Using Local Force Measurements to Guide Construction by Distributed Climbing Robots

Nathan Melenbrink[1,2,*], Panagiotis Michalatos[3,4], Paul Kassabian[5], and Justin Werfel[1]

***Abstract*— Construction automation has historically been driven by top-down implementations of specific tasks, which are neither responsive nor resilient to dynamic situations, and often require centralized control or human supervision. Previous work on robotic assembly has generally neglected to consider forces acting on the structure, whether in the completed structure alone or throughout the building process. In this paper, we investigate the utility of local force measurements in guiding construction by a distributed team of strut-climbing robots, focusing on a scenario involving building an unsupported span out across a gap in a two-dimensional vertical plane, as a step towards building a bridge. We show that such measurements enable robots to build structures that cantilever significantly further than those built by robots without access to such information, while maintainig stability throughout the building sequence. We consider both structures securely anchored to the ground and those resting unanchored atop it, using a counterbalancing approach in the latter case to permit cantilevering. The principles explored in simulation are also demonstrated in hardware, including a prototype strut-climbing robot and truss components, incorporating a cost-effective sensor implementation that reports the requisite force information.**

I. INTRODUCTION

Construction is a major industry that has been slow to incorporate automation, largely due to the fact that most construction projects take place in highly uncontrolled and variable environments. Many construction tasks are repetitive, such as the assembly of truss structures. Truss structures are a fundamental component of the contemporary built environment, enabling spans and cantilevers over unsupported distances much longer than individual building elements. The assembly of truss structures is an ideal entry point to construction automation because it is a process that can yield useful structures such as bridges or towers without requiring additional construction tasks (e.g., building formwork is required for poured concrete, excavation is required for foundations, etc.). Truss structures leverage the structural stability of triangulation, can be assembled into a variety of forms, and are able to cover large spans with a limited quantity of material.

While considerable work has been done towards construction automation in off-site prefabrication facilities, we turn our attention to on-site construction. This neglected area of construction automation might eventually be applicable to disaster relief, outer space construction, or other hazardous environments that would not be conducive for human builders. Even longer-term applications might include facilitating repetitive tasks in the commercial construction industry such as erecting temporary scaffolding, or even building permanent structures.

Typically, bridge construction relies on sturdy scaffolding for support, so little attention needs to be paid to the structural stability of the bridge itself during the construction process. However, installing that scaffolding is itself a challenge that requires structural stability be maintained throughout. We therefore focus specifically on truss structures that can be assembled without the need for an additional scaffolding structure, meaning the truss itself must remain stable throughout the building sequence.

Taking inspiration from some of nature's most successful builders, including beavers, termites, and sociable weaverbirds [1], we look to large numbers of distributed robotic agents as a possible approach to automating bridge construction. Climbing mobile robots are a means of avoiding the workspace limitations imposed by stationary or ground-based robots, and are superior to UAVs in terms of power consumption. We opt for real-time, reactive decision making as opposed to pre-planning, which allows the construction process to be resilient to changing numbers of agents or external perturbations during the building sequence, and can let the process adapt to site conditions without a requirement to collect information about the environment prior to beginning construction. Furthermore, agents operate independently using only local information, avoiding scalability limitations and other difficulties associated with centralized control or global information.

In many scenarios, it may be desirable to build structures that meet user-specified high-level requirements (e.g., span a gap) as opposed to achieving a specific predetermined blueprint. For instance, in emergency relief in volatile environments, site conditions such as the size of a gap may be unknown or subject to change, so that designing a bridge in advance is not feasible. In such cases, robots must respond to encountered conditions in order to achieve the high-level goal.

While the swarm approach offers a robust and adaptable method for assembly, it requires that individual robots be able to detect and prevent common modes of structural failures. In order to provide the robots with information about local forces, the assembly system must be capable of some sort of

This work was supported in part by the Wyss Institute and by Autodesk.
[1]Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, MA 02138, USA
[2]Institute for Computational Design and Construction, University of Stuttgart, Germany
[3]Graduate School of Design, Harvard University, Cambridge, MA 02138
[4]Autodesk, Inc., Boston, MA 02210, USA
[5]Simpson Gumpertz & Heger, Waltham, MA 02453, USA
*nathan.melenbrink@wyss.harvard.edu

real-time structural state sensing.

In this paper, we look at the problem of building a cantilever as far as possible out across a gap in a 2-dimensional vertical plane (Fig. 1). This cantilever scenario considers a building element integral to many applications, where the role of physical forces is particularly important, and serves as a first exploration of principles for this general framework with local force sensing by decentralized robots. Such a cantilevered structure, extending indefinitely, at some point must collapse. We demonstrate that access to local force information lets robots build a cantilever over three times further out than an approach in which robots do not take such forces into account as they build.

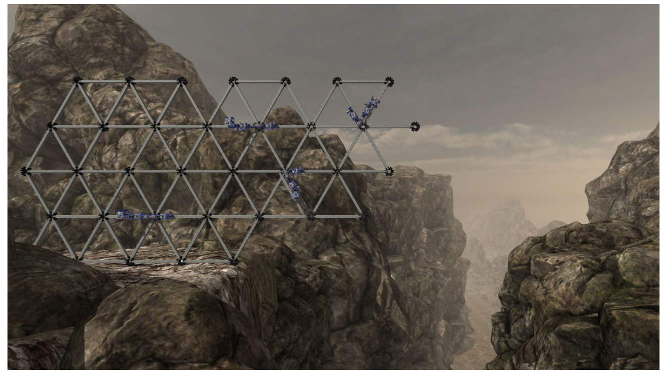

Fig. 1. Rendering of a hypothetical implementation of a team of strut-carrying robots in the process of building a 2D cantilever across a gap, using local force measurements to ensure that struts are placed in a sequence that avoids structural failure.

II. Related Work

A. Robotics

A number of previous studies have considered multi-robot approaches to automating construction tasks. Werfel et al. demonstrated a decentralized team of independent climbing robots limited to local information, able to build structures much larger than themselves using specialized bricks [2]. Augugliaro et al. used a centralized team of aerial vehicles to build structures made of custom bricks according to a preplanned assembly sequence [3]. One limitation of these systems is that the structures buildable by stacking bricks are trivially stable; in order to build spans extending out over a gap, other materials and approaches are required.

Studies focused on truss structures include that of Yun et al., who presented a centralized algorithm for planning the reconfiguration of grounded truss structures [4]. Nigl et al. further developed hardware for this approach, demonstrating an autonomous strut-climbing robot capable of reconfiguring truss elements [5]. Galloway et al. demonstrated a stationary modular robotic system that could assemble truss structures layer by layer from below [6]. Lindsey et al. demonstrated the ability for a team of quadrotors to assemble 2.5D structures (no overhangs) composed of individual struts [7]. The work of Galloway et al., Lindsey et al., and Nigl et al. all operated on square trusses, which are inherently less stable than triangulated trusses.

Most previous work has neglected the issue of stability during assembly. One notable exception is that of McEvoy et al., which presents a centralized approach for finding assembly sequences for truss structures that maintain structural stability throughout [8]. Similarly, Brodbeck et al. consider stability at each step of a preplanned building sequence for an aggregate assembly of blocks [9]. However, for a system intended to work with a potentially large number of independent robots, a preplanned assembly sequence is not feasible; instead, a more dynamic scheme in which robots react to conditions they encounter may be more effective.

B. Structural Health Monitoring

Structural health monitoring, a field that has emerged within the past 20 years, has the goal of using in-situ dynamic response measurements to detect and characterize damage to a structure. Work in this area is focused on the evaluation of complete structures with known topologies. Experimental modal analysis is an area of structural health monitoring where on-site vibration measurements are compared against expected values generated from a finite element model for the purposes of damage detection [10]. More recently, researchers have demonstrated that similarly accurate results can be found using “mobile sensors”—autonomous robots equipped with accelerometers, communication hardware and the necessary degrees of freedom to traverse the structure in question. These principles have recently been demonstrated for the on-site verification of a steel pedestrian bridge [11]. Such techniques require knowing the global topology of a structure, and have not been applied to intermediary stages of a construction sequence.

III. Setup

This paper addresses the problem of building a cantilever as far as possible out across a chasm, as would be necessary to build a bridge without access to the far side. This problem specifically addresses the limitations of previous work that was trivially stable and not able to produce cantilevers or spans. Prior work involving structural force measurements (mentioned above) has not been extended to autonomous robots or real-time reactive planning. Here, we introduce a method for a team of climbing robots to access local force information to inform their building activity, and demonstrate that utilizing this method significantly increases the length of cantilevers that the swarm can build.

A. Assumptions

Construction is performed by a set of identical climbing robots, each of which is capable of locomotion on the truss structure they construct. Each robot is able to carry one strut (with a node pre-attached at one end) at a time, and mechanically attach that strut to a node where a vacant socket exists (Fig. 2). In this study we neglect the details of attachment where some struts would need to infill a gap between two existing nodes, with the extra node being removed and returned. Disassembly is not considered in this work. Robots cannot pass each other on a strut.

We choose an approach in which robots do not communicate directly with each other; instead, they coordinate their activities indirectly, by reference to the shared structure they build. This approach uses the insect-inspired idea of stigmergy, in which individual agents take actions that change the environment, which then affects future actions. Avoiding direct communication also avoids difficulties associated with mobile ad-hoc wireless networks in cluttered environments, requirements for robots to stay in proximity to maintain connectivity, etc.

When located at a node, a robot is able to read sensor information indicating local forces (see below). These forces on the structure change according to robots' movement and building activity. A robot at a node can determine whether each of the available attachment sockets has a strut present or absent, and, if present, whether another robot is currently located at the node at its far end.

For structural stability, we choose the truss geometry to be a triangular lattice, with horizontal rows. In this work we consider only building in two dimensions, in a vertical plane. Robots can keep track of their relative movement along the lattice (and the coordinate system it embodies) as they move from one node to another.

Construction begins from scratch (no struts/nodes initially present in the structure). A supply cache of materials is located at a point taken to be the origin of the coordinate system (Fig. 3). Ground support is available for an unlimited distance in the -x direction and for 3m in the +x direction.

We consider two cases for the structure–ground interface. In the "anchored" condition, we assume that nodes at ground level ($y = 0$) are securely attached there, fully constrained and unable to move in any direction. In the "unanchored" condition, nodes at ground level are unconstrained and free to move.

At each time step, all robots attempt to move from one node to another, with asynchronous update. Attaching a strut to a node, and retrieving a new strut from the cache, occur instantaneously.

For physical plausibility, material assumptions were made based on existing, readily available construction materials. The struts are considered to be standard hollow steel scaffolding tubes, with a 48mm outer diameter and a 4mm wall thickness. The nodes (Fig. 2) comprise a rigid (steel) core capable of handling axial load transfer, as well as a more pliable (polymer) shell with embedded force sensors, which would register slight deformations reflective of the applied bending force. The way in which these assumptions were translated into a finite element model are described in the following section. The unit length of the grid is 1 meter. Nodes have a diameter of 20 centimeters and struts are 90 centimeters in length, such that struts are inset 5 centimeters into nodes.

The mass of one 90cm scaffolding tube is taken to be 4kg, and one node is 2kg. The mass of a single strut-climbing robot is estimated to be 4kg. For simplicity, robots are considered to be in one of two states: laden (carrying one strut with pre-attached node), in which case their mass is modeled as 2kg + 4kg + 4kg = 10kg, and unladen, in which case their mass is modeled as 4kg.

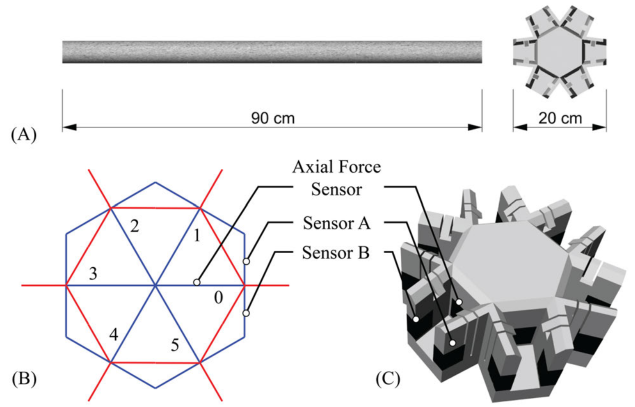

Fig. 2. (A) A node and strut shown to relative scale. (B) Diagram of the node, modeled as finite elements. Blue lines represent polymer elements while red lines represent steel tubes. The 6 sockets of each node are indexed as shown (see Algorithm 1). Each socket has 3 sensors. (C) Rendering of the node, showing positions where corresponding sensors are installed.

B. Simulation Environment

The assumptions outlined above were modeled in a digital simulation environment within the CAD software Rhinoceros, using the Grasshopper parametric solver as a shell for running the simulations. For finite element methods, the Millipede library was used, offering fast structural analysis algorithms for linear elastic systems [12]. Within this environment we calculate forces in Newtons based on the self-weights of the nodes, struts, and robots acting on the system. Finite element models were generated procedurally and analyses were run at every step of the simulation.

Fig. 2 shows an example node design and the abstraction of it used for the simulations.

The software implementation includes a system of global functions that handle finite element methods and identify structural failure. Algorithms are implemented as agent-based behaviors and uniformly assigned to all agents in the simulation. Agents are initiated at the origin (0,0) node and dispatched once per frame until all agents have entered the simulation. If the origin node is occupied by a previously initiated agent, subsequent agents wait for the first frame in which the node becomes unoccupied.

Robots build according to the algorithm described in the next section until a structural failure occurs. Failure is determined by a global routine that checks the maximum normal stress of each element at each frame of the simulation. We model the strut elements as steel, which has a yield strength of 235 MPa. With the proposed materials, it would take an enormous triangular lattice to cause a strut to yield under self-weight alone, so we chose an artificial yield strength of 11.75 MPa (5% of the actual yield strength). This conservative choice ensures that trusses would be robust to an applied load during the construction process. If at any point any element exceeds the artificial yield, the element is

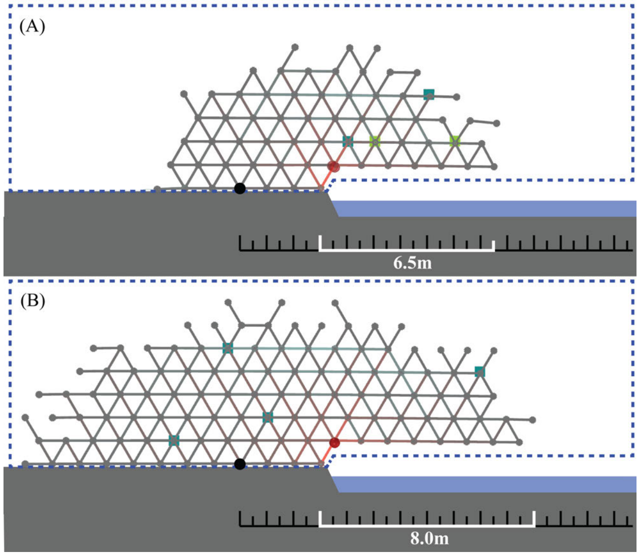

Fig. 3. Typical trials captured at the moment of failure for an anchored structure (A) and an unanchored, counterbalanced structure (B), both built by force-aware agents. The dashed blue line shows the valid workspace where struts can be placed. The red circle over the first cantilevering node indicates that an element in this node has failed, halting the simulation. The origin (0,0) is indicated by a black dot. The dark green squares represent agents that are carrying a strut (laden), while the light green squares represent agents not carrying a strut (unladen). Axial forces acting on the struts are represented by a gradient between red (compression) and cyan (tension), where struts with minimal axial force values are colored grey. Deflection is exaggerated for visual effect. In this example, the length of the cantilever (which is measured from x=3) reached 6.5m at the time of failure in (A) and 8.0m in (B).

considered broken, a failure is registered and the simulation halts. In the unanchored case, a failure can also occur if the cantilever accrues enough mass that the software determines it would topple into the chasm. When either type of failure occurs, we record the number of struts placed, the duration (number of frames in the simulation), and the length of the cantilever achieved in meters. 1000 trials are performed for each condition.

C. Algorithm

1) Anchored structures: Agent behavior (Algorithm 1) can be summarized as follows. From the cache node, a laden agent moves along the lattice, choosing a direction at each step and advancing to the next node in that direction. When its choice would take it along a strut not yet present, it attaches the strut it carries at that location. It then returns to the cache, where it picks up a new strut. To avoid traffic jams and overly narrow (and hence fragile) structures that result when agents move deterministically between targets, stochasticity was introduced by having agents move in a biased random walk: biased in the +x direction when laden, in the direction of the cache when unladen.

We compared two variants of agent behavior. The "force-unaware" variant operates as described above. The "force-aware" variant eliminates from consideration any directions where the forces measured indicate that a potential structural failure might occur if the robot were to move down that strut.

To evaluate the latter condition, a robot reads the values of two force sensors on opposite sides of the socket into which the strut is inserted (Fig. 2B,C), and takes the difference between them. This quantity is a proxy for bending forces in that isolated section of the node, which we found to be an effective metric indicative of impending structural failure. If that difference exceeds a given threshold (in this case a value of 30.0 Newtons, chosen by looking at specific failure cases and selecting a value conservative enough to prevent common failure modes), the robot should not move along that strut. This heuristic holds regardless of whether the node is at the top of a structure and subject only to its own self-weight, or is under kilo-Newtons of force. The insensitivity to loading from above occurs because the axial load from the struts (outer red lines in Fig. 2C) is directly transferred to the rigid steel core. This force could be on the order of tens to thousands of Newtons. The polymer shell and its embedded force sensors, represented by the blue lines in Fig. 2C, register force due to the bending of the strut, which is largely invariant of applied axial load.

Nodes anchored to the ground represent a special loading case. For regular triangular lattice structures, the structural load path travels axially through each node to the struts and nodes below it. The anchored nodes, however, are the terminus of the structural load path, and must transfer those forces to the ground. As a result, considerably higher shear forces are internalized in these nodes, such that their sensors corresponding to struts in the horizontal direction will report high values. To allow horizontal travel between struts along the ground, which would otherwise be forbidden unnecessarily, robots ignore force measurements in this special case (Algorithm 1, second half of the first test in line 5).

To prevent situations where they would get caught in a loop moving back and forth between two nodes, robots are forbidden to return to the last node they came from, unless no alternatives exist (Algorithm 1, lines 10–12).

2) Unanchored structures: To prevent the structure from falling into the chasm when unanchored, we modify Algorithm 1 so that agents build in a way that counterbalances the cantilever. They do this by adding material in other directions, using the weight distribution of the structure to effectively provide anchoring. When a robot at the origin retrieves a new strut from the cache and is assigned a goal location for biasing its random walk (line 8), instead of that goal always being (1000,0), it is assigned randomly from among three possibilities: (1000,0), (-1000,0), or (0,1000) with probabilities of 0.6, 0.3, and 0.1 respectively. This modified algorithm is referred to below as the "balanced" variant.

IV. Results

A. Anchored Structures

Robots with access to local force measurements are able to build trusses that cantilever significantly further than those built by robots without access to local force information (Table I, Fig. 4). Robots building using the force-unaware behavior, with no information about local stresses, create

Algorithm 1 Agent behavior. Strut attachment sites (sockets) at a node are designated 0 through 5, starting with the site in the +x direction and going counterclockwise as shown in Fig. 2B. For force-unaware agents, the first test in line 5 is not applied. Fig. 2C indicates the positions of sensorA and sensorB inside each of the node's 6 sockets. The valid workspace excludes sites with $y \leq 0$ beyond the end of the ground support, as indicated in Fig. 3. Line 8 biases the walk in the direction of a goal, with $\theta(i, laden)$ defined as the difference in angle between $(i\pi/3)$ and the vector from the robot's current position to either (1000,0) if laden or (0,0) if unladen.

1: **loop**
2:   **for** $i = 0$ to 5 **do**
3:     *weight*[i] $\leftarrow$ 1
4:     *bendForce* $\leftarrow$ |sensorA[i]$-$ sensorB[i]|
5:     **if** (*bendForce* $>$ threshold **and not** ($y = 0$ **and** (i=0 **or** i=3))) **or** neighbor[i] is occupied by another robot **or** neighbor[i] is located outside valid workspace **or** (*laden* = false **and** no strut at i) **then**
6:       *weight*[i] $\leftarrow$ 0
7:     **end if**
8:     *weight*[i] = *weight*[i] $\cdot \cos(\theta(i, laden)) + 1.5$
9:   **end for**
10:   **if not** ((only nonzero entry of *weight* corresponds to *previousNode*) **then**
11:     *weight*[*previousNode*]$\leftarrow$ 0
12:   **end if**
13:   select i with probability proportional to *weight*[i]
14:   **if** *laden* = true **and** no strut at i **then**
15:     attach strut at i
16:     *laden* $\leftarrow$ false
17:   **else**
18:     *previousNode* $\leftarrow$ current location
19:     move to neighbor[i]
20:   **end if**
21:   **if** current location = (0,0) **then**
22:     *laden* $\leftarrow$ true
23:     *previousNode* $\leftarrow$ current location
24:   **end if**
25: **end loop**

structures that extend under 2m on average beyond the edge of the ground support before failing; in 9% of trials, a configuration occurs that causes structural failure before the truss even reaches the edge of the ground support. By contrast, the force-aware behavior results in cantilevers that extend over 6m on average beyond the edge; the shortest cantilever observed was 3.5m when failure occurred, while the longest was 7.5m.

Trials are terminated once any element in the configuration fails. Over 99% of trials in the force-aware condition shared the same failing element: the first cantilevering node (red dot in Fig. 3). This common failure mode occurs because that particular node inherits more of the bending force imposed

TABLE I

RESULTS FOR TEAMS OF 4 ROBOTS (ANCHORED STRUCTURES)

	Force-unaware	Force-aware
Number of Struts	32 ± 14	111 ± 9
Cantilever Length (m)	1.9 ± 1.2	6.3 ± 0.5
Number of Steps	60 ± 40	440 ± 60

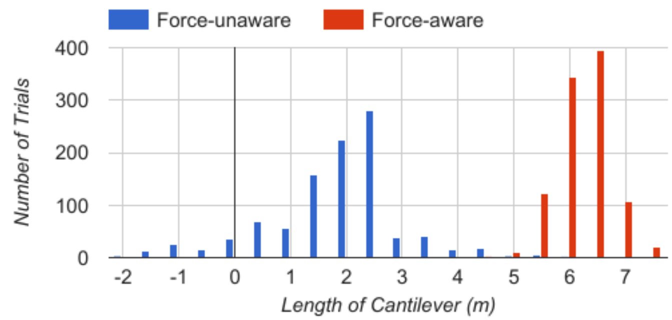

Fig. 4. Distribution of cantilever lengths at time of failure (anchored structures).

by the cantilever than any other node. This observation suggests a possible intervention: creating a unique, specially reinforced node, and putting a special case into the agent behavior so that that node becomes attached at that particular location, could potentially allow the cantilever to be extended significantly further in this particular scenario. Similar locations for targeted reinforcement could potentially be identified in future studies of other construction scenarios.

B. Unanchored Structures

When force-aware agents build using Algorithm 1 (no counterbalancing), approximately a third of trials fail via the structure toppling off the cliff. The balanced variant eliminates this failure mode, and also slightly increases the mean length of the cantilever at structural failure. The cost of these improvements is a considerable increase in use of material and time (Table II).

For comparison, force-unaware agents exhibit very poor performance whether using the original algorithm or the balanced variant (Table II). The trusses built by these agents never extend far enough to topple into the chasm.

TABLE II

RESULTS FOR TEAMS OF 4 ROBOTS (UNANCHORED STRUCTURES)

	Algorithm 1		Balanced variant	
	Unaware	Aware	Unaware	Aware
Num. Struts	30 ± 20	131 ± 10	30 ± 20	220 ± 20
Cantilever(m)	0.8 ± 1.4	6.2 ± 0.5	0.4 ± 1.3	7.3 ± 0.6
Num. Steps	90 ± 60	670 ± 70	90 ± 70	1250 ± 180
Topple (%)	0	34	0	0
Collapse (%)	100	66	100	100

Results of four variants of Algorithm 1 for the unanchored condition, averaged over 1000 trials for each variant. "Aware" and "Unaware" are abbreviations for the force-aware and force-unaware variants. "Topple" refers to the failure mode where the structure falls into the chasm, and "Collapse" refers to failures due to excessive stress.

An obvious extension of this investigation into unanchored structures would be to use more rigid materials, so that unbalanced structures would always fall into the chasm before the structure failed due to stress. Since falling into the chasm is the failure mode best addressed by counterbalancing, we expect that using more rigid materials would provide for a clearer characterization of the utility of counterbalancing for unanchored structures.

V. Hardware

A. Design

To demonstrate key capabilities in a physical system, hardware prototypes were developed for both an autonomous strut-climbing robot (Figs. 5, 6) and an instrumented node-and-strut assembly (Fig. 7). Rather than scaffolding tubes, the system used square-cross-section wooden struts, which allowed for faster design and prototyping of a snap-fit mechanism to attach the struts to the nodes.

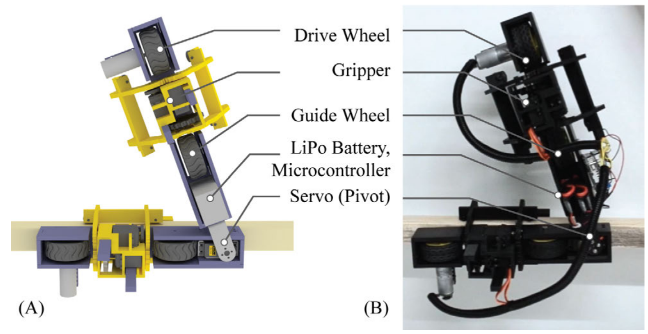

Fig. 5. (A) A rendering of the strut-climbing robot. Not shown is a preliminary design for a strut-carrying module, which would be needed to achieve strut placement. (B) Photograph of initial prototype.

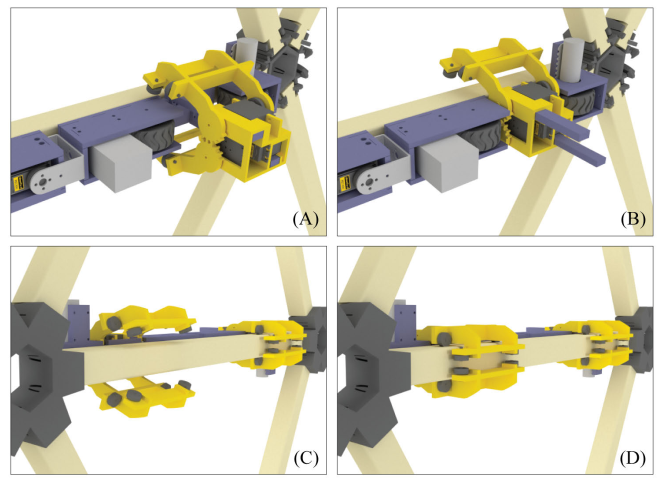

Fig. 6. The sequence for gripping a strut for locomotion: (A) The gripper on the front carriage is in an open and raised position, allowing the front carriage to pivot to any 60-degree increment without colliding with the existing struts. (B) While positioned in front of the desired strut, the gripper is lowered. (C) The same step from the rear view, showing the gripper positioned around the strut. (D) The gripper closes around the strut. Small wheels on the gripper allow it to roll axially along the strut.

The instrumented node described in this section is atypical of the current state-of-the-art technologies that are conventionally used in structural health monitoring. This difference is largely due to the primary utility of structural health monitoring being the comparison of field measurements against known structural models. Measurements are usually taken by commercially available strain gauges. However, there are a number of reasons why strain gauges were deemed impractical for this application, and were abandoned in favor of simple force sensors. Strain gauges are difficult to install and calibrate uniformly over multiple elements, and require additional amplification circuitry that is not needed for force sensors. Independent of amplification circuitry, strain gauges themselves range from tens to hundreds of US dollars per unit, while force sensors can be produced for a fraction of a dollar. As each node may incorporate 12–18 sensors, this difference quickly becomes significant.

Thus we propose instead a slim-package force sensor comprising a strip of force-sensitive material (e.g., Velostat) sandwiched between two copper sheets (Fig. 7). When force is exerted on the sensor, the resistance of the material decreases, yielding an increase in the voltage running through the sensor that is easily detected by a microprocessor. A significant advantage that this method offers over strain gauges is that the force sensors can be incorporated directly into the node, while strain gauges would need to be installed on the struts. The latter condition means that in order for a robot located at a node to evaluate the forces on all attached struts, either there would need to be an electrical connection made between the end of each strut and its corresponding node, or the robot would need to re-position itself repeatedly and read each strut's strain gauge individually. On the other hand, force sensors for all 6 strut positions could be embedded directly into the node, simplifying the fabrication process and allowing robots to access the force data for all 6 possible struts from a single position. The slim profile of the sensors is also advantageous because it allows the sensors to be placed directly in the load path, enabling them to report axial force, which strain gauges are unable to do. While an alternative solution could be achieved using a capacitive force sensor, in order to provide comparable resolution such an approach would require a pliable elastomer to be inserted between the copper sheets, which would introduce an unacceptable degree of deflection to the interface between the node and the strut.

In order to provide reliable information despite the inevitable mechanical variability in sensor attachment, force sensors need to be calibrated when a strut is installed. In a complete system, the sequence by which a robot would install a strut is as follows: (1) The robot arrives at a node, and using pogo pin connections, supplies power to the node's microprocessor and reads the force values for each position by comparing the current reading to the baseline value stored in the node's memory. If no strut has yet been attached at a given socket, there will be no baseline values stored for the corresponding sensors. (2) If the robot identifies a vacant socket that is suitable to receive a strut according to Algorithm 1, the strut-carrying module presses the strut into the appropriate socket until the pliable buckles snap into

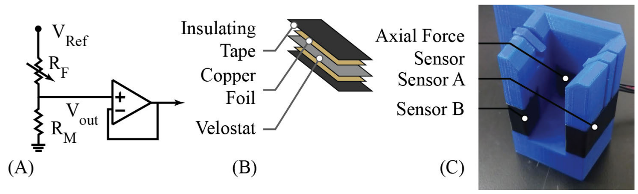

Fig. 7. (A) Circuit diagram for each sensor showing the Velostat as a variable resistor. (B) The layers of the sensor package. (C) A prototype socket used for the Hardware Evaluation. Sensors A and B are installed around the side walls of the socket. The axial force sensor is installed in the back wall of the socket, but this measurement was not needed for the work described in this paper.

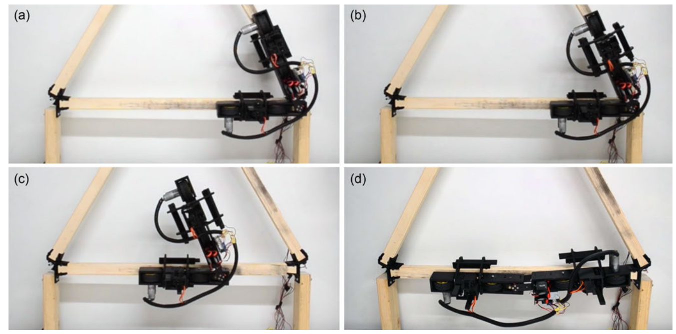

Fig. 9. The locomotion sequence as the robot transitions from one strut to another. (a) The robot has moved using its rear carriage until it makes a hard stop at the next node. The motor encoders cease incrementing, indicating to the microprocessor to power off the motor. The front carriage grips the next strut, (b) the rear carriage releases from the previous strut, (c) the robot traverses down the new strut using its front carriage, (d) the rear carriage pivots into place and attaches to the strut. Next the front carriage will detach and pivot, and the sequence can continue. Note that the nodes are simplified to 2 vacancies as opposed to a complete 6 (i.e., only one third of the node shown in Fig. 2C was fabricated and installed at each vertex of the triangle), as a single triangle was deemed sufficient for early locomotion trials.

place around it. (3) While the robot is still holding the strut, the sensors for that socket are zeroed, such that when the strut is released the values reported by the sensors reflect the force required to support the strut. These new values will then represent the baseline for future measurements. (4) After installing the strut, the baseline sensor values are stored as constants to an array in EEPROM memory, meaning that the values will be preserved even if the microprocessor is powered off. By using passive circuitry and EEPROM memory, we can avoid the need to equip each node with a power source, and can instead retrieve accurate measurements when the robot powers on the node circuitry.

B. Evaluation

Fig. 8 shows force sensor measurements for a prototype node (Fig. 7C) where an installed strut, oriented horizontally, supports increasing loads (via masses hung from its other end). These results show that the proposed sensing system is able to register small variations in applied load. In future work, the node will be redesigned to better account for dimensional variation, and will be constructed of better quality material.

The robot was able to locomote along struts and autonomously transition from one to another (Fig. 9); however, reliability of these operations proved a challenge. Past work developing strut-climbing robots used costly custom struts,

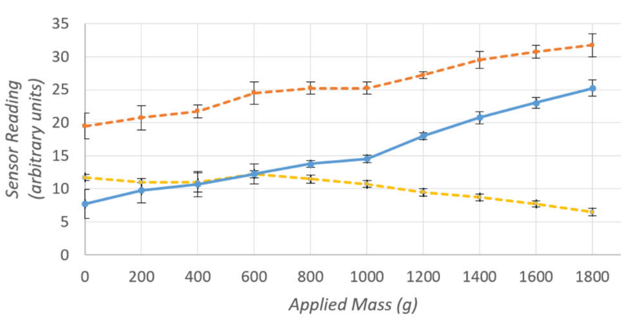

Fig. 8. Force sensor readings as a function of mass providing a bending force to a prototype node. Mass is hung from the far end of a horizontally-oriented strut installed at that node. Dashed lines: yellow, Sensor A (top); orange, Sensor B (bottom, see Fig. 2); solid blue line: difference between the two readings (used in Algorithm 1). Error bars show the standard deviation among 4 trials (removing the hanging masses between trials and leaving the strut in place).

fabricated with notches along their length that could be used by the robots to brace against while moving as well as for odometry [5]. A goal in this work was to use instead low-cost, readily-available building materials for struts. However, slight variations in the dimensions of the wood stock or in the surface friction on the face of the strut had the effect of making it extremely difficult to achieve consistent robot motion across different struts; e.g., adjusting the robot's grip so it moved effectively on one strut might result in it slipping on another, and gripping too tightly to move on a third. While these difficulties were encountered specifically with struts made from wood, such minor dimensional variations also exist in steel scaffolding tubes and other common building materials. The square stock used in this demonstration had the advantage of preventing unwanted rotation around the axis of the strut, a problem that would need to be addressed if considering round stock such as scaffolding tubes. In order to make this type of locomotion feasible for readily-available stock construction materials, the robot would need to be outfitted with additional sensors to ensure that equal gripping force is applied to struts with different material properties. Furthermore, maintaining constant gripping force against the strut while actuating a rolling wheel requires a greater power draw than gripping alone.

Therefore, for future work it may be worthwhile to consider alternative locomotion strategies, such as the extension and contraction technique described by Tavakoli et al. [13] or the hand-over-hand alternating grip approach described by Detweiler et al. [14]. Since these techniques require only gripping in a series of fixed positions, they are likely to perform much more consistently across struts with minor variations in dimensions. It may very well be the case that the ideal robot morphology and locomotion strategy varies with the type of material being used for construction.

VI. Conclusions and Extensions

In this paper we have demonstrated a technique for using local force measurements to guide construction by a team of independent climbing robots. We have shown that such local measurements allow the construction of significantly longer cantilevers. We have further demonstrated that implementing a counterbalancing behavior allows bridges to be built without requiring anchoring to the ground. Finally, we demonstrated prototypes of a strut-climbing robot, node-and-strut hardware, and force sensors.

Extensions to this work will include looking at other, more complete implementations in simulation and in hardware. First, in simulation we intend to generalize the approach to 3-dimensional truss structures. We expect a similar approach to be effective, as the prevailing failure mode should still be due to bending under gravitational force. We also expect that better results could be achieved by certain modifications to the agent behavior—for example, adjusting agent movement such that the bridge becomes built in a tapered shape, which would allow a longer cantilever with the same number of struts. Finally, we could look at the use of heterogeneous materials—for example, more heavily reinforced or lighter materials to be used in suitable locations, as with the case discussed earlier of the typical failure point in the force-aware trials—or the possibility of allowing disassembly, and reuse and repurposing of building materials as the structure grows.

The approach considered here is suited to building in response to a high-level goal, where the detailed form of a desired structure need not be known in advance. For other applications where more is known about the site conditions (e.g., width of the gap to be spanned, so that a more optimal arch shape might be desirable and attainable), or where precise blueprints are required, future work could explore modifying this approach to accommodate those requirements. A stable building sequence would still be needed in such cases, and could potentially be addressed by the kind of force awareness presented here.

In terms of hardware, we are interested to apply these methods of guidance by local force measurements to other types of connections. One example might be instrumenting conventional tube-and-clamp type scaffolding connections with these slim-package force sensors. However, these clamps only connect two struts at a time, presenting a challenge to creating regular lattices. Another approach might be to investigate cellular construction as opposed to strut-based construction, i.e., where the basic building unit is a higher-order pre-assembled form rather than a strut. Such an approach has been explored by Trinh et al. [15]. The primary advantage of such an approach is that the cellular unit itself would be inherently stable. It would be of interest to investigate whether installing force sensors at the interfaces between volumetric units could allow for similarly effective guidance.

References

[1] K. von Frisch and O. von Frisch, *Animal Architecture*. New York: Harcourt Brace Jovanovich, 1974.

[2] 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.

[3] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, pp. 46–64, 2014.

[4] S. Yun, D. A. Hjelle, H. Lipson, and D. Rus, "Planning the reconfiguration of grounded truss structures with truss climbing robots that carry truss elements," in *Proceedings of 2009 IEEE/RSJ IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[5] F. Nigl, S. Li, J. E. Blum, and H. Lipson, "Structure-reconfiguring robots: Autonomous truss reconfiguration and manipulation," *IEEE Robotics and Automation Magazine*, vol. 20, pp. 60–71, 2013.

[6] K. Galloway, R. Jois, and M. Yim, "Factory floor: A robotically reconfigurable construction platform," in *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.

[7] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011, pp. 177–184.

[8] M. McEvoy, E. Komendera, and N. Correll, "Assembly path planning for stable robotic construction," in *Proceedings of IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, Boston, MA, 2014.

[9] L. Brodbeck and F. Iida, "Automatic real-world assembly of machine-designed structures," in *2014 IEEE International Conference on Robotics & Automation (ICRA)*, Hong Kong, 2014, pp. 1221–1226.

[10] C. J. Anumba and X. Wang, Eds., *Mobile and Pervasive Computing in Construction*. London: Wiley-Blackwell, 2012.

[11] D. Zhu, Q. Qi, Y. Wang, K.-M. Lee, and S. Foong, "A prototype mobile sensor network for structural health monitoring," in *Proceedings of SPIE 7294*, 2009, p. 72941A.

[12] P. Michalatos and S. Kaijima, "Millipede 2014 release," http://www.sawapan.eu/sections/section88_Millipede/files/MillipedeMarch2014.pdf, 2014, accessed: 2016-09-30.

[13] M. Tavakoli, M. R. Zakerzadeh, G. Vossoughi, and S. Bagheri, "A hybrid pole climbing and manipulating robot with minimum dofs for construction and service applications," *Industrial Robot: An International Journal*, vol. 32, no. 2, pp. 171–178, 2005.

[14] C. Detweiler, M. Vona, Y. Yoon, S.-k. Yun, and D. Rus, "Self-assembling mobile linkages," *IEEE Robotics & Automation Magazine*, vol. 14, no. 4, 2007.

[15] G. Trinh, G. Copplestone, M. O'Connor, S. Hu, S. Nowak, K. Cheung, B. Jenett, and D. Cellucci, "Robotically assembled aerospace structures: Digital material assembly using a gantry-type assembler," in *Proceedings of IEEE Aerospace Conference*, 2017, pp. 1–7.