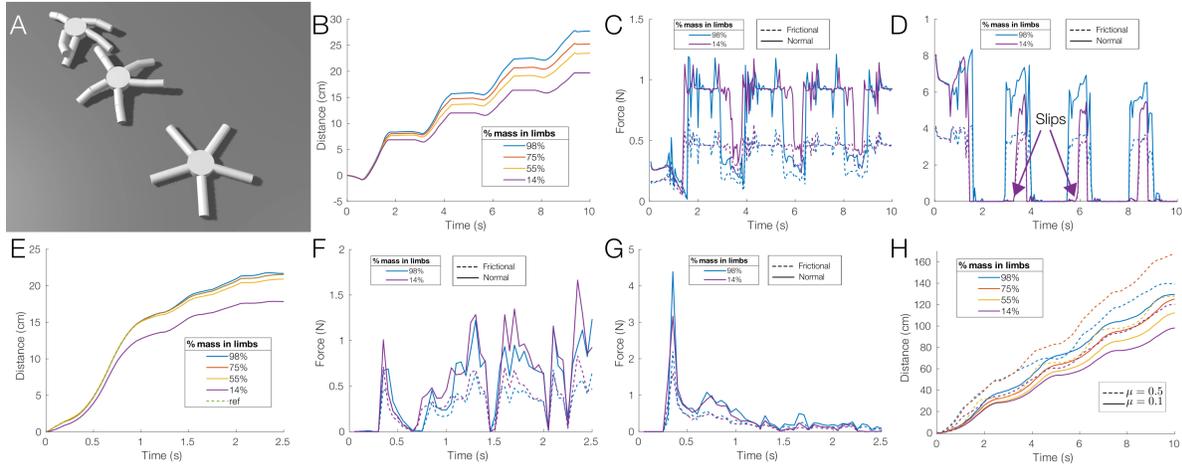
To determine the mechanism that induces this behavior, we can look at the contact forces. Fig. 6.2.3C shows contact forces on the body and inactive limbs from two of the rollouts (98% of the mass in the limbs, referred to from here as distal, and 14% of the mass in the limbs, referred to from here as central). The frictional force from this plot gives a notion of the contact friction resisting locomotion during the rollout. From the plot, we can see that the central case (labeled 14%) has slightly higher friction. Indeed, taking the area under the dashed curves, we find a total impulse of  $3.4608N \cdot s$  for the distal case and  $4.0161N \cdot s$  for the central case, confirming our intuition that there would be slightly more dragging in the central case.

Next, for the same rollouts, Fig. 6.2.3D shows contact forces on the tips of the active arms. These frictional forces give us a notion of the propulsive forces that are being generated by



**Figure 6.2.2:** Snapshots from various simulations. In all cases forward, here called the  $x$  direction, is from left to right. A. A simple, hand-tuned brittle star-like rowing gait performed by a simulated brittle star with 75% of its mass in the limbs. This is quite similar to the gait used later for on-robot experiments. B. A trajectory optimized for a simulated brittle star with 75% of its mass in the limbs. This gait was found by loosely constraining the iLQR algorithm to "go forward", "don't jump", and "end in a similar configuration to the start" all while minimizing the actuation force. The resulting gait has qualitative similarities to brittle star gaits described in the literature [102]. C. A trajectory optimized for a simulated brittle star with 14% of its mass in the limbs. In this case, it is notable that the solution exhibits punting behavior from about 0.67 seconds to 1.63 seconds.

the rowing arms; higher friction here means larger actuation forces can be sustained. Also, if slipping occurs, it will show up here. If we look closely at the beginnings and ends of the contact periods for the central case, it is apparent that slipping is occurring as the limb briefly makes and breaks contact with the substrate. The frictional forces sustained by the central case are also considerably lower here. Again, taking the integral we find total impulses of  $1.5380N \cdot s$  for the distal case and  $0.8867N \cdot s$  for the central case, a difference of almost a factor of two.



**Figure 6.2.3:** Optimal Control Results. A. Simulation of a brittle star-like optimal trajectory. B. Distance traveled during a nominal rollout in simulation for the same inputs. C. Contact forces on the central disc and inactive limbs from the rollouts of extreme cases, where most mass is in the limbs (98%) and most mass is in the central disc (14%). D. Contact forces on the active limbs from the rollouts of the extreme cases. The central disc case has more problems related to slipping. E. Distance traveled over time for optimal trajectories for various mass distributions while attempting to track a given trajectory. F. Contact forces on the central disc and inactive limbs from the optimal trajectories of extreme cases. G. Contact forces on the active limbs from the optimal trajectories of extreme cases. H. Distance traveled for arbitrary optimal gaits that are discovered by the iLQR algorithm. In other words, the reference trajectory is abandoned and the algorithm attempts to find an input sequence that maximizes distance traveled while minimizing the actuator inputs.

We next perform experiments using iLQR to compare performance between the different distributions while performing a more realistic gait. The reference gait and reference input trajectory (also designed with iLQR) are fed as input to the iLQR problem for each configuration. The optimizer is tasked with replicating this reference state trajectory while minimizing the inputs. The cost function at each time therefore takes the following form:

$$\ell_t(x, u) = \frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^{ref})^\top Q_t (\mathbf{x}_t - \mathbf{x}_t^{ref}) + \frac{1}{2}\mathbf{u}_t^\top R_t \mathbf{u}_t, \quad (6.6)$$

where  $\mathbf{x}_t$  and  $\mathbf{u}_t$  are the state and input vectors defined previously,  $\mathbf{x}_t^{ref}$  is the state of the reference trajectory,  $Q_t$  is the (diagonal) state cost matrix, and  $R_t$  is the (diagonal) input cost matrix for time step  $t$ . Similarly, the cost for the final step is

$$\ell_f(x, u) = \frac{1}{2}(\mathbf{x}_f - \mathbf{x}_f^{ref})^\top Q_f (\mathbf{x}_f - \mathbf{x}_f^{ref}), \quad (6.7)$$

where  $Q_f$  is the state cost matrix for the final time step.

For these specific trajectory optimizations,

$$Q_t = Q_f = 50I_{52} \quad (6.8)$$

and

$$R_t = 50I_{20} \quad (6.9)$$

where  $I_n$  is the  $n \times n$  identity matrix.

The forward progress along the trajectory is shown in Fig. 6.2.3E. Performance for the central mass case is much worse than for the other cases, as expected based on previous hypotheses. To figure out if stick-slip behavior is involved, the contact forces can be used as before. Fig. 6.2.3F shows the contact forces on the body and inactive limbs and Fig. 6.2.3G shows the propulsive forces from the active limbs. These plots are harder to interpret, but we can again use integrals to find contact impulses. For the friction on the body resisting motion, the impulse is  $0.6793N \cdot s$  for the distal case and  $0.7846N \cdot s$  for the central case. For the friction supporting the active rowing limbs, the impulse is  $0.6308N \cdot s$  for the distal case and  $0.5281N \cdot s$ . While the differences are more subtle than before because the optimizer does a good job of compensating, these impulses

suggest that the central case cannot use as much force to push itself forward. Note that in these simulations, slipping does not actually occur because the trajectory optimizer avoids it.

Finally, for a third set of simulations, the same reference input trajectory is used as a starting point for the forward rollout of iLQR. However, this time the cost for not following the reference state trajectory is essentially non-existent and instead the cost function is designed to maximize forward movement while minimizing actuator magnitudes. Therefore, while Eq. 6.2.2 still holds, the reference state trajectory is redefined to have a much more sparse structure:

$$\mathbf{x}_t^{ref} = \left[ \frac{1.0t}{50}, 0, z_0, 0, 0, 0, -0.5, 0, 0, 0, \dots \right]^T, \quad (6.10)$$

where  $z_0$  is the height of the robot. Therefore, over the course of the trajectory, the optimizer is free to explore a wider range of behavior. Similarly, at the final step, the reference trajectory is

$$\mathbf{x}_t^{ref} = \left[ \frac{1.0t}{50}, 0, z_0, 0, 0, 0, -0.5, 0, 0, 0, 0, 0, \boldsymbol{\theta}_{1,0}, \dot{\boldsymbol{\theta}}_{1,0}, \boldsymbol{\theta}_{2,0}, \dot{\boldsymbol{\theta}}_{2,0}, \dots \right]^T, \quad (6.11)$$

where  $\boldsymbol{\theta}_{i,0}$  is the (2x1) vector of angles for joint  $i$  at  $t = 0$ . This simply induces the optimization to find trajectories that end in about the same configuration as they started, which is desirable so we avoid motions that are not realistic for a periodic gait.

This results in the optimizer finding the (local) best trajectory for a given mass configuration. Theoretically, if configuration doesn't matter, these should perform roughly the same. On the other hand, the results of these experiments show large qualitative and quantitative differences in optimal trajectories. Snapshots from two such optimized trajectories are shown in Fig. 6.2.2B & C. Fig. 6.2.2B shows a trajectory optimized for a brittle star-like configuration (75% of the mass is in the limbs). The resulting trajectory is qualitatively similar to the brittle star, exhibiting rowing behavior as in [102], along with a commonly observed supplemental kick with a "rear" limb at the apex of the row. This is the fastest trajectory found in our experiments. Conversely,

Fig. 6.2.2C shows a trajectory optimized for the case where only 15% of the mass is in the limbs. It begins with rowing-like behavior, but it notably begins repeatedly bouncing forward on its limbs about 0.67 seconds into the gait. This behavior, usually referred to as punting when it is aquatic, has been well studied in the robotics and animal locomotion literature [167–169]. Calisti et. al make use of an extension of the SLIP model, called Underwater-SLIP (U-SLIP), to describe it rigorously [160]. One assumption of the model is that the mass in the legs is small compared to the body, which is a reasonable assumption for Fig. 6.2.2C but not Fig. 6.2.2B.

Quantitative results from these experiments in Fig. 6.2.3H show that configurations with more mass in the arms tend to perform better than configurations with more mass in the disc, regardless of the environmental coefficient of friction chosen for the simulation, in line with our previous hypotheses and data. An additional notable feature of this data is that at lower friction, the 98% mass in limbs case performs the best, but at higher friction, the case with 75% performs the best. The snapshots in Fig. 6.2.2B correspond to the 75%,  $\mu = 0.5$  case of Fig. 6.2.3H (dashed orange line), while those in Fig. 6.2.2C correspond to the 14%,  $\mu = 0.1$  case of Fig. 6.2.3H (solid purple line).

### **6.3 Brittle Star Weight Distribution**

To discern whether brittle stars have a noticeable pattern of distribution of mass throughout their bodies, five samples of five different species were weighed, both dry and in water. These can be used to calculate the bulk density of each animal. After getting these values for the whole animal, each animal was dissected by cutting off the limbs at the base of the limb where the disc and limb meet. Then, each of the five limbs and the disc were weighed, again both dry and in water, to obtain densities of these separate parts. The data from these brittle stars are shown in Table 6.1. The most striking trend is that the arms are substantially more dense than the discs and tend to contain about 70-80% of the total mass and a comparable value of the weight.

**Table 6.1:** Brittle star mass and density distribution metrics

Sample	Density (g/cm <sup>3</sup> )	Mean arm density	Disc density	% Mass in arms	% Weight in arms
1.	1.30	1.45	1.22	0.72	0.68
2.	1.36	1.51	1.22	0.76	0.71
3.	1.47	1.55	1.25	0.79	0.75
4.	1.58*	1.64	1.34	0.77	0.73
5.	1.50	1.51	1.31	0.79	0.76

\* estimated based on weighing components.

## 6.4 Brittle Star Robot Experiments

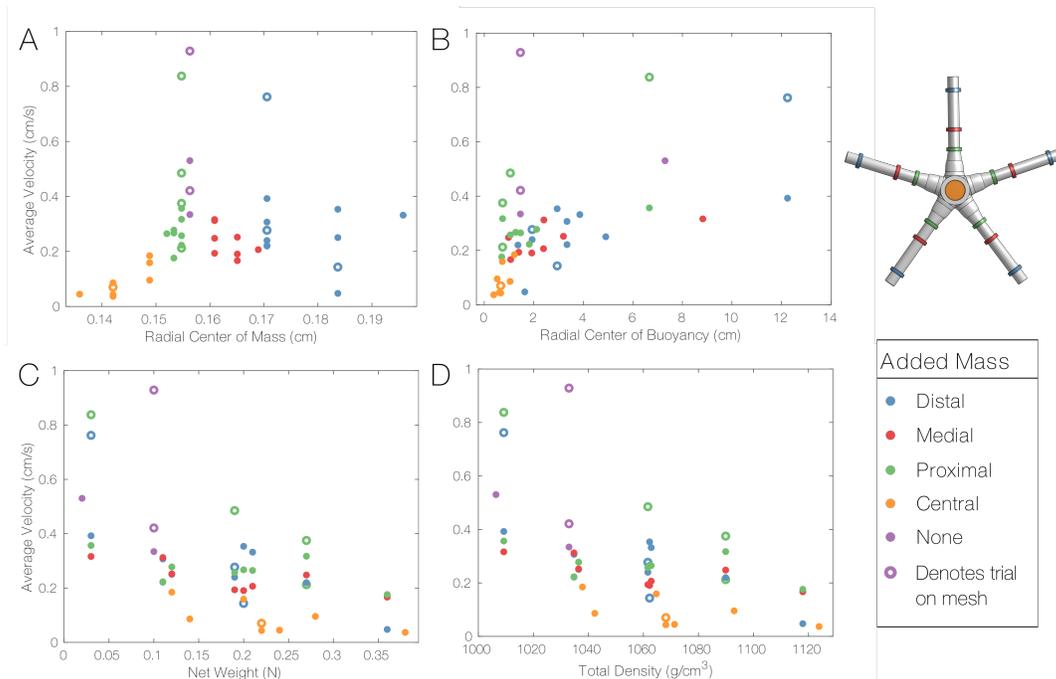
Having verified that brittle stars tend to have much more mass in the arms, we next seek to quantify the effect of mass distribution on real-life locomotion. To test the effect of varying mass on brittle star-like underwater walking, we created a biomimetic brittle star robot based on previous work [64]. While the design of the robot necessarily makes simplifications and trade-offs to accommodate engineering realities, it is designed to be biomimetic in several key senses. Obviously, a five limb design is chosen, with arms radiating out from a central disc at 72° intervals. Since flexibility is a core feature of brittle star morphology, this robot is a flexible robot, meaning that many of its critical structures, most importantly the arms, are made of deformable materials. The robot arms contain flexible shape memory alloy (SMA) artificial muscle coils in quadrants of the cross section that run through the length of the arms. While the brittle star has muscles arranged in similar fashion, it has many sets, each corresponding to a joint between adjacent vertebral ossicles in the arm. We chose to idealize this morphology as a single set of four muscles because of the difficulty of scaling power management, communication, and fabrication for many more actuators.

To test the robot’s performance with different densities and distributions of densities, we needed a simple but unobtrusive way to add and subtract dense and light material from the robot. To accomplish this, we use steel attachments and foam. To add density to the arms, steel rings are placed around it at specified locations. These locations are at the base of the arm, referred to

as a proximal addition, at the middle of the arm (medial), and at the tip of the arm (distal). To add density to the central disc, steel plates are placed into a fixture on top of the disc. Finally, to remove density from the disc, foam is affixed.

To systematically test the effect of varying density distribution, we choose a simple test wherein the robot performs a specified gait for several cycles over a sandpaper-like surface at the bottom of a fish tank. Position data is captured and used to calculate average velocity and a cost of transport estimate. The gait is designed to mimic the rowing gait of the brittle star, although due to the robot's relatively small number of degrees of freedom and to its lack of proprioception, the resulting gait is far less rich. To get an idea of the rowing gait, refer to the diagram in Figure 6.1.1B. Since the executed gait is identical in each case, density distributions with higher average velocity and lower cost of transport are considered superior. After testing on the sandpaper-like surface, the robot was also tested under better contact conditions to minimize slipping. To do this, a plastic grid was placed on the bottom of the tank and a ring with plastic pegs was attached to the underside of the tips of the robot's arms to interface with the mesh.

Data from the whole battery of experiments is presented in Figure 6.4.1. For statistical analysis, total mass, water weight, density, radial center of mass, and radial center of buoyancy are considered as independent variables. The first three of these are highly correlated with each other and all capture a notion of total weight while the last two likewise are highly correlated and capture mass distribution. Thus, one from each set must be chosen for a linear regression. Finally, our dependent variables, average velocity and cost of transport, are correlated as well. Results from a multivariate linear regression with water weight and center of mass versus velocity for locomotion over the sandpaper-like surface are shown in Table 6.2. Independent variables are normalized to compare effect size. Summarizing, we find that the regression supports significant explanatory power (below a  $p$  threshold of 0.01) for both variables. Under the model, a one unit increase in *scaled* water weight is expected to decrease velocity by 0.272 cm/s, whereas a one unit increase in radial center of mass (distributing mass more distally along the limbs) is expected



**Figure 6.4.1:** Results from robot experiments. A. Radial center of mass, B. Radial center of buoyancy, C. Net weight (weight minus the buoyant force), and D. Total density, all plotted against average velocity. The legend refers to where steel weights are added to the robot, if any. Distal, Medial, and Proximal refer to the location along the length of the arms to which steel is added. Central means that steel is added to the central disc. None means that no steel is added. Filled in data-points are from trials that took place on a sandpaper-like surface, while hollow points are from trials on a plastic mesh that increases the robot’s limb’s traction.

to increase velocity by 0.165 cm/s. We also report a similar regression with water weight and radial center of buoyancy as the independent variables in Table 6.3. Effect sizes are similar to those in Table 6.2, but the center of buoyancy just misses the 95% confidence threshold ( $p=0.054$ ) and therefore significance is in question. That said, due to the strong relationship between center of mass and center of buoyancy, this failure to reject the null hypothesis may be due to other factors. Finally, referring again to Figure 6.4.1, we note that the trials on the plastic grid (denoted by hollow circles) do not display any apparent linear or monotonic relationship with radial center of mass or radial center of buoyancy. When mass is added to the center, the robot can barely move even with better contact, but beyond that the previous relationships break down. This supports the hypothesis that stick-slip interactions between the limbs and the substrate play a critical role in governing performance.

**Table 6.2:** Regression Results: Water weight and center of mass versus average velocity

Variable	Coefficient	SE	t Statistic	p Value
Intercept	0.292	0.037	7.975	5.274e-09
Water weight	-0.272	0.049	-5.513	4.938e-06
Center of mass	0.165	0.058	2.833	0.00803

$R^2$ : 0.565

**Table 6.3:** Regression Results: Water weight and center of buoyancy versus average velocity

Variable	Coefficient	SE	t Statistic	p Value
Intercept	0.286	0.048	5.966	1.353e-06
Water weight	-0.181	0.072	-2.525	0.0169
Center of mass	0.171	0.085	2.006	0.054

$R^2$ : 0.515

## 6.5 Discussion

Based on the collection of evidence presented above, it is clear that the mass distribution of brittle stars, in which 70-80% of the animal's mass is in the limbs, plays an important role in locomotion, primarily through the mechanism of slipping behavior. However, the experimental and modeling results bring up an interesting question: is a more distal mass distribution always better? The results presented suggest that more distal at best is marginally better than the distribution seen in the brittle star (70-80% of mass in the limbs). In fact, on the robot the most important boost in performance seems to come from simply displacing mass to any place on the limbs; in Fig. 6.4.1 we observe that the slowest cases overwhelmingly occur when mass is added to the central disc of the robot. Once those cases are removed, the relationship between mass distribution and performance becomes more subtle. In fact, under good contact conditions, more proximal radial centers of mass may even perform better. The trajectory optimization with high friction in Figure 6.2.3H and the trials performed on a better contact surface indicate that under certain conditions (low slip contact), while it's still better to have most of the mass in the limbs, it is not necessarily optimal to have it as distally concentrated as possible. Although we did not calculate the radial centers of mass of the brittle stars, we can clearly see from data shown in

Table 6.1 (and from looking at the volumetric taper of almost every brittle star in existence) that the majority of their limb mass is concentrated in the proximal segment of the arms. While this may be due to biological factors unrelated to the mechanics of locomotion, our data suggest that it is also advantageous for locomotion.

This work also has serious implications for the study of brittle star evolution. The consistent results we find across brittle star data, robotics experiments, and theoretical modeling suggests strong evolutionary pressures to stay within a range of distributions of mass. More studies are necessary to draw strong conclusions on this front, but the preceding results may provide valuable insights into both the initial evolution of brittle stars and may help explain both the consistencies and diversities we see across different brittle star species [146].

There are a number of future directions that could be suggested for this work. While the impetus for this research is that animal studies are difficult due to control precisely due to the quirks of live animal behavior, data from live brittle stars can bolster our arguments. We are also interested in increasing the strength of our arguments related to evolution. To this end, data from fossilized brittle stars would be invaluable to investigate the history of mass distribution in the brittle stars through time and across different environments. While our models are adequate for demonstrating our hypotheses, they could be improved in two key ways. First, drag was neglected. We were comfortable making the assumption that brittle star locomotion is quasi-static and therefore that the contribution of hydrodynamics marginal. Removing this assumption would allow investigation into the interactions between hydrodynamics and contact mechanics. Second, in the introduction we discussed several other models of underwater walking of other organisms such as lobsters [158] and octopuses [160]. It would be worthwhile to compare and unify these models to be able to make broader generalizations about underwater walking. This is especially relevant in light of the results from our trajectory optimizations with relatively light limbs, which converged to a punting gait similar to the one discussed by Calisiti et. al. [154] in their work on the U-SLIP model.

In conclusion, this work presented animal data showing that brittle stars carry most of their mass in their arms. Using brittle star-like robot experiments, the relationship between variations in the distribution of mass and robot performance are characterized, demonstrating that more mass in the arms is much more effective than more mass in the central disc. Finally, two theoretical models are used to investigate the root cause of this relationship. The models suggest that stick-slip behavior is paramount and that more mass in the limbs helps to avoid slipping. This work brings a physics perspective to the study of brittle star locomotion and behavior and hopefully will provide a new tool to paleontologists and evolutionary biologists to explain the history and present of these fascinating creatures.

# Chapter 7

## Conclusions and Future Work

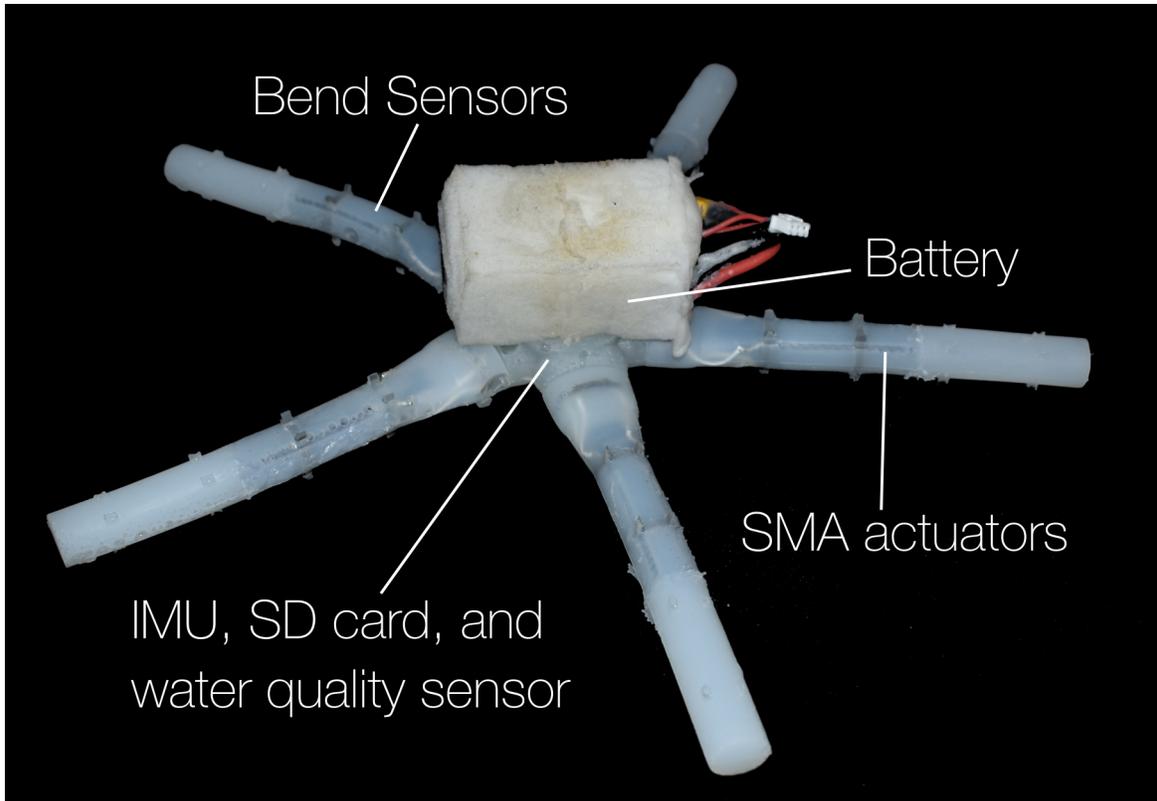
This thesis presented various soft robots that address shortcomings in the soft robotics subfield. Chapter 2 focused on the improvement of the DER physics engine for aquatic soft robot modeling, with a novel motion planning demonstration for path following. Chapter 3 presented PATRICK, a high-dimensional, untethered, brittle star-inspired robot, while Chapter 4 discussed a novel manufacturing workflow for similar robots. Chapter 5 presents the design of a novel control methodology for robust control of soft robot manipulators, with demonstrations of stability and disturbance rejection on PATRICK manipulators. Chapter 6 investigates the physics of underwater walking using the PATRICK robot, looking specifically at the influence of mass distribution.

While at first blush, these different works appear to be at best only mildly related, there are deep connections between them that illuminate the cohesive research program of this thesis and future work that will build upon it. The obvious way to conceptually unify the research of this thesis is that the various topics develop, validate, and apply tools for principled design and control of soft robots (as opposed to the bespoke processes characterizing most recent and historical soft robotics). The more subtle, but more important thread uniting the research of

these works is most apparent in the simulations of the PATRICK robot/brittle star locomotion presented in Chapter 6. These simulations demonstrate the tight interplay between morphology and gaits that optimize locomotion efficiency. The morphology, in the examined case simply the distribution of mass in the limbs, determines the quality of the optimal gait. Thus, we observe that brittle star-like mass distributions (with most of the mass in the limbs) often converge to brittle star like gaits, whereas morphologies with more weight in the central disc converge towards punting gaits, in line with predictions from Calisti et. al. [160] based on limit cycles of their Underwater Spring Loaded Inverted Pendulum (U-SLIP) model. This neatly encapsulates the interrelation of soft robot design and control as well as the consilience of robot and animal locomotion, a connection forged by the hard constraints imposed by classical mechanics and the more subtle influence of evolution. Many soft roboticists are inclined to implicitly place this connection within morphological intelligence, but it seems to be something more. Morphological intelligence as a philosophy aims to understand mechanical structures as either complex adaptive systems or computing systems (really both). The philosophy implicit in Chapter 6 instead seeks to understand (and succeeds in a small, confined, insufficient way) the operation of traditional and optimal control methods on morphologically intelligent structures. Future work will seek to further explore this concept as a research program.

## **7.1 Integrating Sensors and Controllers into PATRICK Platform**

This natural next step is already underway. While the work in Chapter 3 demonstrates that with minimal sensing and control, the robot can still leverage its morphological intelligence to behave in a goal-directed way, additional sensing and control is necessary to execute consistent gaits and to perform other, more complicated tasks. We can achieve this sensing and control by utilizing the findings from Chapter 5, incorporating Bendlabs capacitive bend sensors and the developed robust controllers. Consistent, programmable gaits are now executable in the robot's



**Figure 7.1.1:** The newest version of PATRICK incorporates Bendlabs 2-axis bend sensors in each limb, an IMU for heading information, a total dissolved solids (TDS) sensor for water quality data, an onboard battery for power, and an SD card for saving sensor information.

configuration space. An Invensense ICM20948 IMU is also incorporated so that higher level feedback is available. While the IMU doesn't enable dead-reckoning, compass functionality of a 9-axis IMU allows navigation based on heading. One promising avenue towards utilizing the added sensory and control capability is to test some of the gaits simulated in Chapter 6.

Beyond these changes, an important limitation of the controllers from Chapter 5 is that they directly control the robot in configuration space. Ideally, we would like to move to a more traditional control hierarchy, with low level controllers that operate at the level of the actuator dynamics, taking a desired torque and determining the electrical input necessary under the hood operating in tandem with high level controllers that output desired torques based on task oriented robot configurations. This would allow direct interfacing with conventional robot simulation tools like Dojo.

Finally, we would like to improve DER to allow its use in a simulation-in-the-loop control framework for soft robots. Currently, although the simulation is faster than real-time, it is not fast enough to execute the multiple rollouts necessary for trajectory optimization techniques that are currently on the cusp of being used in the loop for rigid robot control. The Dojo package is close to being able to do this already, but incorporating the same functionality into DER would allow simulation of more flexible structures. Similarly, while we have shown DER to be quite effective for simulation of 2D soft robots, PATRICK and many other soft robots require simulation of the third dimension. In theory, there are no fundamental barriers to doing this but in practice, the implementation used for the works in this thesis has yet to be extended to 3D.

## **7.2 Examining Other Echinoderms and Organisms**

I have demonstrated the utility of soft robots for studying brittle star locomotion in Chapter 6. This approach is readily extended towards the study of other echinoderms and organisms. A particularly interesting area of study is looking at extinct echinoderms. One order of such echin-

oderms is the *Pleurocystitids*, mysterious and ancient flattened echinoderms that have generated debate in the paleontology community about their behavior and mode of life [170]. Recent paleontological [170] and ichnological [171] evidence suggests that these animals had a single, muscularized structure that was used for locomotion, but how the structure was used is unclear. I am working with a team of paleontologists and engineers to adapt the PATRICK robot design to better understand the mode of life of *Pleurocystitids* and I am using the simulation and trajectory optimization approach from Chapter 6 to examine the problem as well. My core findings thus far of this work is that the optimal gait of such a structure is a sculling gait, in which the entire stem is swept from side to side, and that the length of the stem structure improves locomotion efficiency, providing a physical-evolutionary explanation of the fact that derived species of *Pleurocystitids* have longer stems. The ultimate goal is to provide echinoderm biologists and paleontologists with evidence of physical principles guiding evolution, in the same vein as pioneering works of comparative paleontology such as Nyakatura et. al.[172] and the Robophysics work of Dan Goldman and his collaborators [173].

### **7.3 Better Actuators, Better Robots**

Beyond understanding the physical world and locomotion as such, the point of the work presented in this thesis, and indeed the point of all robotics research, is to develop machines that are capable of intervening in the real world as causal actors. Given the fundamental material constraints of human society, of production and consumption, such machines must inevitably be productive sources of labor if they are to fulfill their potential. In the realm of traditional robotics, electric motors represent the cumulative efforts of over 200 years of innovation that has resulted in actuators that are fast, efficient, strong, scaleable, and practical. Thanks in no small part to that powerful technology, motorized robots are finding applications in such critical (and traditional) areas as warehousing [174], construction [175], and agriculture [176], and progress in motion planning of highly dynamic robots [177] promises exciting new frontiers around the corner. That

said, soft robotics has the potential to bring robots from limited, highly controlled industrialized spaces into more messy (and squishy) human spaces [178]. One of the core challenges holding soft robotics back from even the limited scale of the current commercial rigid robotics space is actuation [89]. Currently, no soft robotic actuator combines the attributes that have driven the success of motorized robots [179]. Some, like the Peano-HASEL actuators [180], succeed in matching or exceeding muscle on most metrics, but practical barriers in integration exist that may present headwinds to widespread adoption as the de-facto artificial muscle. Actually existing pneumatic, thermo-electric, dielectric, and electro-magnetic soft actuators all have serious drawbacks in one or more of these critical attributes [181]. It is an excellent demonstration of Tolstoy's Anna Karenina principle [182]: all happy actuators are alike; each unhappy actuator is unhappy in its own way.

In the works presented in this thesis, I used shape memory alloys and showed that they could be deployed on untethered soft robots with unprecedented speed and mobility. In many ways, SMAs are ideal as soft actuators; they are compact and lightweight, strong, and readily used with standard electronics components. Many consider their fundamental drawback to be speed, but the work in this thesis and in prior work shows that they are plenty fast. The core bottleneck is that they are wildly inefficient and do not scale well. The PATRICK robot is mesoscale and capable of moving itself in the water. Manufacturer data and my own experience and materials testing data suggests that it in theory could carry loads on the order of 1kg or less (several Newtons). But to do so for any appreciable length of time, a large, high current capacity battery is required. I have used between 300mAh and 1700mAh batteries, for practical runtimes from five minutes to an hour depending on various parameters. Since SMA is actuated via Joule heating, over 99% of the energy used to power the robot is wasted. In terms of scaling, the larger the structure, the more inefficient and slow the SMA actuation. To come full circle, for soft robots like PATRICK to be broadly economically useful and practical, we require actuators that are similar to SMA (particularly in terms of ease of integration and use) but that operate at motor-like efficiencies -

about two orders of magnitude difference in efficiency - and ideally that scale from insect-size to warehouse-size.

## **7.4 Data Availability**

Raw data and code are available at the following GitHub repository: [https://github.com/zpatty/zach\\_patterson\\_thesis.git](https://github.com/zpatty/zach_patterson_thesis.git).

# Bibliography

- [1] J. Rossiter and H. Hauser, “Soft robotics—the next industrial revolution,” *IEEE Robot. Autom. Mag.*, vol. 23, no. 3, pp. 17–20, 2016.
- [2] C. Majidi, “Soft-matter engineering for soft robotics,” *Advanced Materials Technologies*, vol. 4, p. 1800477, Feb. 2019.
- [3] R. Pfeifer and G. Gómez, “Morphological computation - connecting brain, body, and environment,” in *Creating Brain-like Intelligence: From Basic Principles to Complex Intelligent Systems* (B. Sendhoff, E. Körner, O. Sporns, H. Ritter, and K. Doya, eds.), pp. 66–83, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [4] C. Paul, “Morphological computation: A basis for the analysis of morphology and control requirements,” *Robotics and Autonomous Systems*, vol. 54, no. 8, pp. 619–630, 2006.
- [5] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [6] W.-S. Chu, K.-T. Lee, S.-H. Song, M.-W. Han, J.-Y. Lee, H.-S. Kim, M.-S. Kim, Y.-J. Park, K.-J. Cho, and S.-H. Ahn, “Review of biomimetic underwater robots using smart actuators,” *International journal of precision engineering and manufacturing*, vol. 13, no. 7, pp. 1281–1292, 2012.
- [7] H.-T. Lin, G. G. Leisk, and B. Trimmer, “GoQBot: A caterpillar-inspired soft-bodied rolling robot,” *Bioinspiration & Biomimetics*, vol. 6, no. 2, p. 026007, 2011.
- [8] X. Huang, K. Kumar, M. K. Jawed, A. M. Nasab, Z. Ye, W. Shan, and C. Majidi, “Chasing biomimetic locomotion speeds: Creating untethered soft robots with shape memory alloy actuators,” *Sci. Robotics*, vol. 3, p. eaau7557, Dec. 2018.

- [9] J. Aguilar, T. Zhang, F. Qian, M. Kingsbury, B. McInroe, N. Mazouchova, C. Li, R. Maladen, C. Gong, M. Travers, *et al.*, “A review on locomotion robophysics: The study of movement at the intersection of robotics, soft matter and dynamical systems,” *Reports on Progress in Physics*, vol. 79, no. 11, p. 110001, 2016.
- [10] Y. O. Aydin, J. L. Molnar, D. I. Goldman, and F. L. Hammond, “Design of a soft robophysical earthworm model,” in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 83–87, IEEE, 2018.
- [11] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, “Robust proprioceptive grasping with a soft robot hand,” *Autonomous Robots*, vol. 43, no. 3, pp. 681–696, 2019.
- [12] A. D. Marchese, R. Tedrake, and D. Rus, “Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator,” *The International Journal of Robotics Research*, vol. 35, pp. 1000–1019, Aug. 2015.
- [13] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, “Exploration of underwater life with an acoustically controlled soft robotic fish,” *Sci. Robotics*, vol. 3, p. eaar3449, Mar. 2018.
- [14] M. Calisti, F. Corucci, A. Arienti, and C. Laschi, “Dynamics of underwater legged locomotion: Modeling and experiments on an octopus-inspired robot,” *Bioinspiration & Biomimetics*, vol. 10, no. 4, p. 046012, 2015.
- [15] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario, “An octopus-bioinspired solution to movement and manipulation for soft robots,” *Bioinspiration & Biomimetics*, vol. 6, no. 3, p. 036002, 2011.
- [16] Z. Littlefield, D. Surovik, M. Vespignani, J. Bruce, W. Wang, and K. E. Bekris, “Kinodynamic planning for spherical tensegrity locomotion with effective gait primitives,” *The International Journal of Robotics Research*, May 2019.
- [17] J. Bern, P. Banzet, R. Poranne, and S. Coros, “Trajectory optimization for cable-driven soft robot locomotion,” in *Proc. Robot. Sci. Syst.*, 2019.
- [18] T. Paschal, M. A. Bell, J. Sperry, S. Sieniewicz, R. J. Wood, and J. C. Weaver, “Design, fabrication, and characterization of an untethered amphibious sea urchin-inspired robot,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 3348–3354, Oct. 2019.
- [19] M. A. Bell, J. C. Weaver, and R. J. Wood, “An Ambidextrous STarfish-Inspired Exploration and Reconnaissance Robot (The ASTER-bot),” *Soft Robotics*, Dec. 2021.

- [20] S. I. Rich, R. J. Wood, and C. Majidi, “Untethered soft robotics,” *Nature Electronics*, vol. 1, no. 2, pp. 102–112, 2018.
- [21] X. Huang, K. Kumar, M. K. Jawed, A. M. Nasab, Z. Ye, W. Shan, and C. Majidi, “Chasing biomimetic locomotion speeds: Creating untethered soft robots with shape memory alloy actuators,” *Sci. Robotics*, vol. 3, p. eaau7557, Dec. 2018.
- [22] S. K. Mitchell, X. Wang, E. Acome, T. Martin, K. Ly, N. Kellaris, V. G. Venkata, and C. Keplinger, “An easy-to-implement toolkit to create versatile and high-performance HASEL actuators for untethered soft robots,” *Advancement of Science*, vol. 6, p. 1900178, July 2019.
- [23] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides, “A resilient, untethered soft robot,” *Soft Robotics*, vol. 1, pp. 213–223, Aug. 2014.
- [24] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, Y. Luo, T. Xie, and W. Yang, “Fast-moving soft electronic fish,” *Science advances*, vol. 3, p. e1602045, Apr. 2017.
- [25] H. Jin, E. Dong, M. Xu, C. Liu, G. Alici, and Y. Jie, “Soft and smart modular structures actuated by shape memory alloy (SMA) wires as tentacles of soft robots,” *Smart Materials and Structures*, vol. 25, no. 8, p. 085026, 2016.
- [26] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, Y. Luo, *et al.*, “Fast-moving soft electronic fish,” *Science Advances*, vol. 3, no. 4, p. e1602045, 2017.
- [27] C. Christianson, C. Bayag, G. Li, S. Jadhav, A. Giri, C. Agba, T. Li, and M. T. Tolley, “Jellyfish-inspired soft robot driven by fluid electrode dielectric organic robotic actuators,” *Frontiers in Robotics and AI*, vol. 6, p. 126, 2019.
- [28] R. K. Katzschmann, A. D. Marchese, and D. Rus, “Hydraulic autonomous soft robotic fish for 3D swimming,” in *Experimental Robotics*, pp. 405–420, Springer, 2016.
- [29] A. Arienti, M. Calisti, F. Giorgio-Serchi, and C. Laschi, “PoseiDRONE: Design of a soft-bodied ROV with crawling, swimming and manipulation ability,” in *2013 OCEANS-San Diego*, pp. 1–7, IEEE, 2013.
- [30] C. Armanini, C. Messer, A. T. Mathew, F. Boyer, C. Duriez, and F. Renda, “Soft Robots Modeling: A Literature Unwinding,” *arXiv:2112.03645 [cs]*, Dec. 2021.

- [31] M. A. Graule, C. B. Teeple, T. P. McCarthy, G. R. Kim, R. C. St. Louis, and R. J. Wood, “SoMo: Fast and Accurate Simulations of Continuum Robots in Complex Environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3934–3941, Sept. 2021.
- [32] R. K. Katzschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, “Dynamic Motion Control of Multi-Segment Soft Robots Using Piecewise Constant Curvature Matched with an Augmented Rigid Body Model,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 454–461, Apr. 2019.
- [33] F. Renda, F. Giorgio-Serchi, F. Boyer, C. Laschi, J. Dias, and L. Seneviratne, “A unified multi-soft-body dynamic model for underwater soft robots,” *The International Journal of Robotics Research*, vol. 37, pp. 648–666, May 2018.
- [34] N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, “Elastica: A Compliant Mechanics Environment for Soft Robotic Control,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 3389–3396, Apr. 2021.
- [35] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez, “Software toolkit for modeling, simulation, and control of soft robots,” *Advanced Robotics*, vol. 31, pp. 1208–1224, Nov. 2017.
- [36] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik, “DiffPD: Differentiable Projective Dynamics,” *arXiv:2101.05917 [cs]*, Oct. 2021.
- [37] M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman, “Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics,” *ACM Transactions on Graphics*, vol. 39, Aug. 2020.
- [38] T. Du, J. Hughes, S. Wah, W. Matusik, and D. Rus, “Underwater Soft Robot Modeling and Control With Differentiable Simulation,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 4994–5001, July 2021.
- [39] M. Dubied, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, “Sim-to-Real for Soft Robots Using Differentiable FEM: Recipes for Meshing, Damping, and Actuation,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 5015–5022, Apr. 2022.
- [40] S. Tonkens, J. Lorenzetti, and M. Pavone, “Soft Robot Optimal Control Via Reduced Order Finite Element Models,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12010–12016, May 2021.

- [41] M. Bergou, B. Audoly, E. Vouga, M. Wardetzky, and E. Grinspun, “Discrete viscous threads,” *ACM Transactions on Graphics*, vol. 29, pp. 116:1–116:10, July 2010.
- [42] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, “Dynamic simulation of articulated soft robots,” *Nature Communications*, vol. 11, p. 2233, May 2020.
- [43] N. N. Goldberg, X. Huang, C. Majidi, A. Novelia, O. M. O’Reilly, D. A. Paley, and W. L. Scott, “On Planar Discrete Elastic Rod Models for the Locomotion of Soft Robots,” *Soft Robotics*, vol. 6, pp. 595–610, Oct. 2019.
- [44] C. Della Santina, C. Duriez, and D. Rus, “Model Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges,” Oct. 2021.
- [45] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, “Dynamic control of soft robots interacting with the environment,” in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 46–53, Apr. 2018.
- [46] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, “Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment,” *The International Journal of Robotics Research*, vol. 39, pp. 490–513, Mar. 2020.
- [47] M. K. Jawed, A. Novelia, and O. M. O’Reilly, *A Primer on the Kinematics of Discrete Elastic Rods*. Springer, 2018.
- [48] D. Tong, A. Choi, J. Joo, and M. K. Jawed, “A Fully Implicit Method for Robust Frictional Contact Handling in Elastic Rods,” May 2022.
- [49] X. Huang, K. Kumar, M. K. Jawed, A. Mohammadi Nasab, Z. Ye, W. Shan, and C. Majidi, “Highly dynamic shape memory alloy actuator for fast moving soft robots,” *Advanced Materials Technologies*, vol. 4, no. 4, p. 1800540, 2019.
- [50] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, “Dynamic simulation of articulated soft robots,” *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.
- [51] M. D. Bartlett, N. Kazem, M. J. Powell-Palm, X. Huang, W. Sun, J. A. Malen, and C. Majidi, “High thermal conductivity in soft elastomers with elongated liquid metal inclusions,” *Proceedings of the National Academy of Sciences*, vol. 114, pp. 2143–2148, Feb. 2017.

- [52] X. Huang, K. Kumar, M. K. Jawed, A. M. Nasab, Z. Ye, W. Shan, and C. Majidi, "Highly Dynamic Shape Memory Alloy Actuator for Fast Moving Soft Robots," *Advanced Materials Technologies*, vol. 4, no. 4, p. 1800540, 2019.
- [53] X. Huang, W. Huang, Z. Patterson, Z. Ren, M. K. Jawed, and C. Majidi, "Numerical Simulation of an Untethered Omni-Directional Star-Shaped Swimming Robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11884–11890, May 2021.
- [54] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, May 2011.
- [55] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198, Oct. 2016.
- [56] A. Nagabandi, G. Yang, T. Asmar, R. Pandya, G. Kahn, S. Levine, and R. S. Fearing, "Learning Image-Conditioned Dynamics Models for Control of Underactuated Legged Millirobots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4606–4613, Oct. 2018.
- [57] M. Stolle and C. Atkeson, "Policies based on trajectory libraries," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 3344–3349, May 2006.
- [58] S. Arora, S. Choudhury, D. Althoff, and S. Scherer, "Emergency maneuver library - ensuring safe navigation in partially known environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6431–6438, May 2015.
- [59] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 42–48, Nov. 2007.
- [60] E. Frazzoli, M. Dahleh, and E. Feron, "Robust hybrid control for autonomous vehicle motion planning," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 1, pp. 821–826 vol.1, Dec. 2000.
- [61] M. Watterson and V. Kumar, "Safe receding horizon control for aggressive MAV flight with limited range sensing," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3235–3240, Sept. 2015.

- [62] V. K. Viswanathan, E. Dexheimer, G. Li, G. Loianno, M. Kaess, and S. Scherer, “Efficient Trajectory Library Filtering for Quadrotor Flight in Unknown Environments,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2510–2517, Oct. 2020.
- [63] M. Cianchetti, M. Calisti, L. Margheri, M. Kuba, and C. Laschi, “Bioinspired locomotion and grasping in water: The soft eight-arm OCTOPUS robot,” *Bioinspiration & Biomimetics*, vol. 10, p. 035003, May 2015.
- [64] Z. J. Patterson, A. P. Sabelhaus, K. Chin, T. Hellebrekers, and C. Majidi, “An Untethered Brittle Star-Inspired Soft Robot for Closed-Loop Underwater Locomotion,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8758–8764, Oct. 2020.
- [65] S. Stöhr, T. D. O’Hara, and B. Thuy, “Global diversity of brittle stars (Echinodermata: Ophiuroidea),” *PLoS One*, vol. 7, no. 3, p. e31940, 2012.
- [66] P. Walters and D. McGoran, “Digital Fabrication of “Smart” Structures and Mechanisms - Creative Applications in Art and Design,” *NIP & Digital Fabrication Conference*, vol. 2011, pp. 185–188, Jan. 2011.
- [67] J. M. Jani, M. Leary, A. Subic, and M. A. Gibson, “A review of shape memory alloy research, applications and opportunities,” *Materials & Design (1980-2015)*, vol. 56, pp. 1078–1113, 2014.
- [68] H. C. Astley, “Getting around when you’re round: Quantitative analysis of the locomotion of the blunt-spined brittle star, &em&gt;Ophiocoma echinata&lt;/em&gt;,” *Journal of Experimental Biology*, vol. 215, p. 1923, June 2012.
- [69] M. R. Zakerzadeh, H. Salehi, and H. Sayyaadi, “Modeling of a nonlinear euler-bernoulli flexible beam actuated by two active shape memory alloy actuators,” *Journal of Intelligent Material Systems and Structures*, vol. 22, pp. 1249–1268, July 2011.
- [70] S. G. Shu, D. C. Lagoudas, D. Hughes, and J. T. Wen *Smart Materials and Structures*, vol. 6, no. 3, pp. 265–277, 1997.
- [71] N. N. Goldberg, X. Huang, C. Majidi, A. Novelia, O. M. O’Reilly, D. A. Paley, and W. L. Scott, “On planar discrete elastic rod models for the locomotion of soft robots,” *Soft Robotics*, vol. 6, pp. 595–610, May 2019.
- [72] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, “Dynamic simulation of articulated soft robots,” *Nature Comm.*, 2020.

- [73] M. Cianchetti, M. Calisti, L. Margheri, M. Kuba, and C. Laschi, “Bioinspired locomotion and grasping in water: The soft eight-arm OCTOPUS robot,” *Bioinspiration & Biomimetics*, vol. 10, no. 3, p. 035003, 2015.
- [74] T. Yang, Y. Xiao, Z. Zhang, Y. Liang, G. Li, M. Zhang, S. Li, T.-W. Wong, Y. Wang, T. Li, and Z. Huang, “A soft artificial muscle driven robot with reinforcement learning,” *Scientific Reports*, vol. 8, no. 1, p. 14518, 2018.
- [75] A. D. Marchese, C. D. Onal, and D. Rus, “Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators,” *Soft Robotics*, vol. 1, pp. 75–87, Feb. 2014.
- [76] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Learning, planning, and control for quadruped locomotion over challenging terrain,” *International Journal of Robotics Research*, vol. 30, pp. 236–258, Feb. 2011.
- [77] D. Mellinger and V. Kumar, “Control and planning for vehicles with uncertainty in dynamics,” in *2010 IEEE International Conference on Robotics and Automation*, 2010 IEEE International Conference on Robotics and Automation, pp. 960–965, 2010.
- [78] A. M. Johnson, M. T. Hale, G. C. Haynes, and D. E. Koditschek, “Autonomous legged hill and stairwell ascent,” in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 134–142, 2011.
- [79] V. Vonásek, M. Saska, K. Košnar, and L. Přeučil, “Global motion planning for modular robots with local motion primitives,” in *2013 IEEE International Conference on Robotics and Automation*, 2013 IEEE International Conference on Robotics and Automation, pp. 2465–2470, 2013.
- [80] F. Schmitt, O. Piccin, L. Barbé, and B. Bayle, “Soft Robots Manufacturing: A Review,” *Frontiers in Robotics and AI*, vol. 5, 2018.
- [81] J. Z. Gul, M. Sajid, M. M. Rehman, G. U. Siddiqui, I. Shah, K.-H. Kim, J.-W. Lee, and K. H. Choi, “3D printing for soft robotics – a review,” *Science and Technology of Advanced Materials*, vol. 19, pp. 243–262, Dec. 2018.
- [82] G. Stano and G. Percoco, “Additive manufacturing aimed to soft robots fabrication: A review,” *Extreme Mechanics Letters*, vol. 42, p. 101079, Jan. 2021.
- [83] T. J. Wallin, J. Pikul, and R. F. Shepherd, “3D printing of soft robotic systems,” *Nature Reviews Materials*, vol. 3, pp. 84–100, June 2018.

- [84] T. J. Hinton, A. Hudson, K. Pusch, A. Lee, and A. W. Feinberg, “3D Printing PDMS Elastomer in a Hydrophilic Support Bath via Freeform Reversible Embedding,” *ACS Biomaterials Science & Engineering*, vol. 2, pp. 1781–1786, Oct. 2016.
- [85] J. R. Tumbleston, D. Shirvanyants, N. Ermoshkin, R. Januszewicz, A. R. Johnson, D. Kelly, K. Chen, R. Pinschmidt, J. P. Rolland, A. Ermoshkin, E. T. Samulski, and J. M. DeSimone, “Continuous liquid interface production of 3D objects,” *Science*, vol. 347, pp. 1349–1352, Mar. 2015.
- [86] X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, S. O. Kucheyev, N. X. Fang, and C. M. Spadaccini, “Ultralight, ultrastiff mechanical metamaterials,” *Science*, vol. 344, pp. 1373–1377, June 2014.
- [87] X. Zheng, W. Smith, J. Jackson, B. Moran, H. Cui, D. Chen, J. Ye, N. Fang, N. Rodriguez, T. Weisgraber, and C. M. Spadaccini, “Multiscale metallic metamaterials,” *Nature Materials*, vol. 15, pp. 1100–1106, Oct. 2016.
- [88] D. K. Patel, A. H. Sakhaei, M. Layani, B. Zhang, Q. Ge, and S. Magdassi, “Highly Stretchable and UV Curable Elastomers for Digital Light Processing Based 3D Printing,” *Advanced Materials*, vol. 29, no. 15, p. 1606000, 2017.
- [89] S. I. Rich, R. J. Wood, and C. Majidi, “Untethered soft robotics,” *Nature Electronics*, vol. 1, pp. 102–112, Feb. 2018.
- [90] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, “An integrated design and fabrication strategy for entirely soft, autonomous robots,” *Nature*, vol. 536, pp. 451–455, Aug. 2016.
- [91] C. Richter and H. Lipson, “Untethered Hovering Flapping Flight of a 3D-Printed Mechanical Insect,” *Artificial Life*, vol. 17, pp. 73–86, Apr. 2011.
- [92] P. Phamduy, M. A. Vazquez, C. Kim, V. Mwaffo, A. Rizzo, and M. Porfiri, “Design and characterization of a miniature free-swimming robotic fish based on multi-material 3D printing,” *International Journal of Intelligent Robotics and Applications*, vol. 1, pp. 209–223, June 2017.
- [93] E. B. Joyee, A. Szmelter, D. Eddington, and Y. Pan, “3D Printed Biomimetic Soft Robot with Multimodal Locomotion and Multifunctionality,” *Soft Robotics*, Dec. 2020.

- [94] N. W. Bartlett, M. T. Tolley, J. T. B. Overvelde, J. C. Weaver, B. Mosadegh, K. Bertoldi, G. M. Whitesides, and R. J. Wood, “A 3D-printed, functionally graded soft robot powered by combustion,” *Science*, vol. 349, pp. 161–165, July 2015.
- [95] E. S. Keneth, A. Kamyshny, M. Totaro, L. Beccai, and S. Magdassi, “3D Printing Materials for Soft Robotics,” *Advanced Materials*, vol. 33, no. 19, p. 2003387, 2021.
- [96] D. Drotman, S. Jadhav, M. Karimi, P. de Zonia, and M. T. Tolley, “3D printed soft actuators for a legged robot capable of navigating unstructured terrain,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5532–5538, May 2017.
- [97] T. Umedachi, V. Vikas, and B. A. Trimmer, “Softworms : The design and control of non-pneumatic, 3D-printed, deformable robots,” *Bioinspiration & Biomimetics*, vol. 11, p. 025001, Mar. 2016.
- [98] A. Zatopa, S. Walker, and Y. Menguc, “Fully Soft 3D-Printed Electroactive Fluidic Valve for Soft Hydraulic Robots,” *Soft Robotics*, vol. 5, pp. 258–271, June 2018.
- [99] T. Umedachi and B. A. Trimmer, “Design of a 3D-printed soft robot with posture and steering control,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2874–2879, May 2014.
- [100] Z. J. Patterson, A. P. Sabelhaus, and C. Majidi, “Robust Control of a Multi-Axis Shape Memory Alloy-Driven Soft Manipulator,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 2210–2217, Apr. 2022.
- [101] C. F. LUTKEN, “Bidrag til Kundskab om Slangestjernerne. II. Oversigt over de vestindiske Ophiurer,” *Venskabelige Meddelelser fra Dansk Naturhistorisk Forening i Kjobenhavn*, vol. 7, pp. 1–19, 1856.
- [102] H. C. Astley, “Getting around when you’re round: Quantitative analysis of the locomotion of the blunt-spined brittle star, *Ophiocoma echinata*,” *Journal of Experimental Biology*, vol. 215, pp. 1923–1929, June 2012.
- [103] C. Velez, D. K. Patel, S. Kim, M. Babaei, C. R. Knick, G. L. Smith, and S. Bergbreiter, “Hierarchical Integration of Thin-Film NiTi Actuators Using Additive Manufacturing for Microrobotics,” *Journal of Microelectromechanical Systems*, vol. 29, pp. 867–873, Oct. 2020.
- [104] X. Huang, M. Ford, Z. J. Patterson, M. Zarepoor, C. Pan, and C. Majidi, “Shape memory materials for electrically-powered soft machines,” *Journal of Materials Chemistry B*, vol. 8, no. 21, pp. 4539–4551, 2020.

- [105] M. Zadan, D. K. Patel, A. P. Sabelhaus, J. Liao, A. Wertz, L. Yao, and C. Majidi, “Liquid Crystal Elastomer with Integrated Soft Thermoelectrics for Shape Memory Actuation and Energy Harvesting,” *Advanced Materials*, vol. n/a, no. n/a, p. 2200857.
- [106] M. Zadan, M. H. Malakooti, and C. Majidi, “Soft and Stretchable Thermoelectric Generators Enabled by Liquid Metal Elastomer Composites,” *ACS Applied Materials & Interfaces*, vol. 12, pp. 17921–17928, Apr. 2020.
- [107] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, “Exploration of underwater life with an acoustically controlled soft robotic fish,” *Science Robotics*, vol. 3, Mar. 2018.
- [108] K. C. Galloway, K. P. Becker, B. Phillips, J. Kirby, S. Licht, D. Tchernov, R. J. Wood, and D. F. Gruber, “Soft Robotic Grippers for Biological Sampling on Deep Reefs,” *Soft Robotics*, vol. 3, pp. 23–33, Mar. 2016.
- [109] N. R. Sinatra, C. B. Teeple, D. M. Vogt, K. K. Parker, D. F. Gruber, and R. J. Wood, “Ultragentle manipulation of delicate structures using a soft robotic gripper,” *Science Robotics*, vol. 4, p. eaax5425, Aug. 2019.
- [110] S. Aracri, F. Giorgio-Serchi, G. Suaria, M. E. Sayed, M. P. Nemitz, S. Mahon, and A. A. Stokes, “Soft Robots for Ocean Exploration and Offshore Operations: A Perspective,” *Soft Robotics*, vol. 8, pp. 625–639, Dec. 2021.
- [111] G. Li, X. Chen, F. Zhou, Y. Liang, Y. Xiao, X. Cao, Z. Zhang, M. Zhang, B. Wu, S. Yin, Y. Xu, H. Fan, Z. Chen, W. Song, W. Yang, B. Pan, J. Hou, W. Zou, S. He, X. Yang, G. Mao, Z. Jia, H. Zhou, T. Li, S. Qu, Z. Xu, Z. Huang, Y. Luo, T. Xie, J. Gu, S. Zhu, and W. Yang, “Self-powered soft robot in the Mariana Trench,” *Nature*, vol. 591, pp. 66–71, Mar. 2021.
- [112] J. D. Hubbard, R. Acevedo, K. M. Edwards, A. T. Alsharhan, Z. Wen, J. Landry, K. Wang, S. Schaffer, and R. D. Sochol, “Fully 3D-printed soft robots with integrated fluidic circuitry,” *Science Advances*, vol. 7, p. eabe5257, July 2021.
- [113] G. Cao, B. Huo, L. Yang, F. Zhang, Y. Liu, and G. Bian, “Model-Based Robust Tracking Control Without Observers for Soft Bending Actuators,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 5175–5182, July 2021.
- [114] P. Hyatt, D. Wingate, and M. D. Killpack, “Model-Based Control of Soft Actuators Using Learned Non-linear Discrete-Time Models,” *Frontiers in Robotics and AI*, vol. 6, no. 22, 2019.

- [115] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, “Data-Driven Control of Soft Robots Using Koopman Operator Theory,” *IEEE Transactions on Robotics*, vol. 37, pp. 948–961, June 2021.
- [116] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, “Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators,” *IEEE Transactions on Robotics*, vol. 35, pp. 124–134, Feb. 2019.
- [117] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, “Local Koopman Operators for Data-Driven Control of Robotic Systems,” in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, June 2019.
- [118] M. C. Yip and D. B. Camarillo, “Model-Less Hybrid Position/Force Control: A Minimalist Approach for Continuum Manipulators in Unknown, Constrained Environments,” *IEEE Robotics and Automation Letters*, vol. 1, pp. 844–851, July 2016.
- [119] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, “Soft Robot Control With a Learned Differentiable Model,” in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 417–423, May 2020.
- [120] A. Doroudchi, R. Khodambashi, M. Sharifzadeh, D. Li, S. Berman, and D. M. Aukes, “Tracking Control of a Miniature 2-DOF Manipulator With Hydrogel Actuators,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 4774–4781, July 2021.
- [121] H. Yang, M. Xu, W. Li, and S. Zhang, “Design and Implementation of a Soft Robotic Arm Driven by SMA Coils,” *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 6108–6116, Aug. 2019.
- [122] Y. Wang, L. Xie, and C. E. de Souza, “Robust control of a class of uncertain nonlinear systems,” *Systems & Control Letters*, vol. 19, pp. 139–149, Aug. 1992.
- [123] S. Tarbouriech, G. Garcia, J. M. G. da Silva Jr, and I. Queinnec, *Stability and Stabilization of Linear Systems with Saturating Actuators*. Springer Science & Business Media, 2011.
- [124] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, vol. 2. Citeseer, 2007.
- [125] A. P. Sabelhaus, H. Zhao, E. L. Zhu, A. K. Agogino, and A. M. Agogino, “Model-Predictive Control With Inverse Statics Optimization for Tensegrity Spine Robots,” *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 263–277, Jan. 2021.

- [126] L. Tomholt, L. J. Friesen, D. Berdichevsky, M. C. Fernandes, C. Pierre, R. J. Wood, and J. C. Weaver, “The structural origins of brittle star arm kinematics: An integrated tomographic, additive manufacturing, and parametric modeling-based approach,” *Journal of Structural Biology*, vol. 211, p. 107481, July 2020.
- [127] K. Larson, “Can You Estimate Modulus From Durometer Hardness for Silicones?,” p. 5, 2016.
- [128] A. Ainla, M. S. Verma, D. Yang, and G. M. Whitesides, “Soft, Rotating Pneumatic Actuator,” *Soft Robotics*, vol. 4, pp. 297–304, Sept. 2017.
- [129] A. N. Gent, “On the Relation between Indentation Hardness and Young’s Modulus,” *Rubber Chemistry and Technology*, vol. 31, pp. 896–906, Sept. 1958.
- [130] M. Hovd, R. D. Braatz, and S. Skogestad, “SVD controllers for  $H_2$ -,  $H_\infty$ - and  $\mu$ -optimal control,” *Automatica*, vol. 33, pp. 433–439, Mar. 1997.
- [131] K. J. Åström and R. M. Murray, *Feedback Systems*. Princeton university press, 2010.
- [132] S. Tarbouriech and M. Turner, “Anti-windup design: An overview of some recent advances and open problems,” *IET Control Theory & Applications*, vol. 3, pp. 1–19, Jan. 2009.
- [133] R. Hanus, M. Kinnaert, and J. L. Henrotte, “Conditioning technique, a general anti-windup and bumpless transfer method,” *Automatica*, vol. 23, pp. 729–739, Nov. 1987.
- [134] R. D. Braatz and M. Morari, “Stability and Performance Analysis of Systems Under Constraints.” <https://resolver.caltech.edu/CaltechCDSTR:1993.009>, June 1993.
- [135] P. J. Campo and M. Morari, “Robust control of processes subject to saturation nonlinearities,” *Computers & Chemical Engineering*, vol. 14, pp. 343–358, May 1990.
- [136] M. V. Kothare and M. Morari, “Multiplier theory for stability analysis of anti-windup control systems,” *Automatica*, vol. 35, pp. 917–928, May 1999.
- [137] T. Wang, Y. Zhang, Z. Chen, and S. Zhu, “Parameter Identification and Model-Based Nonlinear Robust Control of Fluidic Soft Bending Actuators,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, pp. 1346–1355, June 2019.
- [138] C. Jarrett and A. J. McDaid, “Robust Control of a Cable-Driven Soft Exoskeleton Joint for Intrinsic Human-Robot Interaction,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, pp. 976–986, July 2017.

- [139] A. A. Alqumsan, S. Khoo, and M. Norton, “Robust control of continuum robots using Cosserat rod theory,” *Mechanism and Machine Theory*, vol. 131, pp. 48–61, Jan. 2019.
- [140] K. Suzumori, S. Iikura, and H. Tanaka, “Development of flexible microactuator and its applications to robotic mechanisms,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1622–1623, IEEE Computer Society, 1991.
- [141] H. Wang, C. Wang, W. Chen, X. Liang, and Y. Liu, “Three-Dimensional Dynamics for Cable-Driven Soft Manipulator,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 18–28, Feb. 2017.
- [142] S. P. Timoshenko, “LXVI. On the correction for shear of the differential equation for transverse vibrations of prismatic bars,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 245, pp. 744–746, 1921.
- [143] J. D. Shackleton, “Skeletal homologies, phylogeny and classification of the earliest asterozoan echinoderms,” *Journal of Systematic Palaeontology*, vol. 3, pp. 29–114, Jan. 2005.
- [144] B. Mueller, A. R. Bos, G. Graf, and G. S. Gumanao, “Size-specific locomotion rate and movement pattern of four common Indo-Pacific sea stars (Echinodermata; Asteroidea),” *Aquatic Biology*, vol. 12, pp. 157–164, Apr. 2011.
- [145] G. Hendler and J. E. Miller, “Swimming ophiuroids—Real and imagined,” in *Biology of Echinodermata*, pp. 179–190, CRC Press, 2020.
- [146] S. Stöhr, T. D. O’Hara, and B. Thuy, “Global Diversity of Brittle Stars (Echinodermata: Ophiuroidea),” *PLOS ONE*, vol. 7, p. e31940, Mar. 2012.
- [147] Aristotle, *On the Gait of Animals*. 350 B.C.E.
- [148] M. H. Raibert, “Hopping in legged systems — Modeling and simulation for the two-dimensional one-legged case,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, pp. 451–463, May 1984.
- [149] R. Blickhan, “The spring-mass model for running and hopping,” *Journal of Biomechanics*, vol. 22, pp. 1217–1227, Jan. 1989.
- [150] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, Oct. 2018.

- [151] A. V. Birn-Jeffery and T. E. Higham, “The Scaling of Uphill and Downhill Locomotion in Legged Animals,” *Integrative and Comparative Biology*, vol. 54, pp. 1159–1172, Dec. 2014.
- [152] S. M. Reilly, E. J. McElroy, and A. R. Biknevičius, “Posture, gait and the ecological relevance of locomotor costs and energy-saving mechanisms in tetrapods,” *Zoology*, vol. 110, pp. 271–289, Sept. 2007.
- [153] R. E. Ritzmann, R. D. Quinn, and M. S. Fischer, “Convergent evolution and locomotion through complex terrain by insects, vertebrates and robots,” *Arthropod Structure & Development*, vol. 33, pp. 361–379, July 2004.
- [154] M. Calisti, F. Corucci, A. Arienti, and C. Laschi, “Dynamics of underwater legged locomotion: Modeling and experiments on an octopus-inspired robot,” *Bioinspiration & Biomimetics*, vol. 10, p. 046012, July 2015.
- [155] B. L. Coughlin and F. E. Fish, “Hippopotamus Underwater Locomotion: Reduced-Gravity Movements for a Massive Mammal,” *Journal of Mammalogy*, vol. 90, pp. 675–679, June 2009.
- [156] M. A. Ashley-Ross, R. Lundin, and K. L. Johnson, “Kinematics of level terrestrial and underwater walking in the California newt, *Taricha torosa*,” *Journal of Experimental Zoology Part A: Ecological Genetics and Physiology*, vol. 311A, no. 4, pp. 240–257, 2009.
- [157] C. Pond, “The Role of the ‘Walking Legs’ in Aquatic and Terrestrial Locomotion of the Crayfish *Austropotamobius Pallipes* (Lereboullet),” *Journal of Experimental Biology*, vol. 62, pp. 447–454, Apr. 1975.
- [158] J. Ayers and J. Witting, “Biomimetic approaches to the control of underwater walking machines,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Nov. 2006.
- [159] T. Kano, D. Kanauchi, H. Aonuma, E. G. Clark, and A. Ishiguro, “Decentralized Control Mechanism for Determination of Moving Direction in Brittle Stars With Penta-Radially Symmetric Body,” *Frontiers in Neurobotics*, vol. 13, 2019.
- [160] M. Calisti, E. Falotico, and C. Laschi, “Hopping on Uneven Terrains With an Underwater One-Legged Robot,” *IEEE Robotics and Automation Letters*, vol. 1, pp. 461–468, Jan. 2016.
- [161] B. M. Ashby and S. L. Delp, “Optimal control simulations reveal mechanisms by which arm movement improves standing long jump performance,” *Journal of Biomechanics*, vol. 39, pp. 1726–1734, Jan. 2006.
- [162] G. Hicks and K. Ito, “A method for determination of optimal gaits with application to a snake-like serial-link structure,” *IEEE Transactions on Automatic Control*, vol. 50, pp. 1291–1306, Sept. 2005.

- [163] S. Schaal, P. Mohajerian, and A. Ijspeert, “Dynamics systems vs. optimal control — a unifying view,” in *Progress in Brain Research* (P. Cisek, T. Drew, and J. F. Kalaska, eds.), vol. 165 of *Computational Neuroscience: Theoretical Insights into Brain Function*, pp. 425–445, Elsevier, Jan. 2007.
- [164] K. Wampler and Z. Popović, “Optimal gait and form for animal locomotion,” *ACM Transactions on Graphics (TOG)*, July 2009.
- [165] W. Li and E. Todorov, “ITERATIVE LINEAR QUADRATIC REGULATOR DESIGN FOR NONLINEAR BIOLOGICAL MOVEMENT SYSTEMS,” in *First International Conference on Informatics in Control, Automation and Robotics*, vol. 2, pp. 222–229, SCITEPRESS, Aug. 2004.
- [166] T. A. Howell, S. L. Cleac’h, J. Z. Kolter, M. Schwager, and Z. Manchester, “Dojo: A Differentiable Physics Engine for Robotics,” Mar. 2022.
- [167] M. Martinez, R. Full, and M. Koehl, “Underwater punting by an intertidal crab: A novel gait revealed by the kinematics of pedestrian locomotion in air versus water,” *Journal of Experimental Biology*, vol. 201, pp. 2609–2623, Sept. 1998.
- [168] D. M. Koester and C. P. Spirito, “Punting: An Unusual Mode of Locomotion in the Little Skate, *Leucoraja erinacea* (Chondrichthyes: Rajidae),” *Copeia*, vol. 2003, pp. 553–561, Sept. 2003.
- [169] L. J. Macesic and S. M. Kajiura, “Comparative punting kinematics and pelvic fin musculature of benthic batoids,” *Journal of Morphology*, vol. 271, no. 10, pp. 1219–1228, 2010.
- [170] P. Gorzelak and S. Zamora, “Understanding form and function of the stem in early flattened echinoderms (pleurocystitids) using a microstructural approach,” *PeerJ*, vol. 4, p. e1820, Apr. 2016.
- [171] I. A. Rahman, R. P. S. Jefferies, W. H. Südkamp, and R. D. A. Smith, “Ichnological Insights into Mitrata Palaeobiology,” *Palaeontology*, vol. 52, no. 1, pp. 127–138, 2009.
- [172] J. A. Nyakatura, K. Melo, T. Horvat, K. Karakasiliotis, V. R. Allen, A. Andikfar, E. Andrada, P. Arnold, J. Lauströer, J. R. Hutchinson, M. S. Fischer, and A. J. Ijspeert, “Reverse-engineering the locomotion of a stem amniote,” *Nature*, vol. 565, pp. 351–355, Jan. 2019.
- [173] B. McInroe, H. C. Astley, C. Gong, S. M. Kawano, P. E. Schiebel, J. M. Rieser, H. Choset, R. W. Blob, and D. I. Goldman, “Tail use improves performance on soft substrates in models of early vertebrate land locomotors,” *Science*, vol. 353, pp. 154–158, July 2016.

- [174] G. Q. Huang, M. Z. Q. Chen, and J. Pan, “Robotics in ecommerce logistics,” *HKIE Transactions*, vol. 22, pp. 68–77, Apr. 2015.
- [175] K. S. Saidi, T. Bock, and C. Georgoulas, “Robotics in Construction,” in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), Springer Handbooks, pp. 1493–1520, Cham: Springer International Publishing, 2016.
- [176] R. Sparrow and M. Howard, “Robots in agriculture: Prospects, impacts, ethics, and policy,” *Precision Agriculture*, vol. 22, pp. 818–833, June 2021.
- [177] E. Guizzo, “By leaps and bounds: An exclusive look at how Boston dynamics is redefining robot agility,” *IEEE Spectrum*, vol. 56, pp. 34–39, Dec. 2019.
- [178] S. Kim, C. Laschi, and B. Trimmer, “Soft robotics: A bioinspired evolution in robotics,” *Trends in Biotechnology*, vol. 31, pp. 287–294, May 2013.
- [179] D. R. Higuera-Ruiz, K. Nishikawa, H. Feigenbaum, and M. Shafer, “What is an artificial muscle? A comparison of soft actuators to biological muscles,” *Bioinspiration & Biomimetics*, vol. 17, p. 011001, Dec. 2021.
- [180] N. Kellaris, V. Gopaluni Venkata, G. M. Smith, S. K. Mitchell, and C. Keplinger, “Peano-HASEL actuators: Muscle-mimetic, electrohydraulic transducers that linearly contract on activation,” *Science Robotics*, vol. 3, p. eaar3276, Jan. 2018.
- [181] C. Majidi, “Soft Robotics: A Perspective—Current Trends and Prospects for the Future,” *Soft Robotics*, vol. 1, pp. 5–11, Mar. 2014.
- [182] L. Tolstoy, *Anna Karenina*. Oxford ; New York : Oxford University Press, 1980., 1980.