Material-Robot System for Assembly of Discrete Cellular Structures

Benjamin Jenett[1], Amira Abdel-Rahman[1], Kenneth Cheung[2], and Neil Gershenfeld[1]

***Abstract*— We present a material-robot system consisting of mobile robots which can assemble discrete cellular structures. We detail the manufacturing of cuboctahedral unit cells, termed voxels, which passively connect to neighboring voxels with magnets. We then describe "relative" robots which can locomote on, transport, and place voxels. These robots are designed relative to and in coordination with the cellular structure-- the geometry of the voxel informs the robot's global geometric configuration, local mechanisms, and end effectors, and robotic assembly features are designed into the voxels. We describe control strategies for determining build sequence, robot path planning, discrete motion control, and feedback, integrated within a custom software environment for simulating and executing single or multi-robot construction. We use this material-robot system to build several types of structures, such as 1D beams, 2D plates, and 3D enclosures. The robots can navigate and assemble structures with minimal feedback, relying on voxel-sized resolution to achieve successful global positioning. We show multi-robot assembly to increase throughput and expand system capability using a deterministic centralized control strategy.**

***Index Terms*— Assembly; Space Robotics and Automation; Path Planning for Multiple Mobile Robots or Agents**

I. Introduction

AUTOMATED processes for material deposition and manipulation are becoming more prevalent in state-of-the-art production of structural systems. Additive manufacturing can produce hierarchical, architected metamaterials with novel properties unattainable with traditional engineering materials [1]. High performance, continuous fiber composite aircraft components can be made by automated tape laying, utilizing large gantries, tooling, and autoclaves [2]. In these examples, high repeatability is achieved with computational control systems and stationary machines with stiff motion axes. However, the scale of the built object is limited to a fixed bounding envelope.

Manuscript received: February 24, 2019; Revised May 18, 2019; Accepted June 15, 2019. This paper was recommended for publication by Editor Dan Popa upon evaluation of the Associate Editor and Reviewers' comments. Research supported by NASA STMD Game Changing Development Program through the Automated Reconfigurable Mission Adaptive Digital Assembly Systems (ARMADAS) project, NASA Space Technology Research Fellowship (NNX14AM40H), and CBA consortia funding.

[1]B. Jenett and A. Abdel-Rahman, and N. Gershenfeld are with the Center for Bits and Atoms, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (emails: bej@mit.edu, amira.abdel-rahman@cba.mit.edu, gersh@cba.mit.edu). Corresponding author: bej@mit.edu

[2]K. Cheung is with NASA Ames Research Center, Intelligent Systems Division (Code TI), Moffett Field, CA 94035 USA (email: kenny@nasa.gov)
Digital Object Identifier (DOI): see top of this page.

Mobile robots promise larger scale construction. Aerial, wheeled, and ambulatory platforms combine various material deposition techniques, yet tradeoffs remain between robotic complexity and the quality of the resulting artifact. Even with automation, some limits, such as stochastic errors, are inherent to the continuous, or monolithic, materials typically used.

An alternative strategy is based on the assembly of discrete, building-block elements into larger functional structures, using reversible connections to allow for disassembly and reuse. Here, the modularity of the structure can inform design of mobile robots for assembly. By ensuring local robot precision and reliability, robots can build structures larger and more precise than themselves. This approach is studied here, for assembly of a modular, space-filling lattice structure. We present the resulting material-robot system, control strategies, experiments to demonstrate core functionalities, and simulations to study system tradeoffs and scaling.

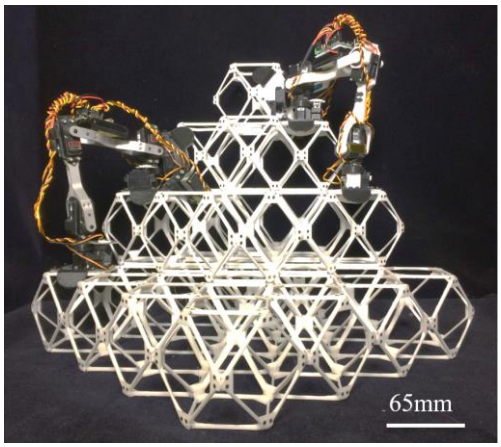

Figure 1: Mobile robots which assemble a discrete cellular structure. This pyramid structure consists of 30 cuboctahedral voxels.

II. Background

In this paper, we are motivated by the automated production of large scale, lightweight, high performance structures, with applications in aviation and aerospace. Aircraft manufacturing relies on expensive and precise tooling to produce designs which stray little from the "tube with wings" motif, primarily due to the reliance on legacy as a basis for cost, safety, and performance risk reduction [3]. Non-standard geometries are economic non-starters, even with performance gains [4], due to the cost-sensitive and risk-averse nature of the airline industry. The ability to produce arbitrary aerostructure geometries at low-cost can potentially disrupt this trend [5].

Space missions require high stiffness-to-weight booms and trusses for baselines, apertures and collecting surfaces, whose performance and output increase with characteristic length scale [6]. However, mass and volume constraints of rocket payloads necessitate innovations such as deployable structures, whose complexity increases risk and adds to the already high cost of sending material to orbit (~$\$10^4$/kg). On-orbit assembly seeks to bypass this, relying on incremental construction using raw materials [7] or pre-formed structural modules [8], and is seen as a key technology for the future of space exploration missions [9]. Automating these processes can reduce risk and cost. In terms of robot control, algorithms which enable autonomy and robustness are desirable to avoid failure and to enable response to unforeseen conditions.

Robotic platforms for producing such structural systems can be described in terms of motion systems, material format, mechatronics, sensing and positioning, and construction algorithms. Additive manufacturing has become a standard for rapid prototyping, with commercially available platforms offering highly repeatable, cost-effective production. More custom platforms can scale down to nm scale features on cm scale parts [10], while desire for larger (>1m) parts has led to larger machines [11]. Macro-scale printing [12] deposits cementitious material with building-scale gantries, though the resolution and efficiency seen in smaller scale printing worsen. Part quality from deposition processes can suffer from anisotropy due to build orientation [13], and stochastic errors inherent to the continuous medium that are difficult and costly to detect [14]. Stationary platforms for discrete assembly typically use strut-and-node or brick elements. The former is exemplified by a multi-DoF robotic arm mounted to linear motion systems to construct tetrahedral truss plates for substructure of space apertures [15]. Smaller, gantry-based motion systems for assembly of interlocking, functional electronic bricks [16], or parallelized deposition of spherical voxels joined with an adhesive binder [17], have been shown.

Mobile robotic approaches seek to extend the boundaries of these processes. A mobile crane, mounted with a smaller robot arm, can reach large extents while maintaining local precision for deposition [18], though tests were performed from a single stationary position. Within a LiDAR boundary, collaborative wheeled robotic arms can perform collective deposition [19]. Fiber-based systems have also been demonstrated, including mobile robots which act as an inflatable mandrel for construction of fiber-composite tubes [20], and multi-robot systems for constructing fiber structures [21]. Repeatable deposition quality and rigorous quantification of structural performance of the built objects remain open challenges.

Mobile robots building discrete structures also tend to use either modularized strut-and-node truss elements or some form of brick. UAVs guided by LiDAR assemble magnetically connected struts and nodes [22] or passively stacked bricks [23], though these systems have a bounded work envelope. For systems without global positioning, localized laser scanning [24] or optical camera fiducial recognition [25] can be used for bricks or truss modules, respectively. In order to reduce sensing complexity, tactile or IR sensors can be used to recognize local positioning markers, while robot movement is assisted by passive geometric alignment features. This has been demonstrated on both truss [26][27] and brick structures. The latter are especially relevant here and are closest to our approach of "relative" robotics. AMAS [28] is a similar robot morphology to ours, with bricks that transmit data, power and mechanical interconnect between neighbors. TERMES [29] is capable of 2.5D motion and assembly, using local markers and rules to enable single and multi-robot construction. Algorithmically, we see tradeoffs between centralized and distributed systems. This was studied by Costa et al [30], and results indicate that systems with large numbers of robots and parts benefit from a distributed control architecture, while a centralized approach can provide near-optimal solutions for smaller systems. The latter will be employed in this work.

Recently, discrete, modular lattice elements with reversible connections have been assembled into materials and structural systems with record setting mechanical properties [31], large-scale reconfigurability [32], and mission-adaptive performance [33]. These lattice elements can be mass produced with best-practice manufacturing techniques, ensuring repeatability and affordability [34]. Stationary robotic platforms have been used to automate assembly on small (<1m) scales [35]. Further, it has been shown that mobile, task-specific robots can be designed relative to the discrete cellular structure for locomotion, manipulation, and inspection [36] [37]. For these material systems, it is accepted that the proper joining of elements is critical to achieving desired global behavior [34], and that this process can be accomplished with a variety of fastening systems, such as nuts and bolts [32], shear clips [31], or cable-ties [33]. However, the approach taken here is to use simple, passive magnetic connections to reduce robot end effector complexity and allow for rapid development of assembly strategies. A more robust design will be addressed in the *Discussion* section.

The main contribution of this paper is a coordinated material-robot system consisting of 1) discrete voxels for 3D isotropic construction, 2) mobile robots designed relative to the voxel structure, enabling precise locomotion and assembly in all directions, regardless of gravity vector, with minimal feedback, and 3) a deterministic centralized control architecture for single and multi-robot simulation and control.

III. Methodology

Central to the approach described here is the coordinated design of the voxels and the robots: while the voxels are optimized for properties such as structural performance, steps are taken to ensure compatibility with the robotic platform. Similarly, the robots are designed relative to the specific geometry of the voxels and the resulting cellular structure.

A. Discrete cellular material system

The voxel geometry used here is a cuboctahedron (referred to herein as Cuboct), with lattice pitch P = 101.6mm (4in), and strut length l = 71.84mm (2.83in). While there are numerous lattices and corresponding cell geometries [38], this cell geometry is selected for favorable scaling of mechanical

properties such as specific stiffness [34] and aspects pertinent to robotic assembly, as described in [39]. To summarize, flat faces, as opposed to vertices, provide more contact area for alignment. Also, the addition of a single voxel provides 4 joined vertices, ensuring proper load transfer, as opposed to the octahedral voxel which requires 4 voxels to achieve this condition. The Cuboct cell is decomposed into 6 uniform faces (Figure 2). The faces are 3D printed PLA, oriented parallel to the build plate to promote continuous material orientation along the strut axes. 3.175mm cube magnets are then press-fit into pockets at the nodes, with opposing magnet poles oriented as shown in Figure 2. This ensures that the faces, and thus the voxels, are rotationally symmetric and orientation agnostic. Faces are then joined at their vertices using interlocking features and a screw which acts in shear, with partial cold-tapping into the plastic for retention. Each voxel consists of 6 faces, 12 screws, and 48 magnets, and weighs 35g.

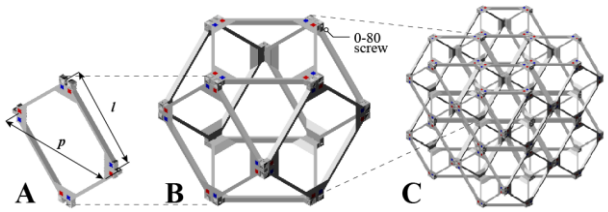

Figure 2: Discrete cellular material system. A) Modular face with press-fit magnets, colored to indicate orientation. B) Cuboct voxel consisting of 6 faces, joined at their vertices with interlocking features and a 0-80 screw, C) Assembled 2x2x2-voxel cube

B. *Mobile robotic platform*

Our robot is based on an inchworm archetype [40]. The geometric configuration is driven by functional requirements for navigating the cellular environment, as described in [36]. Besides requiring a new end effector to interact with voxels, the most critical update from the previous version is a redesign of the leg motor hubs. Motors have been relocated to align with the robot primary axis, reducing the robot width to that of a single voxel, which allows the robot to pass by a wall without collision. This requires a new fork-type interface to attach legs to the servo horns and a low-profile bearing pressed into the back of the motor housing. The robot comprises nine servo motors, is powered by a 6V rechargeable battery, and communicates wirelessly with a central controller using 2.4 GHz radio (Figure 3).

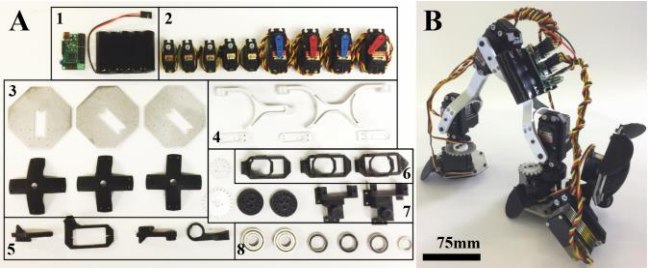

Figure 3: Robotic platform and components. A) 1. Control board and battery, 2. Motors, 3. Foot end effectors, 4. Legs, 5. Part placement mount, 6. Main motor hubs, 7. Ankle hubs and gears, 8. Bearings. B) Built robot.

The end effectors for foot attachment and voxel placement are the same. The gripping mechanism is designed to use as many passive and as few active degrees of freedom (DoFs) as possible. Alignment features constrain translation in *x* and *y*, and rotation in *z*. A single actuator rotates a cruciform "key" 45º to create four contact points with customized surfaces on the underside of each beam of the top square face of the voxel (Figure 4C). This constrains *z* translation and *x* and *y* rotation. The actuation DoF is orthogonal to the loading direction, thus decoupling these functions and requirements (Figure 4B). When the key is in the locked position, the stiffness of the gripper is a function of the key's geometric and material properties, as opposed to the torque capacity of the motor.

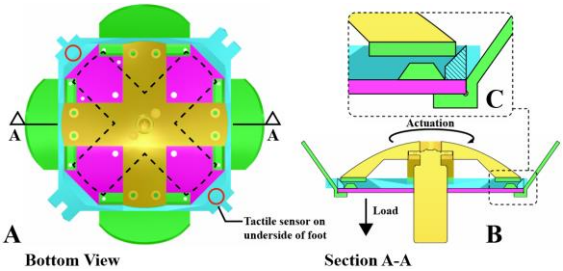

Figure 4: Gripping end effector. A) Bottom view of voxel face, shown in blue. Ungripped key position is shown in dashed lines, allowing entry/exit to/from a voxel. B) Section view of key in gripped position, C) Magnified view showing passive and active constraints engaging with voxel feature.

C. *Software environment*

Our software environment for simulation and robot control was developed in JavaScript. This allows us to leverage existing visualization packages and be web-enabled. A screenshot of the environment is shown in Figure 5 and can be seen further in the video supplement. To the left is a 3D visualizer of the voxel structure, and a simplified model of the robot. To the right are several inputs and outputs. Inputs include robot geometric parameters and start and end target locations. Outputs include motor angles and step numbers.

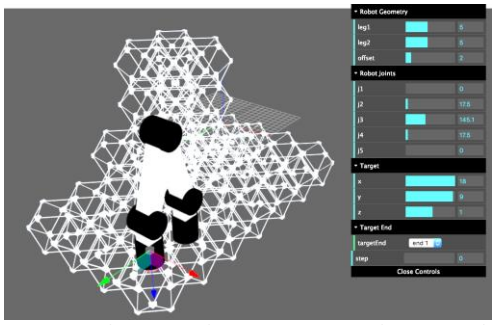

Figure 5: Screenshot of custom software environment for design of discrete cellular structures, and simulation and control of mobile robotic assembly.

D. *Control architecture*

The overall control architecture is shown schematically in Figure 6. Individual control levels will be described in order from high to low level in the following subsections. To summarize, our control scheme works as follows: CAD models of a given discrete cellular structure can be represented as a 3D bit matrix, or as functional representations (f-reps) in our custom software environment. Next, a build sequence is determined, based on environmental/operational constraints. Depending on single or multi-robot construction, path

planning for each voxel in the build sequence is sent to the robot. Each path consists of linked primitive moves, such as "step" or "turn 90º". Primitive moves consist of a list of motor positions, which when executed in order, result in the desired motion. Lastly, sensors detect errors during motion execution, and thus can provide feedback to avoid failure.

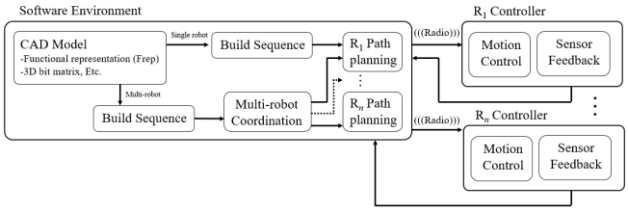

Figure 6: Schematic control hierarchy. The software environment centrally computes build sequences and path planning and delivers strings of motion commands to robots through serial over 2.4 GHz radio.

We employ a centralized control system, whereby a single platform (in this case, a laptop), computes and schedules single and multi-robot coordination. While this is well understood to be susceptible to a single point of failure, it allows for near-optimal path computation, and reduces robot sensing and computing requirements [30]. Development of a distributed approach is addressed in the *Discussion* section.

E. Build sequence

We determine the order in which voxels are placed based on three main considerations. First is the number of robots. Multiple robots allow multiple build fronts, or group construction on a single build front, but require coordination to avoid collision and deadlock, which will be addressed later. Second is environmental factors such as gravity. In a 1g environment, it is reasonable to build in layers parallel to the ground plane, moving up in the *z* direction. In a zero-gravity environment, this restriction would not exist. Algorithms for zero-g construction will be addressed in future work. Lastly, we must consider where the part source, or pickup location is, and how many locations there are, as this will drive where construction begins as well as possible build directions.

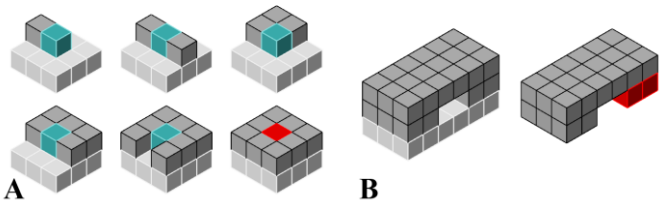

Figure 7: Geometry Specifications and Building Constraints: A) Placement constraints based on the current robot design, B) Unconnected areas for a build layer are possible with connectivity below, under-hangs are not possible.

The proposed building sequence algorithm imposes geometric constraints of what can be built. We assume the built geometry is a connected geometry, though horizontal layers can have unconnected areas, as long as each area has at least one voxel connected to the layer underneath (Figure 7B). A voxel can be built at layer *n* as long as it has at least one empty neighbor in the same layer (Figure 7A). If a sacrificial layer is required as shown, this could be removed later by a robot similar to [37], but would not be needed in zero gravity.

Two algorithms for choosing the building sequence will be presented, one is a deterministic algorithm that is more suited for single robot construction, and the other is more flexible for multi-robot construction. The first algorithm assumes the simple case of single robot with one set pickup station. The building sequence is layer by layer, and in each layer the voxels are sorted based on the Manhattan distance to the pickup station location. In this case the entire build sequence can be precomputed. For structures with no overhangs and no unconnected areas at any layer, no extra consideration is needed to determine the building sequence (Figure 8A).

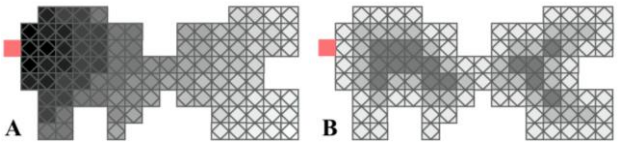

Figure 8: Layer Building Sequence comparison between the proposed algorithms: A) Algorithm 1: Manhattan distance values to the pickup station B) Algorithm 2: Distance field values/ranks

For structures with overhangs and cantilevers, the voxels are sorted differently. At each layer, we find the voxel with both the smallest Manhattan distance to the pickup station and an existing neighbor underneath it. Then, the rest of the voxels are sorted based on the Manhattan distance to this first voxel. In the case of layers with unconnected areas, for each area, we have to find the voxel with both the smallest distance and connected to the layer underneath it and start to build around it. This sorting algorithm is presented here in pseudo code:

Algorithm 1: Sorting algorithm for single robot construction

```
1: For each layer I in structure
2:     set v_not_sorted to voxels in layers i
3:     set v_reference to pickup_location
4:     set v_sorted to empty list
5:     While v_not_sorted length not = to v_sorted length
6:         If v_reference is the pickup_location then
7:             V is the closest voxel to v_reference and
                   neighbor at layer i-1 exists
8:             set v_reference to V
9:             add V to v_sorted
10:        Else
11:            V is the closest voxel to v_reference and at
                   least one neighbor in v_sorted
12:            If V doesn't exist then
13:                set v_reference to pickup_location
14:            Else
15:                add V to v_sorted
16:            End if
17:        End if
18:    End while
19: End for
```

The second algorithm is a more general approach that is suited to complex structures and multi-robot construction. Here, we rank the voxels layer by layer based on a distance field to the closest edge (Figure 8B). Each voxel has a discrete rank and the robots are forced to follow this rank order, hence

building the center of the shape first then layer by layer around it in order to prevent deadlocks. For multi-robot construction, the centralized system does not have a fully pre-determined building sequence. All voxels with the same rank can be built simultaneously allowing for more parallelization and flexibility in terms of the number of robots and pickup in the system. At any point in time, when a robot picks up the voxel at the pickup station, it is assigned to build the voxel with the rank *r* that is closest to the pickup station. If another robot is completing the last voxel of rank *r*, the robot at the pickup station will wait for these intermediate voxels to be built and only start building the new voxels with rank *r+1* when they are done. This dependency is shown in Figure 10C. While our algorithms allow for overhangs, this relies on structural properties that exceed those of our current system. This will be useful for future versions with high performance connections.

F. Path planning

After having determined the build sequence, an individual robot needs a path from a pickup location to a target location, given an existing voxel configuration. Here we employ a method that uses A* search which minimizes the function:

$$f(n) = g(n) + h(n) \quad (1)$$

Where *g(n)* is the cost of moving from the start to voxel *n* and *h(n)* is the estimated cost of moving from voxel *n* to the goal. The Manhattan Distance between two voxels is used as the heuristic function. For single robot construction, all paths are computed offline, which are then sent to the robot. Individual path sequences consist of linking together primitive motions to move the robot from starting location to target location. In the case shown in Figure 9, the goal is to place the voxel in the bottom left corner of the image. To accomplish this, the robot follows a path sequence which is exported as a string to be read and executed by the motor controller.

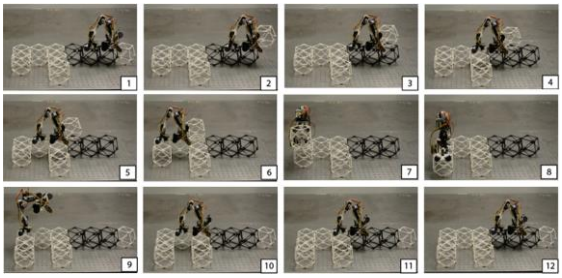

Figure 9: Step-by-step path sequence. 1) Stand, 2) Pick up voxel, 3) Step back, 4) Step back, 5) Step back, 6) Step back, 7) Turn 90º, 8) Place voxel, 9) Turn -90º, 10) Step forward, 11) Step forward, 12) Step forward

G. Multi-robot coordination

In case of multiple robot construction, when using the second algorithm, we wish to avoid robot collision. The centralized system is aware of the location of the deployed robots at each timestep *t* (based on the previously calculated paths) and uses this when performing the next path search. If no such path exists, the robots wait for a certain amount of timesteps until a path is cleared. We call this strategy spatio-temporal scheduling, and it is similar to a strategy studied by Murata and Terada [41]. Whereas in that example they rely on distributed robot communication and independent agents to perform individual searches, in our centralized approach we search for a collection of robots, noting performance decrease for larger systems. We look at several cases and show how enforcing certain rules will prevent collision (Figure 10). Cases A and B look at beginning and ending intersections, case C looks at construction dependencies, and case D looks at mid-path intersection:

1) *Case A:* $t_{start\ 1,3} \neq t_{start\ 2,4}$; $t_{end\ 1,3} \neq t_{end\ 2,4}$; $t_{start\ 1,2} \neq t_{end\ 4,3}$
2) *Case B:* $t_{start\ 1} \neq t_{start\ 2,\ end\ 4}$; $t_{end\ 3} \neq t_{start\ 2,\ end\ 4}$
3) *Case C:* $t_{end\ 2} > t_{end\ 1,\ start\ 3}$
4) *Case D:* $t_{1,3\ between\ i\text{-}j} \neq t_{2,4\ between\ i\text{-}j}$

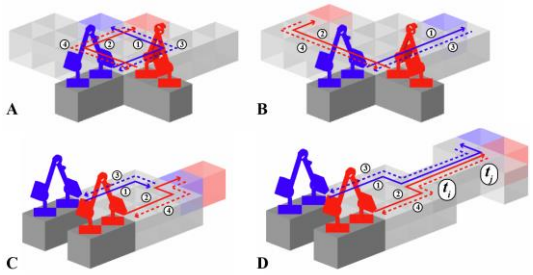

Figure 10: Spatio-temporal scheduling. Paths are determined to include updated built structure and to avoid collision.

H. Motion Control

Primitive motions consist of individual motor positions, linked as a sequence to execute the full motion. For example, stepping forward one cell consists of the following primitives: 1) Grip back foot, ungrip front foot, 2) Step front foot forward, 3) Grip front foot, ungrip back foot, 4) Step back foot forward. A list of the main primitive motions is presented in Table 1.

TABLE I. PRIMITIVE MOTION DESCRIPTIONS

Motion	***Description***
Grip/ungrip	Rotation of end effector key between 0-45º
Step forward/back	Parabolic trajectory described by $y = -x^2+Px$, where *P* is lattice pitch
Turn 0, 90, 180, 270º	Turning occurs at back foot, front can turn +/- 90º due to placement end effector
Pick up/place voxel	Voxels can be placed on plane or above plane
Step up/down	Stepping up/down more than 1 voxel level requires concave/convex cornering
Concave/convex corner	Convex corner requires furthest reach, which determines minimum leg length

For multi-motor motions, we use inverse kinematics to determine motor angles, which are simplified by enforcing end effector orientation normal to the travel plane (Figure 11):

$$\theta_1 = \tan^{-1} 2(x, y) - \tan^{-1} 2(a_2 \sin\theta_2, a_1 + a_2 \cos\theta_2) \quad (2)$$

$$\theta_2 = 2\tan^{-1}\sqrt{(a_1^2 + a_2^2)^2 - (x^2 + y^2)/(x^2 + y^2) - (a_1^2 + a_2^2)^2} \quad (3)$$

$$\theta_3 = 540° - \theta_1 - \theta_2 \quad (4)$$

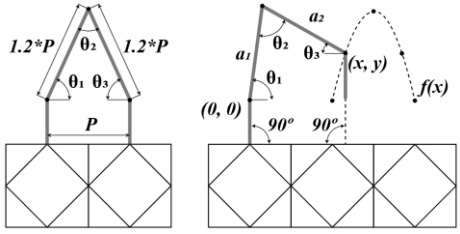

Figure 11: Inverse kinematics parameters. Here the robot follows a path using the parabolic function $y = -x^2 + Px$, where P is lattice pitch.

I. Feedback

Feedback is provided by tactile sensors on opposite corners of each foot (Figure 12). When both sensors are engaged, we can be confident that the foot is correctly placed. A control loop enforces this as a condition to be satisfied prior to locking the newly placed foot and unlocking the other foot to be moved next. Feedback for part placement can be obtained by monitoring torque of the placement motor while pulling on a newly attached voxel. It is assumed that this will be more critical for a fully structural connection scheme, and thus placement verification is left for future work.

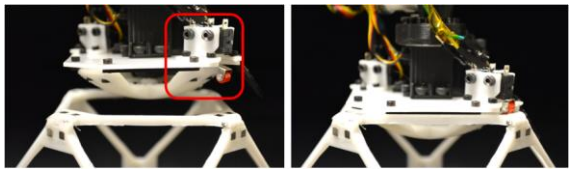

Figure 12: Tactile sensors on foot end effectors. Unengaged (L) and engaged (R). Feedback provides confirmation of successful foot placement.

IV. Experiments

The experiments presented herein were performed on an optical table, which has two important properties to note. One is flatness, which is cited as +/- 0.1 mm (0.004 in.) over a 60 x 60 cm (2 x 2 ft) area. The other is a steel surface the voxels magnetically adhere to it. This provides a controlled build environment, which contributes to successful assembly. Regardless of the configuration of the final structure, there is a minimum base "seed", which consists of a row of three voxels (colored black) upon which the robot initially stands and picks up new voxels, which are refilled manually. Video of these experiments can be found in supplementary material.

A. Single robot construction

We used a single robot to build three kinds of structures: a 1D beam, a 2D plate, and a 3D enclosure (Figure 13 A-C). Results for these experiments are shown in Table 2. Also, we demonstrated 3D locomotion where a robot transports a voxel over a "wall", requiring concave and convex cornering and vertical stepping (Figure 13F), which is possible due to the 3D isotropic cellular structure and gripping end effectors.

B. Multi robot construction

Two robots were used to build a branching structure and a pyramid (Figure 13D, 13E). In the former, the robots build separate single-voxel beams, which then turn to intersect. The successful demonstration of independent robot construction on an interconnected structure is significant because it supports the idea that local part metrology determines global precision. The pyramid structure is made up of three layers (5x5x1, 3x3x1, 1x1x1), and demonstrates implementation of multi-robot coordination with spatio-temporal scheduling. While this structure is essentially 2.5D and therefore could be built using other robotic assembly systems, it shows that we can use our approach to build a structure with a part count comparable to state of the art (on the order of 2^5) [29].

We can compare these initial demonstrations to prior art for mobile robotic assembly and more traditional manufacturing approaches. In Figure 14, we compare various fabrication process, including additive manufacturing, stationary and mobile robotic discrete assembly, and manual assembly, looking at scale and volumetric throughput. Due to the assembly of sparse structures, we demonstrate a comparatively high volumetric throughput, while still being comparatively limited in scale. In future work, our system will be used to build larger scale structures to access new, unoccupied areas of this parameter space (large scale with high throughput).

TABLE II. Robot Experiment Results

	1D	*2D*	*3D*	*Branch*	*Pyramid*
# of voxels	6	9	23	23	35
Motion types	5	7	17	8	10
Total motions	36	70	207	338	352
Avg motion time (sec)	10.5	10	9.5	10	10.5
Build time (min)	6.67	12	33	28.17	30.75
Avg time/voxel (min)	1.1	1.33	1.41	1.21	0.87

V. Simulations

In order to study tradeoffs and scalability of our system, we perform a variety of simulations in our software environment. This allows us to rapidly compare parameters such as number of robots, number of pickup stations, and number of voxels, while monitoring both number of steps and overall construction time (Figure 15). First, we assess the impact of number of robots and part pickup locations (both ranging from 1 to 4) for a 5x5x5 cube. In Figure 15C, we focus specifically on how to decrease time, comparing between the effect of increasing robots versus increasing pickup stations. Although both decrease time, robot count has a greater impact.

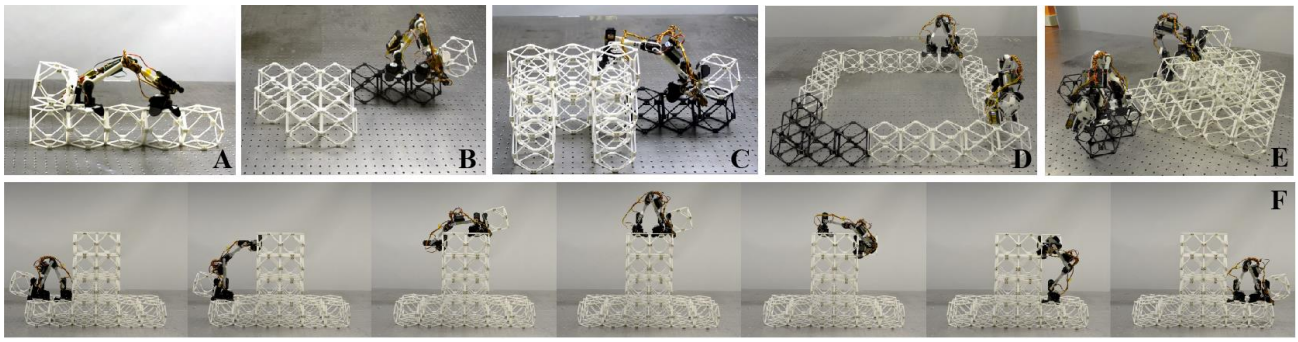

Figure 13: Robot experiments. A) 1D beam reconfiguration, B) 2D plate assembly, C) 3D enclosure assembly, D) Multi-robot construction of branching structure showing global precision from part metrology, E) Multi-robot construction of 35 voxel pyramid, F) 3D locomotion and voxel transport.

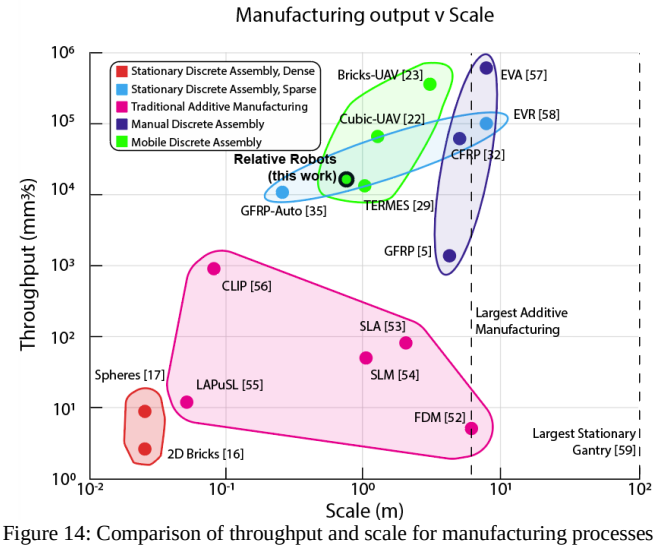

Figure 14: Comparison of throughput and scale for manufacturing processes

Lastly, in Figure 15D, we look at building cubes with side lengths of 5, 10, 25, and 50 voxels, containing 125, 1,000, 15,625, and 125,000 voxels, respectively, using one to four robots. We see an expected improvement with more robots, but we can also observe construction time increases exponentially with structure scale, due to the increased distance a given robot travels for part pickup and placement. We will discuss aspects of scaling in the following section.

VI. Discussion

Future research for this material-robot system will focus on aspects related to structural performance, scaling, and controls. The material system will need several upgrades to the joint and constituent material. The current magnetic joint does not meet the criteria of a structural connection. It is, however, useful for alignment. A reversible connection with sufficient stiffness and strength while also being amenable to robotic assembly is currently being developed. The current low-fidelity 3D printed material will be replaced with an injection molded GFRP, which has been shown to offer low-cost and high performance at comparable scales [32].

Our current robot scale is driven by two actuation-specific considerations. The lower scale limit is driven by external dimensions of commercial-off-the-shelf (COTS) motors which must interface with the voxel (such as entering and gripping a face). The upper scale limit is driven by torque density limits of COTS motors, which, while shown to be roughly linear [42], does not account for the associated scaling of robot mass (linkages, end effectors, etc.). Arbitrarily large robots are impractical, though this would be less of an issue in zero gravity. For large scale structures, the physical distance a robot travels from part source to build front can become problematic, as number of steps (s) increases with the square of the number of voxels (n), where $s = n(n+1)/2$, so for large values of $n, s \propto n^2$. We showed this can be improved with multiple part sources, though very large structures could benefit from incremental relocation of part sources.

We showed that a locally precise robot can build a globally precise structure at small scales. However, this demonstrates relative, not absolute, precision. It has been shown that a pyramid made of n passively stacked, spherical voxels with a random standard deviation σ would result in a structure with a sub-linear error scaling [43]. For sparse structures, it has been shown experimentally [44] and numerically [45] that increasing the cross section of an $n \times n$ beam will decrease beam length error and beam tip deflection. These benefits of elastic averaging, however, have limits. Locally, we find that if part error exceeds a threshold relative to part compliance (primarily a function of beam bending stiffness), it may be impossible to align joints for assembly [34]. Alignment with hardware using a traditional global reference system may be impacted by small part error stacking over long distances. While this can be predicted with characterization of mean error distribution or addressed with the addition of compliant interface parts, ideally, the entire system could be built using a discrete material system and thus share a relative reference frame, such as a standalone solar electric transport vehicle.

A controlled environment contributes greatly to the success of our deterministic control experiments. In more dynamic environments, such as on an exoplanet, the ability to adapt to changing conditions will be critical. In the future, we can use techniques such as machine learning and neural networks to model the non-linear relationship between these factors and control strategies, as shown in [46][47], or use variations of neural network architectures to build stable and robust adaptive control models [48]. Moreover, online learning and hierarchal models could be very useful in trajectory planning and modeling and controlling a hierarchy of systems and subsystems [49][50]. Distributed control strategies for full 3D construction have been explored algorithmically [51], but this remains an open challenge for hardware implementation. We expect that our developments here can be leveraged for the future design and demonstration of a scalable construction system for modular, high performance cellular structures.

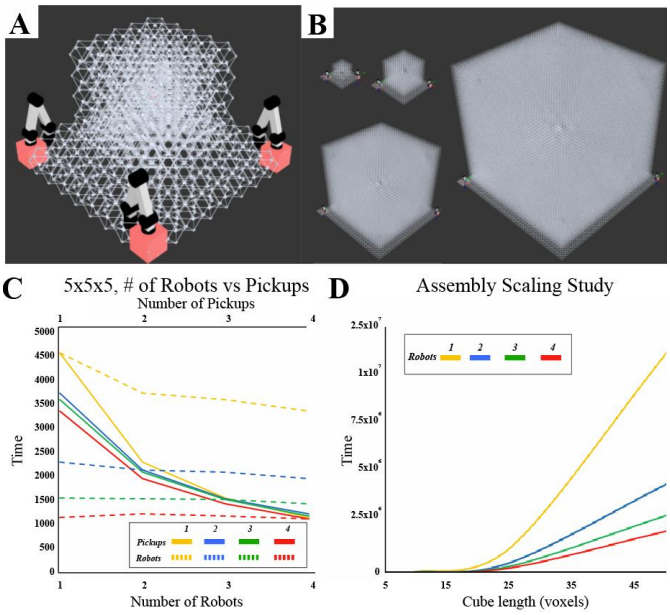

Figure 15: Simulation studies. A) 5x5x5 cube, B) 5^3, 10^3, 25^3, and 50^3 cubes, C) 1-4 Robots v Pickup stations for assembly of 5^3 cube, D) 1-4 Robot assembly time for 5^3, 10^3, 25^3, and 50^3 cubes.

REFERENCES

[1] T. A. Schaedler and W. B. Carter, "Architected Cellular Materials," *Annu. Rev. Mater. Res.*, vol. 46, no. 1, pp. 187–210, 2016.

[2] Z. August, G. Ostrander, J. Michasiow, and D. Hauber, "Recent developments in automated fiber placement of thermoplastic composites," *SAMPE J.*, vol. 50, no. 2, pp. 30–37, 2014.

[3] J. Markish and K. Willcox, "Value-Based Multidisciplinary Techniques for Commercial Aircraft System Design," *AIAA J.*, 2008.

[4] R. H. Liebeck, "Design of the Blended Wing Body Subsonic Transport," *J. Aircr.*, 2008.

[5] N. Cramer *et al.*, "Elastic Shape Morphing of Ultralight Structures by Programmable Assembly," *Smart Mater. Struct.*, 2019.

[6] L. Puig, A. Barton, and N. Rando, "A review on large deployable structures for astrophysics missions," *Acta Astronautica*, 2010.

[7] R. P. Hoyt *et al.*, "SpiderFab: An architecture for self-fabricating space systems," in *AIAA SPACE Conference*, 2013.

[8] K. C. Cheung, "ARMADAS,". [Online]. Available: https://gameon.nasa.gov/projects/automated-reconfigurable-mission-adaptive-digital-assembly-systems-armadas/. [Accessed: 12-Jan-2018].

[9] K. Belvin *et al.*, "In-Space Structural Assembly: Applications and Technology," in *AIAA SciTech*, 2016.

[10] X. Zheng *et al.*, "Multiscale metallic metamaterials," *Nat. Mater.*, 2016.

[11] C. E. Duty *et al.*, "Structure and mechanical behavior of Big Area Additive Manufacturing (BAAM) materials," *Rapid Prototyp. J.*, vol. 23, no. 1, pp. 181–189, 2017.

[12] B. Khoshnevis, D. Hwang, K.-T. Yao, and Z. Yah, "Mega-scale fabrication by contour crafting," *Int. J. Ind. Syst. Eng.*, 2006.

[13] A. R. Torrado and D. A. Roberson, "Failure Analysis and Anisotropy Evaluation of 3D-Printed Tensile Test Specimens of Different Geometries and Print Raster Patterns," *J. Fail. Anal. Prev.*, 2016.

[14] D. Lévesque, C. Bescond, M. Lord, X. Cao, P. Wanjara, and J. P. Monchalin, "Inspection of additive manufactured parts using laser ultrasonics," in *AIP Conference Proceedings*, 2016, vol. 1706.

[15] J. T. Dorsey, W. R. Doggett, R. a Hafley, E. Komendera, N. Correll, and B. King, "An Efficient and Versatile Means for Assembling and Manufacturing Systems in Space," *AIAA Space Conf.*, 2012.

[16] W. Langford, A. Ghassaei, and N. Gershenfeld, "Automated Assembly of Electronic Digital Materials," in *ASME MSEC*, 2016.

[17] J. D. Hiller and H. Lipson, "Fully Recyclable Multi-Material Printing," *Solid Free. Fabr. Symp.*, 2009.

[18] S. J. Keating, *et al*, "Toward site-specific and self-sufficient robotic fabrication on architectural scales," *Sci. Robot.*, 2017.

[19] X. Zhang and et al, "Large-scale 3D printing by a team of mobile robots," *Autom. Constr.*, 2018.

[20] M. Kayser *et al.*, "FIBERBOTS: an autonomous swarm-based robotic system for digital fabrication of fiber-based composites," *Constr. Robot.*, 2018.

[21] M. Yablonina and A. Menges, "Towards the Development of Fabrication Machine Species for Filament Materials," in *Robotic Fabrication in Architecture, Art and Design 2018*, 2018.

[22] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Auton. Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[23] F. Augugliaro *et al.*, "The flight assembled architecture installation: Cooperative contruction with flying machines," *IEEE Control Syst.*, vol. 34, no. 4, pp. 46–64, 2014.

[24] K. Dörfler, T. Sandy, M. Giftthaler, F. Gramazio, M. Kohler, and J. Buchli, "Mobile Robotic Brickwork: Automation of a Discrete Robotic Fabrication Process Using an Autonomous Mobile Robot," in *Robotic Fabrication in Architecture, Art and Design 2016*, 2016.

[25] S. Karumanchi, "Payload-centric autonomy for in-space robotic assembly of modular space structures," *J. Feild Robot.*, 2018.

[26] Y. Yoon and D. Rus, "Shady3D: A Robot that Climbs 3D Trusses," in *IEEE International Conference on Robotics and Automation*, 2007.

[27] F. Nigl, S. Li, J. E. Blum, and H. Lipson, "Structure-reconfiguring robots: Autonomous truss reconfiguration and manipulation," *IEEE Robot. Autom. Mag.*, vol. 20, no. 3, pp. 60–71, 2013.

[28] Y. Terada and S. Murata, "Automatic Modular Assembly System and its Distributed Control," *Int. J. Rob. Res.*, 2008.

[29] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science (80-.).*, 2014.

[30] A. Costa, A. Abdel-Rahman, B. Jenett, N. Gershenfeld, I. Kostitsyna, and K. C. Cheung, "Algorithmic Approaches to Reconfigurable Assembly Systems," in *IEEE Aerospace Conference*, 2019.

[31] K. C. Cheung and N. Gershenfeld, "Reversibly assembled cellular composite materials.," *Science*, vol. 341, no. 6151, pp. 1219–21, 2013.

[32] B. Jenett, D. Cellucci, C. Gregg, and K. C. Cheung, "Meso-scale digital materials: modular, reconfigurable, lattice-based structures," in *Proceedings of ASME MSEC*, 2016.

[33] B. Jenett *et al.*, "Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures," *Soft Robot.*, vol. 4, no. 1, 2017.

[34] C. Gregg, J. Kim, and K. Cheung, "Ultra-Light and Scalable Composite Lattice Materials," *Adv. Eng. Mater.*, 2018.

[35] G. Trinh *et al.*, "Robotically Assembled Aerospace Structures: Digital Material Assembly using a Gantry-Type Assembler," in *IEEE Aerospace Conference Proceedings*, 2017.

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

[37] B. Jenett and D. Cellucci, "A mobile robot for locomotion through a 3D periodic lattice environment," in *Proceedings of ICRA*, 2017.

[38] V. Deshpande, M. Ashby, and N. Fleck, "Foam topology: Bending versus stretching dominated architectures," *Acta Mater.*, 2001.

[39] M. Ochalek, G. Trinh, O. Formoso, B. Jenett, C. Gregg, and K. Cheung, "Geometry and Joint Systems for Lattice-Based Reconfigurable Space Structures," in *IEEE Aeroconf*, 2019.

[40] K. D. Kotay and D. Rus, "The Inchworm Robot: A Multi-Functional System," in *Autonomous Robots 8*, 2000.

[41] Y. Terada and S. Murata, "Modular Structure Assembly Using Blackboard Path Planning System," in *International Symposium on Automation and Robotics in Construction.*, 2006.

[42] K. Dermitzakis, J. P. Carbajal, and J. H. Marden, "Scaling laws in robotics," in *Procedia Computer Science*, 2011.

[43] J. Hiller and H. Lipson, "Design and analysis of digital materials for physical 3D voxel printing," *Rapid Prototyp. J.*, 2009.

[44] G. A. Popescu, T. Mahale, and N. Gershenfeld, "Digital materials for digital printing," in *International Conference on Digital Fabrication Technologies*, 2006.

[45] B. Jenett, C. Gregg, D. Cellucci, and K. Cheung, "Design of multifunctional hierarchical space structures," in *IEEE Aerospace Conference Proceedings*, 2017.

[46] V. T. Yen, W. Y. Nan, and P. van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Comp. and Applications*, 2018.

[47] W. He and Y. Dong, "Adaptive Fuzzy Neural Network Control for a Constrained Robot Using Impedance Learning," *IEEE Trans. Neural Networks Learn. Syst.*, 2018.

[48] J. J. Rubio, "Modified optimal control with a backpropagation network for robotic arms," *IET Control Theory Appl.*, 2012.

[49] J. de Jesús Rubio, "Discrete time control based in neural networks for pendulums," *Appl. Soft Comput. J.*, 2018.

[50] J. R. García-Sánchez *et al.*, "Robust switched tracking control for wheeled mobile robots considering the actuators and drivers," *Sensors (Switzerland)*, 2018.

[51] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *Int. J. Rob. Res.*, 2008.

[52] "Stratasys Mojo."Https://www.stratasys.com/3d-printers/mojo, [Accessed: 12-Jan-2018].

[53] "PROX 950." https://www.3dsystems.com/3d-printers/prox-950/specifications, [Accessed: 12-Jan-2018].

[54] "SLM" https://slm-solutions.com/en/products/machines/selectivelasermeltingmachines/. [Accessed: 12-Jan-2018].

[55] B. Moran, "Large-Area Projection Micro-Stereolithography," *Lawrence Livermore National Labs*, 2015.

[56] J. R. Tumbleston *et al.*, "Additive manufacturing. Continuous liquid interface production of 3D objects.," *Science*, 2015.

[57] M. S. Lake, W. L. Heard, J. J. Watson, and T. J. Collins, "Evaluation of Hardware and Procedures for Astronaut Assembly and Repair of Large Precision Reflectors," Langley Research Center, NASA /TP-2000-210317, 2000.

[58] M. D. Rhodes, R. W. Will, and C. Quach, "Baseline Tests of an Autonomous Telerobotic System for Assembly of Space Truss Structures," Langley, 1994.

[59] "EEW." [Online]. Available: http://www.eew-protec.de/110.0.html?&L=1. [Accessed: 12-Jan-2018].