Recursive Robotic Assemblers

by

Miana M. Smith

S.B. Mechanical Engineering, Massachusetts Institute of Technology (2021)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author____

Miana M. Smith
Program in Media Arts and Sciences
May 19, 2023

Certified by____

Prof. Neil Gershenfeld
Director
MIT Center for Bits and Atoms
Thesis Supervisor

Accepted by____

Prof. Tod Machover
Academic Head
Program in Media Arts and Sciences

Recursive Robotic Assemblers

by

Miana M. Smith

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Biology efficiently builds across size scales: at the scale of tens of nanometers, ribosomes assemble more ribosomes, enabling the highly parallelized production of proteins that make up living systems ranging from prokaryotes at the scale of microns, to blue whales at the scale of tens of meters. At a level above ribosomes, we might consider cell division as another type of assembly process: as the size scale of the assembled parts grows, the assemblers also grow. This represents a recursive and hierarchical assembly process. In contrast, current robotic and CNC construction processes, though often parallelized, are constrained to preset, limited assembly rates and sizes. Inspired by biology, this thesis considers how we might develop recursive and hierarchical robotic assembly systems. That is, similar to a biological assembly system, can we develop a robotic assembly system that is able to build robots, structures, and robots integrated in structures?

To this end, we decompose both the robot and the structures into a set of compatible building blocks, or voxels, that can assemble and reassemble into more complex structures. The decomposition of the robot is based on a "functional voxel" that routes electrical signals and power, in addition to mechanical forces. Robotic modules are made by incorporating actuation, which then assemble into reconfigurable robots using a reversible solder joint. An additional set of construction voxels, which do not contain electrical features, enables the robot to assemble higher performance structures. This work exists at the intersection of modular robotics and collective robotic construction, prioritizing scalability— our ability to produce many robots that then build useful structures.

A set of functional voxels, robot modules, and construction voxels have been developed and characterized. The robotic system is characterized by its function: the robot is able to assemble another robot and the robot is able to assemble construction voxels into small structures. The construction voxel system is characterized using mechanical testing, which verifies that the material system is performant. Together, this demonstrates all the elements required for recursive robotic assembly, in which a robot is able to assemble both more robots and larger structures.

Thesis Supervisor: Prof. Neil Gershenfeld
Title: Director, MIT Center for Bits and Atoms

Recursive Robotic Assemblers

by

Miana M. Smith

The following people served as readers for this thesis:

Thesis Reader____

Prof. Caitlin Mueller
Associate Professor of Civil and Environmental Engineering, Architecture
Massachusetts Institute of Technology

Thesis Reader____

Prof. Russ Tedrake
Toyota Professor of EECS, Aero/Astro, MechE
Massachusetts Institute of Technology

Acknowledgements

Thank you to Neil, for creating this unique space and pushing this research. I feel very lucky to be a part of the CBA.

Thank you to my readers, Prof. Caitlin Mueller and Prof. Russ Tedrake, for your time and guidance in developing this thesis.

To the CBA students, postdocs, researchers, UROPs, and friends of the lab— Dave, Amira, Alfonso, Jake, Camron, Eric, Zach, Chris, Natalie, Nicole, Patritsia, Eyal, Jack, Quentin, Leo, Jiri, Kristof, and Sophia— thanks for making the CBA fun, thanks for making it interesting, and thanks for making it a community.

Thank you to Amira for all the work you have done building a foundation for this project. I'm very grateful to have gotten to work with you. Thank you to Chris for your work on the first version of this project, and thank you for onboarding me to this project so well. Thank you to Alfonso for your expertise on Instrons, projectiles, and vernacular. Thank you to Dave for your help with the hard-hitting technical questions, water jug excursions, and your friendship in the long basements nights and days.

To Kara, Tom, John, Sherry, Joe, Candace— thank you for keeping the CBA running.

To all my friends, thank you for connecting me back to the outside world. A special thank you to Rachel, for your mentorship, and to Serena, for your enabling.

To my family, thank you for your love, support, and all the snacks you've sent.

To Milka— thank you for reading the first drafts and the final drafts, thank you for the adventures, and thank you for much more than I know how to say.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomously making large structures is challenging: as the structure increases in size, so too do the size, cost, and complexity of the machines needed. This growth is driven in part by the challenge of keeping error low as the structure grows and in part by the practice of using machines the size of, if not larger than, the constructed structures. In contrast, nature offers many examples of organisms that prolifically build structures much larger than themselves, such as honeycomb [2], the nests of weaver birds [3], or the temporary self-assemblies of army ants [4]. Taking inspiration from these biological systems, collective robotic construction offers an alternate approach to building big, where many robotic agents collaborate to build and maintain structures [5]. However, many collective robotic construction approaches use fixed or known swarm sizes with fixed robot geometries, which is unlike the flexibility that natural systems exhibit when building. Recent research from the MIT Center for Bits and Atoms (CBA) suggests that to efficiently assemble large structures it is critical that the system be recursive and hierarchical— that is, the robots need to be able to build more robots and larger robots [6]. This thesis seeks to demonstrate a first iteration of a self-replicating robotic swarm system for assembly in hardware.

Though automation is becoming more prevalent in construction— such as in 3D-printing houses or in the development of bricklaying robots— these processes remain limited by the size and complexity of the machines needed [12] [13]. Figure 1-1 shows, for example, a 3D

Figure 1-1: Comparison of natural and synthetic building processes. A) Army ant bridge [7]; B) Honeybees festooning [8]; C) Sociable weaver nests [3]; D) WASP 3D printer [9]; E) HadrianX bricklayer [10]; F) Airbus Beluga [11].

printer for a house which is three times the size of the printed structure [9]. In fields in which automation is the norm, the construction of large structures still poses additional challenges and costs: composite parts for jumbo jets are made using tooling the size of a house [14], and once assembled, are transported using even larger jets [11]. As the size of the machine gets larger, the demands on the relative precision of the machine also get larger; the machine must travel larger distances while maintaining a low error threshold in material placement.

To address this error scaling, we can adopt an assembly-based approach to building large structures using discrete elements, or voxels, with error-correcting features [15]. The canonical example of this approach is Lego Bricks, in which an imprecise assembler— a child— can repeatably assemble complex and large structures because of the alignment features on the bricks. By varying the local geometry of a voxel, we can design materials with specific properties, such as varied compliance in different orientations [16]. To decouple assembler size from structure size, we take a swarm robotics approach to assembly, where many small robots locomote over the structure and place new voxels. The digital nature of the underlying structure enables error-correction for the robot, which then decreases the need for complex feedback systems [17]. Together, these features represent a strategy for building large, high performance structures.

However, this approach is limited by the throughput of single robots that place single voxels at a time. To address this, the robot itself can also be decomposed into modules that then enable the assembly of more robots, or even larger robots that can manipulate more material at once. In this way, the swarm can scale itself to the construction task at hand. Simulations have demonstrated that taking this self-replicating hierarchical approach can increase the efficiency of constructing large structures [6]. Together, these features form the basis of a robotic assembly system in which the distinction between material, assembler, and built structure are blurred: robots are made from the parts that they assemble. The work in this thesis primarily draws from and combines three research areas: discrete material systems, modular robotics, and relative robotic construction.

1.1 Discrete Material Systems

Prior research has demonstrated that we can build large structures with desirable and design-able material properties using discrete materials. [15] achieved a record for light-weight high-stiffness materials using a cuboctahedron ("cuboct") lattice made from discretely assembled units of looped carbon fiber. [16] developed a library of injection molded GFRP voxels for achieving highly varied material properties— using the cuboct geometry as a base, a set of compliant, auxetic, and chiral voxels are demonstrated and characterized. See figure 1-2 for reference. Further, by using discrete materials, we can disassemble (reassemble) our structures— this means that component parts can have lifespans beyond the structures they comprise. Significantly, this also means that structures can be easily repaired in contrast to more monolithic styles of fabrication [18].

This set of materials has been applied toward diverse applications. A morphing wing (Mission Adaptive Digital Composite Aerostructure Technologies or MADCAT) was developed with NASA, which leveraged the variable stiffness of a voxel structure to achieve an

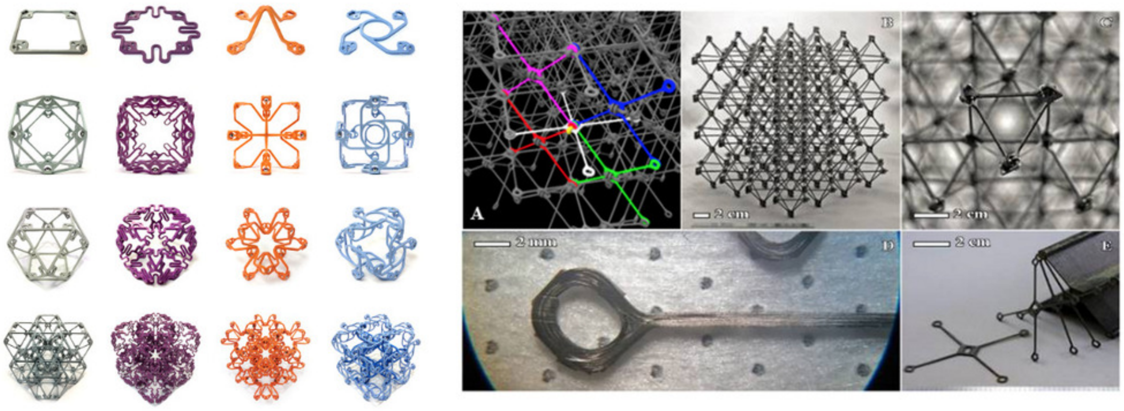

Figure 1-2: Prior CBA voxel systems. (Left) A library of voxel types with varied material properties from [16]. (Right) Digital composites— oriented carbon fiber assembled into a cuboct lattice, from [15]

aero structure with continuous deformations. This wing was composed of 1000s of voxels and was assembled by hand, according to algorithmic steps [19]. Similarly, a high efficiency race car was developed with Toyota as part of an annual ultra-high mileage race car competition. While previous composite layup style cars could take up to a year to design and make the tooling, the body of this car was assembled in weeks from a voxel feedstock [20]. These examples are shown in figure 1-3.

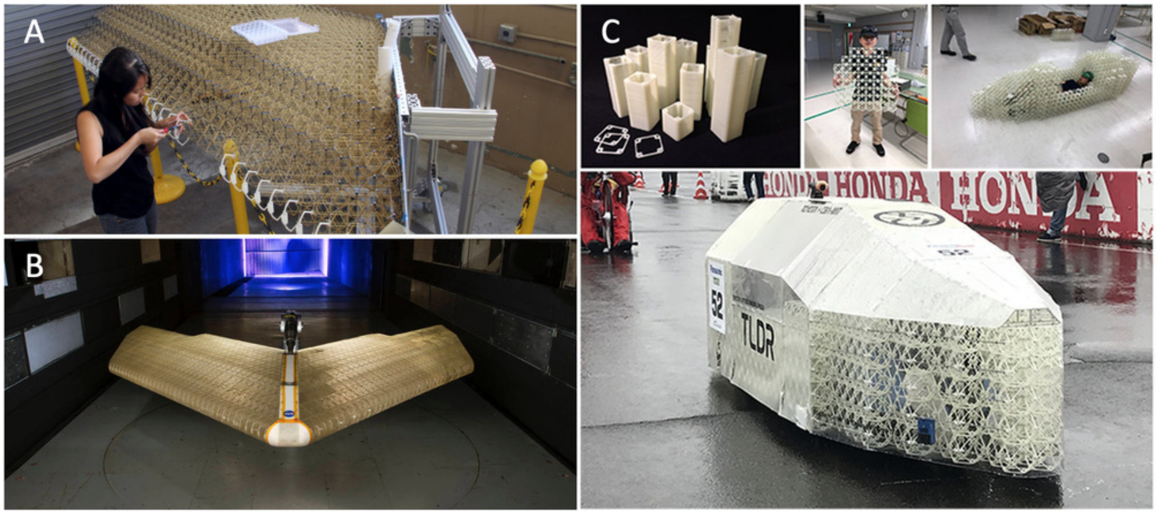

Figure 1-3: Large-scale voxel assemblies. A) Hand assembly of MADCAT [21]; B) Completed MADCAT [19]; C) TLDR, a lightweight voxel based race car built with Toyota [20]

1.2 Modular Robotics

Modular robotics is a well established field that provides many useful references from which this thesis builds. The goal of modular robotics is to enable highly customizable robotic geometries for a variety of challenges using a common set of building blocks, or modules. These challenges could include developing robotic modules that enable multiple locomotive strategies, such as walking in one configuration and slithering in another [22]. To these ends, a variety of actuation and attachment strategies have been developed, some of which are shown in figure 1-4. A shows M-TRAN III, a highly reconfigurable modular robotic system that utilizes a chain-like geometry [22]. B shows Roombots, a modular robotic system designed to act as reconfigurable furniture [23]. The Roombots system demonstrates a high degree of autonomy and assemble into structural systems; it possesses many of the characteristics this thesis seeks to achieve. C shows the Soldercubes system, which notably introduces a reversible low-melt solder joint between modules to transmit power and signal [1]. The joint system in this thesis is based on the joint system presented in Soldercubes. D shows the M-blocks system, which uses a unique flywheel-brake system to achieve locomotion, demonstrating one of the more novel modes of actuation employed in modular robotics [24]. E shows the FireAnt3D system which aims to make a reconfigurable robot not limited to a lattice geometry, as the prior examples are [25].

Many of these prior systems aim to make modular robotic systems for the sake of modular robotic systems. That is, they are fully committed to realizing a very high degree of reconfigurability and a very high degree of distributed function. Often, each individual module must function as its own robot— i.e., a single module that can do it all. This type of requirement leads to much more complex (and multi-functional) modules, which in turn make it challenging to scale to much higher module counts. Because this thesis is concerned with making a modular robotic system for the purpose of using modular robots to assemble larger structures, I am not subject to these constraints. The modules in this thesis do not need to be of all one type, they do not need to employ highly decentralized communications and controls strategies, and they do not need to be able to be arbitrarily composed. Instead, they need to be able to self-assemble (given an initial seed robot) and to assemble lattice

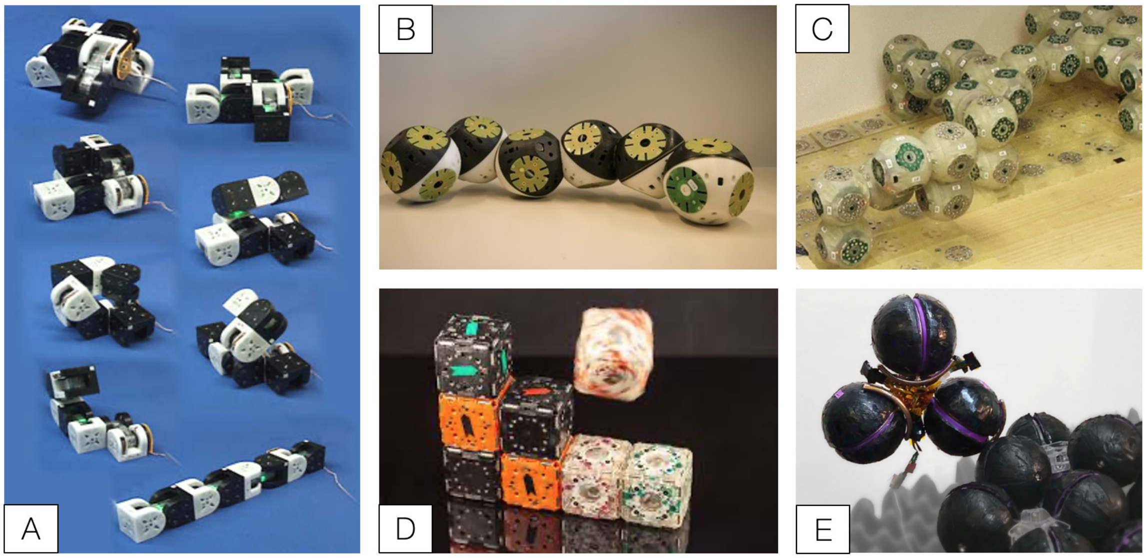

Figure 1-4: Example modular robotic systems. A) M-TRAN III [22], image from [26]. B) Roombots [23], image from [27]. C) Soldercubes, image from [28]. D) M-Blocks [24], image from [29]. E) FireAnt3D, image from [25].

structures. So— we can build from the strategies and technologies already developed and apply them toward this purpose.

1.3 Relative Robotic Construction

To maintain global accuracy over the voxel structure, we adopt a “relative robot” approach to assembly [20], in which our robot and material system are designed together, such that the material system is able to correct errors in the positioning of the robotic system. Relative robotic construction represents a subset of collective robot construction, where swarms of robot assemble structures much larger than themselves. The key feature of a relative robotic construction system as opposed to a general collective system is the interface between the robot and the assembled material. Some examples of robotic construction systems that use this strategy are shown in Figure 1-5.

A shows the TERMES system, which uses termite-inspired robots to assemble a brick-like substrate. The robots are made with ”whegs” (wheel-legs) specifically designed for

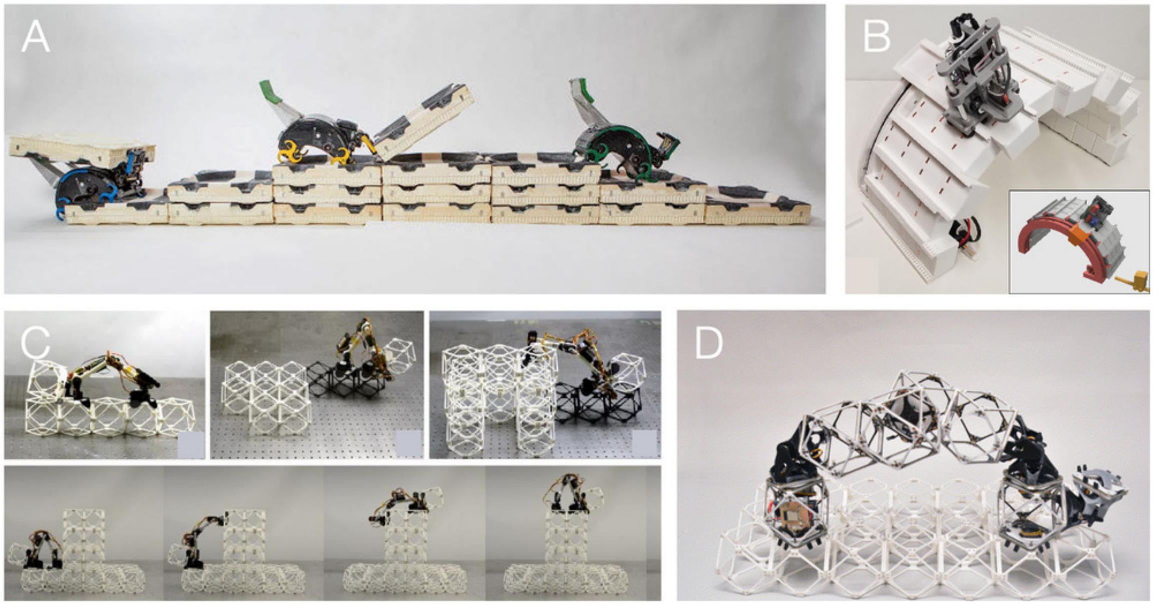

Figure 1-5: Overview of example relative robotic systems. A) TERMES system [30]; B) VaultBot [31]; C) BILL-E [17]; D) Magnetic BILL-E [6].

locomoting over their assembly substrate. Bricks are then outfitted with both indexing and magnetic features to facilitate assembly [30]. B shows the VaultBot system, which assembles spanning structures by extruding bricks out from an initial arch. This system also features a high degree of integration between the robot and the assembly substrate, and makes use of two different robot geometries [31].

C and D show iterations of the BILL-E voxel assembly system, the immediate predecessors of the assembly robot described in this thesis. The first BILL-E system (shown in C) consists of an inchworm style robot that is able to pick and place magnetically connected voxels. The voxel system enables the robot to have good global positional accuracy despite having minimal to no feedback systems. D shows magnetic BILL-E, which aimed to decompose BILL-E into a magnetic modular feedstock. Both of these systems are further described in the background section of Chapter 2.

The magnetic BILL-E system represents a first pass at a hardware system for recursive and hierarchical assembly of voxel systems. Because of various limitations further discussed later, the hardware system was not fully able to demonstrate these goals. However, sim-

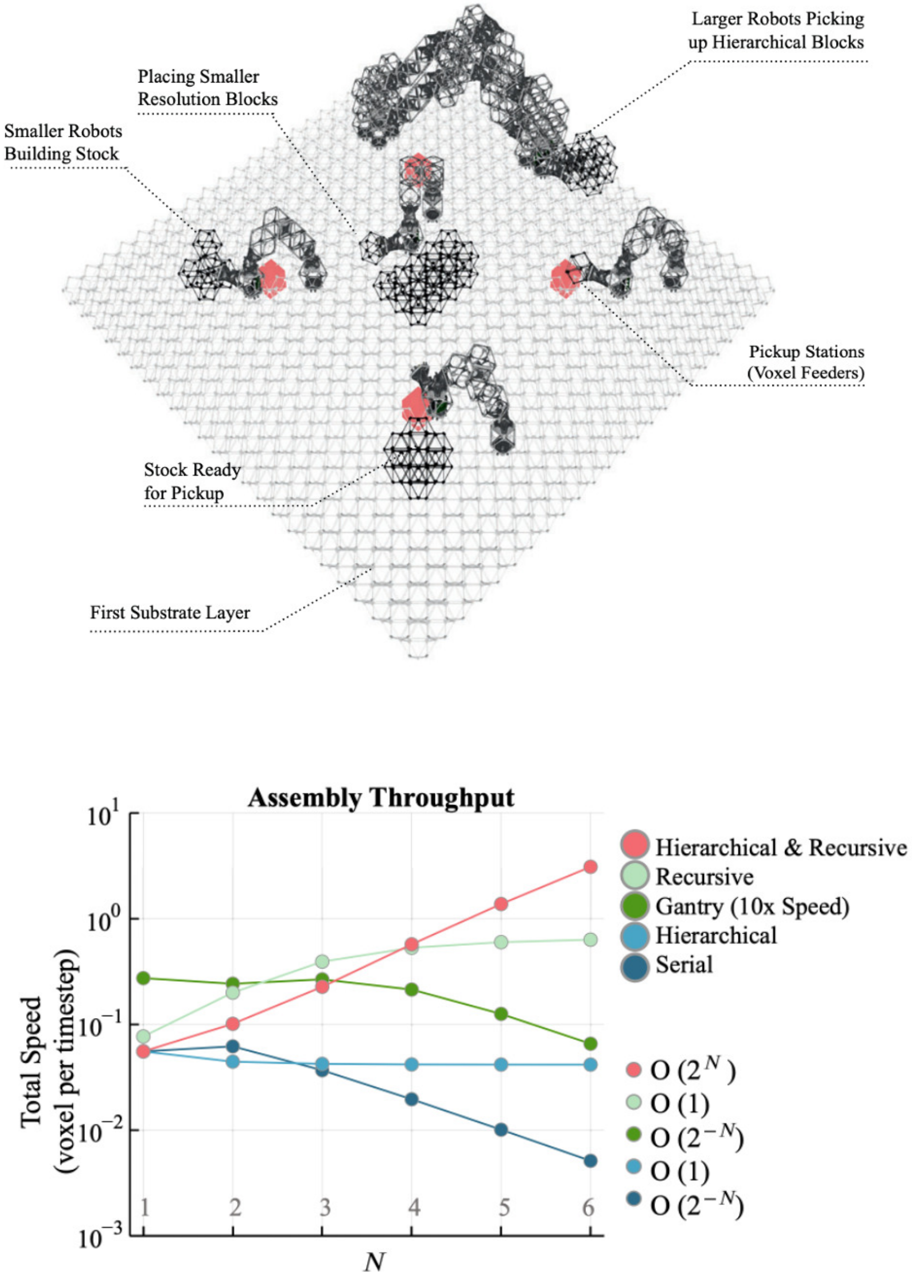

Figure 1-6: (Top) Simulated hierarchical and recursive robotic swarm. (Bottom) The assembly throughput of hierarchical and recursive robot swarms, recursive swarms, hierarchical robots, serial robotic construction, and gantry construction for voxel structures of increasing size. Both images from [6], image credits: A. Abdel-Rahman.

ulation workflows for optimizing the path planning and swarm behaviors of this style of robotic system were developed [6]. This style of robotic system introduces a question of how to know when a robot should build more structure, build more robots, or build larger robots. [6] explores these questions and shows, in simulation, that a hierarchical and recursive robotic assembly swarm is able to achieve highly efficient assembly throughput for large voxel structures.

1.4 Contributions

The goal of this thesis is in many ways to realize a hardware version of the recursive and hierarchical robotic assembly system presented in [6] and summarized in figure 1-6. I begin with demonstrating a recursive or self-replicating robotic assembly system, with the potential for hierarchical growth. To this end, this thesis presents and demonstrates a set of hardware for making robots that can assemble (useful) structures as well as more robots. Chapter 2 describes the robotic system, including relevant background, system design, and robot performance. Chapter 3 describes the construction voxel material system, including voxel design, mechanical performance, and comparisons to other cellular materials.

Chapter 2

Assembly Robots

Extending on the prior BILL-E systems developed at the CBA, this thesis demonstrates a self-replicating robotic assembly system. As in these prior projects, this thesis focuses on a robot that locomotes over the lattice, deriving accuracy from self-correcting features between the lattice and robot, and places new voxels. The robot is additionally comprised of a modular feedstock such that it may self-replicate.

2.1 Background

The self-replicating assembly robots in this thesis draw from prior literature in collective robotic construction and modular robotics, introduced in the Introduction. In this section, I will give a deeper overview of the prior CBA voxel assembly systems that this thesis builds from as well as an overview of prior modular robotic connection schemes which inform the robotic module design of this robot.

2.1.1 BILL-E

The most immediate predecessor of this thesis is the BILL-E voxel assembly system developed at CBA. BILL-E is a relative robot designed to locomote over a voxel structures

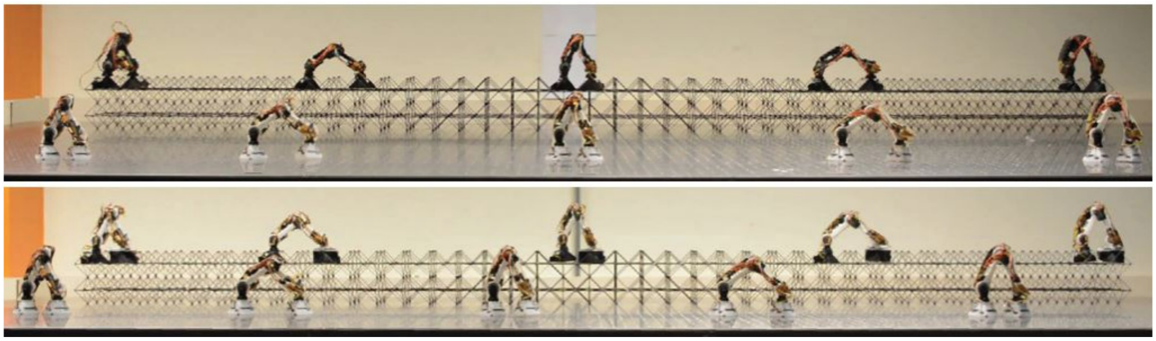

Figure 2-1: BILL-E v0 locomotes over a lattice versus a plane with intentional error. (Top) Both robots locomote without added error. (Bottom) With intentional added error, the robot on the plane accumulates substantial error over the course of the walk while the lattice walker does not. Image from [20].

and pick and place new voxels. The first version of BILL-E, detailed in [32], introduced BILL-E as a relative robotic assembler, though was not used for voxel assembly. This work demonstrated that voxel lattice in combination with gripper compliance enables local error correction for the robot, preventing the accumulation of error as the robot locomotes.

The most capable version of this system assembles a magnetic voxel feedstock and is described in [17]. Voxels from [17] consist of face-connected cuboct cells made from 3D printed ABS at a lattice pitch of 101.6 mm. The faces are 3D printed as they have a triangular beam cross-section, to better facilitate mating with BILL-E's grippers. Faces manually assemble into voxels using a finger joint fixed with a screw. BILL-E then assembles voxels together face-to-face using attraction between the eight magnets embedded per voxel face. Magnets are installed such that the voxels assemble together agnostic of their relative orientation. Additionally, assembly is done on top of a stainless steel optical table, which provides a flat, controlled ground plane the magnets adhere to. BILL-E utilizes three grippers: two act as the feet of the robot, while the last acts as the voxel pick-and-placer. These are all actuated using hobby servos and commands to the robot are sent over radio. This work demonstrated the assembly of voxel structures on the order of 25 voxels using 1 - 2 robots. The utility of this system is limited by the use of magnetic connectors, which place a low upper limit on the strength of the resultant structure [17].

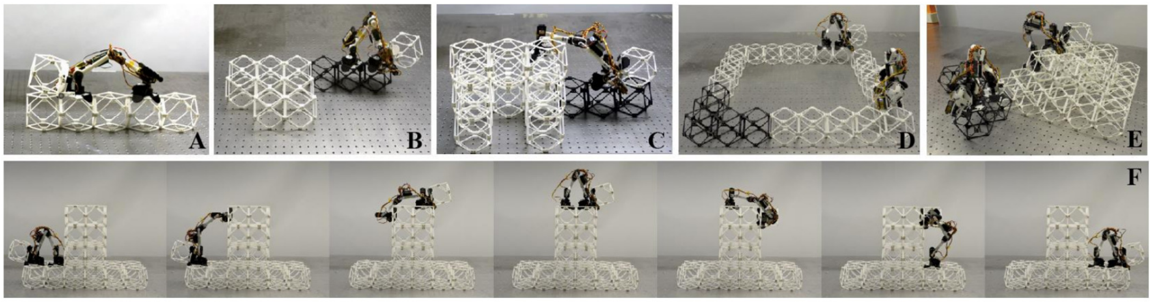

Figure 2-2: A) BILL-E iteratively moves 6 voxels forward. B) BILL-E constructs 3x3 plane. C) BILL-E constructs 3D structure with overhang. D) Two BILL-Es build lines that meet. E) Two BILL-Es construct a pyramid (30 voxels). F) BILL-E climbs a wall while holding a voxel payload. Image from [17].

2.1.2 Magnetic Exponential BILL-E

The starting point for this thesis work is a magnetic modular version of BILL-E. The hardware was developed by Chris Cameron, with contributions from Zach Fredin, Jiri Zemanek, and the author. Path planning, optimization, and simulation for this system was done by Amira Abdel-Rahman. This work is described in [6]. Like the prior BILL-E system, it was targeted toward the assembly of the same previously described magnetic voxels. Similar to the design of the magnetic voxels, this version used magnetic connections between modules for the structural connection and spring-loaded pins for the electrical connection, transmitting power, ground, and one signal line between modules. The configuration of the robot is similar to that of the non-modular BILL-E, though because each module must occupy voxel space to satisfy the self-assembly requirement, modular BILL-E is subject to tighter joint limits and more limited accessible voxel space.

The primary issue with this system lies in the strength of the connections between modules. The magnetic connections do not provide sufficient strength for modular BILL-E to reliably support its own weight during routine movements, such as taking a step while carrying a single voxel. Though the magnets nominally provide 56 N of holding force (assuming perfect contact between all 8 magnets), in practice the magnets provided an average of 20 N. Attempts at increasing this value by adding back irons, increasing magnet thickness, and improving installation process resulted in an approx. 10 N increase to 30

Figure 2-3: The module types in magnetic BILL-E. (From left) Primary module including ESP32 microcontroller board. Elbow module. Wrist module. Gripper module.

N of average holding force— still too low. Further, the use of magnetic joints of similar strength in both the robot and the structure makes the removal of voxels unreliable: a robot joint may detach instead of the target voxel.

The lack of feedback from the actuators also proved to be an issue— there is no way for the system to know if a step has been completed. Additionally, because the communications run one-way, the system has no way of knowing if e.g., one of its legs has fallen off. For the smallest version of the robot to be able to address any voxel space in a plane (instead of being limited to diagonals) it is necessary for it to do a step where it cantilevers a large portion of its weight— this maneuver is near both the torque limits of the servos and the failure strength of magnetic connections, resulting in inconsistent behavior caused by brown outs. Though this system demonstrated a set of parts potentially capable of self-assembly and assembly, the hardware was not able to fully realize that goal.

Figure 2-4 A shows a CAD render of the version of the robot deployed in hardware. The additional voxel gripper in these images is outdated, using the same elbow type module for voxel removal was not feasible as the rotation point drives the voxel into the ground plane as it swings, usually resulting in the gripper detaching. A different hinging module was designed, similar to that in the original BILL-E instead. Figure 2-4 additionally shows renders of hypothetical larger versions of BILL-E more able to take either larger steps or carry more voxels at once. In this work, it is still unclear how to scale the wrist style rotational joint for a larger robot.

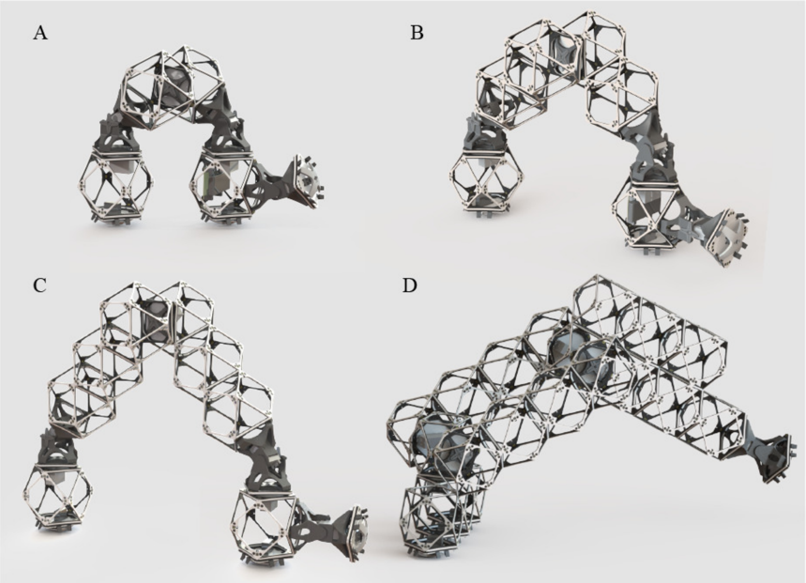

Figure 2-4: CAD renders of potential hierarchical reconfigurations of magnetic BILL-E. Image credit: C. Cameron.

However, the functional voxels used in this system were adaptable to other electromechanical systems, such as a reconfigurable drone, which was assembled by hand in approximately twenty minutes using rivets instead of magnetic joints. The speed of assembly is enabled by the simultaneous creation of the wiring harness and mechanical system with the placement of each voxel and beats other approaches to rapid on-demand fabrication of drone systems, which typically employ 3D printing [33].

2.1.3 Modular Robot Connectors

The field of modular robotics presents us with many options for connector types with a range of mechanical strengths. At one end of the spectrum of connectivity is the Kilobot system, which is a 2D reconfigurable swarm in which the individual robots can communicate wirelessly but cannot make mechanical connections [34]. At the other end of the mechanical strength spectrum (for systems meant to be reconfigurable) are robotic systems that employ active latching, such as the motor driven latching in Roombots [23] or ATRON [35]. The active latching schemes add significant weight and complexity overhead to the module design, especially if the actuation is done via motor.

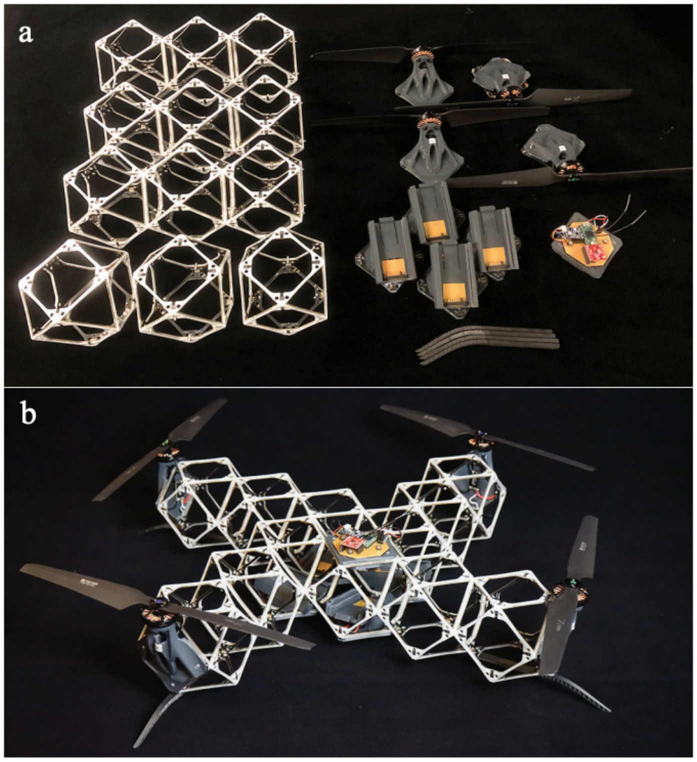

Figure 2-5: A reconfigurable quadcopter built from functional voxel. A) All the component parts. B) The assembled "Voxelcopter". Image from [33]

Between these two extremes, we have magnetic connectors and phase-change driven connectors. Magnetic connectors may be passive, such as in magnetic BILL-E [6] and M-Blocks [24], or it may be driven, using e.g. electropermanent magnets, as in Robot Pebbles [36]. Because the one of the primary limiting factors in the magnetic BILL-E system, magnet-based connectors were not used. Phase-change connectors rely on driving a phase change in a material at the connection point, such as by freezing water in [37] or by reflowing solder in the soldercubes system [1] or by melting a polymer in [25].

The solder-reflow process presented in [1] and demonstrated in [28] represents a high strength, light weight, reversible connection mechanism able to simultaneously make a mechanical and electrical connection. This is everything that I want for this robotic system. The system presented in [1] uses Field's alloy, which melts at 62° C and 10 Ω resistors on the opposite side to reflow it. Their system has a tensile strength of 173 ± 46 N [1]. A connector from the Soldercubes system is picture in figure 2-6.

2.2 Functional Voxels

Prior CBA research has established a library of voxels able to achieve diverse mechanical material properties. I now add to this library a "functional voxel" which is able to route electrical power and signals in addition to mechanical forces. [6] and [33] demonstrate a first iteration of the functional voxel, which consists of PCB faces laminated to acetal. Intra-voxel connections are made using a finger joint to index voxel connections and a right-angle soldeered connection to form the electrical and mechanical connection. Inter-voxel connections are made using a spring-loaded connector (Molex 0788641001) for the electrical connection and magnets for the mechanical connection. The connectors and magnets are arranged such that the voxels are orientation agnostic, and transmit three electrical lines (power, ground, and one data line). To improve the mechanical performance of the inter-voxel connection, the magnets may be substituted for a more robust connector, such as rivets, as done in [33].

This thesis uses the prior functional voxel system as a starting point with the following

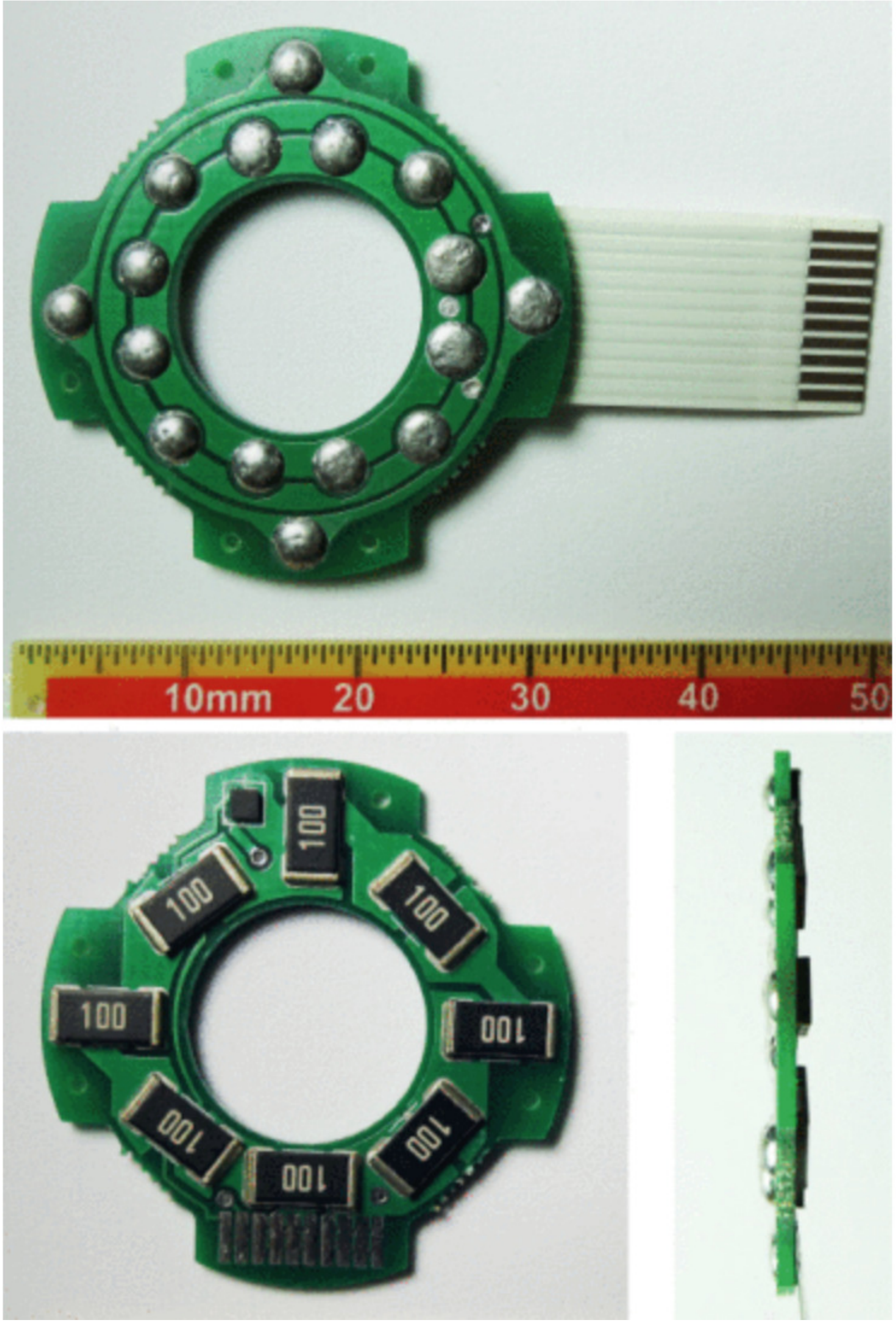

Figure 2-6: A face from the soldercubes system. Image from [1].

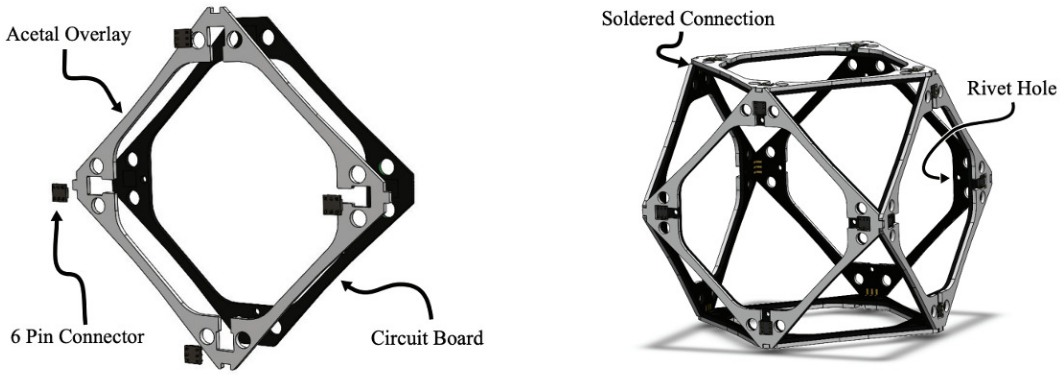

Figure 2-7: Functional voxel design from magnetic BILL-E. Images from [33]

changes:

- The addition of a second data line to enable two-way communications
- A more mechanically stable (though still robotically assemble-able) joint between voxels
- Optional integration of microcontrollers directly onto the voxel frame

The functional voxel used in this thesis consists of six PCB faces assembled into a cubcot geometry. The mechanical joint within a voxel (as in the prior functional voxel) consists of an epoxied finger joint, while the electrical ones are soldered. Between voxels, the electrical and mechanical connection is made using a reversible low-melt solder connection, based on the connector previously described in [28]. Pads for power, ground, and two data lines are exposed, as shown in figure 2-8.

The size of the functional voxel face is additionally reduced, such that the bounding box of the assembled cuboct has side length 65 mm. This is done to reduce the moment arms of the assembled robot, to improve the spatial resolution of assembled voxel structures, and to facilitate faster and cheaper prototyping. Unlike the voxels from [33], these functional voxels are not orientation agnostic. For the reversible solder connections to align, the voxels can only be assembled in a given orientation. To enforce this orientation, spaces for magnets are added, such that only the correct orientation results in an attractive force between all the magnets. The magnets additionally help facilitate fine alignment of the functional voxels,

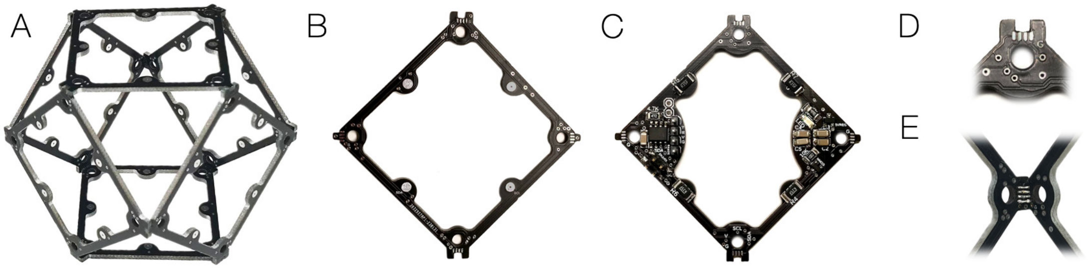

Figure 2-8: A) A functional voxel; B) Functional voxel face; C) ATtiny412 voxel face; D) Close-up view of intra-voxel connection point; E) Close-up view of a soldered intra-voxel joint.

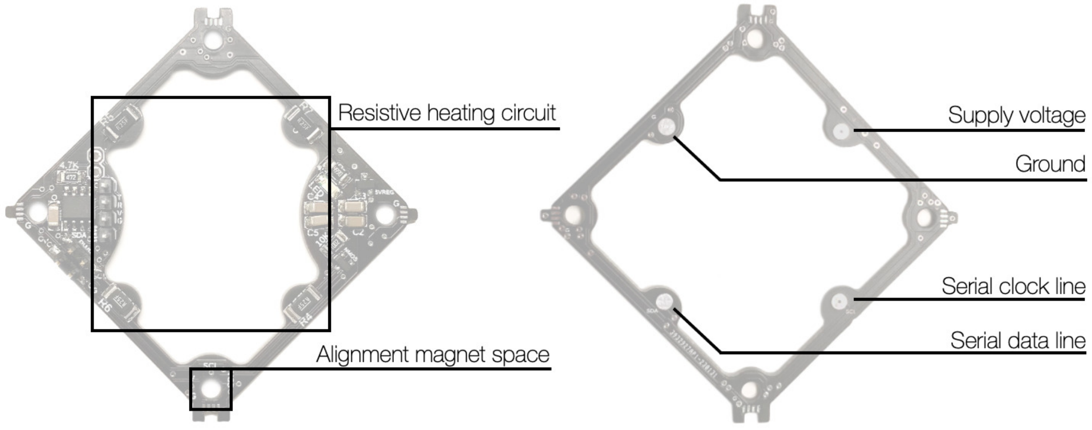

Figure 2-9: Inter-voxel connection system. (Left) ATtiny412 face with resistive heating circuit and alignment magnet spaces outlined. (Right) Basic functional voxel frame with inter-voxel connection points labelled.

which lowers the necessary spatial resolution of the assembly robot.

To form a connection between voxels, a voxel face containing a microcontroller with a resistive heating circuit must make contact with the basic functional voxel face frame. The microcontroller board, such as the ATtiny412 based face shown in figure 2-9, switches on a resistive heating circuit to reflow low melt solder— Chipquick Indium (IN52/SN48) solder— on the opposite side. Unlike [28], the Chipquick solder does not require the multistep process of priming and rinsing that Field's alloy does, simplifying the construction process. Additionally, I only reflow the solder from one side of the connection, which reduces the amount of electrical components at the cost of increasing the reflow time. The ATtiny412 board is additionally designed with a PWM output and an analog input, enabling its use for a variety of functions, such as hobby servo control, ESC control, or for a sensor input. In the context of this thesis, this board is primarily used to control hobby servos with the potentiometer line broken, as actuators in the assembly robot.

An additional two functional voxel face types were designed: an ESP32 face and an ATTiny1624 face (see figure 2-10). The ESP32 face enables Wi-Fi communication with a central computer (or with other ESP32 functional voxels) and acts as the primary controller on an i2c bus. This means that any robot (or other functional voxel construct) must include at least one ESP32, which disseminates instructions to the other microcontrollers on the lattice. The ESP32 specifically is built around the ESP32-WROOM-32 microcontroller and includes circuitry for level-shifting the i2c lines, power-cycling the attached modules for bus recovery, and additional breakout pins for adding extra features to the board. The ATTiny1624 board is designed for closed loop brushed DC motor control. It includes an A4953 H-bridge for current control, inputs for a quadrature encoder on the motor shaft, a Hall effect sensor for global motor zeroing, and a transistor switch for resistive heating solder reflow. The motor inputs and encoder outputs are connected via soldered wires to the board. In practice, the global zeroing with the Hall effect sensor was not implemented.

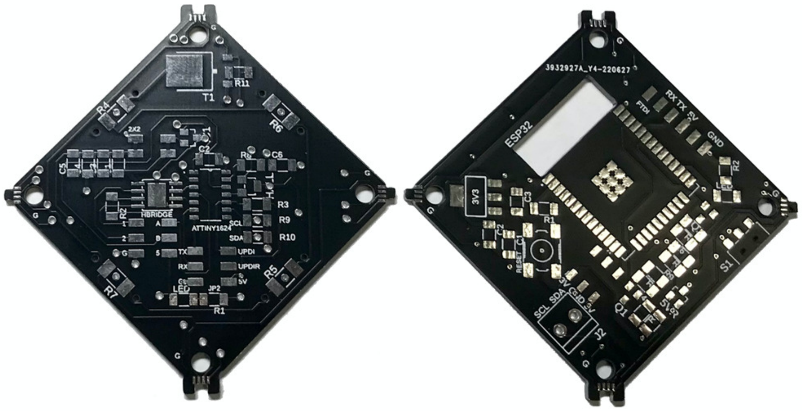

Figure 2-10: (Left) ATtiny1624 functional voxel face. (Right) ESP32 functional voxel face.

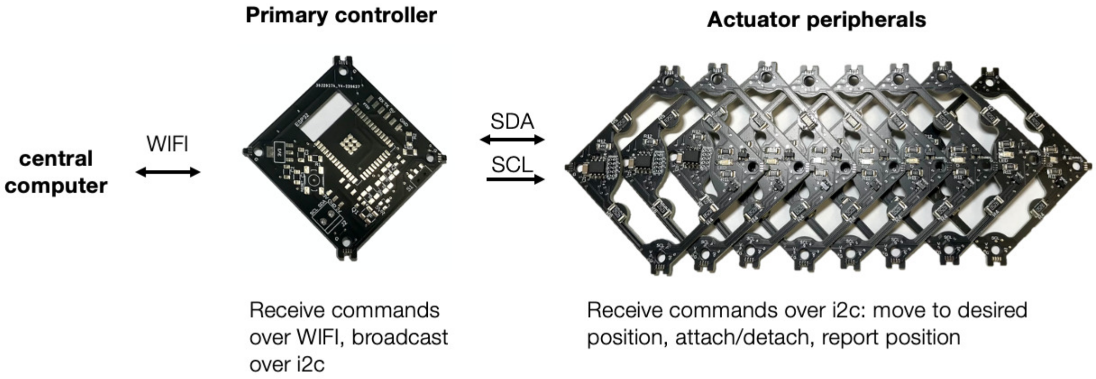

Figure 2-11: Diagram depicting the communications structure between voxel faces.

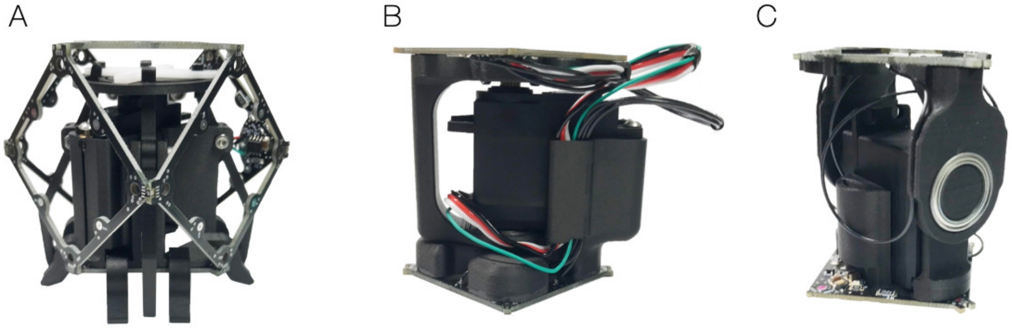

Figure 2-12: Robotic module types. A) Gripper module; B) Wrist type joint; C) Elbow type joint.

2.3 Robotic Modules

Using this set of faces, we can build robotic modules by adding actuation. The robotic assembler makes use of three types of modules: a gripper, a "wrist" type joint, and an "elbow" type joint. The gripper allows the robot to attach to voxels, enabling it to locomote or to manipulate voxels. This thesis primarily explores using a base 6 degree of freedom configuration robot, consisting of grippers at either end, two wrist type joints, and four elbow type joints. See figure 2-12 for the module types and figure 2-16 for two 6DoF robot configurations.

To select an appropriate motor for the modules, I consider the required torque for the worst-case loading scenario, which occurs when the module directly above the gripper cantilevers the fully extended robot gripping an extra module against gravity. I approximate each of the modules as equal mass, with 50g of mass contributed by the various hardware before the addition of a motor (see 2.1 for a breakdown of the actual masses of the robot components). I then approximate the minimum required torque by calculating the torque applied by gravity on the mass of extended arm at its midpoint. In this loading scenario, I use 7 modules (6 extended in cantilever, with an additional module held at the end). This minimum torque approximation is plotted in figure 2-13 in addition to the stall torques and masses of a selection of Pololu motors as well as a few low KV BLDC motors. Figure 2-13

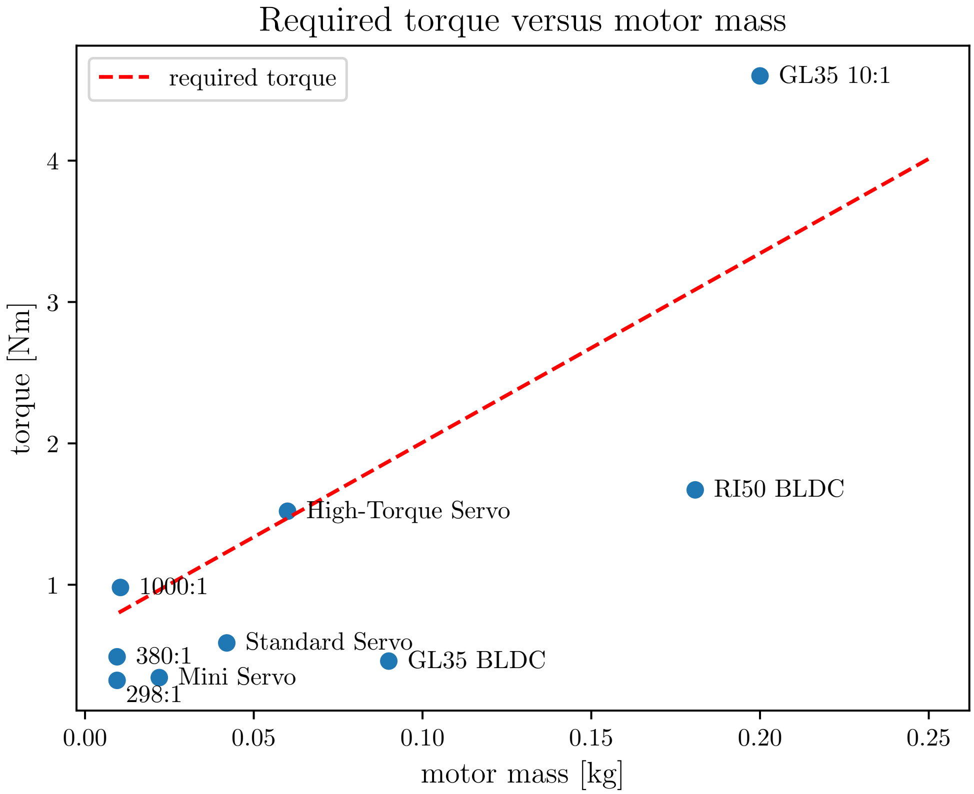

Figure 2-13: Required torque is plotted for the worst case loading scenario (dashed red line) as it varies versus added motor mass. The stall torques of a selection of Pololu motors are plotted (blue). The ratio labels correspond to the micro metal gearmotor family, at that gear ratio. The servos are FEETECH brand with the following product numbers. High-torque servo: FS5115M-FB, Standard servo: FS5106B-FB, Mini servo: FT1117M-FB.

plots three brushed DC gearmotors from Pololu's micro-metal gear motor family, as well as three servos their FEETECH line of servos with feedback lines broken out. Two low-KV BLDC motors from CubeMars— intended for use as gimbal motors or for robotic joints— are also plotted (the GL35 and RI50). A nominal value for the GL35 with reduction is also plotted. This value is based on average masses reported for planetary gearboxes made for Nema 11 stepper motors in a range from 3:1 to 100:1, which are appropriately sized and torque rated for this application.

Three of the motors are above the minimum requirement— the high-torque servo (FEETECH FS5115M-FB), the 1000:1 micro metal gearmotor, and the 10:1 geared GL35 BLDC motor. For convenience, the servo option is used— it is cheaper than the other options and additionally features the feedback on its potentiometer broken out. Of course, the worst case loading scenario is not necessarily something that the robot will ever need to do, so some of the other motor options may also function. In an earlier version of the hardware (further discussed later in the section on early prototypes), a gearmotor version was implemented using the 380:1 motor. In future work, I plan on further exploring using a BLDC motor system, with some reduction. As the plot shows, without some reduction, the robot cannot support the worst case loading scenario— however, with relatively low reduction, the maximum torque can be boosted without substantially increasing the motor weight. The plotted notional 10:1 reduction represents a worst case for the added weight— because the system does not require large reduction, a more lightweight reduction system is likely feasible. The interest in eventually switching to quasi-direct drive system stems from plans to build out the robotic module system to support robot geometries other than just the assemblers, which would benefit more from actuation capable of achieving more dynamic motion (this is further discussed in the Conclusion chapter).

In an effort to keep the robot cost low (after all, we want these to be swarms!), I primarily considered lower-end hobby type servos— all the servos plotted are in the 10-25USD range. So, the actuation for the robot in total comes to 163.60 USD (a full bill of materials is available in table 2.1). In contrast, the BILL-E version in [17] uses a set of four Hitec HS-7950TH servos (120USD each) and five Hitec 5085MG servos (40 USD each) for a total of 680USD spent on actuators. Similarly, magnetic BILL-E in [6] uses six Hitec D950TW servos (130USD each) and three Hitec HS-5087MH servos (35USD each) for a total of 885USD spent on actuators. If cost is of no concern, the higher torque ratings and greater programmability of these servos may be desirable, though I have been happy with the performance of the FEETECH servos.

The wrist and elbow modules both use the FEETECH High-Torque Servo FS5115M-FB from Pololu that has the feedback line broken out. Each of these modules uses one ATtiny412 face and one basic frame face. FDM 3D printed parts interface between the boards and motors. The 3D printed parts are made from either PLA filament printed on a Prusa MK3S or Onyx filament (nylon with carbon fiber) on a Markforged Mark 2. Printing with PLA on the Prusa is a cheaper and faster process, though the resulting parts are slightly heavier with worse dimensional tolerance. The Onyx parts are higher performance at about 4x the cost, which results in approximately 40USD differential over the whole robot. See 2.1 for the complete BOM. Additional hardware per module includes one bearing, assorted screws and heat set inserts for affixing the motor to the 3D printed parts, and wiring to go between faces within a module.

The gripper module uses the FEETECH Mini Servo FT1117M-FB from Pololu. A smaller servo is used here as the required torque is much lower. This module requires one ATtiny412 board for controlling the gripper. It can be outfitted with additional microcontroller faces to form the complete cuboct structure. If additional microcontrollers are not needed, the basic frame face may be used. In the 6DoF robot, the first gripper contains one ATtiny412 board, one ESP32 board (which acts as the primary controller for the entire robot), and four basic frame faces. The other gripper omits the ESP32 board and uses five basic frame faces. The design of the gripper is based on that of a gripper designed by Zach Fredin for the CBA DICE project (see [38] for reference), which uses a flexure to open and close a set of gripping legs around the underlying substrate. The adapted version of this design used in this thesis is shown in figure 2-14. Aside from adapting the design for the relevant dimensions for this application, the design is simplified to eliminate the bearings to keep the weight low, and all parts are FDM 3D printed to keep fabrication consistent

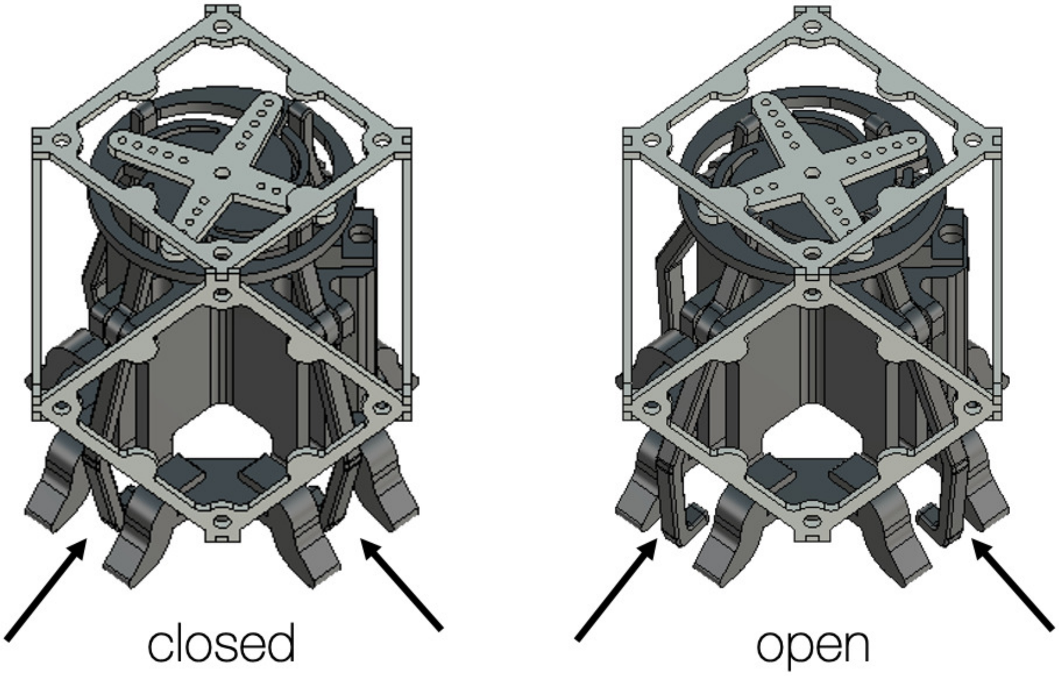

Figure 2-14: CAD render of open and closed gripper positions. Left: closed. Right: open.

with other modules.

Modules are assembled into robots by stacking. The wrist and elbow type modules only accept connections from opposing ends (as in a ”chain” type modular robot), while the grippers can be configured to accept connections from any of the faces aside from the gripping surface. Though the robots explored in this thesis only make one connection to the grippers, future versions of the robot could add additional features by expanding from the grippers. Specifically, a separate gripping arm (as in [17]) could be added to improve the payload carrying ability of the robot.

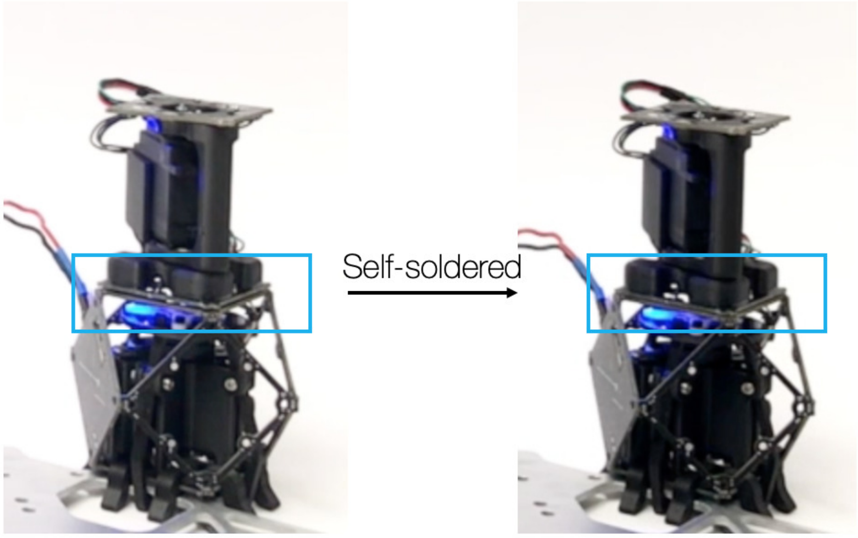

Figure 2-15: Wrist module before and after self-soldering.

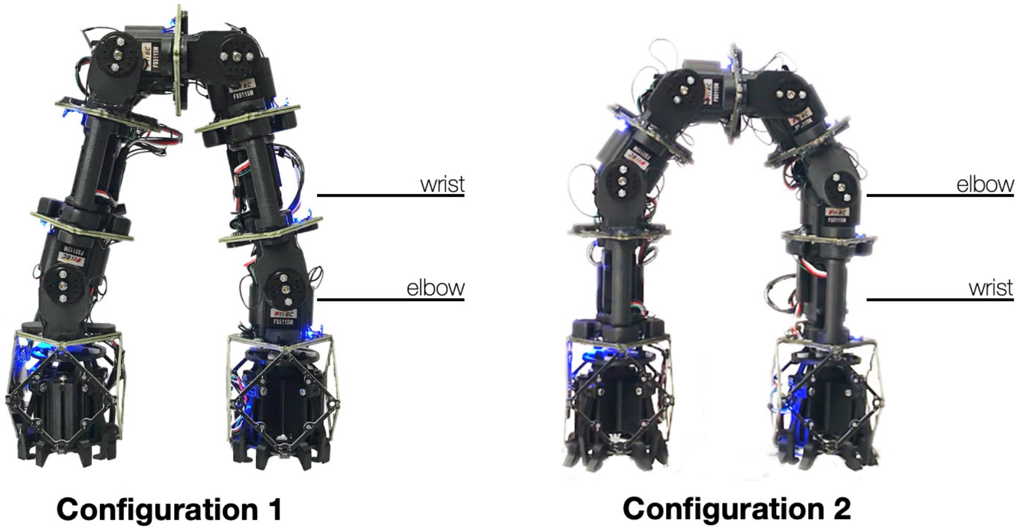

Figure 2-16: Two different 6DoF robot configurations. (Left) Robot modules are stacked as follows: gripper, elbow, wrist, elbow, elbow, wrist, elbow, gripper. (Right) Robot modules are stacked as follows: gripper, wrist, elbow, elbow, elbow, elbow, wrist, gripper. The configurations enable different accessible spaces.

Item	Mass [g]	Cost [USD]	Amount	Vendor	Notes
FEETECH High-Torque Servo	60	22.95	6	Pololu	15.5 kg-cm stall torque at 6V
FEETECH Mini Servo	22	12.95	2	Pololu	3.5 kg-cm stall torque at 6V
ATtiny412 PCB	4.5	3.22	8	JLCPCB, Digikey	board cost: 0.72 @ 50 boards, electronics cost $<$ 2.50.
ESP32 PCB	10.9	4.90	1	JLCPCB, Digikey	board cost: 0.40 @ 5, electronics cost: $<$ 4.50
Basic frame PCB	2.2	0.356	9	JLCPCB	board cost: 0.356 @ 50
3DP elbow parts	29.19 - 21.60	1.15-5.62	4	N/A	ranges are for PLA to Onyx
3DP wrist parts	35.70 - 27.14	1.47-8.28	2	N/A	ranges are for PLA to Onyx
3DP gripper parts	43.15 - 34.35	1.53-8.24	2	N/A	ranges are for PLA to Onyx
Ball bearings	3.4	1.00	6	Amazon	uxcell 6702-2RS 15mm x 21mm x 4mm
Additional hardware	$\approx$ 3.5 - 6.5	$\leq$ 1.00	8	N/A	screws, wires, etc.
Total	797.56 to 724.08	222.06 to 266.98	1	N/A	ranges: PLA to Onyx

Table 2.1: Bill of materials for the complete 6DoF robot.

2.4 Robot Performance

2.4.1 Basic voxel assembly

As with prior BILL-E systems, the assembly robot requires an initial seed of voxels from which it can build out. So that the system might be more portable to environments beyond the lab (or at least beyond the optical table), a build plate is made (see figure 2-17). The build plate holds the seed voxels, voxel feed locations, and a few extra features that help with robotic assembly of more robots. The robot starts on the seed voxels, picks up voxels from the feed locations (which contain indexing features such that the manual placement of voxels is relatively consistent), and can then place the voxels off the build plate, extending the surface it can stand on. The build plate needs to be rigidly affixed to a ”ground” surface, otherwise robot motions may result in sliding or even tipping of the build plate. Even with this constraint, this system offers greater flexibility in build locations than the prior system, without requiring a large seed voxel plane.

Basic assembly sequences have been demonstrated with the robotic system. 1-D assembly using robot configuration 2 is shown in figures 2-18 and 2-19. This sequence demonstrates some of the foundational moves that the robot uses. Both configurations pick up voxels from the feed by flipping the length of their body over (i.e., the robot goes from standing to vertical to back down). I do this (instead of simply rotating 180° about one of the wrists) because the servos used only have a controllable range of 180° but are mechanically able to sweep a larger range, which makes it practically difficult to guarantee that the 0 of the servo is aligned with the robot’s zero. In contrast, the flipping motion is much less sensitive to servo motion limits, making it more practical to implement. Freeze frames from this sequence are shown in figure 2-18.

Following from 2-18, figure 2-19 continues the assembly process of the 1D beam. Once the desired voxel placement location is beyond the reach of the robot, it steps forward onto the assembled voxels, which enables it to reach forward another voxel length (frames 2 through 6). At this point, it can no longer far enough forward to place another voxel

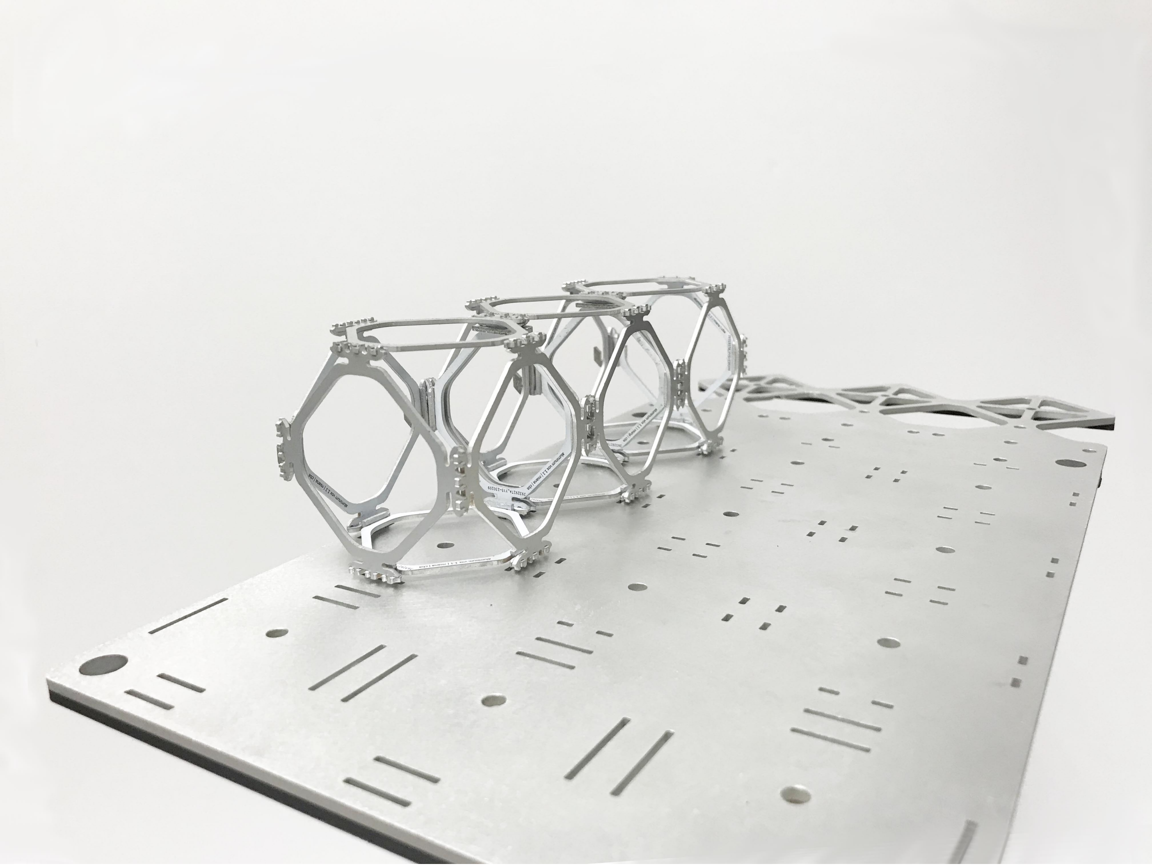

Figure 2-17: Build plate for the assembly robot. The robot starts with three voxels affixed to the build plate and can then extend the structure off the plate. The plate is outfitted with voxel feed locations on the left and additional voxel face surfaces at the right to enable robotic assembly of another robot. This build plate has extra rows that could be used if using additional feed locations.

directly, so it places the new voxel at a diagonal. It then steps forward, picks back up the diagonal voxel, and places it. Given that the current robotic system does not have an additional gripper, this represents a functional, though inefficient, method of carrying voxels forward. An alternate strategy using only the currently developed robotic system would be to have the robots pass voxels between each other, like a bucket brigade, as described in [32].

The repeatability of this system is evaluated by having a robot walk back and forth on a voxel structure. Specifically, the configuration 2 style robot walked back and forth over a five voxel beam (to include voxels not anchored to the build plate), using three different step types. The failure rate was then recorded. The robot failed 7.1% of the time. Of these failures, 3/4 consisted of misalignment between the foot and the lattice— the step for no clear reason just failed— which is a recoverable error. With a better feedback system, this type of error could be detected and fixed. Currently, in semi-autonomous operation, keeping a human checking the success of steps and running a "fix step" convenience routine acts as a stop gap measure to help the robot continue building even after failure. The remaining 1/4 of the time consisted of failures that the system cannot recover from without significant intervention, or a major addition to its feedback system. This type of error is caused by the gripper foot getting caught on the voxel despite nominal disengagement. This usually results in the robot browning out, requiring a reboot and re-positioning to continue running. Another few careful iterations of the gripper foot geometry will likely eliminate this issue.

Error in the robotic system largely stems from compliance in the system. The two primary types are shown in figure 2-21. In the first, the underlying voxel structure significantly flexes. The image shows a Delrin voxel structure, and this compliance contributed to the move to aluminum voxels. A similar issue can occur on the robot side, where because of the designed (and un-designed) compliance in the robot, certain maneuvers become difficult to accomplish. In the image, the angular difference between the robot's foot and the voxel substrate is highlighted. In this example, the robot is trying to push down on the voxel in its other gripper foot, but is instead hinging about its back foot. This type of issue may be resolved through more careful iterations and optimizations of the robot hardware, as well.

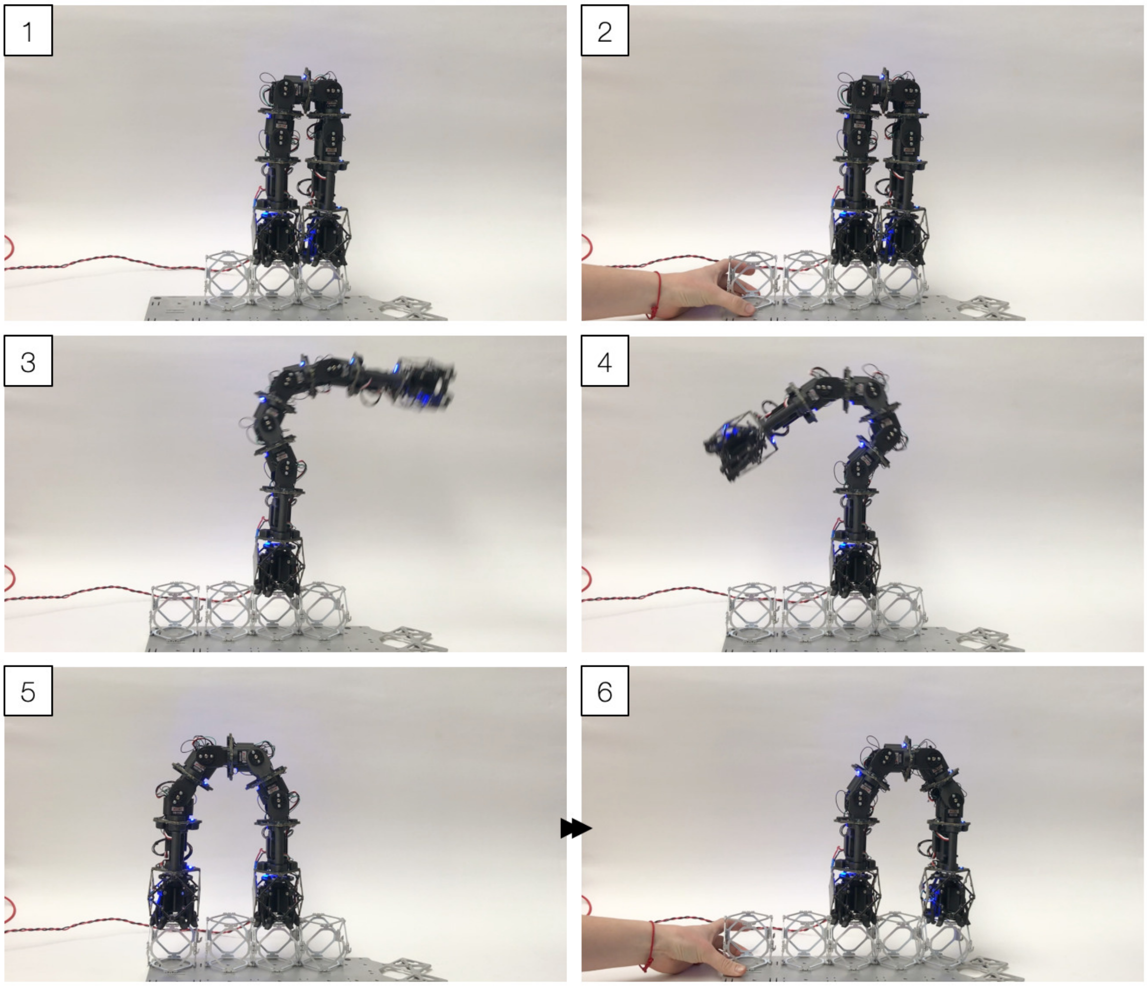

Figure 2-18: Freeze frames from a basic assembly maneuver. 1) Starting robot configuration on the build plate. 2) New voxel is manually added to the feed location. 3-4) Robot flips over to reach the feed location. 5) Robot grips new voxel. 6) Robot flips back to place the voxel and a new voxel is fed in. The sequence is continued in figure 2-19.

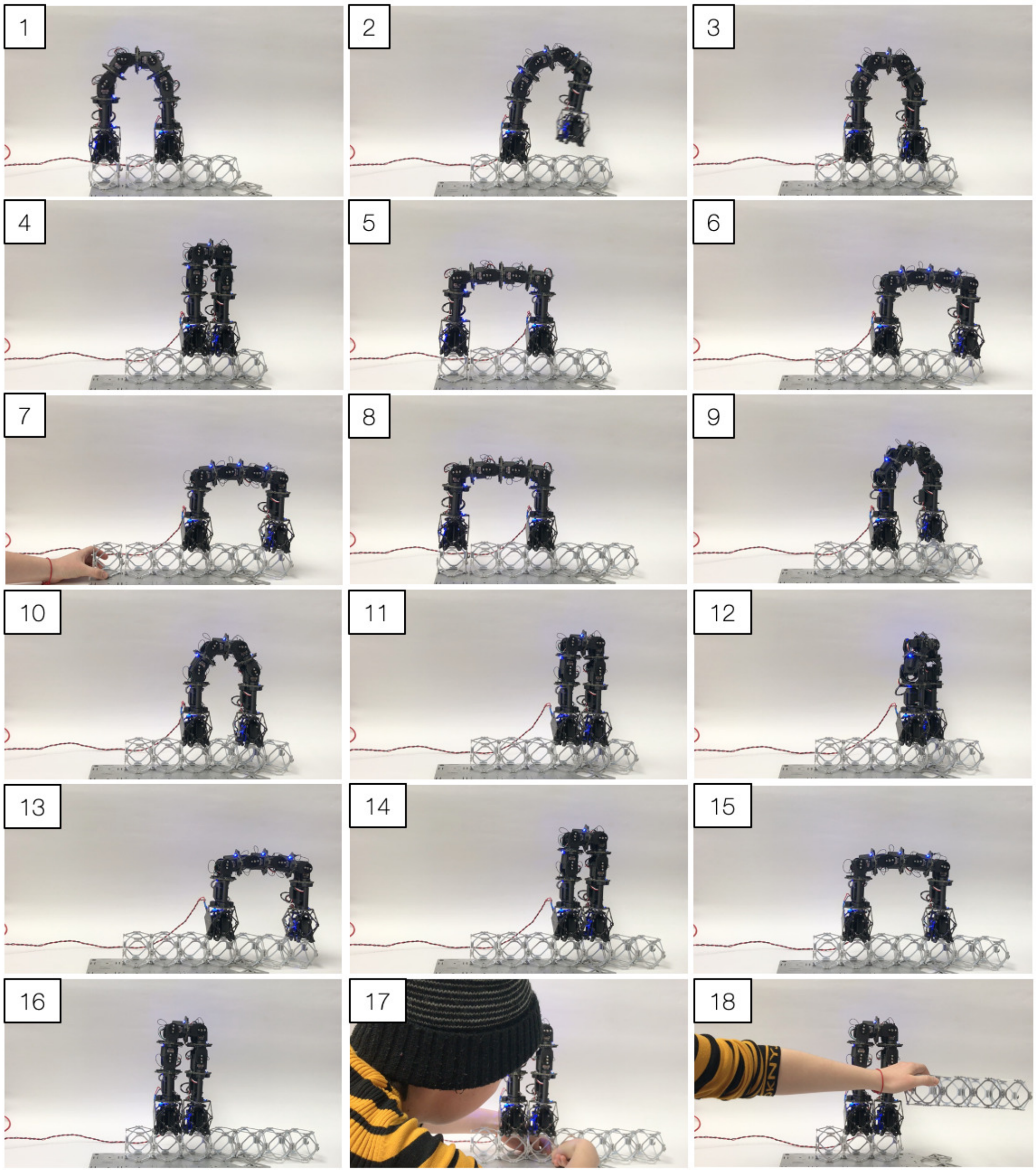

Figure 2-19: Continued freeze frames from basic assembly sequence. 1) The picks up the next voxel and 2) places the voxel and steps back. 3) Robot re-grips substrate and 4) steps forward. 5) From the new anchor, the robot grabs the next voxel and 6) places it. 7-8) A new voxel is added and the robot grips it. 9) The robot temporarily places the new voxel at the diagonal. 10) The robot steps back from the diagonal and 11) steps forward. 12) The robot re-grips the target voxel. 13) The robot moves the voxel to the front of the the voxel line. 14-16) The robot returns to its initial configuration. 17-18) The clips holding the first voxel are disengaged and the beam is removed.

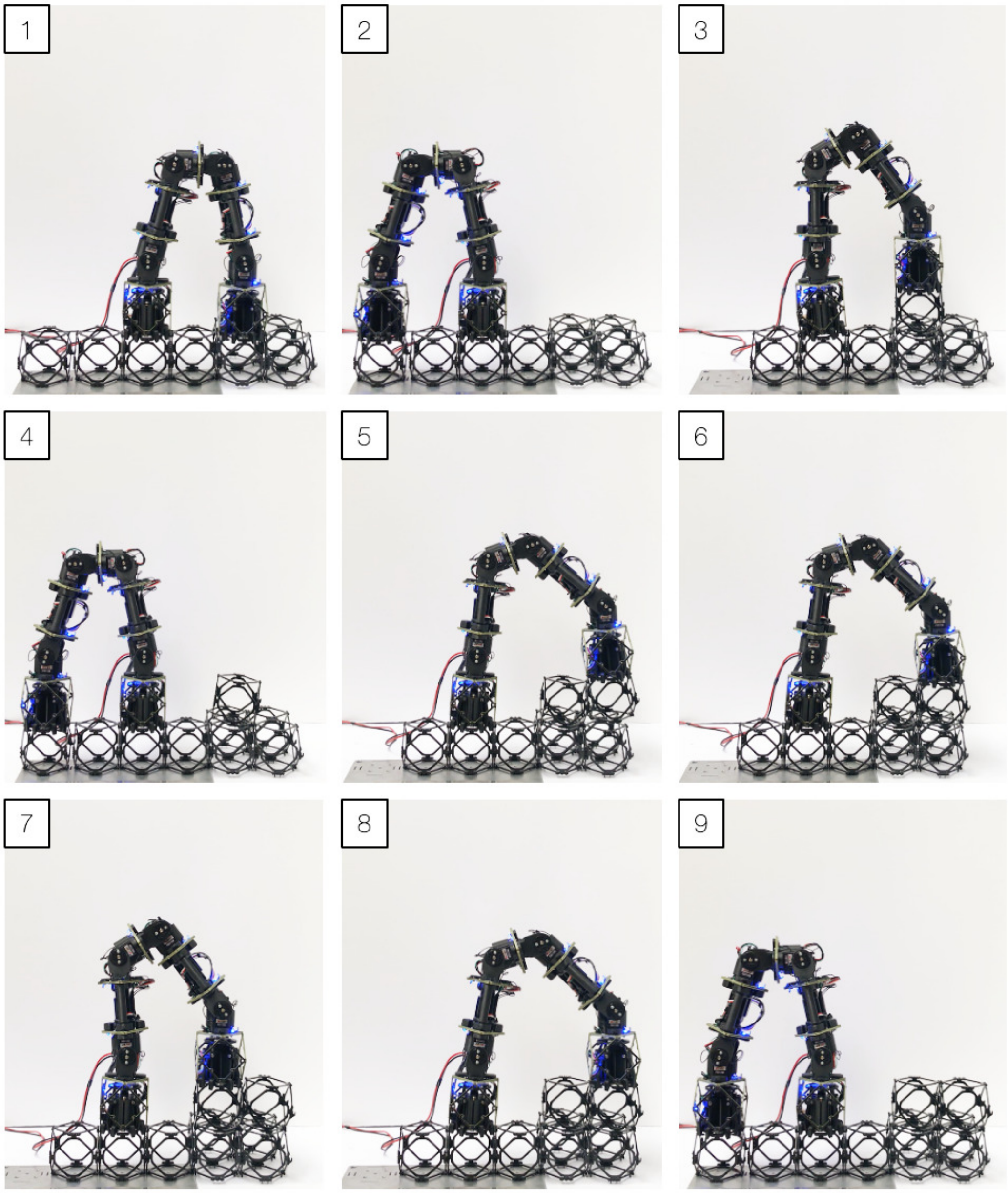

Figure 2-20: Voxel self-correction in a small 3D structure. 1) Initial configuration. 2) Pickup voxel. 3) Place voxel. 4) Robot picks up the next voxel. Note that the previously placed voxel is at an angle— it was not fully installed by the robot and was knocked off position. 5) The robot places the next voxel— there is some interference between it and the prior voxel, but both are sufficiently close to their correct positioning that when the robot releases the voxel in 6), the voxel falls over the correct position and helps to push the first voxel back. 7) To fully install the voxel, the robot steps (stomps) onto the first voxel. 8) It does the same to the next voxel. 9) The robot returns to get the next voxel and the first two voxels are installed.

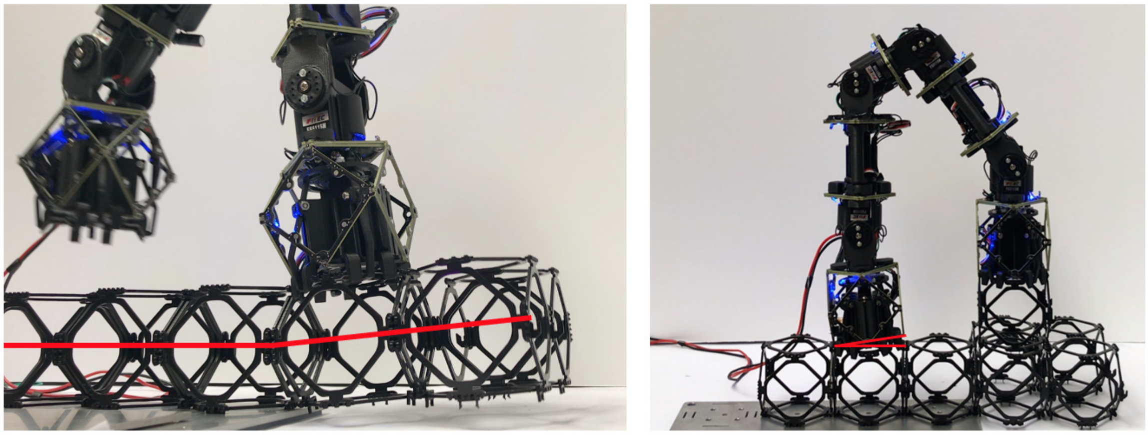

Figure 2-21: Examples of compliance in the assembly system. (Left) Underlying Delrin voxel structure flexes under the weight of the robot as it walks. (Right) Robot's gripper foot is flexed at an angle to the underlying voxel structure.

2.4.2 Robot assembles robot

The robotic system presented here is capable of self-configuration. The assembly robot assembles a second robot as a demonstration of its ability to self-replicate. Freeze frames from the video of this process are shown in figures 2-22 and 2-23. Note that this system is not yet fully autonomous— new modules are fed into the build plate by hand and the success of each module attachment is manually verified.

The configuration 1 version of the 6DoF robot starts in its initial stance on a build plate with a three voxel base (frame 1). The build plate additionally features a voxel-top surface at two voxel distances away from the third build plate voxel and one level lower. Modules are fed in at the voxel feed location. Unlike with voxel assembly, there are no alignment features on the build plate that match the modules. Because the outline of the modules is slightly thinner than that of the construction voxels, and because the modules are much heavier, the higher inconsistency in module feed location does not appear to practically matter, as the alignment features and flexural gripping mechanism on the gripper are able to account for fairly large displacements. (This does not tend to work out for the construction voxels largely because the robot is wont to make contact with the modules with some force, which bounces the construction voxels in a way the modules do not, because of their heavier

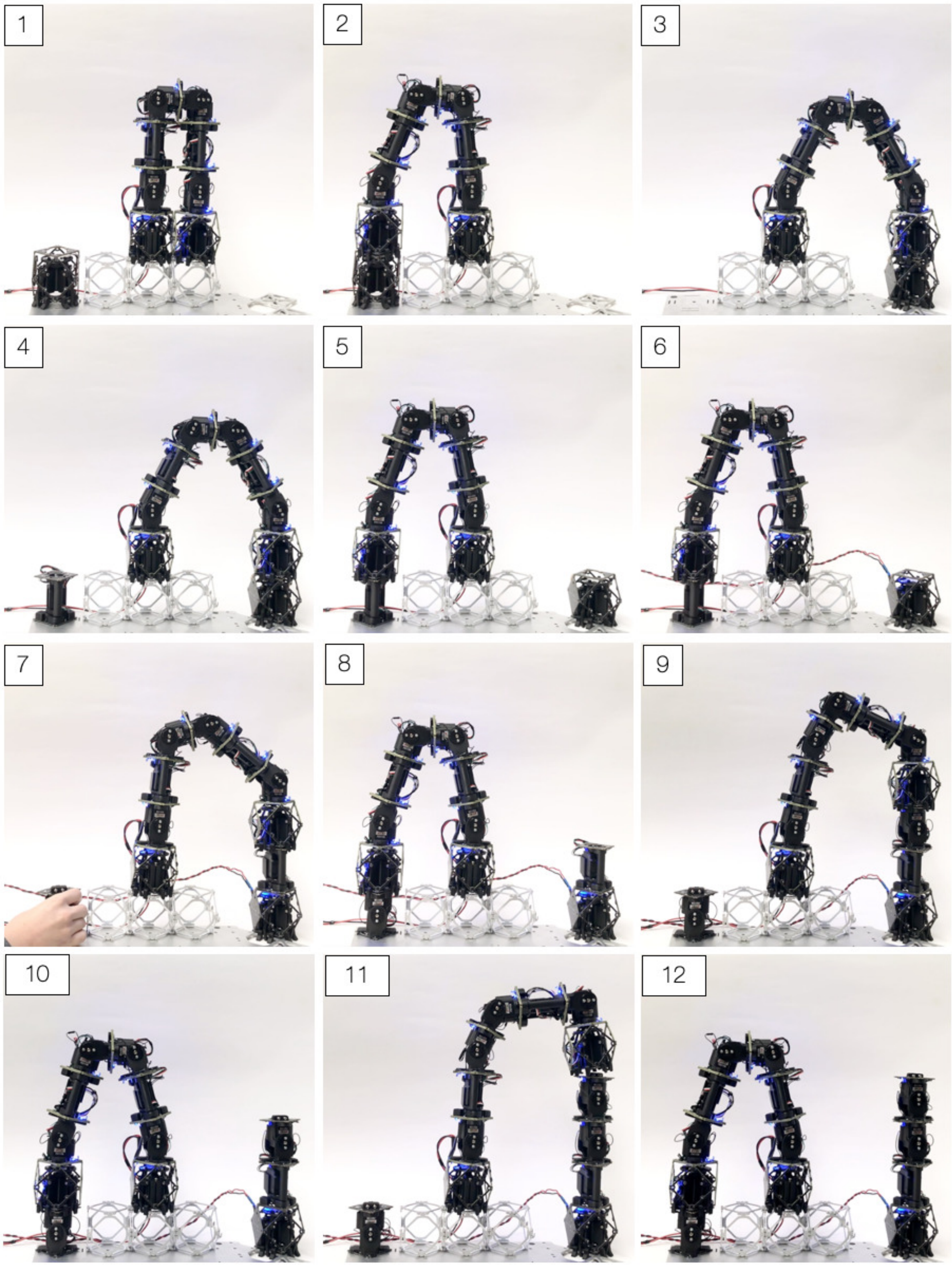

Figure 2-22: First half of freeze frames from the robotic assembly of a second robot.

Figure 2-23: Second half of freeze frames from the robotic assembly of a second robot.

weight.)

The robot picks up the first gripper of the next robot, which contains its ESP32 primary face (frame 2). This module is placed onto the lower voxel top surface of the build plate (frame 3). The next module (a wrist) is delivered to the feed location, and robot 1 swings back to it (frames 4-5). Power is delivered manually, via wires to a DC power supply (frame 6). The first module of the second robot then grips the build plate. Robot 1 drops the wrist module onto the gripper (frame 7). Robot 2 self-solders the wrist module. It takes approximately three minutes of heating to get a well soldered connection. (Note that the solder usually flows at around the one-minute mark, but the opposing PCB pads will not be hot enough to form a reliable connection at this point!) Robot 1 then picks up and places the next module, an elbow (frames 8-9). The soldering process is repeated. The next module (another elbow) is then placed and attached (frames 10-12).

At this point, the second robot has reached a height that the first robot cannot address in the vertical orientation. So, the second robot bends its first two elbows such that the next attached platform is at a lower height. (frame 1 of the second half build sequence). Robot 1 places the elbow onto the new platform (frame 2). The elbow needs to be soldered before robot 1 retracts, or else the module may be knocked off— this position is at the limit of the robot's accessible space, and it cannot approach the voxel drop in a purely vertical direction. Once again, robot 2 is too tall for robot 1 to reach. Luckily, robot 2 has also reached a length where it can bend to reach modules placed on the build plate voxels. Once the next module is soldered, robot 1 returns to the next module and robot 2 bends such that it is out of the way for robot 1 to place the next module onto the build plate (frames 3-4). The next module (the final elbow) is placed onto the build plate voxel surface (frame 5). Note that this build plate voxel has an insulator over its top surface, preventing shorting. Additionally, magnets are installed underneath it such that they can provide some fine alignment of the module when placed— the functional voxel faces on the modules do not yet index in this orientation with the construction voxels. Robot 1 then goes to pick the next module (a wrist) while robot 2 bends back to solder the current module on (frames 6-7).

This process is then repeated for the next set of modules. Robot 1 places the second wrist module and robot 2 attaches it (frames 8-12). Robot 2 places the second gripper module and robot 2 attaches it (frame 11-14). The second robot is then complete! (frame 15)

As stated previously, do note that this system is not fully autonomous yet— automatic error checking and correction are not yet implemented in the hardware! In the video, the soldering dead time is cut out (approximately 6 minutes per module to heat and cool). Further, the second robot is removed and replaced on the build plate after each module attachment to check that the attachment and module work.

This process is additionally reversible. Figures 2-22 and 2-23 are from the second time that robot 1 assembled robot 2. It was not my intention to do this twice, but the first time trying to do the assembly went so poorly that I slowly came to realize I would have to do it a second time, with all the lessons learned. Freeze frames from this process are shown in figure 2-24. So, robot 2 was manually disassembled by sending the solder command to each module while applying a bit of tension— an autonomous version of this is not yet implemented— and then reassembled by robot.

The first attempt did not go well for myriad reasons, primarily that it was the first time I tried robotic assembly of a robot structure. Another confounding factor was that I decided that the second robot should use a different joint configuration (meaning that it would need new paths for motions) and that it should further use a re-structured software set up. Trying to deploy and debug the new system live proved overly difficult and time-consuming, and my phone died before the last two modules were placed, hence the jump in the last two modules.

More useful information about the limitations of the current module design I learned from doing this twice:

- The low melt solder highly benefits from a re-application of flux before being reflowed again— this means that in-the-field reconfiguration or the robotic system may be limited with the current design.

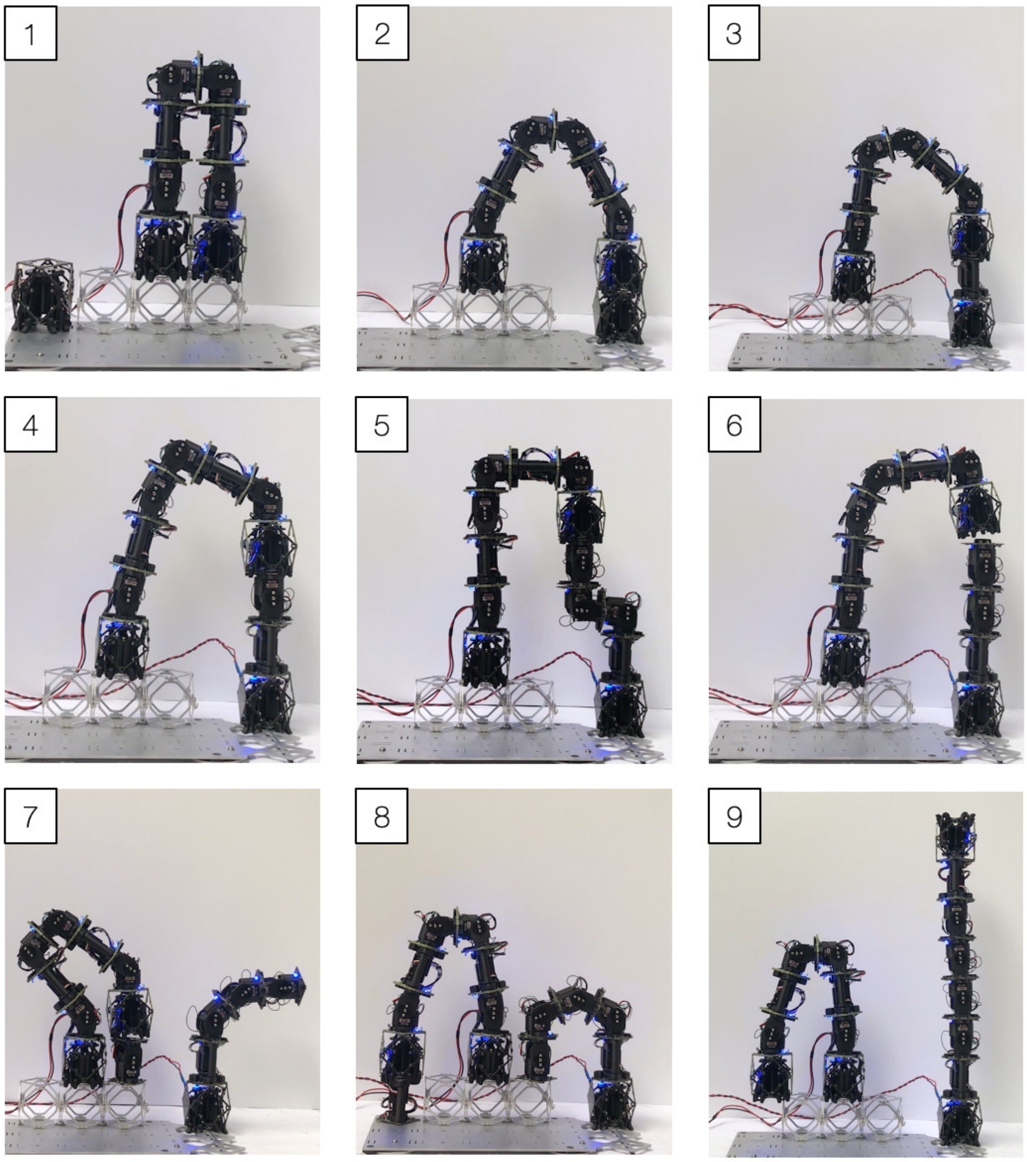

Figure 2-24: The initial assembly attempt.

- On an aluminum lattice, the solder joints are the weakest element of the entire system— if the robot collides with itself, the other robot, or the build plate lattice, it will break at the module joints. This is good because these are designed to be reversible joints, but also bad because it is highly inconvenient.
- The tops of the servos of the elbow modules protrude slightly from the 65x65x65mm bounding box and dock into an empty space on lower module— this indexing does a better job of fine alignment from a wider initial position range than the magnets do, and it may be practical to intentionally switch to using mechanical indexing instead of the magnets.
- We must be careful to let the low melt solder fully cool or else the assembly robot is liable to become a disassembly robot.

2.5 Control System

2.5.1 Single Robot

A central computer send commands over Wi-Fi to the ESP32 face, which then disseminates these over the I2C bus to the ATtiny faces. These commands include move to desired joint angles, attach/detach module, and report current joint angle. For multi-module moves (e.g. taking a step), the desired joint angles are generated by running an inverse kinematics routine in Drake and generating a trajectory. This trajectory is then sampled and sent to the ESP32, which streams the desired positions with a set time-delay. So that I can take into account control inputs, a workflow for generating optimized trajectories has also been developed. However, this work is not currently integrated in the hardware system as the optimization process requires some further adjustments.

Trajectories for the hardware system are generated by producing a library of key frame positions using a simulated version of the robot in PyDrake. The key frames are found using Drake's InverseKinematics method, which formulates solving an inverse kinematics problem as an optimization, which gives us greater flexibility in generating robot configurations without needing to specify the desired target location in absolute terms, or in generating good-enough configurations for positions the robot may not be able to actually reach. (I largely do not take advantage of these features for the hardware system). Because the robot is symmetric between the lower and upper four modules, flipping trajectories across this axis generates trajectories for motions in the opposite direction. In simulation, I anchor one gripper at the origin, and dictate desired initial and final positions as cartesian coordinates with desired orientation. I.e, of the form $[x, y, z, rot_x, rot_y, rot_z]$. The system then runs four IK problems, the first to find the starting configuration, then to find an offset position over the starting configuration, then to find an offset over the final configuration, and last to find the final configuration. After the initial configuration, each problem uses the robot configuration from the prior step as its initial guess, with a cost on the difference between the final position and initial position, which helps prevent the generation of technically correct though very disjointed trajectories. The offset positions are offset in the z-axis of the gripper, or perpendicular to the surface it is gripping. That is, if the robot is stepping on a horizontal plane, the offset will be vertical, while if it is approaching a vertical plane, the offset will be horizontal. Joint limits are applied to each of the joints as constraints on the optimization and the target final Cartesian position is given as a constraint, while the rotational orientation is permitted to have a bit of slack. I then interpolate these key frames using Drake's PiecewisePolynomial method. As an optional step, these trajectories can be simulated using a simulated PID controller within Drake, meant to behave similarly to the hardware— for complex moves, this can help check for self-collisions.

The simplified robot model used in Drake is shown in figure 2-25. Note that the model was put together when the hardware still utilized a wrist type module that more closely resembled a functional voxel with a rotating upper platform, so the model robot looks a bit different from the actual robot. The model was not updated with the approximate new geometry because the modules would still use the same bounding box for collisions.

This process runs fast enough (less than 100ms) to be run in the loop with the robot, though because the robot does repetitive moves in a pre-defined lattice space, the ability to formulate desired configurations on the fly does not offer much advantage. Further,

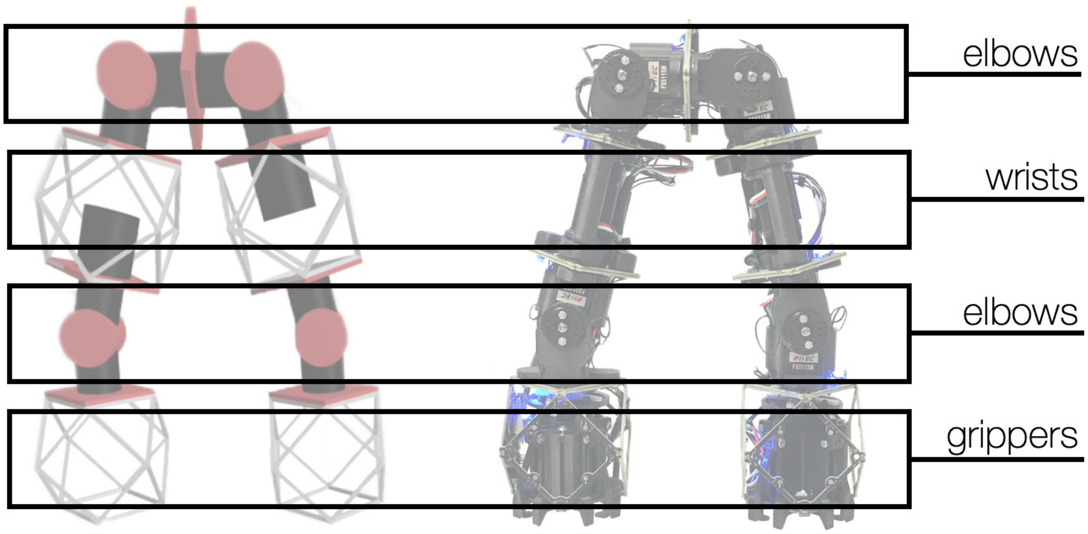

Figure 2-25: Simplified robot model in Drake versus actual robot. Corresponding components are labelled. Note that the wrist actuators are visualized based on an earlier version of the module that more closely resembled a functional voxel.

the discrepancies between the simulated and actual system are such that most of the generated trajectories need to be tuned. Primarily, the compliance of the underlying voxel structure, the compliance in the robot's grip, the backlash of the motors, and the slight weight imbalance contribute to error. So, instead, I store a library of tuned key frames and interpolate through these to send to the robot. A future step may be to re-work the simulation methodology such that it is easier to reconfigure the software as the hardware is reconfigured.

A workflow has also been established for generating optimized trajectories using Drake's direct collocation method [39]. The direct collocation method works by simultaneously optimizing the state and control trajectories at a series of knot points, enforcing the dynamics and initial conditions as constraints. From [39], this formulation is:

$$\min_{x[\cdot],u[\cdot]} \quad \ell_f(x[N]) + \sum_{n_0}^{N-1} h_n \ell(x[n], u[n])$$

$$x(\dot{t}_{c,n}) = f(x(t_{c,n}), u(t_{c,n})), \forall n \in [0, N-1]$$

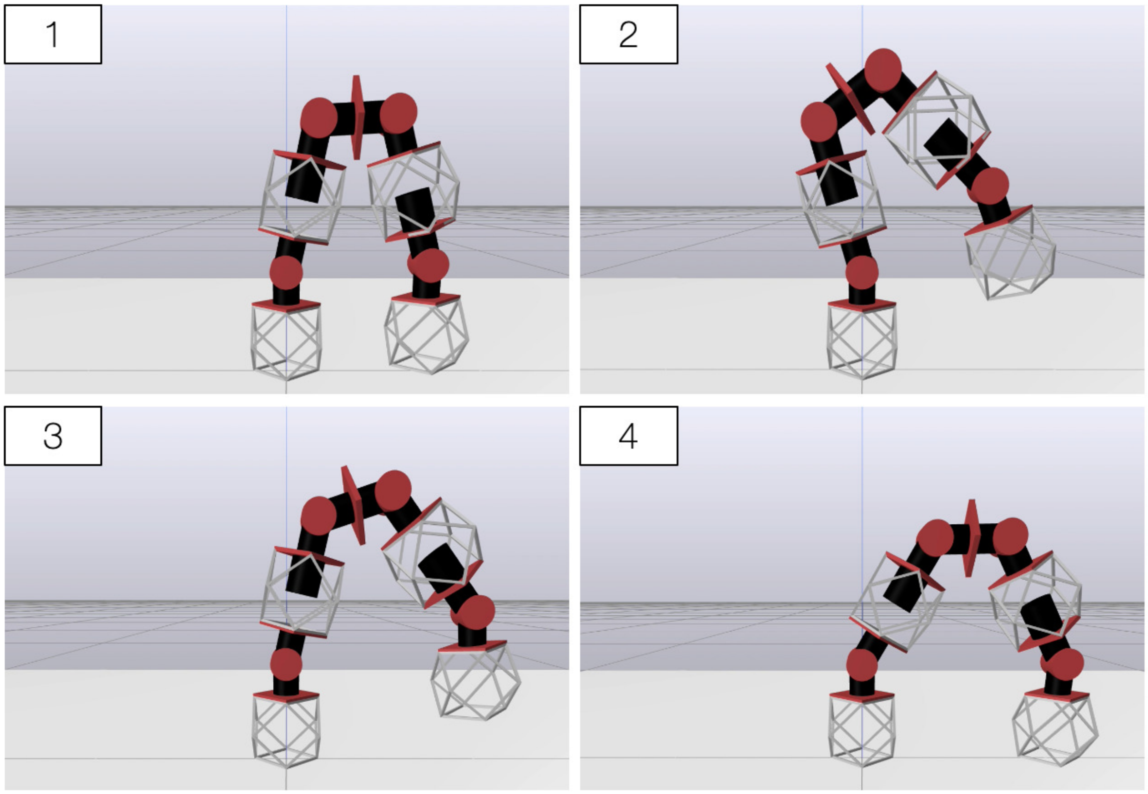

Figure 2-26: Example key frames from IK method for configuration 1 style robot. 1) Starting position; 2) Vertical offset above start position; 3) Vertical offset above end position; 4) End position.

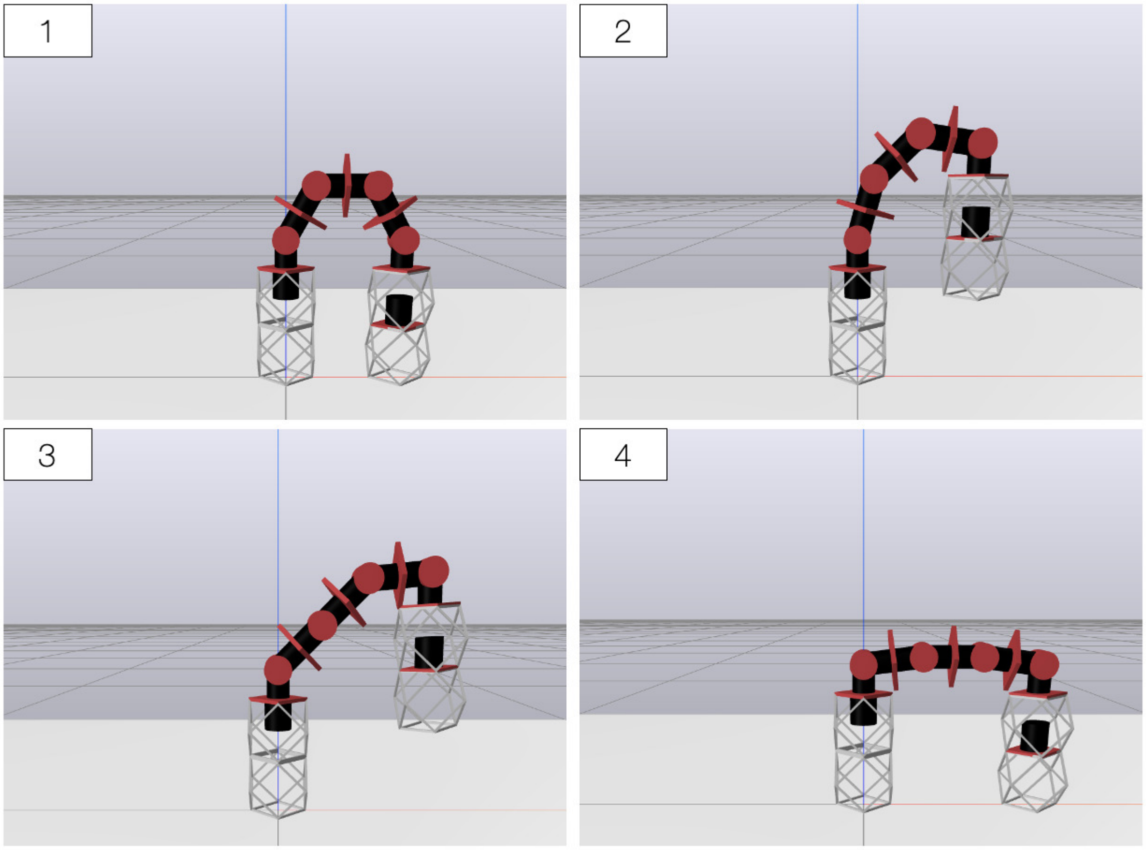

Figure 2-27: Example key frames from IK method for configuration 2 style robot. 1) Starting position; 2) Vertical offset above start position; 3) Vertical offset above end position; 4) End position.

$$x[0] = x_0$$

$$+\text{additional constraints.}$$

Using an optimized trajectory may be of utility in situations in which the system is torque limited (or otherwise doing motions that may be hard on the robot's joints). As an example, I compare in Drake the total control input of a trajectory generated using the above described IK method versus one produced by the direct collocation method. I use as a test case the robot going from its default standing position to vertical (see figure 2-28). I send the generated trajectories to the simulated PID controller, and sum over the absolute value of the control input over the simulation. In this example, the direct collocation trajectory uses approximately 60% less control input. Though this has not been replicated in hardware, qualitatively, the naive IK method results in more hardware failures (e.g. solder joints failing) than a trajectory modelled off the optimized version.

Though the direct collocation strategy shows some process, I do not primarily use it because it often requires fine-tuning of the costs and constraints to produce reasonable, collision-free trajectories. A trajectory based on grabbing a few key frames of the optimized standing-to-vertical move has been implemented, but beyond that, the optimization process does not currently offer enough benefit for wider use. In the future, such as if the robot ever needs to interface with the "real world" it may be more relevant to re-visit these workflows.

A simulation based workflow for the low-level coordination of multiple robots was also developed. Because the robots alone are relatively simple, collaborative behaviors have the potential to substantially improve their performance. Rotating a voxel's orientation, for example, could be difficult for a single robot to do as it would likely need to involve placing the voxel without attaching it onto the lattice, risking losing lattice pitch indexing. In contrast, two robots can more easily alter the orientation by passing the voxel between each other, without risking losing the known position of the voxel. An example of this type of coordination is shown in figure 2-29, where one robot picks up a stack of voxels and passes it to another robot. In this example, robot configurations are found simultaneously using an IK method similar to the one described above, though without pre-defining desired

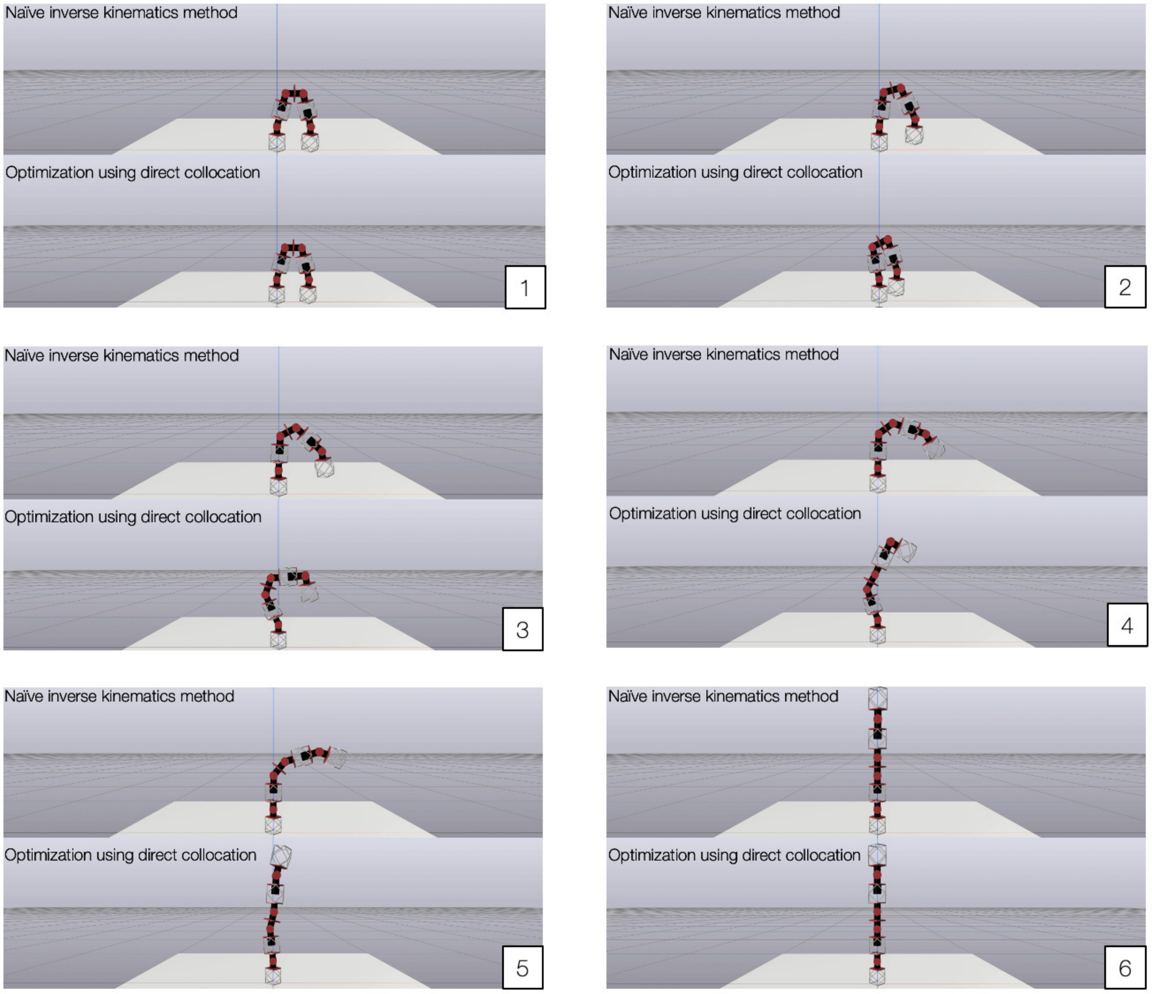

Figure 2-28: Freeze frames of a comparison of direct collocation trajectory (lower split screen) versus the naive inverse kinematics method (upper split screen).

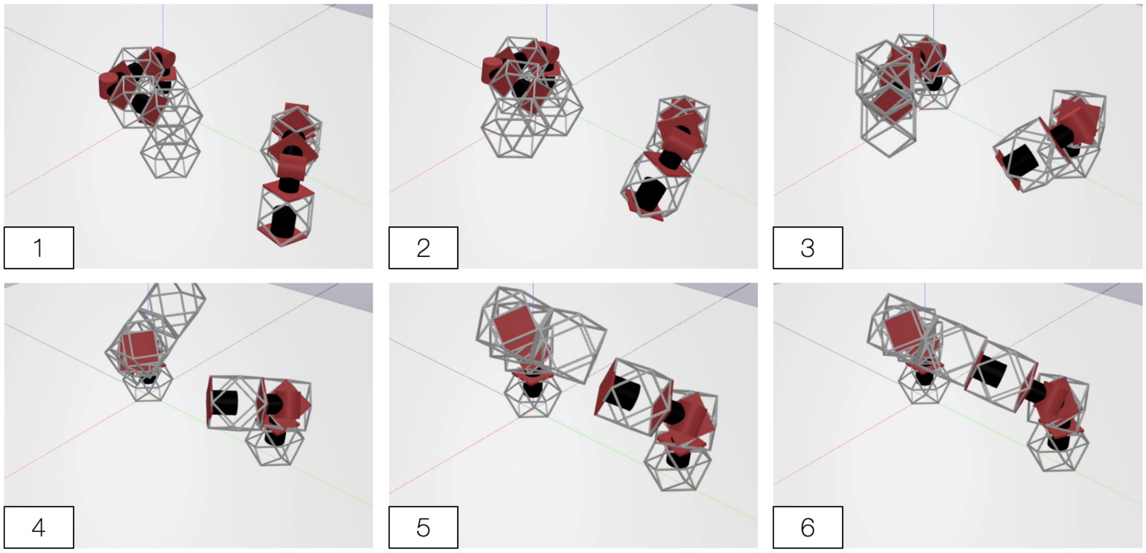

Figure 2-29: Freeze frames from a two robot collaborative move. Robot 1 starts gripping a stack of 2 voxels, which it then lifts and passes to robot 2.

positions in global coordinates. Instead, desired positions are given as relationships between faces on the separate robots and the IK method handles finding appropriate configuration to satisfy these requirements, as well as other constraints. An optimization workflow for both robots was additionally developed, though suffers from the previously described issues—specifically, the amount of tuning of costs and constraints likely makes it impractical for generating a large amount of motion types. Though the two robot simulated trajectories have not yet been applied to the hardware system, this work may be useful as the robotic swarm matures to include more robots and more non-trivial robot interactions. Specifically, the utility of this workflow derives from how it enables the easy generation of simultaneous trajectories to satisfy constraints in robot relationships, as opposed to the position dictated approach currently used for locomotion.

2.5.2 Many Robots

Path planning for multiple robots derives from the work previously done by A. Abdel-Rahman in [17] and [6]. The algorithms developed for these earlier BILL-E versions are largely compatible with the new system, with some adjustment of the types of steps that

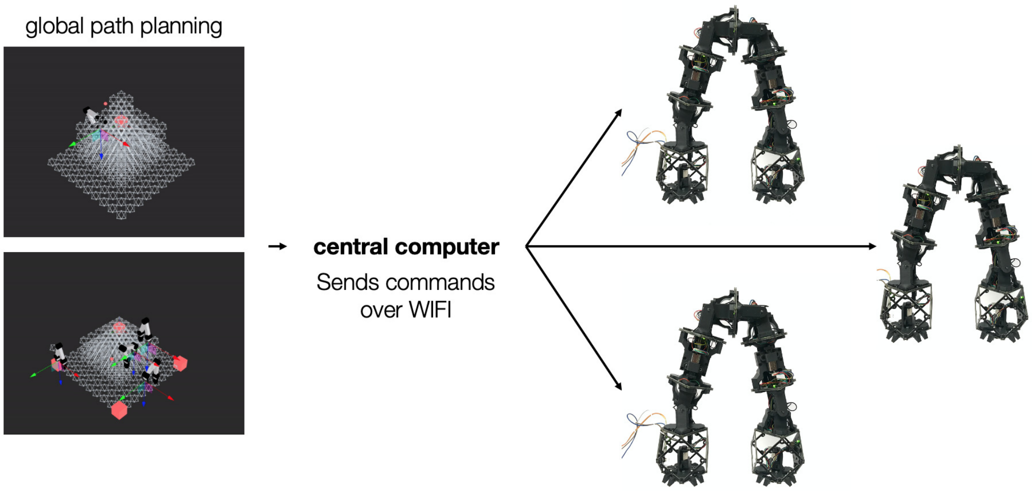

Figure 2-30: Path planning and communications diagram for a robot swarm. Path planning for a given structure is done using algorithms previously developed by A. Abdel-Rahman in [6] and [17]. The series of steps is then disbursed to the robot swarm over wifi from a central computer. Global path planning images credit: A. Abdel-Rahman.

a robot may do and the requirements for voxel placement. The prior workflows establish a process for going from a desired structure to a voxel decomposition to a robot path for various swarm sizes. The output of this can then be fed to the current software as CSV file and disseminated to the robot swarm. See figure 2-30 for reference. This workflow is not yet integrated with the hardware.

At present, a maximum of two robots have been simultaneously operated. The integration of these assembly algorithms becomes necessary once the assembled structures grow to non-trivial sizes. Currently, the focus has been on developing motion and structural primitives, building out the hardware to the point where it is ready for integration with the more mature planning algorithms.

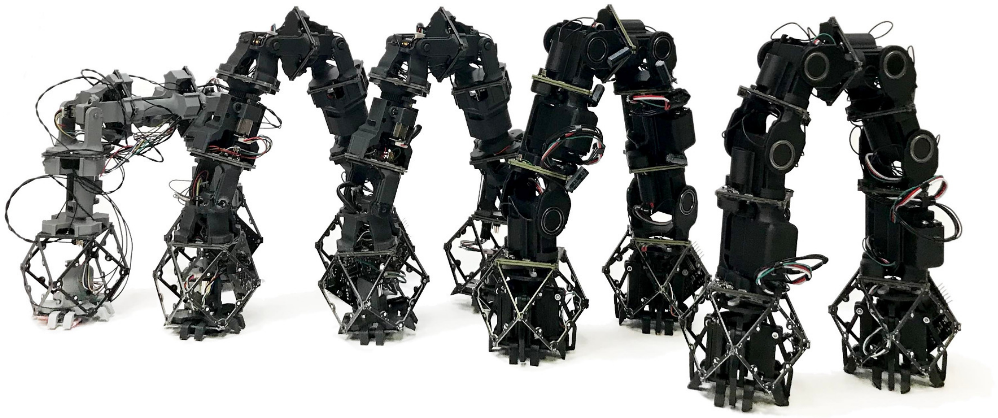

Figure 2-31: Photograph of five iterations of the robotic assembler. (From left) Earliest 5DoF configuration, first 6DoF gearmotor robot, second 6DoF gearmotor robot, first servo robot (configuration 1), second servo robot (configuration 2).

2.6 Early Prototypes

This section covers some prototypes of this robotic system. The initial plan was to use DC gearmotors with encoders on the motor shafts as the actuators for this system. This decision was based on their being cheap, lightweight, and high-torque.

The first iteration of robot based on the gearmotor modules is shown in figure 2-31. This was a fast 5DoF configuration similar to that of magnetic BILL-E, with the middle wrist joint inline with the other modules instead. The two significant takeaways from this version were 1) the solder joint functioned well enough to pursue and that 2) the 5DoF configuration was overly limiting in the robot's addressable space.

Based on that, the next version of gearmotor robot was developed with a 6DoF configuration matching that of the configuration 1 style servo robot. Module boards used the previously described ATtiny1624 boards and implemented a PID controller to control motor shaft angle. Because this version used a quadrature magnetic encoder as feedback, various methods for zeroing the motors were developed. In one zeroing method, the motors slowly turn until they contact a stop, and reset zero at that point— this method worked well for

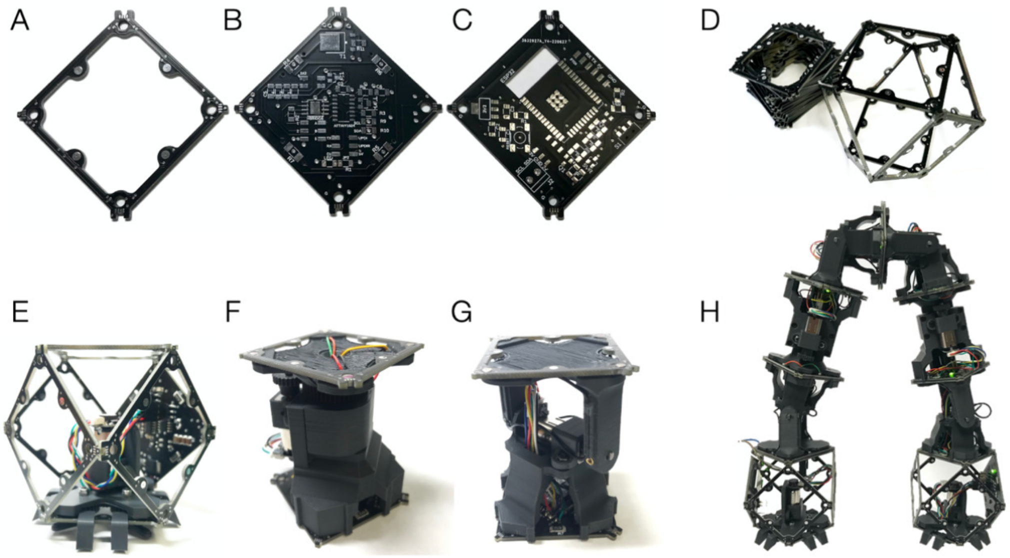

Figure 2-32: Components of the gearmotor version of the assembly robot. A) functional voxel face (the same as used in the servo version); B) ATtiny1624 face; C) ESP32 face; D) Assembled functional voxel; E) Gripper; F) Wrist; G) Elbow; H) Assembled robot.

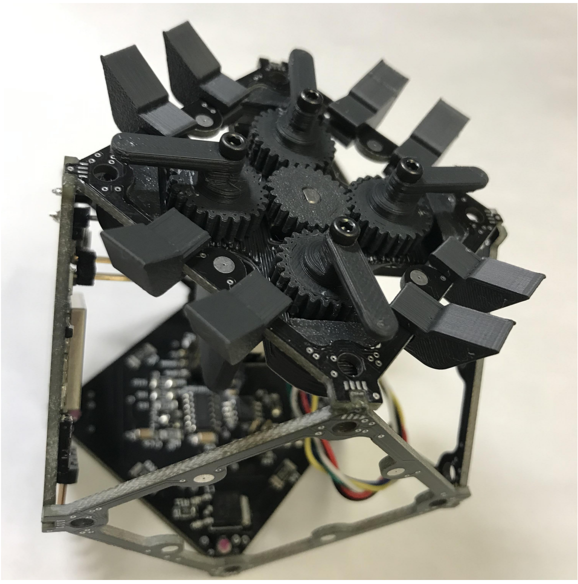

Figure 2-33: An early gripper design.

the elbows but was not suitable for the wrists, which were continuous rotation. To this end, the boards contained a hall effect sensor for zeroing off of an attached magnet to the wrists— because of challenges in fixing a magnet to the wrist at an appropriate height to the sensor without later colliding with other parts of the module, this functionality was not successfully implemented. The modules could also accept user input zeros, so pre-positioning the robot at joints limits could result in a reasonable global zero position.

Additionally, because the encoders were on the motor shaft and not the output shaft, I can only extrapolate the output shaft position, which makes it difficult to address issues such as backlash. The gearboxes on the gearmotors suffered from a large amount of backlash— about 6-10 degrees— which over the length of the robot results in many centimeters of play. This was largely unworkable. A second version of the robot was made with rubber bands to bias the backlash direction, but between the challenges in zeroing the system (i.e. starting it from a known position), the backlash of the motors, and the slop in the motor to printed part connections, I decided that this version was not suitable. So, I moved to using the servos with feedback lines as described in prior sections.

These earlier versions of the robot also used a different gripper design. The gripping element in magnetic BILL-E and the original BILL-E is designed as a rotating cross arm that contacts the beams of the voxels when closed and misses the beams at the diagonals of the voxel square when open. This design puts significant restrictions on the design of the construction voxel face and, because of the relatively small clearance when open, can result in the robot tripping if it does not step off of the voxel in a sufficiently vertical way. To address this, these gearmotor versions of the system use gears with flanges, giving a much wider length difference between the open and closed positions. See figure 2-33 for an image.

This designed struggled to provide a sufficiently stable grip on the lattice. First, because the parts are FDM 3D printed, the part is weak along its layer lines, and primarily it experienced forces that pulled in this weak direction. Next, the intention was to add a bit of soft material (e.g. polyurethane foam) as a kind of sock to the gripper flanges to provide a better grip on the voxels. It was difficult to adhere the sock well enough for it to stay attached for more than 10 open-close cycles. Without the added padding, the flanges did not constrain the gripped voxel sufficiently well to prevent some rotation and sliding over the voxel. These challenges prompted the gripper re-design that is used in the servo version.

Another useful lesson learned from the gearmotor iterations was the utility of leaving microcontroller boards accessible. Because they were largely occluded by other module hardware, it was extremely impractical to re-program the microcontrollers. In the servo version, the programming pins are left accessible such that firmware can be updated easily. A feature of the gearmotors that was lost with the servos is the ability for the wrists to continuously rotate— generally, path planning is easier with less restrictive motion limits, as imposed by the servos. In the future, it may be beneficial to revisit using specifically designed servo mechanisms for this system.

2.7 Conclusion

This chapter introduces hardware for making recursive robotic voxel assembly systems. Robotic assembly of another robot is demonstrated— representing the first level of robotic recursion. Future work will address scaling the robots to larger sizes and different geometries. Robotic assembly of basic voxel structures is shown and evaluated— unlike in prior BILL-E systems, these assemblies are load bearing in both tension and compression. The details of the construction voxel system are described in the following chapter.

The current system has only been used to demonstrate small-scale assemblies (maximum 8 voxels or robotic modules) as the larger-scale path planning algorithms have not yet been incorporated. Additionally, collaborative behaviors in hardware are still minimal for the same reason. Future work will include integrating the path planning systems developed for prior BILL-Es into this system, and building larger and more collaboratively. The current system also suffers from a lack of feedback; though I have access to the servo feedback it is not currently being used in a meaningful way. Improving this system, and potentially incorporating other feedback mechanisms (such as sensors to verify gripper foot placement) will be explored. Significantly, the current system does not have a module for installing shear clips into the construction voxel structure; this will need to be remedied. Additionally, the wired power connections begin to pose issues as the robot moves around over a larger area— these will need to eventually be replaced with battery power.

Future work at the scale of the whole system (i.e. the robotic assemblers and the voxels they work with) is discussed in the Conclusion chapter.

Chapter 3

Construction Voxels

A set of construction voxels has also been developed so that the robot may assemble structures using more standard materials: we do not have to build everything from FR4 and solder! The construction voxels, like the functional voxels, are a face connected cuboctahedron (cuboct) geometry. This chapter covers relevant background, including prior CBA voxel designs, the details of this voxel design, and the mechanical testing and simulation results.

3.1 Background

Cellular materials represent a solution for distributing material efficiently and effectively in space. In nature, cellular materials enable many lightweight to high strength and stiffness structures, such as in bone. In engineering applications, cellular materials are often employed when mass-efficient solutions are desired. Figure 3-1 shows some example natural and synthetic cellular materials, such as honeycomb, familiarly produced both by bees from wax and by humans, from many other materials, such as aluminum. The figure additionally shows micrographs of bone and of a polymer foam, two examples of stochastic cellular materials.

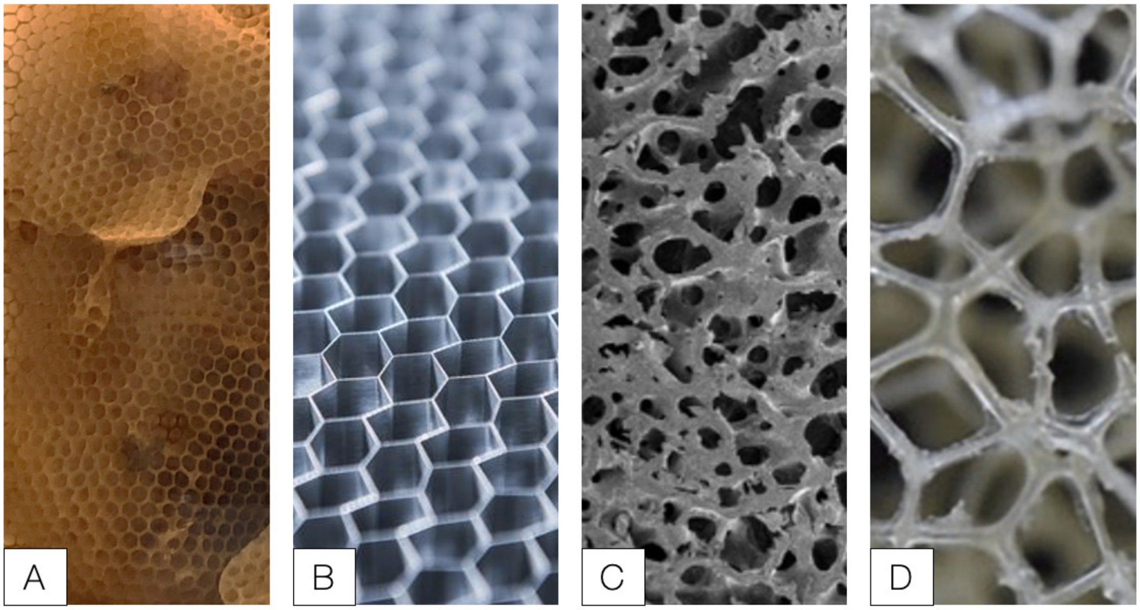

Figure 3-1: Examples of natural and synthetic cellular materials. A) Honeycomb (from bees). B) Aluminum honeycomb. Image source:[40]. C) Micrograph of bone tissue. Image source: [41]. D) Micrograph of polymer foam. Image source: [42].

Though there are many stochastic cellular materials in use in engineering [43], non-stochastic ones have better architectural efficiency [44]. These materials span a range of sizes and purposes— at the very large scale, we might recognize a truss bridge as a cellular material assembly, and at the other size scale, are microlattices with features on the order of 100s of microns [45]. Figure 3-2 shows some examples of non-stochastic open-cell cellular materials, including CBA lattices. Much of the research in producing this type of structure is focused on materials with very small features— lattice pitches on the size scales of 100s-1000s of microns, which then form material samples on the size scale of centimeters [44] [45] [46] The manufacturing techniques required for producing material at this scale, such as two-photon lithography 3D printing, would be very difficult to scale up to sizes past the 10s of centimeter scale [16]. To leverage the benefits of cellular materials at larger size scales, the CBA approach has been to discretize the lattice into building blocks which can then be assembled into arbitrarily large structures [15].

Figure 3-2 A shows a cuboctahedron (”cuboct”) lattice made from oriented carbon fiber frames which assemble together, from [15]. B shows injection molded face-connected cuboct

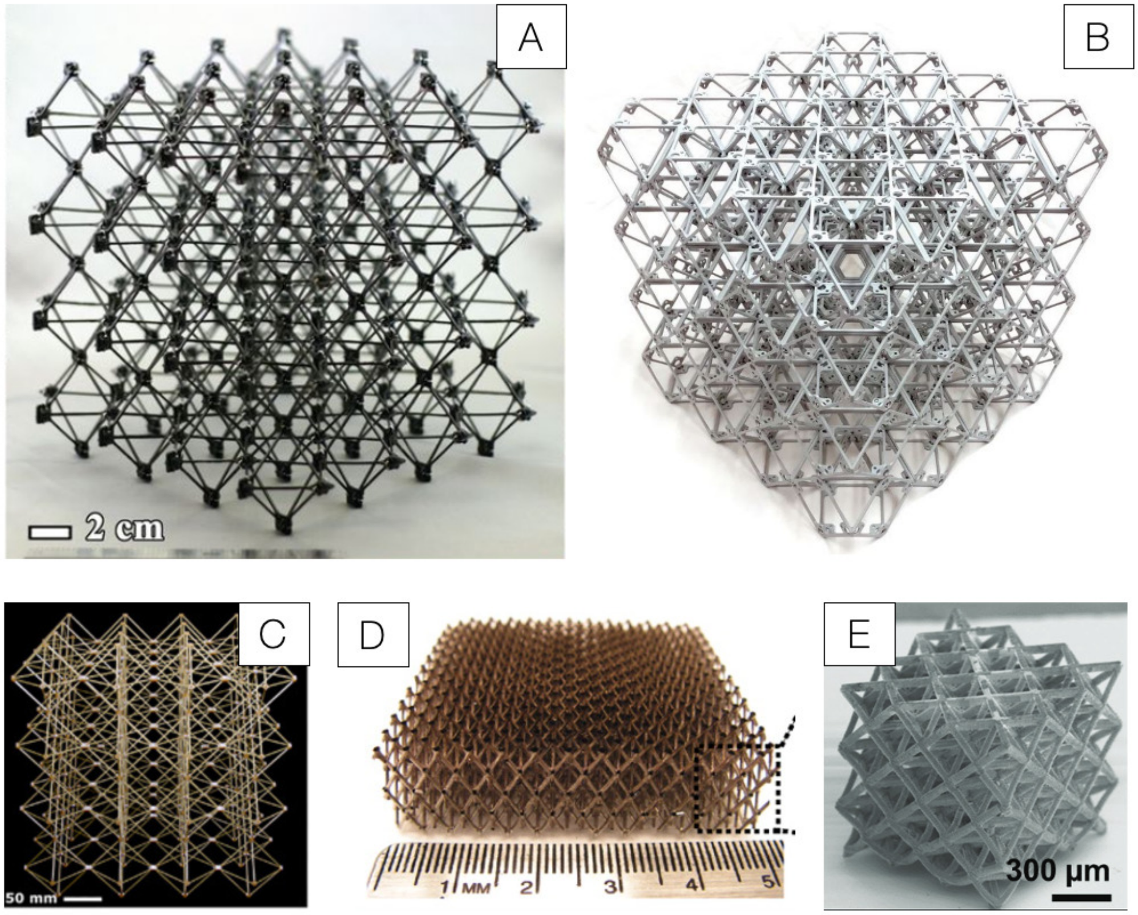

Figure 3-2: Examples of non-stochastic open-cell cellular materials. A) Discrete carbon fiber assembly from [15]. B) Injection molded discrete metamaterial from [20]. C) Injection molded node-connected cuboct from [47]. D) Metal microlattice from [45]. E) Microlattice from [46].

from [16], which uses a voxel type decomposition. C shows a node connected voxel style cuboct lattice from [47]. D and E both show examples of microlattices, from [45] and [46] respectively.

The CBA systems shown in figure 3-2 all use a cuboct lattice. The reason for this is that the cuboct lattice exists at the boundary between a stretch dominated and bending dominated lattice. Stretch dominated lattices have better mechanical performance, and typically require higher connectivity (and therefore more weight)[45]. Since the cuboct sits at this boundary, when sufficiently tiled, it represents a highly mass efficient geometry. The three cuboct lattices demonstrate different decompositions. In the first, image A, the discrete elements are X shaped frames which consist of struts of the lattice. These parts interlock and are fixed with a shear pin. The next, B, uses a voxel-style decomposition. The lattice is decomposed into blocks or voxels (in this case, face-connected cuboctahedra), which are then assembled together. The voxels themselves may either be the fundamental unit, such as in image C, where the entire node-connected cuboct/octet geometry is made via injection molding, or the voxels may additionally be decomposed into faces, struts, or nodes. In B, the voxels are decomposed into injection molded faces.

As this thesis extends on prior CBA work, I also focus on the cuboct lattice and use the face-connected geometry with a voxel decomposition as it is the most compatible with robotic assembly. The voxels offer a well-defined anchoring system for the robots, and the face-to-face connections offer a larger space for error correcting features and connection mechanisms as compared to the node-connected decomposition. Prior CBA work has explored a variety of connection systems for the voxels, including nut and bolt [47], aluminum rivets [16], various version of interlocking joints with shear pins [15] [48], as well as the magnetic system from prior BILL-E systems [17]. A variety of material types have also been used, such as the carbon fiber composite and injection molded GFRP examples shown above.

3.2 Voxel Design

Design Overview

The primary goal of the construction voxel design was to create a cuboct voxel compatible with robotic assembly with better joint strength than the prior magnetic systems. A secondary goal was to use a 2D decomposition of the voxel so that the design could be more easily ported to different materials, as well as to facilitate faster prototyping.

The version of the construction voxel used by the robot for assembly consists of six cuboct faces which snap fit together. The resulting cuboct voxel is then decorated with mechanical latching features, which are also made from sheet stock. The latching features facilitate voxel-to-voxel connections via vertical assembly (see figure 3-6 for reference). The voxels fit in a 65x65x65mm bounding box and are designed to have beams with 3 mm x 1.6 mm rectangular cross-section. The thickness of the beams varies a bit depending on the sheet stock used, though within the range 1.5 - 1.7 mm.

The design is compatible with multiple material types. For this thesis work, I initially used acetal (Delrin or Acetron) for ease of prototyping and later switched to aluminum for greater stiffness. This voxel design has been made from acetal, acrylic, aluminum, steel, sintered nylon, and printed nylon-carbon fiber composite. Some of these voxel types are shown in figure 3-3. Porting between material types only requires minor adjustments to the snap fit features (or even no adjustments).

The mechanical connection system used here enforce a specific orientation for voxel assembly. Voxels can be only be added to sides of voxels with the latching clips installed. This is a limitation the prior CBA voxel systems did not have— the voxel based assemblies were typically orientation agnostic. However, given that this thesis uses an iterative assembly process where voxels are placed one at a time, and that it is reasonable to assume that the assembly system knows the orientation of a voxel, this limitation is not as practically significant as it may seem. It does, however, limit the way the structure can be disassembled, which may be a significant issue for a structure that is reconfigured at a high

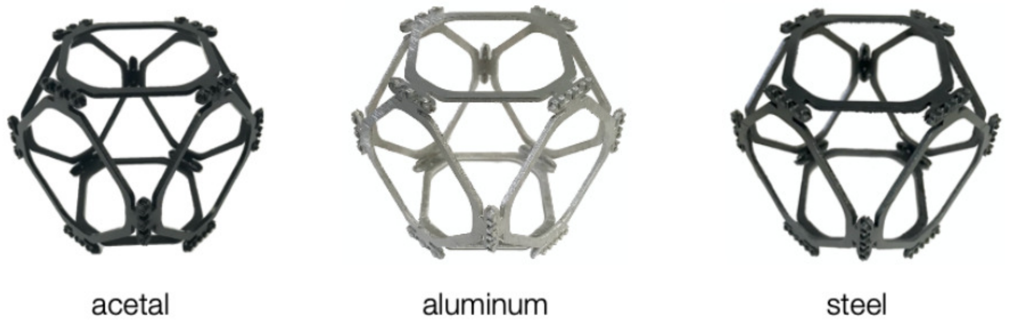

Figure 3-3: Construction voxels made from laser cut acetal, aluminum 6061, and mild steel.

frequency. Figure 3-4 highlights the assembly directions and clips. Note that this system is not constrained to use the same orientation of voxel throughout the structure— so long as there are no collisions between the clips, the clip orientation can be altered, or even left off. Figure 3-5 shows two examples of different assembly directions for a small plane of voxels. Voxel orientation can be chosen based on what is most convenient for the robot to access. The clips on the voxels may additionally be flipped upside down for further flexibility in assembly. Specifically, this means that if needed, this system is still able to add voxels in any orientation to a structure (i.e., by adding a voxel with a flipped set of clips, the system can build out in the opposite directions than what is indicated in figure 3-4.

Use of the previously pictured clips constrains the voxel in all but the vertical assembly direction. To fully constrain the voxel, an additional shear clip can be added. (Note that the robot is not currently equipped to do this; adding this functionality is a goal for future work). If a slightly weaker and anisotropic joint between voxels is acceptable, an alternate version of the clips has been made which employs a snap fit latch. These clips completely constrain the position of the voxel once placed. The voxel assembly sequence for the clip and shear clip combination is shown in figure 3-6. These clips are shown in figure 3-7. A and B show the snap fit version, while D and E show the standard clip that requires an additional shear clip, C. For this system, I use two clip sizes because the larger clip permits a larger margin of error in the robot's voxel placement. Using exclusively the large clips would result in collision, so a smaller clip is used in each side as well. We could instead use

Figure 3-4: Construction voxel with clips installed highlighting build direction. Clips are circled. Note that the voxels do not need to have the same clip orientation— the assembly direction can be flipped during assembly.

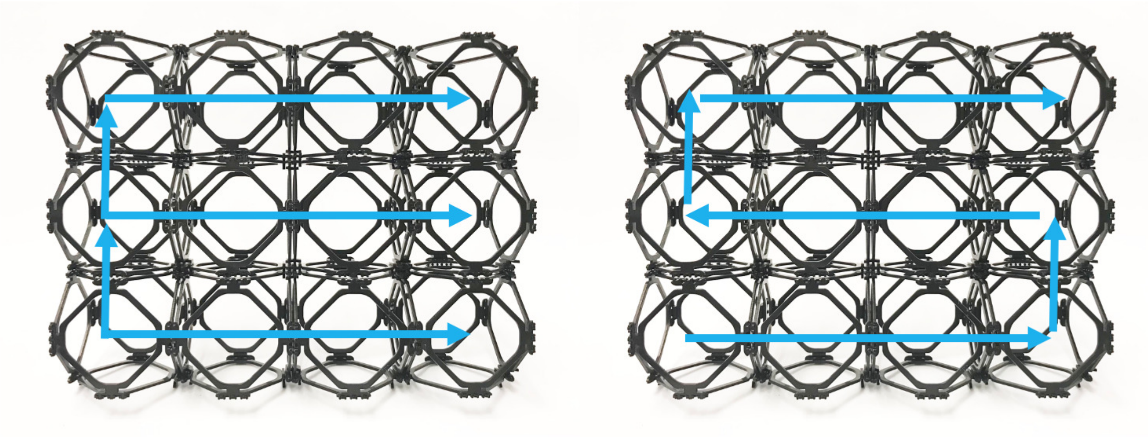

Figure 3-5: Examples of construction voxel assembly directions.

only the small clips with a more precise robotic assembler.

This voxel design, unlike past CBA face-connected cuboct designs, does not have the faces touching. The faces are instead offset from each other, to facilitate the snap-fit connection. Figure 3-8 shows a close up image of the intersection of four voxels in which this offset (as well as the clip joints) are visible. The cuboct system shown in [49] uses an offset version of the design as well. Permitting an offset between the faces makes joint design much easier without necessarily negative changes to the mechanical performance as compared to when the faces are touching (see the section on simulation results for more information). In practice, the joint system implemented in this thesis is not as stiff and strong as would be desirable, though is functional and shows promise for further improvements. Alternate designs without offset faces are discussed in the section Early Design Explorations.

Manufacturing Processes

Delrin voxels were made by laser cutting 1.6 mm sheet stock on a Trotec CO2 laser and snap fitting the resulting parts together. This is an easy and fairly time-efficient process— laser cutting the parts for 8 voxels takes approximately an hour (7.5 minutes per voxel), which can be passive, and the manual assembly of each voxel (including clips) takes less than two minutes on average. Because of this convenience, the original plan was to use Delrin voxels for most of the work in this thesis. However, the stiffness (or lack thereof) of the voxels became a problem during the robotic assembly process— it simply became easier to switch to something stiffer to cut out some slack in the system. To this end, I decided to switch materials to something more performant.

I surveyed a set of other plastics easy to prototype with first, which did not yield useful results. I tried a laser cut acrylic, which would have comparable behavior to the Delrin, though at a marginally higher stiffness. The permissible strain for acrylic is much lower though, which made the snap fit connection close to untenable. I then tried 3D printed nylon with chopped carbon fiber (Markforged Onyx filament). The SDS suggests that the filament has comparable properties to Delrin (which mechanical testing results in the

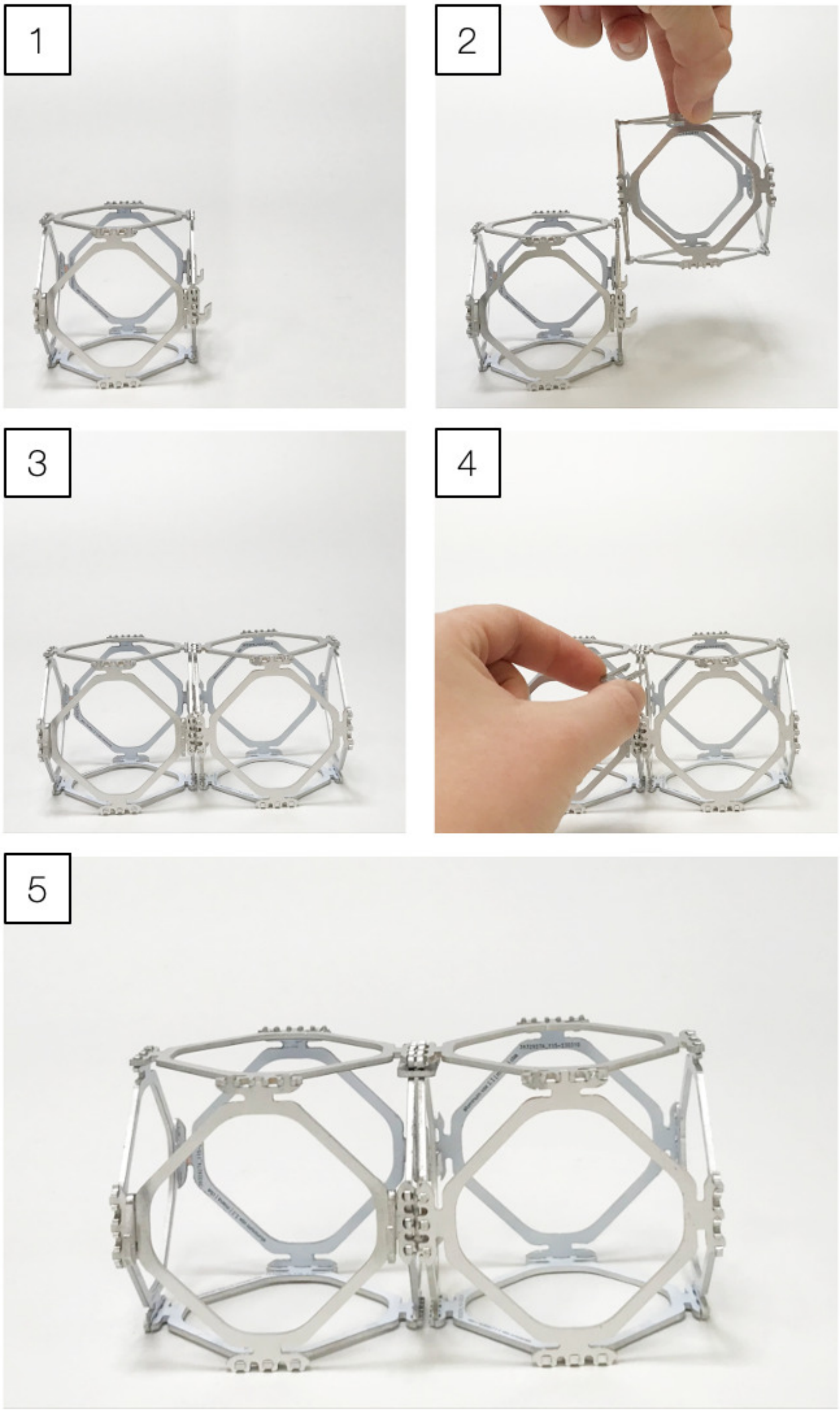

Figure 3-6: Voxel assembly sequence. 1) The first voxel. 2) Second voxel is added from vertical direction. 3) Second voxel is placed. 4) The shear clip is added. 5) Fully constrained set of voxels.

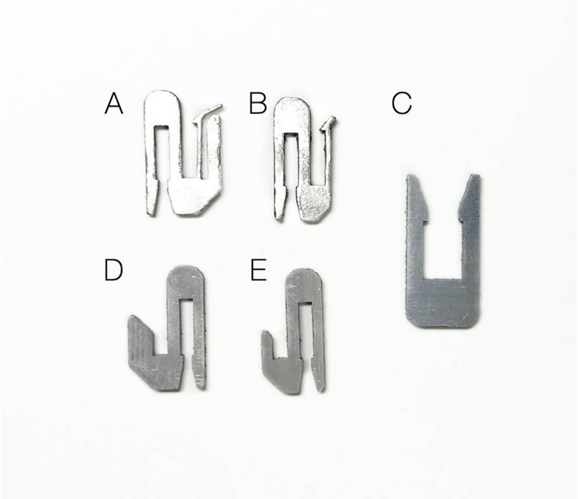

Figure 3-7: Voxel clip types. A) Snap fit large horizontal clip; B) Snap fit small horizontal clip; C) Shear clip; D) Large clip; E) Small clip.

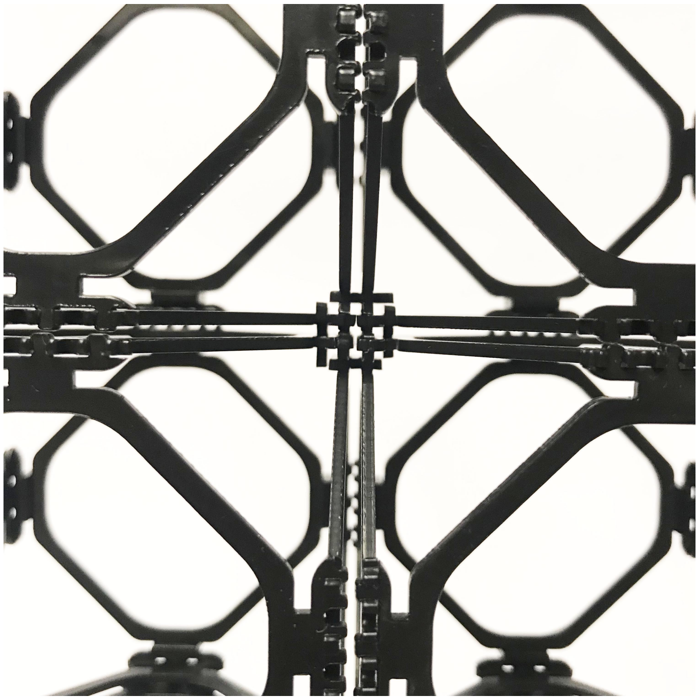

Figure 3-8: Close up shot of the joint between four Delrin voxels.

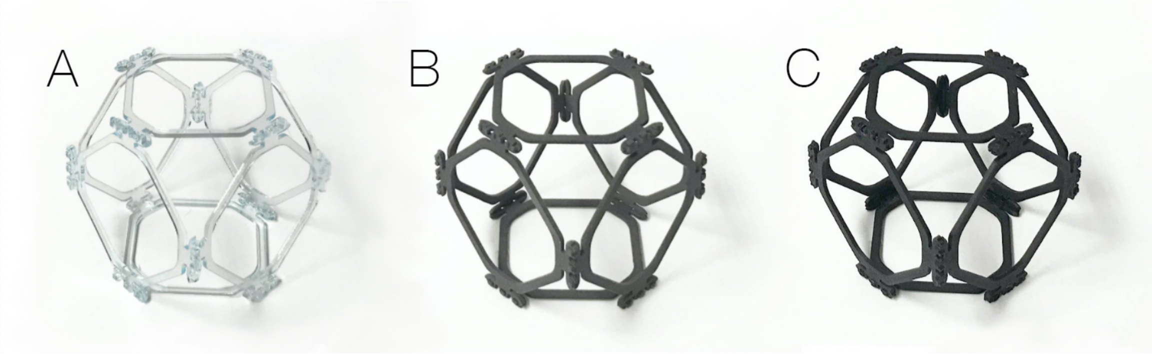

Figure 3-9: Construction voxels made from laser cut acrylic, 3D printed nylon (sintered on a Formlabs Fuse), and 3D printed nylon composite (FDM printed on a Markforged Mark 2).

following sections agree with). The FDM printing x-y resolution was not good enough for the snap fit features to survive, and it was not worth it to redesign the voxel for essentially the same stiffness at a lower throughput manufacturing process. I also tried printing the voxel in a Formlabs Fuse with sintered nylon, which is significantly less stiff than Delrin (this was more an exercise in using the printer). The faces printed on this printer assemble nicely, and its resolution is good enough that a pre-assembled voxel printed on it quite closely resembles a voxel assembled from parts it printed.

I then tried laser cutting metal faces on a Fablight fiber laser. On this, I tried 6061 aluminum and mild steel, both at approx 1.6 mm thickness. This process was fairly successful, except for the slag left on the parts. Additionally, the snap fits benefit from slight re-tuning such that they are not plastically deformed (or so that the mating force is in the range of whatever method the faces will be assembled by). The slag left on the aluminum is not especially well adhered to the parts, so the removal is not that labor-intensive. The slag left on steel parts, on the other hand, is properly welded on and can be quite time-consuming to remove. Overall, though, the process was fairly easy and results in parts with much higher strengths and stiffnesses, in addition to much higher ductility.

I also surveyed methods for getting parts made. The voxels in [16] were made from injection molded GFRP by protolabs. At a quantity of 300 voxel faces, using this service results in a per-face cost of approx 2.20 USD and a mold cost of around 3k USD. Depending on the percentage glass fiber infill of the GFRP, we might expect a 1.5-2.5x increase in stiffness from the acetal. This (to me) is an overly large investment in a voxel design that is not fully settled— as the primary goal of this thesis is the robotic assembly of voxel structures, the voxel design is not as mature as it would be for a thesis focused on the materials. Quotes for getting the parts made from aluminum sheet stock (5056 and 6061) ranged from 4.50 per face at the lowest to on the order of 10s of dollars at the highest. I then realized that aluminum PCBs are cheap. Though the cutting tolerances for the aluminum PCBs are quite bad, the error is fairly consistent, and the snap fit joints support an amount of error. Additionally, though the aluminum used in the PCBs does not appear to be an especially stiff or strong alloy, it is still much higher performance than acetal (or GFRP).

For this thesis, I decided it was most convenient to get the voxel faces manufactured as aluminum PCBs from JLCPCB. At a quantity of 300 voxel faces, each face costs 0.23 USD. Including the most expensive shipping option and other fees increases this to 0.57 USD per face. This is very cheap and very convenient for me. The parts are pictured in figure 3-10. Unfortunately, the dimensional tolerance of the clips matters more and includes smaller features definitely beyond what the PCB fab is willing to do. So, the clips are made in-house from 6061 aluminum on the Fablight laser cutter. The snap fits are sized such that they are difficult (though not impossible) to assemble by hand. The faces are instead assembled with the help of pliers, which increases the time spent on assembly (though still below 5 minutes per voxel). Because of the dimensional tolerances of PCB fab, some snap fits are not especially tight. In this scenario, they can simply be deformed such that they are tight, as the aluminum is pliable. Assemblies of 2x2x2 and 3x3x3 aluminum PCB voxels are shown in figure 3-11.

As a practical aside, slag removal on small parts cut from on the Fablight can be a laborious process. For this work, I tried many processes for efficiently removing slag from many parts at once. The working processes are summarized below:

- Orbital sanding (effective, mid-high throughput): Small parts with many edges tend to stay in place after cutting, so I take advantage of this by leaving the parts in (replacing any that have fallen out), clamping the stock material, and using an orbital sander

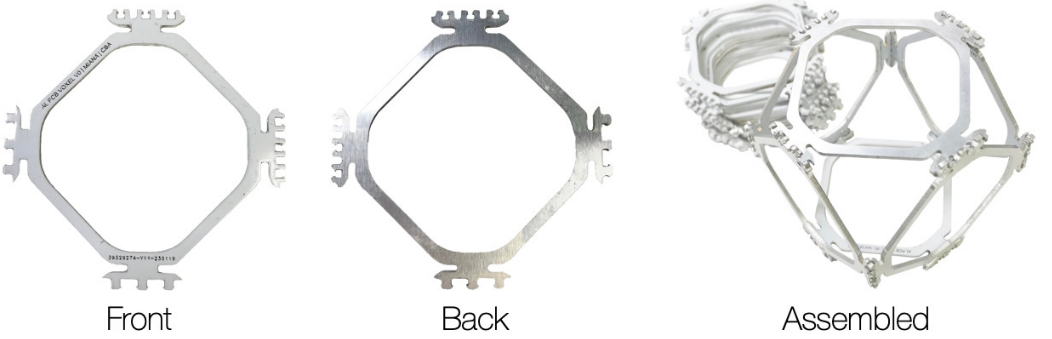

Figure 3-10: Aluminum PCB voxel. (From left) Top surface of aluminum PCB with solder mask and silkscreen. Bottom surface. Assembled voxel.

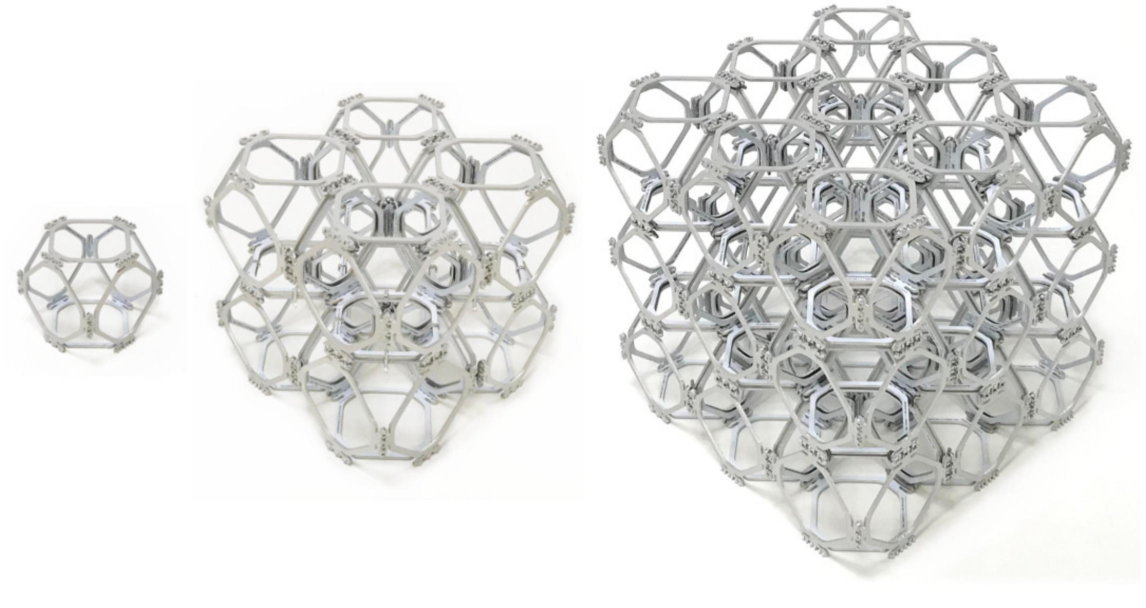

Figure 3-11: Single aluminum PCB voxel, 2x2x2 cube, and 3x3x3 cube. Note this is a composite image; the scales of each assembly may not be matched.

to remove slag. This is very effective, fast, and suitable for high part counts. As a bonus, this results in a nice surface finish.

- Hand sanding (effective, low throughput): Sanding by hand is always an option— the fastest version of this (for me) used a 3D printed jig to hold the part while sanding.
- Tumbling (somewhat effective, high throughput): Tumbling the parts with an abrasive media allegedly should work well, though the precise process is still unknown to me. Using ceramic triangular media in a rotary tumbler for 12 hours was the most effective, though left some slag hardened on. A layer of grime additionally formed on the parts, so tumbling was followed by 6 hours in a sonicator. Other media I tried included walnut, steel of strange shapes, and carbide polishing sand— these were all ineffective.

Vertical Assembly

The clips I have so far described only support in-plane construction. Another set of clips is used for building up. These index into the same notching features on the voxel face using snap fits. Several designs for these clips have been developed and evaluated, in terms of mechanical properties and robot compatibility.

The first design used is shown in figure 3-12 B and uses a staple like geometry, with two stacked snap fits to attach the two voxel faces. This design was awkward to install, so I moved onto the style of clip shown in figure 3-12 A and C, which horizontally installs onto the initial voxel for assembly convenience. This horizontal installation introduces compliance into the clip that the prior design did not have— the convenience of the easier install outweighed this concern. In the future, as this design evolves to prioritize performance, the compliance in this joint (under tension it will hinge open in the installation snap fit joint), will have to be eliminated. The clips then snap fit onto the lower voxel, with extra length on the flanges to assist in positional correction. The mechanical behavior of these clips is discussed in greater detail in the Mechanical Testing section. Specifically, the installation and removal forces are found— the downward force required to install these proved difficult to consistently achieve with the robot. So, a non-structural clip design was made, shown in figure 3-12 D. This clip acts as an aligner, with a weak snap fit which holds the voxel

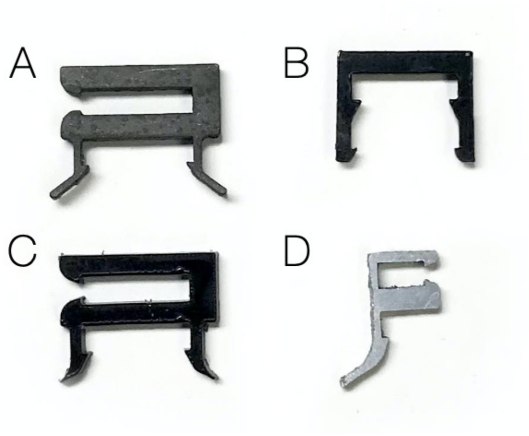

Figure 3-12: Vertical clip types. A) metal snap fit; B) Delrin "staple"; C) Delrin snap fit; D) metal aligner.

in place until it can be structurally constrained using the shear clips shown before. The location and function of the aligner clips is shown in figure 3-13.

3.3 Voxel Behavior

3.3.1 Mechanical Testing Overview

Instron testing was done to characterize the mechanical behavior of the voxels. To characterize the behavior of single voxels and the clips, a series of compression and tension tests were done on an Instron 4411 with a 500 N load cell using Delrin parts. Fixtures were made using water-jet aluminum such that voxels could be constrained in a way similar to how they would be in an assembly. Photos of the mechanical testing set up for single voxels and joints is shown in figure 3-14. A and B show the basic fixturing with a voxel installed—removable bars are screwed in over where the clips would install. The lower portion is then screwed into the static Instron base while the upper flange is gripped, as shown in C. The same set up was used for testing two voxel stacks, as shown in D. To better characterize the behavior of the clips as installed, the double-voxel set up was used with 6061 aluminum voxels, which are much stiffer than the Delrin joints as shown in E. A shearing set up was

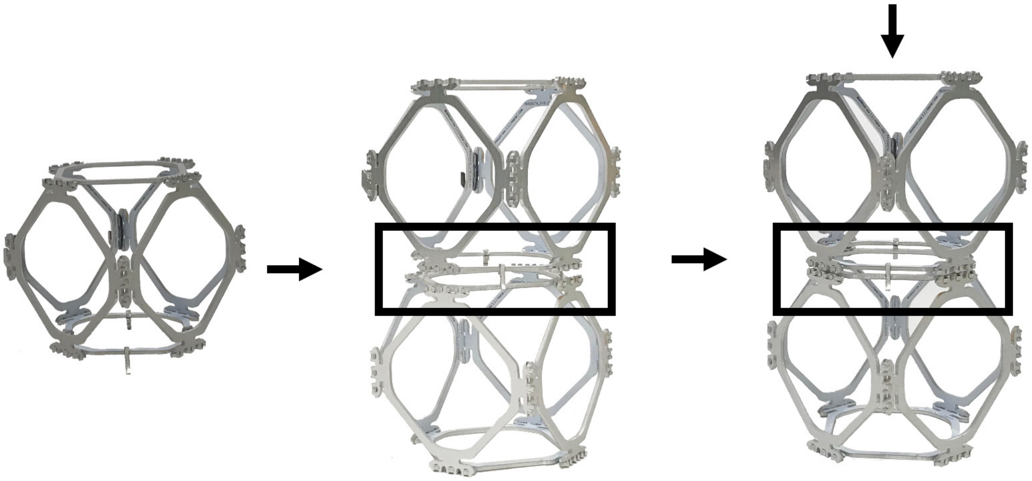

Figure 3-13: Aluminum aligners designed for the robot to use to facilitate assembling upward.

also made this way, as shown in F. Each of these tests was run at least three times to verify consistent behavior in the parts.

To characterize the bulk material behavior, compression tests were done on 2x2x2 and 3x3x3 voxel blocks, for both Delrin and aluminum. This was done on an Instron 5985 with a 250 kN load cell. Note that these were not fully constrained on their loading surfaces, so compression test of single voxels were re-done on this Instron without constraints. These tests were each only done once, as prior testing indicated that the voxels behaved consistently, the amount of voxels needed to replicate these tests is limiting, and prior literature frequently only employs single runs for larger voxel assemblies: [16], [48].

The resulting force-displacement data from the Instron testing is used primarily to extract the stiffness and/or effective modulus for the tested sample. Note that for this work, the stress-strain responses presented are engineering stress-strain, not true stress-strain. An example plot of the full stress-strain response of a 3x3x3 aluminum PCB voxel assembly is shown in figure 3-15. After the toe region, there is a clear elastic region. The slope of a best fit line to this area is taken as the effective modulus for that sample. In

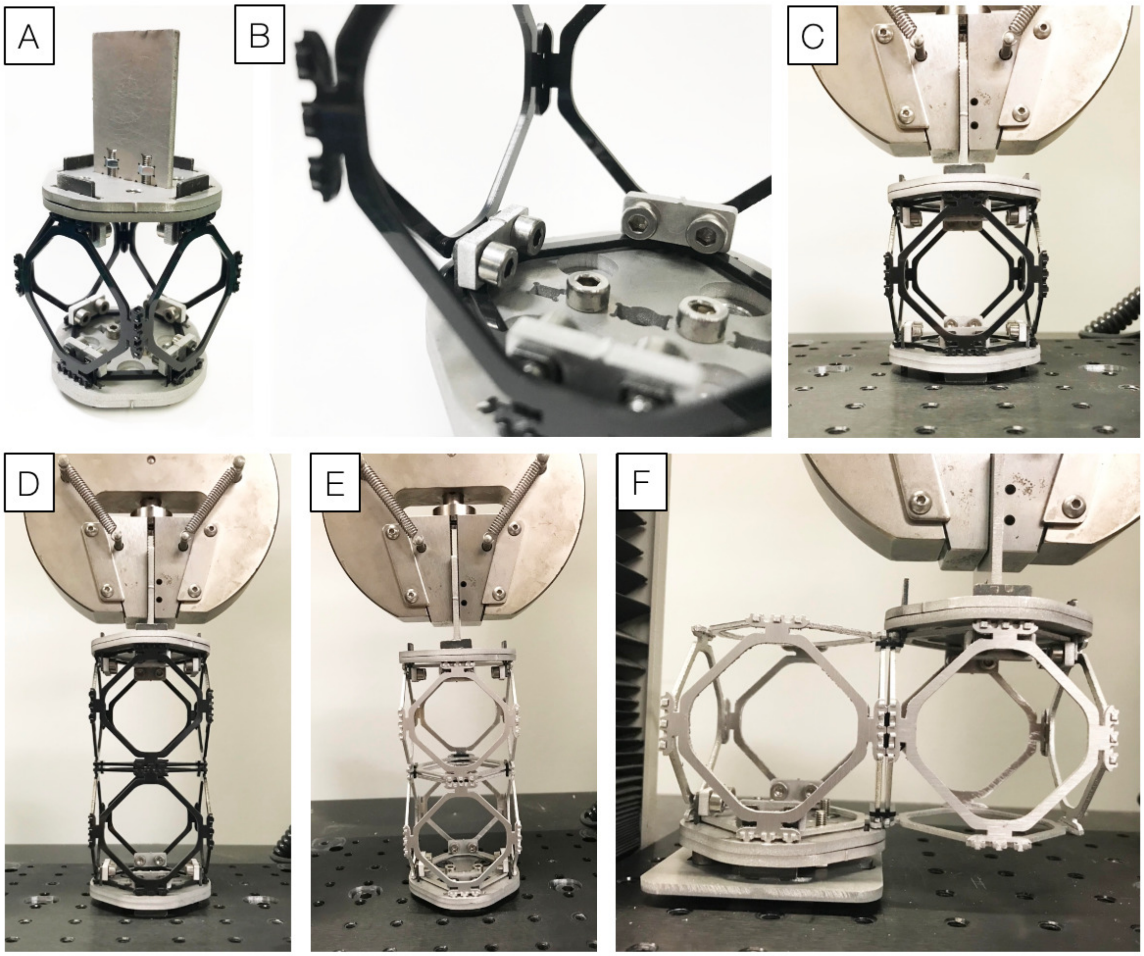

Figure 3-14: Mechanical testing set up for single voxels and joints. A) Fixtures with voxel installed. B) Close up view of voxel attachment to testing fixture. C) Fixtures installed in Instron. D) Two voxel stack tensile and compressive set up. E) Testing set up for acetal clips in tension, using aluminum voxels. F) Shear testing set up for acetal clips.

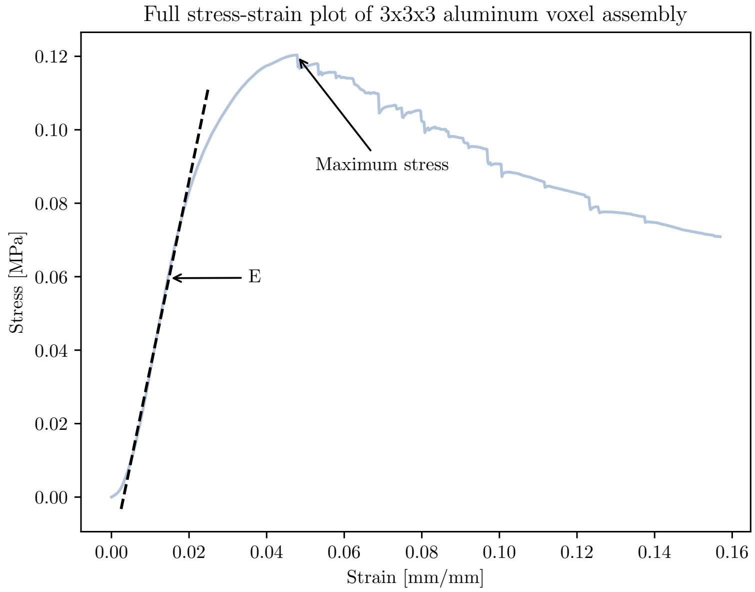

Figure 3-15: Stress-strain curve for 3x3x3 aluminum voxel structure. The engineering tensile or compressive modulus is found via best-fit line to the elastic region of the response. The modulus is plotted as a dashed line. The maximum stress is also annotated.

figure 3-15, this line is plotted as a dashed line (note that for clarity it is plotted over a larger region than it is actually fit to). The maximum strength or stress is simply taken as the peak stress reached (annotated). The yield stress is not calculated. A summary of these results is shown in 3.2.

3.3.2 Simulation Methods

This section covers the simulation methods used.

FEA simulation

For high fidelity simulation, I use mesh FEA simulation. For this work, I used the simulation environment in Fusion 360 to do linear static stress simulations. (I also attempted to use the nonlinear static stress simulation package, though it failed to converge for any of the systems of interest). Example results from applying a 10 N tensile load to an aluminum

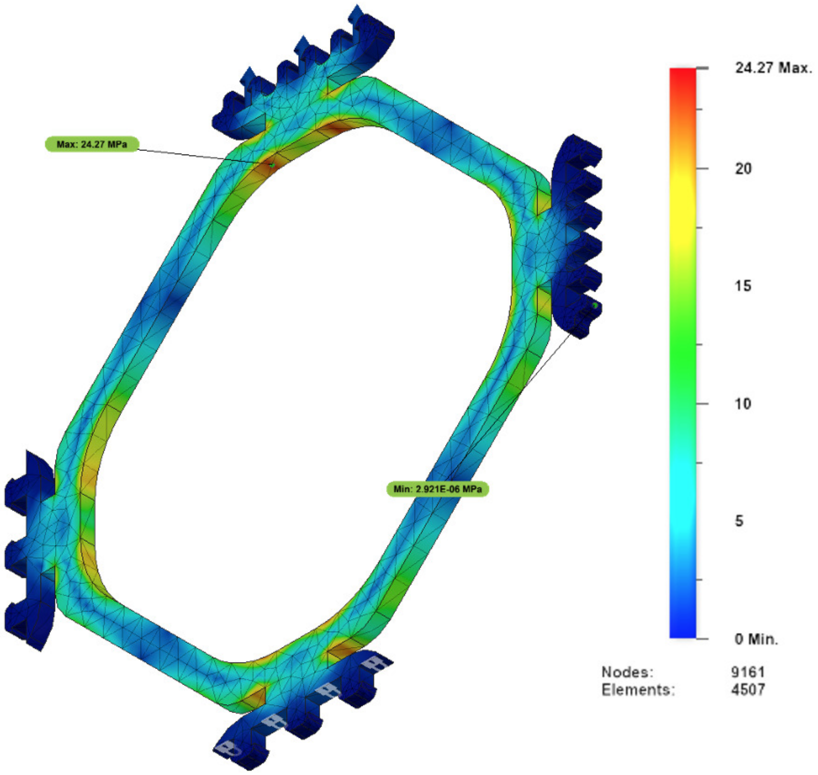

Figure 3-16: A single voxel face in tension in the Fusion 360 FEA environment. The applied force is 10 N with the opposite side anchored. The peak stress (24.27MPa) occurs near the indexing features, as seen in the failures of actual test specimens.

voxel face are shown in figure 3-16. The highest stress concentration occurs under the indexing features, matching the failure modes shown in figures 3-21 and 3-23.

This environment was also used to simulate the behaviors of larger voxel assemblies, including 2x2x2 and 3x3x3 voxel assemblies under compression. The results from these are shown in figure 3-30 and discussed in the Results and Discussion section. At the 3x3x3 voxel system, it became nearly untenable to run the simulation, as the software would often crash. It is likely not feasible to run a simulation for voxel structures larger than 3x3x3 voxels without making simplifications to the geometry and connection scheme. A 3x3x3 voxel structure using this design does, admittedly, represent many parts: Each voxel has six faces, and requires and assortment of clips, totaling around 300 separate components.

Simplified beam model

Though the FEA simulation is able to provide accurate results for small voxel systems, to simulate larger voxel structures, I need to use a different approach. [16] uses Frame3DD, an open source software for analyzing frame and truss structures. For [16], C. Cameron wrote an additional library, VoxFrame, which adds functionality for conveniently building out cuboct lattice structures using pyFrame3DD, a python wrapper for Frame3DD. The results from this library have been validated against results from Oasys, a commercial software suite for structural analysis. This work builds from that code to build a simplified beam model of the voxels in this thesis.

For the simplified beam model, each voxel consists of a cuboct voxel with short beams extending from the corners where additional voxels connect to. This is illustrated in figure 3-17, in which the connector lengths are exaggerated and colored green for visibility. In the simulation model, they are set to a length of 0.8 mm such that when put together, they match the offset distances of the actual voxels. The lattice pitch is 65 mm. I am able to separately toggle the Young's modulus of the connector beam elements. The stiffness of the actual clip connectors can be extracted, and then find the equivalent Young's modulus such that the modelled connectors have the same overall stiffness as the actual ones. Note that this beam model makes enough simplifications that it may be more useful to think of it as representing the ideal behavior for this voxel system. Comparisons between the simulated results and measured results are shown in the Results and Discussion section.

This simulation environment also gives us a chance to explore how assembling the voxels with offset connectors compares to assembling them face-to-face, without an offset. In simulation, I explore a four voxel stack with offsets (this voxel geometry), and without. In the cantilevered beam, the tip displaces 1.80 mm for the no offset style voxel, and 1.92 mm for the offset voxel. For the compressive load, the tip displaces 0.53 mm for the no offset voxel, and 0.51 mm for the offset voxel. For the tensile load, the tip displaces 0.52 mm for the no offset voxel, and 0.51 mm for the offset voxel. These results are shown in figure 3-18. These simulations are done with beams with Young's modulus 3 GPa and shear modulus 1 GPa, similar to values for Delrin. The stiffness of the connector beams is equal to that

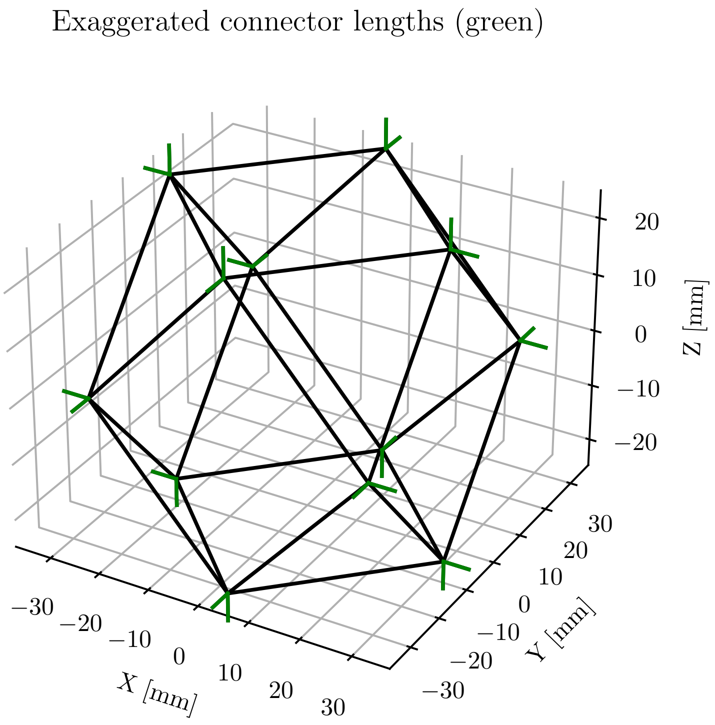

Figure 3-17: A single simulated voxel with connectors colored green.

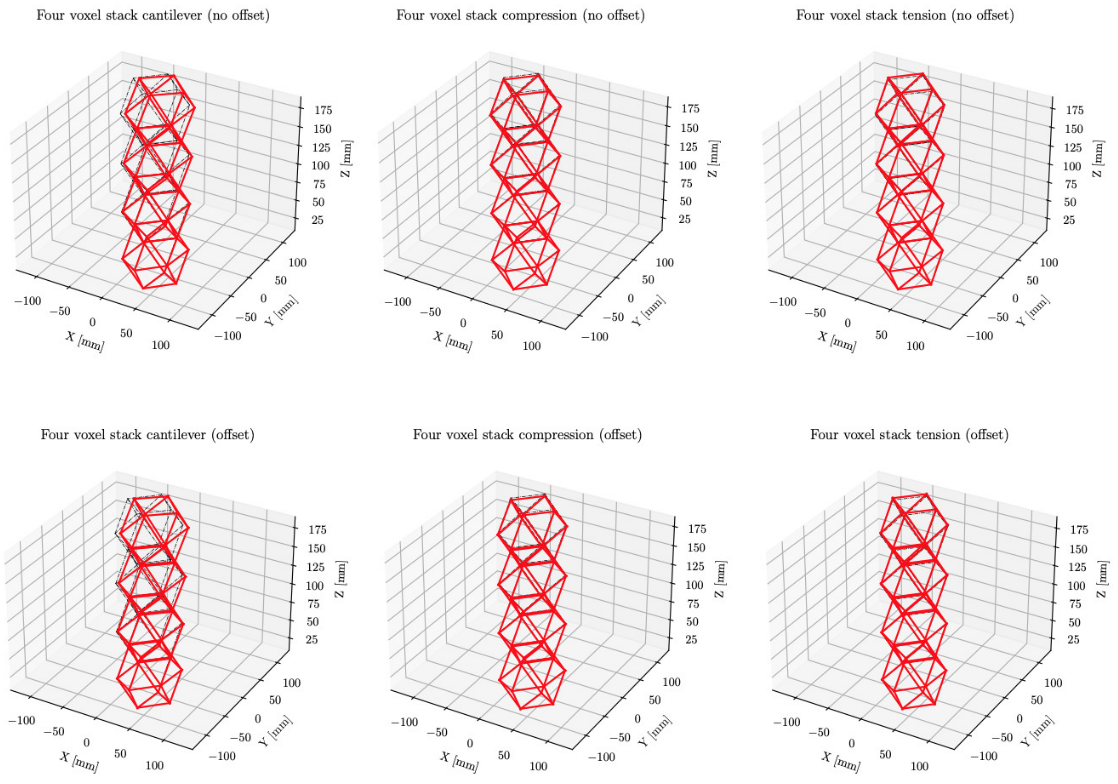

Figure 3-18: A four voxel stack subject to a cantilever, compressive, and tensile load. "Offset" refers to the style of voxel picture in figure 3-17 while "no offset" refers to the face-to-face connection from prior voxel systems. In the cantilevered beam, the tip displaces 1.80 mm for the no offset style voxel, and 1.92 mm for the offset voxel. For the compressive load, the tip displaces 0.53 mm for the no offset voxel, and 0.51 mm for the offset voxel. For the tensile load, the tip displaces 0.52 mm for the no offset voxel, and 0.51 mm for the offset voxel.

of the voxel beams— this is not the case for the current physical system, but these early simulations are aimed at understanding relationships and not predicting absolute behavior, so this is fine. The largest difference is for the cantilevered load case, in which the offset version displaces more than the non-offset case.

Based on this, I ran a sweep of relative connector stiffnesses for the offset faces geometry and compare against the non-offset face case. The results from this are plotted in figure 3-19. The relative stiffness of joints to beams refers to how stiff the joint beam elements are compared to the cuboct beam elements (i.e., 10^{-1} on the x-axis means the joint beams are set to 0.1 the stiffness of cuboct beams.) As the relative stiffness increases, the displacement

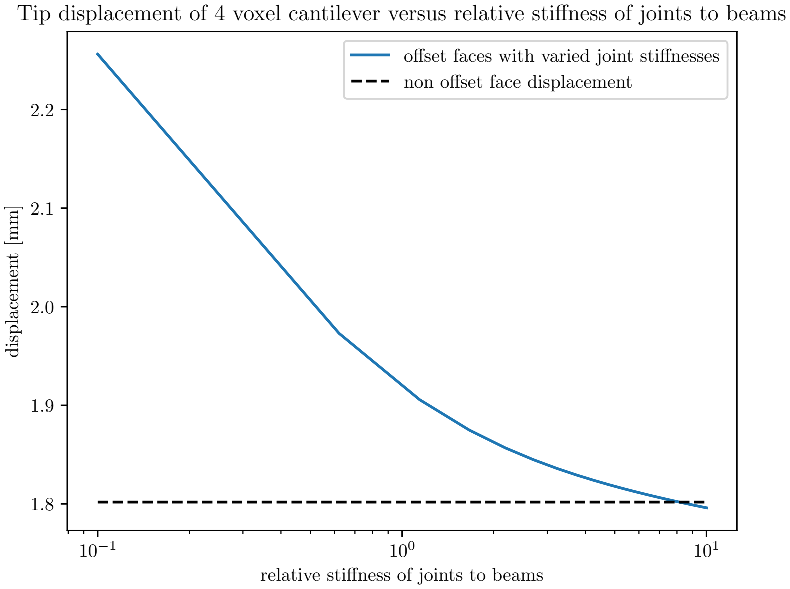

Figure 3-19: Tip displacement of a four voxel cantilevered stack versus the relative stiffness of joints to beams (e.g. a relative stiffness of 1 means that the joints and beams are set to have the same value). The beams are set with Young's modulus = 3000 MPa and shear modulus = 1000 MPa.

of the offset 4 voxel stack approaches that of the non-offset faces, eventually crossing at approx. 8x the stiffness. It is worth noting though, that through this sweep of three orders of magnitude of connector stiffnesses, the displacement changes less than 0.5 mm (or less than 0.8% strain).

We can then consider the impacts of the clips on larger voxel assemblies. Figure 3-20 shows the shear deflection for NxNxN voxel blocks with offset and non-offset faces. For this study, the Young's modulus of connector beams was swept from 0.001*E to 0.02*E, where E is the Young's modulus of the voxel beams (set to 3GPa). The dashed line plots the resultant strain from applying a constant shear force to non-offset faces at the given voxel amount. For the single voxel, the offset faces deflect less than the non-offset version. The resultant strain for the non-offset faces at larger voxel amounts then decreases more rapidly than for the offset version. It appears though, that as N increases, the offset versions still approach the same value, suggesting that the behavior of these connectors should become less relevant as the structure increases in voxel count.

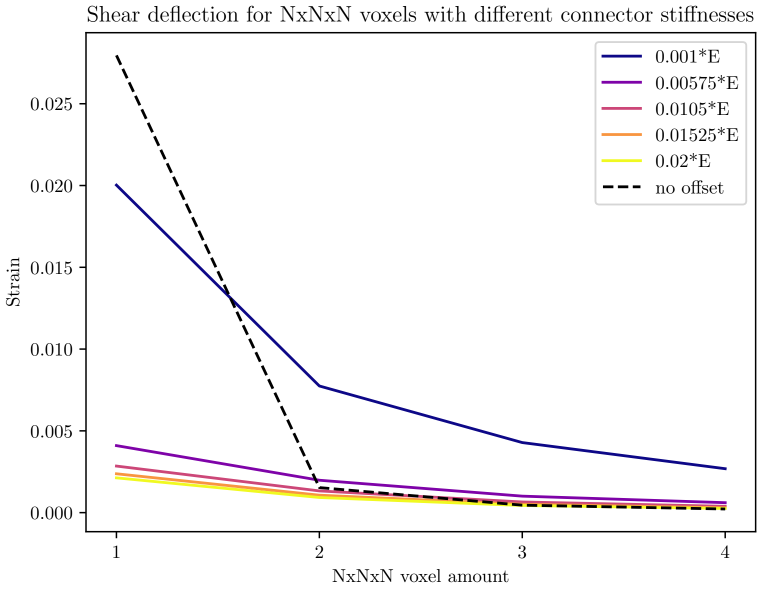

Figure 3-20: Shear deflection for NxNxN voxel blocks with offset and non-offset faces subject. A sweep of connector stiffnesses is shown, ranging from 0.0001 to 0.02 times the modulus of the beam elements. The dashed line is the response from a voxel structure with no offsets.

3.3.3 Results and Discussion

This section covers the results from mechanical testing and compares them to simulated behavior.

Failure modes

The Delrin voxels primarily experienced brittle failure at the end of the beam near the clip indexing features. This region features a width change coupled with a sharp angle, so it is not surprising that failure would occur here. Figure 3-16 shows FEA simulation of a single voxel face indicating this region. Some example images of this type of failure are shown in figure 3-21. The left image shows a single voxel failing in tension and a double voxel stack failing in compression. The double voxel stacks, in tension, did fail at the joints more frequently, which is further discussed later. The Delrin voxels often failed simultaneously and dramatically in multiple locations— a fun feature for entertainment, and a bad feature for a structural system. In contrast, the Markforged Onyx voxels and the aluminum voxels are ductile.

Individual Voxel Behaviors

This section covers the behavior of individual voxels and the function of clips. To characterize the individual voxel, compression and tension tests were done. To understand the impact of the joint system on the voxel performance, compression and tension tests were done on two-voxel stacks. Clips were additionally individually compression tested using much stiffer voxel fixtures. Some shear and other off-axis testing was done as well, which is further described later.

The tensile and compressive modulus for one and two voxel stacks is shown in figure 3-24. For one voxel in tension, $E = 0.367 \pm 0.012$ MPa. For one voxel in compression, $E = 0.404 \pm 0.009$ MPa. For two voxels in tension, $E = 0.314 \pm 0.009$ MPa. For two voxels in compression, $E = 0.372 \pm 0.003$ MPa. These results are additionally summarized in 3.2.

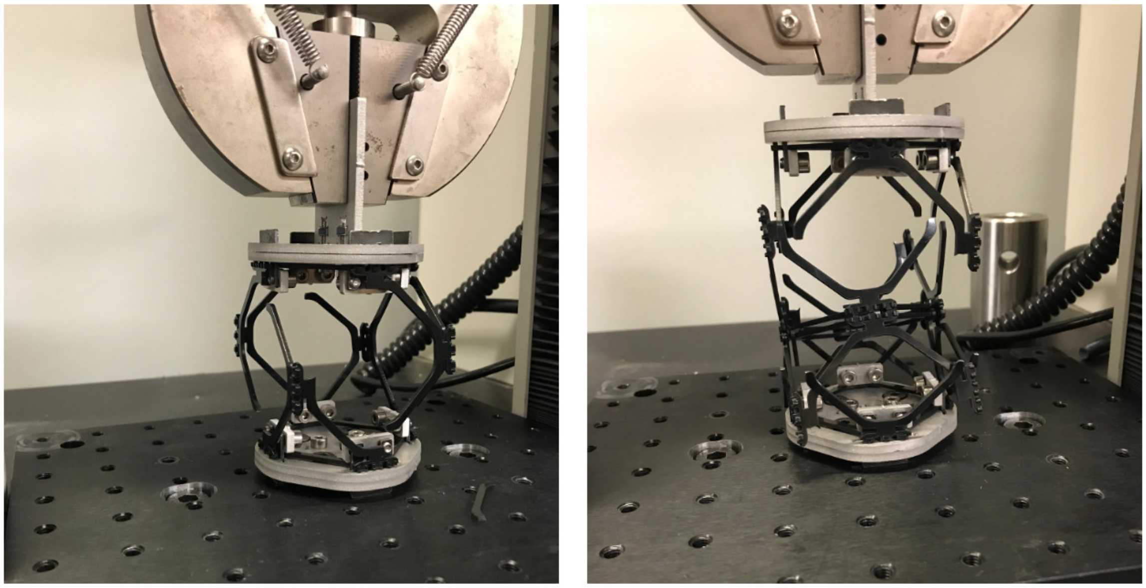

Figure 3-21: Photo of prevalent failure mode for acetal voxels. (Left) A single voxel loaded in tension until failure. (Right) A two voxel stack loaded in compression until failure.

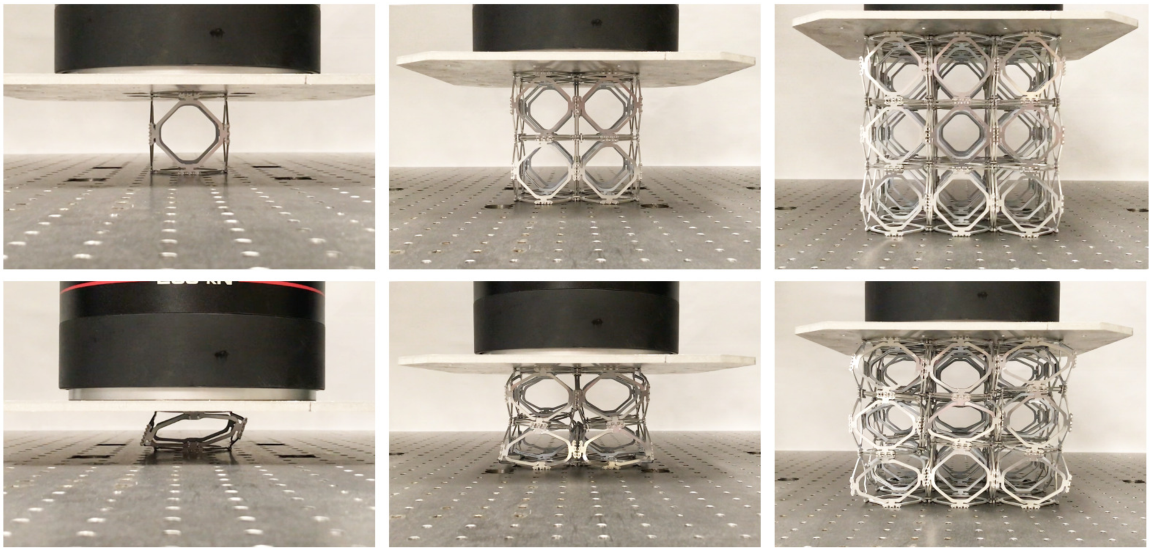

Figure 3-22: Compressive tests of aluminum PCB voxels. Upper images are un-deformed samples and lower images are compressed 30 mm.

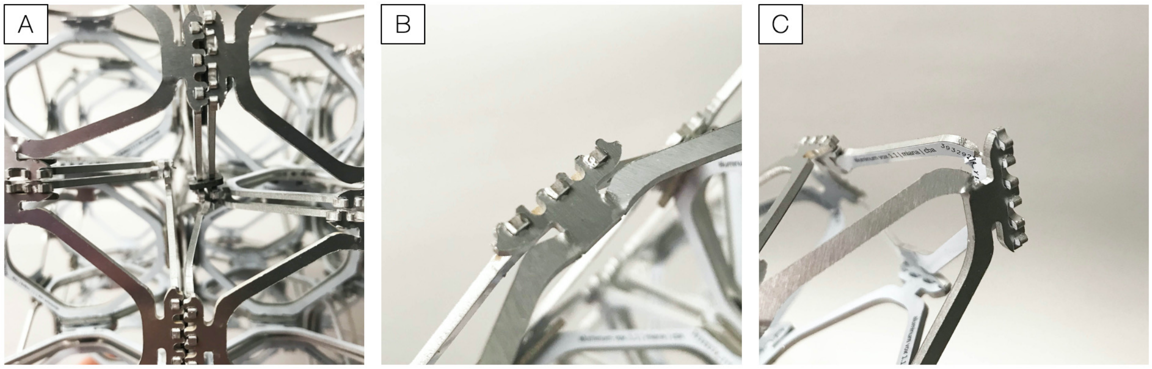

Figure 3-23: Photo of failure in the aluminum PCB voxels. A) Snap fit joint separation in 3x3x3 compression test. B) Plastic deformation in a voxel face in the 2x2x2 test— note that copper and silkscreen layers have fractured; not the aluminum. C) Plastic deformation and the beginning of fracture in the aluminum in a single voxel.

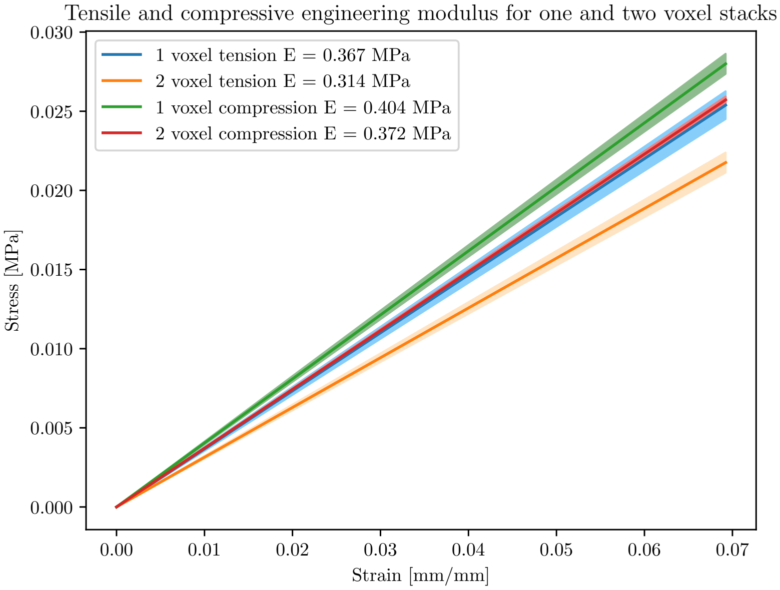

Figure 3-24: Tensile and compressive engineering modulus for one and two voxel stacks. (Delrin).

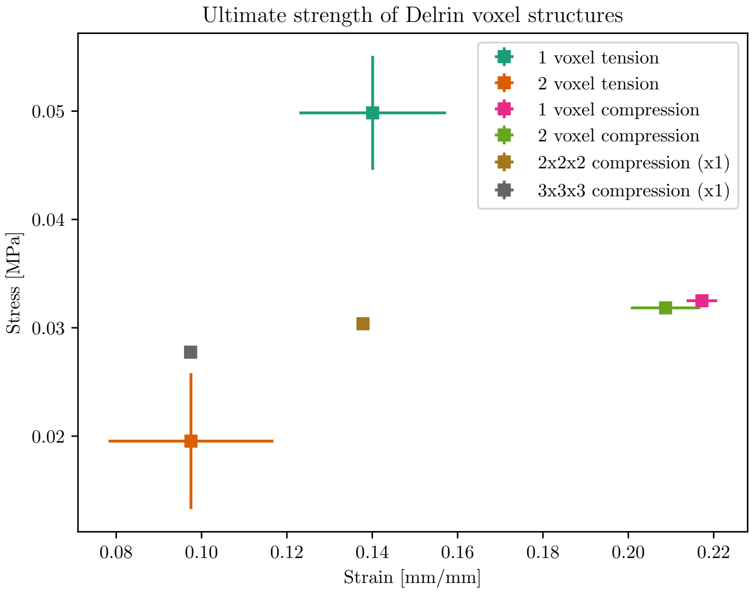

Figure 3-25: Ultimate strength of Delrin voxel structures.

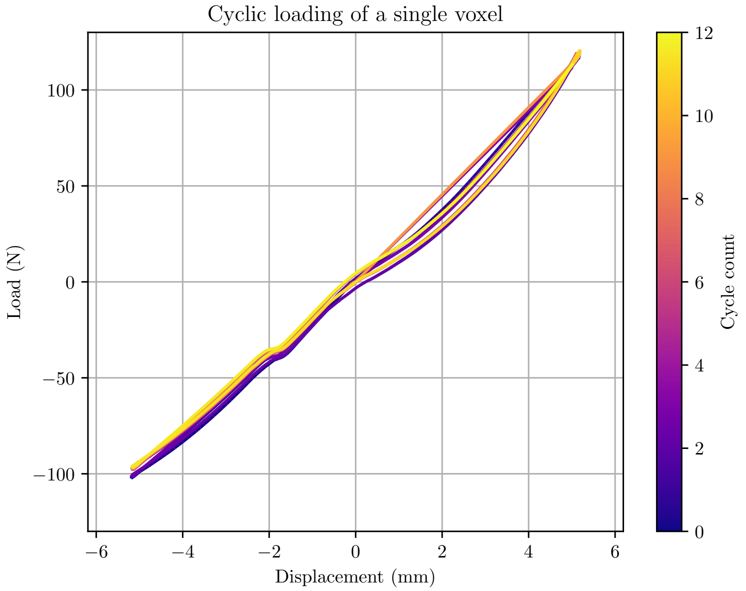

Figure 3-26: Cyclic loading of a single Delrin voxel, 12x cycles in the elastic region.

The difference between the one and two voxel stack moduli or stiffnesses can help us characterize the behavior of the joints between the voxels. The voxel stack can be approximated as a spring series, where the voxels are a spring with one stiffness and the joints are another spring. (This could be further extended such that within the joint, the clip is one spring, and the region in which it interfaces is another.) From the prior results, the nominal spring stiffness of a two voxel stack in tension with an infinitely stiff connector would be 11.92 N/mm. Its actual value is 10.21 N/mm, giving us a connector stiffness of 71.48 N/mm. The series spring stiffness equivalent for two voxels of equal stiffness with a connector in series is given by:

$$\frac{1}{k_{eq}} = \frac{2}{k_{vox}} + \frac{1}{k_{conn}}$$

So:

$$k_{conn} = \frac{1}{\frac{1}{k_{eq}} - \frac{2}{k_{vox}}}$$

The compressive moduli are higher than the tensile moduli (this difference is significant, though small). This difference is likely caused by the flexing at the constrained corners of the voxel that occurs in tensile loading but does not occur during compressive loading. Calculating out the approximate equivalent spring series where the compressive voxel is represented by one spring stiffness k_{vox} and the tensile one is represented as a series of k_{vox} with two springs on either end k_{joint} supports this conclusion. From this, I get $k_{joint} = 511.73 N/mm$. I can compare this to the k_{joint} stiffness I get if I compare the extrapolated k_{conn} value with the measured k_{conn} value from the connector-only testing set up (with compensation for the elasticity of the testing fixture). This calculation gives us $k_{joint} = 458.34 N/mm$, in a similar range to the prior calculation. These are both fairly approximate calculations, but they suggest that compliance from flexing at the constrained corners contributes to the mismatch in compressive and tensile stiffness and modulus.

Cyclic loading of a single voxel for 12 cycles, staying within the elastic range, shows minimal relaxation in the Young's modulus. A single voxel is displaced 5 mm in the compressive (negative) and tensile (positive) direction, where each compression-tension pair is counted as 1 cycle. A force-displacement plot of the single voxel behavior is shown in figure 3-26. This response shows the voxel under tension has a slightly larger hysteresis loop and more variability than the voxel in compression. Note that the small plastic region in the compressive regime is due to some slack in the testing set up.

The voxels also have different compressive and tensile ultimate strengths, which are shown in figure 3-25. Note that these reflect the maximum stress reached by the test specimen, not the yield stress. Figure 3-25 plots results for 1- and 2- voxels in tension and compression, as well as 2x2x2 and 3x3x3 voxel structures in compression (all for Delrin). Because the NxNxN tests were only done once, error bars are not plotted for them. These results are additionally summarized in 3.1. The voxels under tension have much higher variance in their ultimate strength than the compressive tests do. This is likely because in the 1- and 2- voxel tests, compressive failure occurs consistently at the beam-index joint, while in the tension tests, failure sometimes occurs in the clips, snap fits, or joints. In tension, the clip joints results in a much lower UTS— the joints fail before the beams do. In compression, the clip joints do not significantly contribute to a change in the ultimate strength. In compression, the NxNxN stacks show a small reduction in the ultimate strength of the structure as compared to the 1- and 2- voxel stacks. This difference may not be statistically signifcant, though likely reflects that the NxNxN structures experience failure in the inter-voxel connections. The premature joint failures are clearly visible in figure 3-28, which shows the full stress-strain results for all the Delrin NxNxN voxel tests.

The snap-fit style side clips and the basic side clips have comparable performance, as shown in 3.2. The stiffness for the basic side clips is 14.77 ± 0.37 N/mm and 13.89 ± 0.56 N/mm for the snap fit version. The maximum load reached for the basic clips is 46.63 ± 1.38 N and 48.14 ± 3.24 N for the snap fit version. This suggests I can switch between these clips without expecting significant change in the structure performance.

With the shear clips added, this joint reaches a stiffness of 82.65 ± 0.69 N/mm and a peak load of 209.64 ± 7.11 N. From this, the shear clips have an extrapolated stiffness

Test type	Ultimate strength	Strain
1 voxel tension	0.0498 ± 0.0053	0.140 ± 0.017
2 voxel tension	0.0195 ± 0.0063	0.098 ± 0.019
1 voxel compression	0.0325 ± 0.0003	0.217 ± 0.004
2 voxel compression	0.0318 ± 0.0001	0.209 ± 0.008
2x2x2 compression (x1)	0.0304	0.138
3x3x3 compression (x1)	0.0277	0.097

Table 3.1: Ultimate strengths of Delrin voxels.

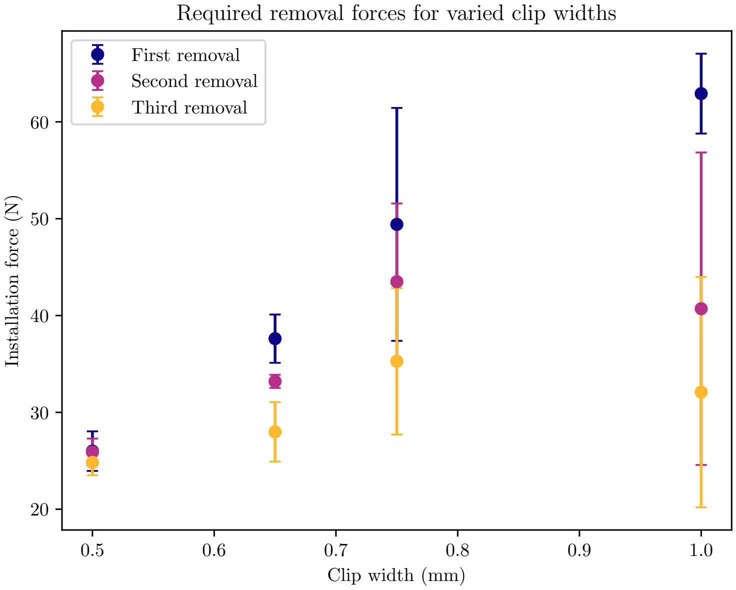

Figure 3-27: Average required removal force over three cycles versus vertical clip width.

of 67.88 ± 1.05. Along the snap axis (that is, if the voxel is pulled in the direction that disengages the snap fit), the shear clip has a stiffness of 17.24 ± 0.84 N/mm. This value is quite low, though this is likely due to a loose interface between this clip and the aluminum fixtures. The peak load in this orientation is 50.07 ± 8.39 N. The side clips and shear clips were additionally tested in shear— the fixturing proved insufficiently stiff for this and clear also deformed, so the test was only run once and the stiffness value represent a lower bound on the actual stiffness of this configuration. The stiffness was 35.15N/mm and the peak load was 403.09 N.

The snap fit vertical clips shown in figure 3-12 were tested to determine how their snap-

Part	Material	Test	Stiffness (N/mm)	Modulus (MPa)	Maximum load (N)
1 voxel	Delrin	compression	26.28 ± 0.60	0.404 ± 0.009	137.29 ± 1.45
1 voxel	Delrin	tension	23.84 ± 0.83	0.367 ± 0.012	210.56 ± 22.22
2 voxel	Delrin	compression	12.07 ± 0.10	0.372 ± 0.003	134.52 ± 0.46
2 voxel	Delrin	tension	10.21 ± 0.30	0.314 ± 0.009	82.55 ± 26.52
Side clips	Delrin	tension	14.77 ± 0.37	N/A	46.63 ± 1.38
Side snap fit clips	Delrin	tension	13.89 ± 0.56	N/A	48.14 ± 3.24
Side clips w/ shear clips	Delrin	tension	82.65 ± 0.69	N/A	209.64 ± 7.11
Side clips w/ shear clips	Delrin	shear	> 35.15	N/A	403.09
Shear clips	Delrin	tension	67.88 ± 1.05	N/A	N/A
Shear clips	Delrin	tension (snap axis)	17.24 ± 0.84	N/A	50.07 ± 8.39
1 voxel	Onyx	compression	20.09 ± 2.39	0.309 ± 0.037	105.34 ± 9.75
1 voxel	Aluminum PCB	compression	292.55	4.50	353.27
2x2x2 voxel	Delrin	compression	67.357	0.518	513.30
3x3x3 voxel	Delrin	compression	101.88	0.522	1055.14
2x2x2 voxel	Aluminum PCB	compression	658.64	5.07	2028.4
3x3x3 voxel	Aluminum PCB	compression	994.93	5.10	4576.68

Table 3.2: Mechanical properties of voxels and voxel components.

fit beam width impacted the required removal force for the voxel. This was done by testing 0.5 mm, 0.65 mm, 0.75 mm, and 1.0 mm width clips in tension using Delrin voxels. Clips were removed and re-inserted to see how the required force changes with use. The original intention behind these vertical clips was that their snap fits would be weak enough such that they could still be removed and reinstalled.

The clips have high variability in their required removal force— the snap fits often disengage at different times and in different amounts. The removal force increases as the beam width increases, as expected. However, as the clip is reused, the removal force (or strength) decreases more significantly for the wider clip widths. This is because the thicker clips are more likely to experience actual failure during removal.

The 0.65 mm width clips were determined to be the best selection for robotic assembly, as the 0.5 mm width clips suffered from some manufacturing challenges. These clips required an averaged insertion force of $26.96 \pm 2.02N$, and an average removal force of $37.61 \pm 2.49N$. Neither of these forces is especially tenable to the robot to achieve. The robot is able to insert voxels with this type of joint because it does not have to make every connection at the same time— by mating the voxel at a slight angle, the clips do not need to engage simultaneously, lowering the necessary force. The robot is not able to remove voxels with this type of joint (in a controllable way). These results contributed to the decision to move away from using this joint for the vertical connections.

At the level of a single voxel, the voxel behaviors are dominated by beams at the external boundary. The cuboct geometry only satisfies the Maxwell rigidity criteria when it becomes reasonable to assume its nodes are fixed, that is, when the voxel amount in a given axis is fairly large. If we count the boundary beams and internal beams for the cuboct geometry, the crossover point, where the structure has an equal amount of boundary and internal beams, occurs at N=3 in an NxNxN structure [20]. To better understand the ”bulk material” behavior of these voxels, compression tests of 2x2x2 and 3x3x3 voxel structures was done. It is left as future work with a more refined voxel design to extend the testing to 4x4x4 structures, in which the internal beam count has surpassed that of the boundary beams. The NxNxN tests are done with both Delrin and aluminum PCB voxels. The Delrin

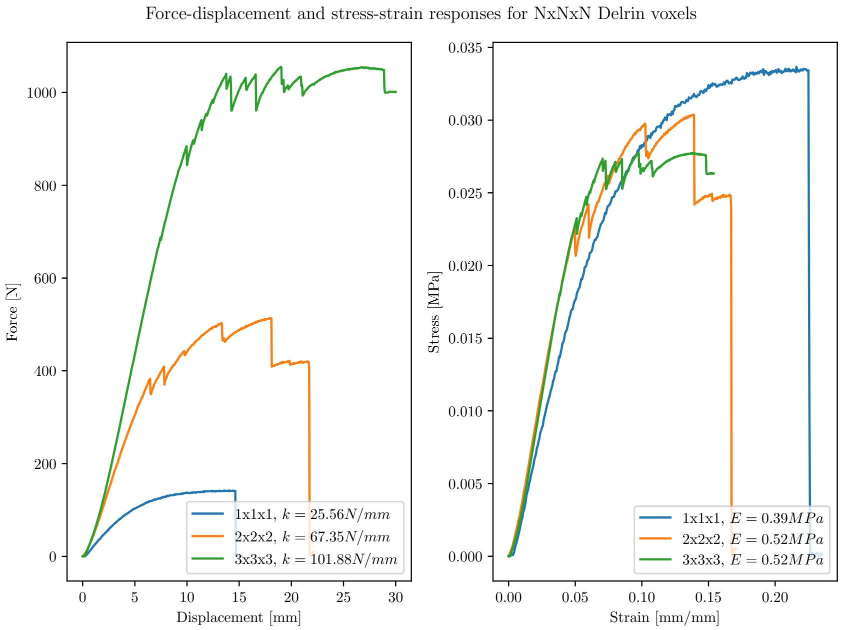

Figure 3-28: Force-displacement (left) and stress-strain (right) responses for NxNxN Delrin voxels. Stiffnesses (N/mm) and compressive modulus (MPa) are noted as well.

stress-strain response is shown in figure 3-28 and the aluminum one is shown in figure 3-32. Loading is done at a rate of 20 mm/min and each specimen is compressed 30 mm, or to failure, whichever occurs first.

As the voxel count increases, I expect the modulus to increase towards its continuum modulus. For Delrin, the compressive modulus is $E_1 = 0.39MPa$ for a single voxel, $E_2 = 0.52MPa$ for a 2x2x2 voxel assembly, and $E_3 = 0.52MPa$ for a 3x3x3 voxel assembly. So, for the Delrin system, the modulus levels off at $E = 0.52MPa$, by the 2x2x2 assembly, which is premature for the cuboct geometry. A plot of the measured modulus versus the simulated modulus using the simplified beam model is shown in figure 3-29. The simplified beam model, in which connectors are modelled as small beams at the corners of the cuboct geometry connecting voxels, represents the ideal behavior for this system. For the Delrin version, using the measured connector stiffnesses and the measured Delrin sample modulus, the simplified beam model substantially underpredicts the stiffness of the single voxel. As the voxel side length grows, it shows a similar approach to a continuum value as prior face-to-face cuboct does [16]. The measured values, instead, appear to level off. If I instead

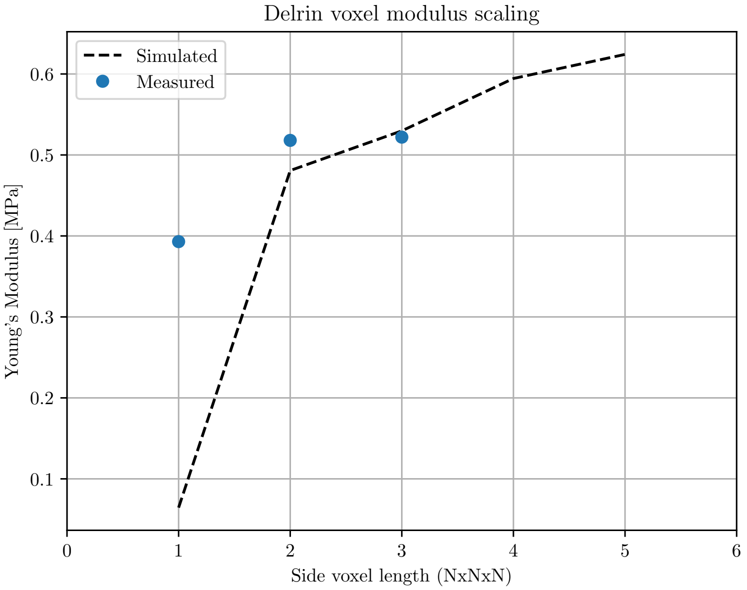

Figure 3-29: Compressive modulus scaling of NxNxN Delrin voxels.

simulate the system using a fully meshed FEA model of the voxel system, I see good agreement (as expected). The scaling of the compressive modulus for the measured versus FEA simulated results is shown in figure 3-30. The FEA version slightly underpredicts the stiffness at each side length; this is because the top surface where the compressive load is applied is not fully constrained, so the simulated beams are able to flex further.

The ultimate strength of the Delrin voxel assemblies decreases as N increases— the single voxel reaches a peak stress of 0.0325MPa while the 2x2x2 reaches 0.0304MPa and the 3x3x3 reaches 0.0277MPa. This is because the connectors and joints fail before the beams do, lowering the ultimate strength of the assembly. (The maximum load achieved increases; the 3x3x3 assembly reaches 1055.14 N, or sufficient to support up to approx. 107 kg (235 lbs)— 395x its own weight.) The stress-strain response more clearly shows that the first joint failure happens at the same stress for both the 2x2x2 and 3x3x3 system, with some later connector failures also occurring in similar places. These failures represent a mix between failures in the intra-voxel snap fits (the snap fits disengage) and failures in the clips (the clips disengage, or break.)

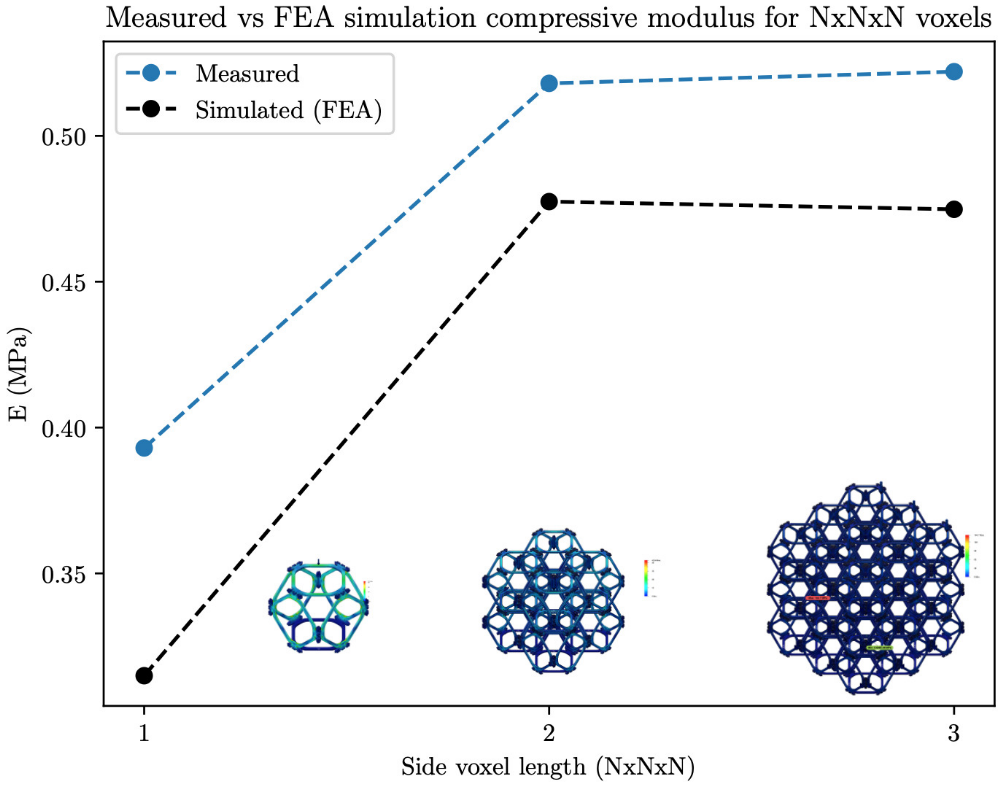

Figure 3-30: Measured versus FEA simulated compressive modulus results for Delrin NxNxN voxels.

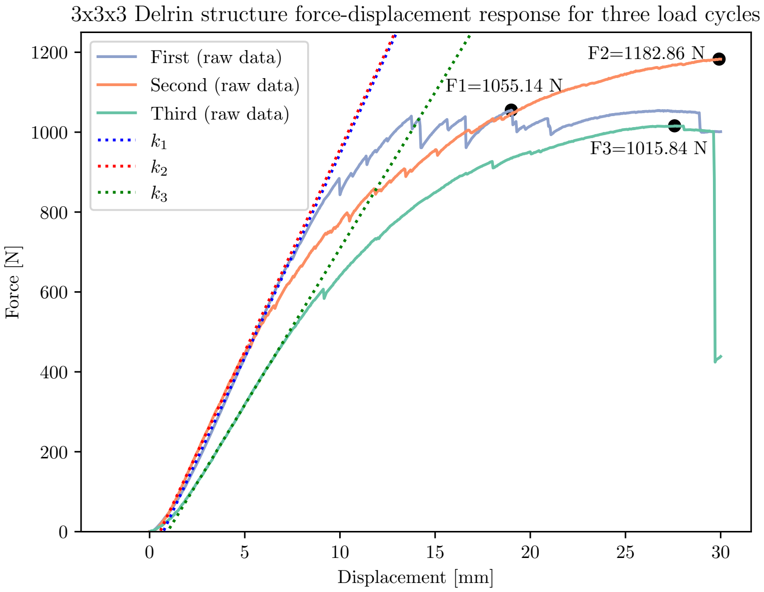

Figure 3-31: Force-displacement response of 3x3x3 Delrin voxels over three load cycles. The stiffnesses (slopes of the dotted lines) are: k1 = 101.88 N/mm, k2 = 101.21 N/mm, and k3 = 78.23 N/mm.

Because the 3x3x3 Delrin structure did not fail at 30mm displacement, the structure was re-loaded an additional 2 times. During the unloading, the failed snap fits appear to heal. Audibly, the snap fits re-snap, and visual inspection suggests that the joints and voxels were not significantly damaged. The force-displacement response of these is shown in figure 3-31. Between the first and second loading, the stiffness does not change very much— from k1 = 101.88 N/mm to k2 = 101.21 N/mm— though the elastic region is shortened. Surprisingly, the second loading achieves much higher maximum load. The second loading reached F2 = 1182.86 N, as compared to the first loading at F1 = 1055.14 N. The response also does not feature as many steep drops as the joints or connectors fail— this suggests that perhaps the first loading-unloading cycle may have somehow annealed the joints, enabling a higher strength. The third loading is more sensible— the stiffness decreases to k3 = 78.23 N/mm and the ultimate strength decreases to F3 = 1015.84 N. The third loading experienced major failure at approx. 29.5 mm displacement, in which multiple beams and joints failed simultaneously.

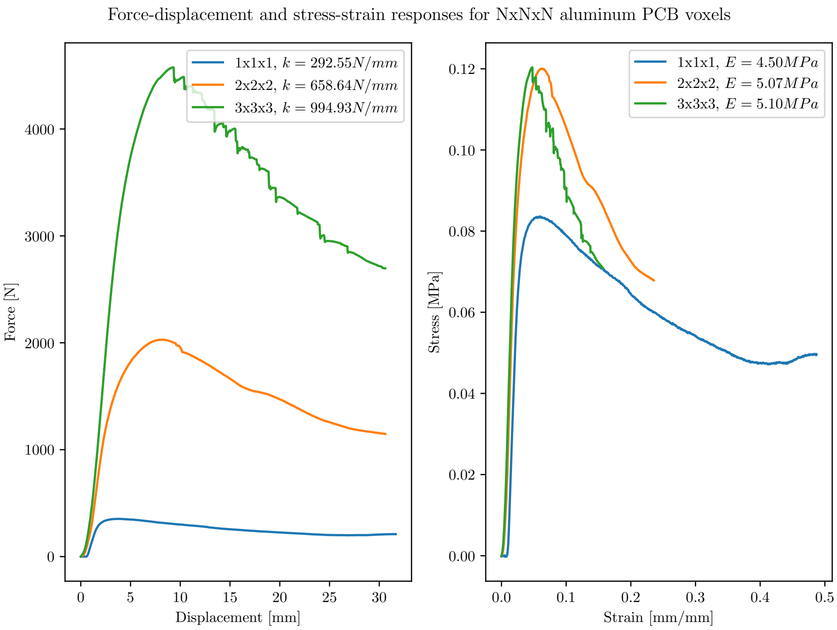

Figure 3-32: Force-displacement (left) and stress-strain (right) responses for NxNxN aluminum PCB voxels. Stiffnesses (N/mm) and compressive modulus (MPa) are noted as well.

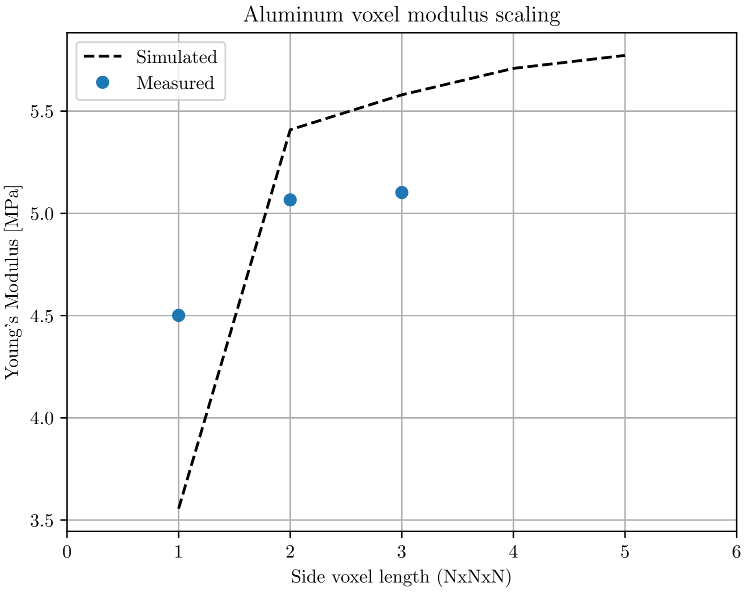

Figure 3-33: Compressive modulus scaling of NxNxN aluminum voxels.

The same tests were done for the aluminum PCB voxels. (The repeated loading was not done, as the aluminum voxels plastically deformed). The force-displacement and stress-strain responses are shown in figure 3-32. The compressive modulus scaling of NxNxN aluminum voxels as compared to the simplified beam model simulation is plotted in figure 3-33. The single voxel has compressive modulus E1 = 4.50MPa, the 2x2x2 has compressive modulus E2 = 5.07MPa, and the 3x3x3 has compressive modulus E3 = 5.10MPa. Similar to the Delrin response, the aluminum version also levels off early, as can be seen in the comparison to the simulated version. Note that the values for the simplified beam model of the aluminum version are likely less accurate than that of the Delrin one, as the aluminum clips were not tested.

The strength of the aluminum voxel assemblies does increase as the voxel amount increases, though this also levels off. The single voxel reach a peak stress of 0.08 MPa, the 2x2x2 reaches 0.12 MPa, and the 3x3x3 reaches 0.12 MPa. In terms of peak force, the 3x3x3 voxel structures reaches 4576.68 N, which means that it can support approximately 10^3 times its own weight (or around half a US ton at maximum). Unlike the Delrin voxels, the maximum strength of the aluminum voxels is not limited by critical failure in the joints— the entire structure has begun plastically deforming by the time two joints disconnect (visible as two small sharp changes in the force-displacement graph). None of the other joints or clips in the 2x2x2 structure disconnect— the whole structure is failing in a more ductile way. In the 3x3x3 structure, the joints within voxels do fail. Many of the snap-fits disconnect, which is visible in the very jagged response of the 3x3x3 structure. The aluminum PCB voxels are more susceptible to this kind of failure because the PCB manufacturing process used does not support making the sharp internal angles needed for a good snap fit. The snap fits are instead rounded, making them easier to disassemble.

I suspect the reason that the measured modulus does not scale more similarly to the way the ideal one does is that the connector indexing features are acting like a flexure. It is clear that the beams are hinging at those corners (or in the case of Delrin, snapping), and this behavior is likely preventing the better scaling. Similarly, the simulated modulus scaling for a compliant cuboct geometry in [16] shows a flatter scaling as the flexural element is

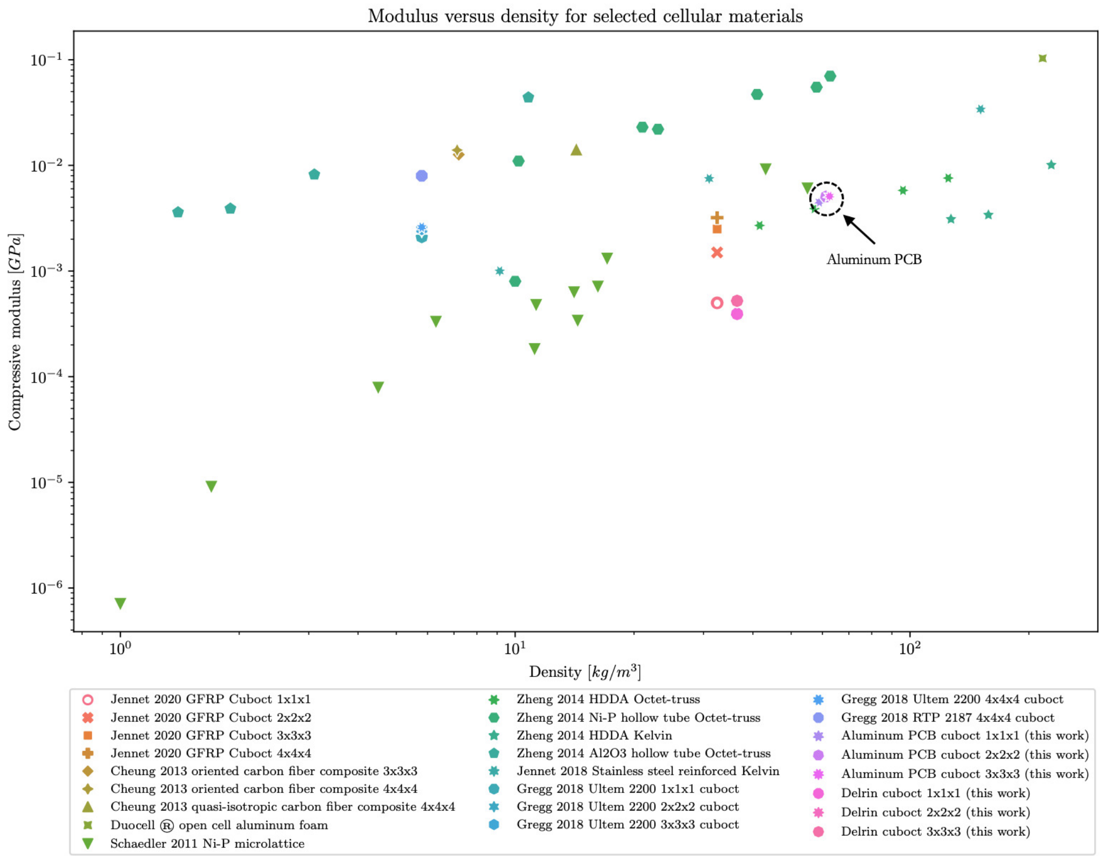

Figure 3-34: Ashby plot of compressive modulus versus density for selected cellular materials, including voxels from this thesis. Aluminum PCB voxels are highlighted. Selected cellular materials from: Jennet 2020 materials from [16], Cheung 2013 materials from [15], Duocell ®open cell aluminum foam from [50], Schaedler 2011 materials from [45], Zheng 2014 materials from [46], Jennet 2018 materials from [48], and Gregg 2018 materials from [47].

increased. By modifying the voxel face and the connectors to reduce the size of the indexing feature, I may be able to achieve closer to the simulated scaling.

Another limiting factor is the strength of the joints— their premature failure limits the strength of voxel assemblies. For plastic and composite voxels, using clips made from e.g. aluminum or stainless steel would improve this. Choosing a much stiffer material has the added benefit of shifting the mechanical performance closer to that of the non-offset voxel case. For example, based on the results for a simulated cantilevered beam in figure 3-18, using stainless steel clips with an acetal voxel with the current clip geometry would result in the clip-equivalent beam having the same stiffness as the beam elements, which puts the displacement difference between the offset and non-offset faces within 95%. The stiffness of the current clip geometry is limiting though— future work should include re-visiting this design such that the clip geometry itself can be stiffer, without needing to switch to stiffer base material.

An Ashby plot of compressive modulus versus density is shown in figure 3-34, including the voxels discussed in this work, as well as other similar cellular materials, such as prior CBA voxel systems and various microlattices from literature. The aluminum PCB voxels are specifically annotated. These are in a similar range as other cellular material systems. If the system were able to achieve the ideal cuboct scaling behavior, I would expect to be even more competitive with the other existing cellular materials.

3.4 Early Design Explorations

Some prior voxel designs are shown in figure 3-35. A) and B) show a version of the voxel that used stacked friction fits coupled with a clip for constraining the remaining degrees of freedom. This version utilized a similar assembly scheme as the current voxel system, though featured face-to-face connections and friction fits instead of snap fits. I moved away from this version because I determined that the voxel-to-voxel connection system would be too difficult for the assembly robot to achieve— the voxels would have required structures to have been built up hierarchically, from voxels to beams, from beams to planes, and

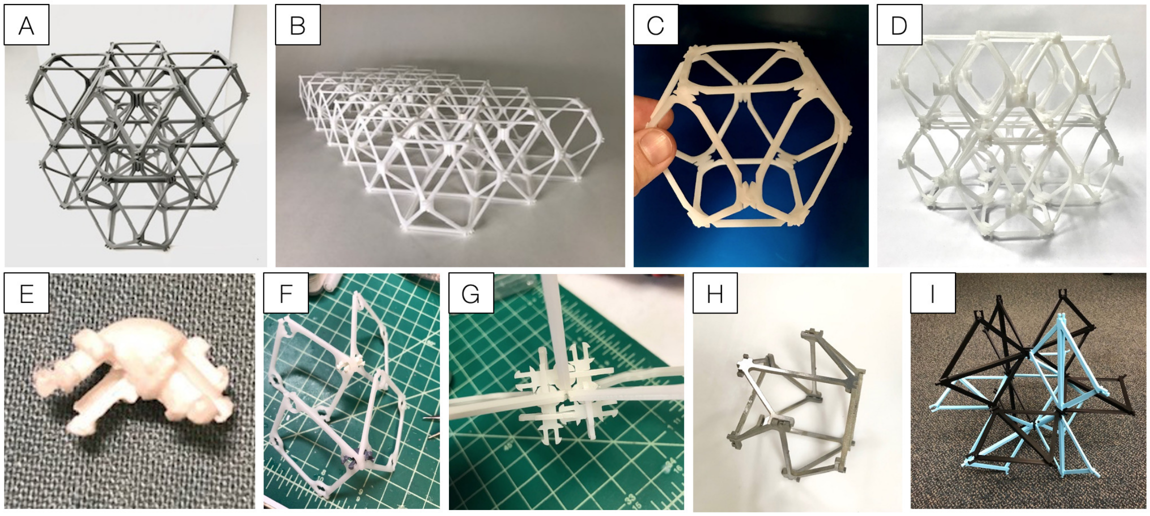

Figure 3-35: Some prior voxel designs for this thesis.

from planes into structures, to prevent collision in the joint system. C) and D) show early iterations of the construction voxels used in the thesis. C) shows a single voxel— the initial version used a single snap fit at the joints, this was expanded to three primarily to increase the stability of the cuboct during assembly, making it easier to assemble. D) shows some earlier versions of the clip system used in this thesis.

E), F), and G) explore a face-to-face voxel connection scheme in which I have one set of planar voxel faces and one set of one type of connector element, which could either be 3-dimensional (E and F) or planar as well (G). E) shows an exploration of a 3D printed connector element for the voxels (the production scale version would have been injection molded), inspired by plastic rivets. This element used stacked snap-fits and inserted into the corners of voxels, fully constraining the joint within the voxel. It would then snap-fit into the face of the next voxel. This system would also necessitate horizontal assembly, which would be more challenging for the robot to assemble (especially to generate the necessary force to make the snap fits). A similar system to this is shown in G, in which planar parts assembly into the connector instead. This would have enabled easier use of a stronger material for the connector. In an effort to keep part count down (only two unique parts— faces and joints), the design in G created a lot of material overhead for each joint. It additionally did not constrain the faces well until most of the faces were attached (i.e., two voxels were connected to each other).

H) and I) show earlier explorations into alternate version of the cuboct geometry. To mitigate the double-faces, and to enable better flat-packing, this design splits the cuboct face using a bracing beam. Face-to-face connections complete the cuboct lattice, while the extra beam can provide additional support. H) uses press-fits as a temporary joint for prototyping. I) uses a snap fit and clip design, which necessitated a voxel decomposition as shown in the image, which is not especially compatible with the style of robot explored in this thesis. The concept for this design was originally developed by A. Abdel-Rahman.

3.5 Conclusion

This chapter demonstrates that the construction voxel system used in this thesis is structural. Voxels made from both acetal and aluminum are characterized, and their behavior is compared against simulated ideal behaviors. The voxels perform well— a 3x3x3 assembly of aluminum voxels is able to reach a peak load of 4576.68 N with compressive modulus 5.1MPa at a density of $62.5kg/m^3$. This means that the robotic assembly system in this thesis is able to assemble lightweight, high stiffness structures. See figure 3-36 for a ”qualitative” mechanical test of an aluminum voxel structure.

The current system has limitations that will be addressed in future work. As the voxel count increases, the effective modulus of the current design does not scale the way a cuboct lattice is expected to. This is likely caused by unintentional compliance in indexing features in the voxels. Next steps include revising and refining the voxel face design and testing the new design. The strength of the structures is limited by the strength of the joints. Future work will focus on revising the geometry of the clips to improve their ultimate strength. Additionally, minor geometry changes should improve the stiffness of the clips as well, and will be done in future work.

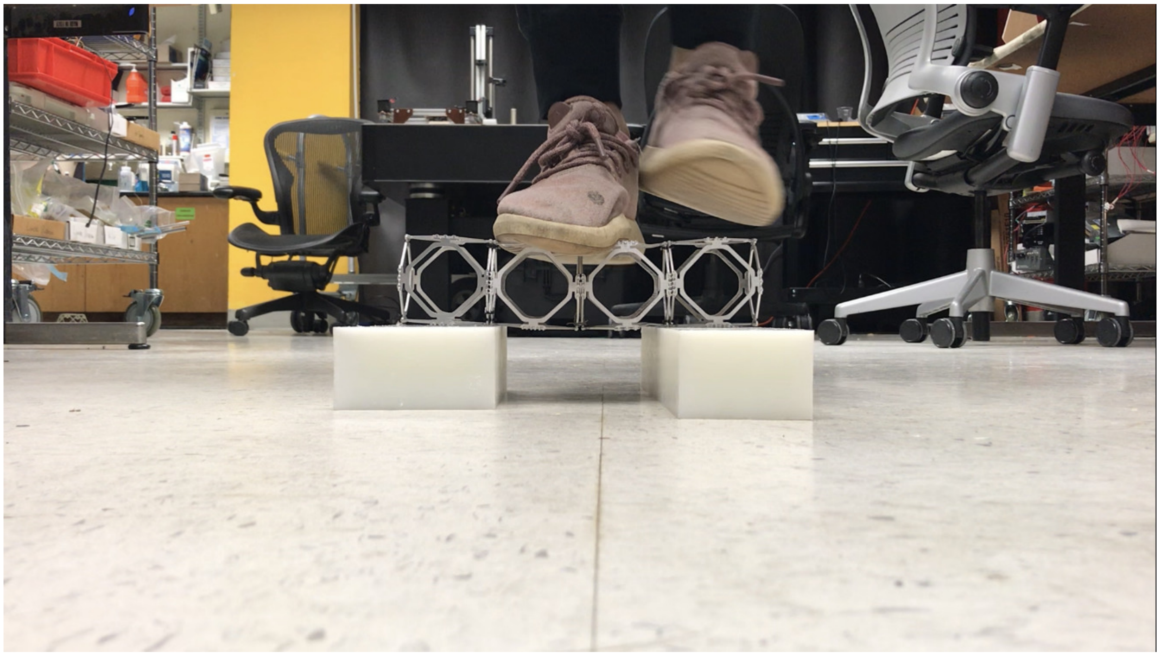

Figure 3-36: A three point bending test of sorts. Freezeframe from video of qualitative mechanical testing of the voxel system.

Chapter 4

Conclusion

We can approach the construction of large, complex structures the way nature does: with (somewhat) simple, massively parallel, iterative assembly systems. By adopting a voxel feedstock and designing our assembly robot to live within this voxel ecosystem, we can achieve structures with design-able and desirable material properties while simultaneously lowering the design requirements for the robots. The underlying lattice provides the robots with a repetitive, known environment in which to operate. The interface between the voxels and the robots self-corrects, enabling the construction of systems larger than the robot itself. By additionally decomposing the robot into a modular feedstock, we can reconfigure the size and nature of our robotic swarm to the construction task at hand. This reconfiguration— in which the robot follows a recursive assembly process to build structures, self-reproduce, and build larger robots— enables a time-to-size scaling of construction similar to that of natural systems.

This thesis demonstrates an early hardware version of this type of system. A robotic system for assembling voxels with structural connections has been demonstrated. The material system has been characterized via mechanical testing, which verifies that it is structural, lightweight, high-stiffness system. The robotic system has further demonstrated one level of self-reproduction, in which the robot assembles another robot out of a modular feedstock. This system comprises the necessary elements for recursive and hierarchical assembly system, though large-scale assembly of structures with and without electromechanical functionality has not yet been implemented. The goal is to ultimately produce a recursive and hierarchical robotic assembly system such that we can efficiently and autonomously construct large and complex structures.

4.1 Next steps

There is still, however, quite a distance to go! Much of the current work has been focused on achieving structural assembly and the first layer of robot-recursion. Future plans are to not only improve on the current assembly abilities, but to deploy hierarchy in the hardware system. That is, the project is working toward the construction of larger or more diverse robots as well as the integration of robots or other electromechanical functionality directly into assembled structures. I split the next steps between currently ongoing efforts, and longer-term arcs.

In the near term, there are still challenges associated with both the robotic assembler and construction voxel system, which need to be addressed (these are discussed in more detail in their respective chapters). The robotic assembler has not yet been pushed to assemble large structures (or even medium structures for that matter), so many of the difficult questions associated with scaling the system up still remain unknown. To push this system forward, a near term project will focus on assembling a small bridge, at the scale shown in figure 4-1. The shown bridge assembly would comprise eighty voxels, an order of magnitude more than the current system has assembled. These larger assembly projects will also likely provide opportunities to further explore collaborative behavior between robots. As the structure grows, the limitations of the individual robot become more pronounced, necessitating collaboration, as well as the implementation of the algorithms required for it.

Currently, the robotic modules and functional voxels presented have only been used as modules for the robotic assembler. However, these components are intended to have broader functionality as a framework for building electromechanical functionality into structures and for exploring alternate robotic configurations. The construction and reconfiguration of these

Figure 4-1: Immediate next steps. A) Plot of a small voxel bridge. B) Partially assembled voxelcopter. C) Partially assembled ant robot.

active structures can be automated using the robotic assembly, enabling both robot and swarm level optimization for a given task. To this end, the modules need to be tested in other robotic systems, and the family of module types needs to be expanded.

An ongoing project on this front is a second iteration of the Voxelcopter project, using the hardware developed here. A partially assembled version of this voxelcopter is shown in figure 4-1. This provides an opportunity to test the robotic assembly of an electromechanical system, which will help elucidate the assembly requirements for the functional voxels in varied configurations. It additionally provides the opportunity to develop further module types, and to evaluate the electrical and mechanical performance of the functional voxels under very different loads from the assembly robot.

Another ongoing project on this front is the development of a robotic ”ant” using these robotic modules. The partially assembled ant is shown in figure 4-1. The ant is a quadruped with four two-DoF arms, using elbow joints oriented perpendicular to each other. This work is in collaboration with A. Abdel Rahman, who is developing reinforcement learning workflows for finding robotic gates, such that the robots can reconfigure and function more autonomously. The ant represents a starting geometry toward this work, which will also necessitate the development of likely a new actuator set to achieve more dynamic locomotion, as well as sensor modules for feedback on e.g. robot orientation and contact forces.

Longer time frame goals are aimed toward improving the autonomy of the recursive and hierarchical process. The current system uses modules which leverage off-the-shelf components and have much of the assembly complexity shifted from the steps the robot does to the steps I do (e.g., the robot only needs to stack modules and run current, while I have assembled the many components together into the modules). Future work will eventually have to address this and shift the balance of complexity to the robots, or else the system will not be able to scale to truly large quantities or applications where its autonomy would be most beneficial. On one front, this may be addressed via the development of other robots or machines specifically for putting together robot components. On another front, more conceptually consistent with a recursive, hierarchical system, the active elements of the robots themselves may also become discrete assemblies, as in [51].

The current ways in which we manufacture objects, structures, almost anything at a large size scale or amount, is not sustainable. This thesis presents an alternate strategy for building, but does not evaluate it in terms of sustainability (admittedly, much of this is too early stage for this to be necessarily productive). Future work will eventually have to evaluate this, and we are currently working on shifting feedstocks toward more renewable sources for voxels. The ultimate hope of this system is that it, or something like it, can improve the way we build big complex things in a way that is beneficial to our broader ecosystem. In very broad strokes, future work will be directed toward achieving that.

Bibliography

[1] J. Neubert, A. Rost, and H. Lipson, "Self-soldering connectors for modular robots," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1344–1357, 2014.

[2] F. Nazzi, "The hexagonal shape of the honeycomb cells depends on the construction behavior of bees," *Scientific reports*, vol. 6, no. 1, pp. 1–6, 2016.

[3] S. D. Z. W. A. Animals and Plants, "Sociable weaver." `https://animals.sandiegozoo.org/animals/sociable-weaver`. Accessed: 2022-09-16.

[4] C. R. Reid, M. J. Lutz, S. Powell, A. B. Kao, I. D. Couzin, and S. Garnier, "Army ants dynamically adjust living bridges in response to a cost–benefit trade-off," *Proceedings of the National Academy of Sciences*, vol. 112, no. 49, pp. 15113–15118, 2015.

[5] K. H. Petersen, N. Napp, R. Stuart-Smith, D. Rus, and M. Kovac, "A review of collective robotic construction," *Science Robotics*, vol. 4, no. 28, p. eaau8479, 2019.

[6] A. Abdel-Rahman, C. Cameron, B. Jenett, M. Smith, and N. Gershenfeld, "Self-replicating hierarchical modular robotic swarms," *Communications Engineering*, vol. 1, no. 1, pp. 1–10, 2022.

[7] C. Reid, "Army ant bridge." `https://images.theconversation.com/files/395419/original/file-20210416-23-1axomur.JPG?ixlib=rb-1.1.0&rect=46%2C496%2C5184%2C2592&q=45&auto=format&w=1356&h=668&fit=crop`, 2021. Accessed: 2022-11-16.

[8] C. Honeybees, “Festooning bees.” `https://carolinahoneybees.com/wp-content/uploads/2022/02/bees-festooning-in-empty-frame.jpg`, 2022. Accessed: 2022-11-16.

[9] WASP, “Big delta.” `https://www.3dwasp.com/wp-content/uploads/2016/08/bigdelta12m_campo-16-uai-667x667.jpg`, 2016. Accessed: 2022-11-16.

[10] InceptiveMind, “Hadrian x.” `https://www.inceptivemind.com/wp-content/uploads/2020/07/Hadrian-X.jpg`, 2020. Accessed: 2022-11-16.

[11] Airbus, “Airbus belugast loading helicopter.” `https://mediaassets.airbus.com/permalinks/563971/wpr/beluga-transport-test-loading.jpg`, 2022. Accessed: 2022-11-16.

[12] M. Talagani, S. DorMohammadi, R. Dutton, C. Godines, H. Baid, F. Abdi, V. Kunc, B. Compton, S. Simunovic, C. Duty, *et al.*, “Numerical simulation of big area additive manufacturing (3d printing) of a full size car,” *SAMPE J*, vol. 51, no. 4, pp. 27–36, 2015.

[13] R. D. Caballar, “Bricklaying robot brings automation to construction sites.” `https://redshift.autodesk.com/articles/bricklaying-robot`, 2022. Accessed: 2022-11-16.

[14] S. AeroSystems, “Distinctive capabilities (rd).” `https://www.spiritaero.com/programs/research-technology/distinct-capabilities/`. Accessed: 2022-11-16.

[15] K. C. Cheung and N. Gershenfeld, “Reversibly assembled cellular composite materials,” *science*, vol. 341, no. 6151, pp. 1219–1221, 2013.

[16] B. Jenett, C. Cameron, F. Tourlomousis, A. P. Rubio, M. Ochalek, and N. Gershenfeld, “Discretely assembled mechanical metamaterials,” *Science advances*, vol. 6, no. 47, p. eabc9943, 2020.

[17] B. Jenett, A. Abdel-Rahman, K. Cheung, and N. Gershenfeld, “Material–robot system for assembly of discrete cellular structures,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4019–4026, 2019.

[18] C. Drew and J. Mouawad, "New challenges for the fixers of boeings 787," *The New York Times*.

[19] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, *et al.*, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, no. 5, p. 055006, 2019.

[20] B. B. E. Jenett, *Discrete mechanical metamaterials*. PhD thesis, Massachusetts Institute of Technology, 2020.

[21] K. Cheung, "Assembly of madcat." `https://news.mit.edu/2019/engineers-demonstrate-lighter-flexible-airplane-wing-0401`, 2019. Accessed: 2022-11-16.

[22] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of m-tran iii modular robotic system," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 373–386, 2008.

[23] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A. Ijspeert, "Roombots: A hardware perspective on 3d self-reconfiguration and locomotion with a homogeneous modular robot," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1016–1033, 2014. Reconfigurable Modular Robotics.

[24] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, "3d m-blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1925–1932, 2015.

[25] P. Swissler and M. Rubenstein, "Fireant3d: a 3d self-climbing robot towards non-latticed robotic self-assembly," 11 2020.

[26] S. GINGICHASHVILI, "The real transformer." `https://thefutureofthings.com/3309-the-real-transformer/`. Accessed: 2023-04-16.

[27] C. Atwell, "Epfls modular roombots can create different pieces of furniture." `https://www.hackster.io/news/`

`epfl-s-modular-roombots-can-create-different-pieces-of-furniture-bbd5348a273c`, 2021. Accessed: 2023-04-16.

[28] J. Neubert and H. Lipson, "Soldercubes: a self-soldering self-reconfiguring modular robot system," *Autonomous Robots*, vol. 40, no. 1, pp. 139–158, 2016.

[29] J. D. CSAIL, "Self-transforming robot blocks jump, spin, flip, and identify each other." `https://news.mit.edu/2019/self-transforming-robot-blocks-jump-spin-flip-identify-each-other-1030`, 2019. Accessed: 2023-04-16.

[30] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[31] N. Melenbrink, A. Wang, and J. Werfel, "An autonomous vault-building robot system for creating spanning structures," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7066–7072, IEEE, 2021.

[32] B. Jenett and K. Cheung, "Bill-e: Robotic platform for locomotion and manipulation of lightweight space structures," in *25th AIAA/AHS Adaptive Structures Conference*, p. 1876, 2017.

[33] C. G. Cameron, Z. Fredin, and N. Gershenfeld, "Discrete assembly of unmanned aerial systems," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 339–344, IEEE, 2022.

[34] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3293–3298, 2012.

[35] D. Brandt, D. J. Christensen, and H. H. Lund, "Atron robots: Versatility from self-reconfigurable modules," in *2007 International Conference on Mechatronics and Automation*, pp. 26–32, 2007.

[36] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2485–2492, 2010.

[37] S. Miyashita, M. Hadorn, and P. E. Hotz, "Water floating self-assembling agents," in *Agent and Multi-Agent Systems: Technologies and Applications* (N. T. Nguyen, A. Grzech, R. J. Howlett, and L. C. Jain, eds.), (Berlin, Heidelberg), pp. 665–674, Springer Berlin Heidelberg, 2007.

[38] Z. Fredin, "Assembling integrated electronics," Master's thesis, Massachusetts Institute of Technology, 2021.

[39] R. Tedrake, *Underactuated Robotics.* 2023.

[40] T. A. Composites, "Aluminum honeycomb core—aerospace grade." `https://www.toraytac.com/product-explorer/products/UCD6/Aluminum-Honeycomb-CoreAerospace-Grade`. Accessed: 2023-04-16.

[41] D. Kytyr, V. Petranova, and O. Jiroušek, "Assessment of micromechanical properties of trabecular bone using quantitative backscattered electron microscopy," pp. 119–122, 06 2012.

[42] C. Gaulon, J. Pierre, C. Derec, L. Jaouen, F.-X. Bécot, F. Chevillotte, F. Elias, W. Drenckhan-Andreatta, and V. Leroy, "Acoustic absorption of solid foams with thin membranes," *Applied Physics Letters*, vol. 112, p. 261904, 06 2018.

[43] M. F. Ashby, T. Evans, N. A. Fleck, J. Hutchinson, H. Wadley, and L. Gibson, *Metal foams: a design guide.* Elsevier, 2000.

[44] T. A. Schaedler and W. B. Carter, "Architected cellular materials," *Annual Review of Materials Research*, vol. 46, pp. 187–210, 2016.

[45] T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, and W. B. Carter, "Ultralight metallic microlattices," *Science*, vol. 334, no. 6058, pp. 962–965, 2011.

[46] 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, no. 6190, pp. 1373–1377, 2014.

[47] C. E. Gregg, J. H. Kim, and K. C. Cheung, "Ultra-light and scalable composite lattice materials," *Advanced Engineering Materials*, vol. 20, no. 9, p. 1800213, 2018.

[48] B. Jenett, N. Gershenfeld, and P. Guerrier, "Building block-based assembly of scalable metallic lattices," in *International Manufacturing Science and Engineering Conference*, vol. 51388, p. V004T03A053, American Society of Mechanical Engineers, 2018.

[49] O. Formoso, C. Gregg, G. Trinh, A. Rogg, and K. Cheung, "Androgynous fasteners for robotic structural assembly," in *2020 IEEE Aerospace Conference*, pp. 1–8, 2020.

[50] E. Aerospace, "Aluminum foam: Duocel open cell foam."

[51] W. Langford and N. Gershenfeld, "A discretely assembled walking motor," in *2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, pp. 1–6, IEEE, 2019.