TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction

Share Your Story

TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction

Kirstin Petersen
SEAS/Wyss Institute
Harvard University
Cambridge, MA 02138
kirstin@eecs.harvard.edu

Radhika Nagpal
SEAS/Wyss Institute
Harvard University
Cambridge, MA 02138
rad@eecs.harvard.edu

Justin Werfel
Wyss Institute
Harvard University
Cambridge, MA 02138
justin.werfel@wyss.harvard.edu

Abstract—Collective construction is the research area in which autonomous multi-robot systems build structures according to user specifications. Here we present a hardware system and high-level control scheme for autonomous construction of 3D structures under conditions of gravity. The hardware comprises a mobile robot and specialized passive blocks; the robot is able to manipulate blocks to build desired structures, and can maneuver on these structures as well as in unstructured environments. We describe and evaluate the robot's key capabilities of climbing, navigation, and manipulation, and demonstrate its ability to perform complex tasks that combine these capabilities by having it autonomously build a ten-block staircase taller than itself. In addition, we outline a simple decentralized control algorithm by which multiple simultaneously active robots could autonomously build user-specified structures, working from a high-level description as input.

I. INTRODUCTION

Construction traditionally involves direct human operation of tools and equipment, careful preplanning, and little or no true automation. Bringing automation to construction has the potential to improve measures like its speed and efficiency, as well as enabling construction in settings where it is difficult or dangerous for humans to work, e.g., in extraterrestrial environments or disaster areas. Nature provides us with impressive examples of animal construction: many species of termites build complex mounds several orders of magnitude larger than themselves. Inspired by termites and their building activities, our goal is to develop swarm construction systems in which large numbers of autonomous robots build human-scale structures according to desired specifications.

In this paper we present TERMES, a hardware system that is a first step toward this goal of automated construction by swarms of robots. The system comprises a mobile robot and specialized passive building blocks; the robot can autonomously manipulate the blocks, build structures with them, and maneuver on these structures as well as in unstructured environments. We describe the key capabilities of the robot regarding climbing, navigation, and manipulation on structures, and report its reliability in tests associated with each of these three actions. To illustrate the robot's ability to perform complex tasks combining these functions, we demonstrate it autonomously building a ten-block structure significantly larger than itself (taller than its height and $> 15\times$ its volume).

We further present a sample decentralized control algorithm for multiple simultaneously active robots to autonomously build user-specified structures, working from a high-level description as input.

Several key features characterize our approach. Structures can be built at scales much larger than that of the robots that build them, in three dimensions with gravity. The robot is designed with a philosophy of simplicity: it needs to perform only a few basic tasks, has relatively few sensors and actuators (10 and 3 respectively), and responds only to local conditions. Robots act independently, responding as necessary to the presence of others but each carrying out its own tasks and capable of building the entire structure alone. These choices are influenced by the social insect colonies that inspire our work. Finally, by designing the robots and building material together, we can incorporate passive elements into both in order to exploit "mechanical intelligence". Doing so increases robustness and makes it possible for robots to reliably carry out their tasks while remaining simple. Together these features enable an approach which we believe will be both useful and feasible for automated construction systems using robot swarms.

The organization of the paper is as follows. In §II we discuss related work, and in §III outline our approach. We describe the core tasks of climbing, navigation, and manipulation in §IV–§VI, and in §VII put them together in a demonstration of building a ten-block staircase. In §VIII we discuss the details of fabricating the robot and blocks. We discuss issues of high-level control for a full multi-robot collective construction system in §IX. In §X we conclude and discuss future work.

II. RELATED WORK

Collective construction of large-scale structures requires robots that can manipulate building materials as well as maneuver over and around the structures they create from them. While swarm robotics and mobile manipulation are both very large research areas, work in their region of overlap is relatively scant. This may be in part because each poses difficult enough problems on its own without the addition of those from another major area.

A few groups have demonstrated hardware systems intended for automated construction by distributed mobile robots, gen-
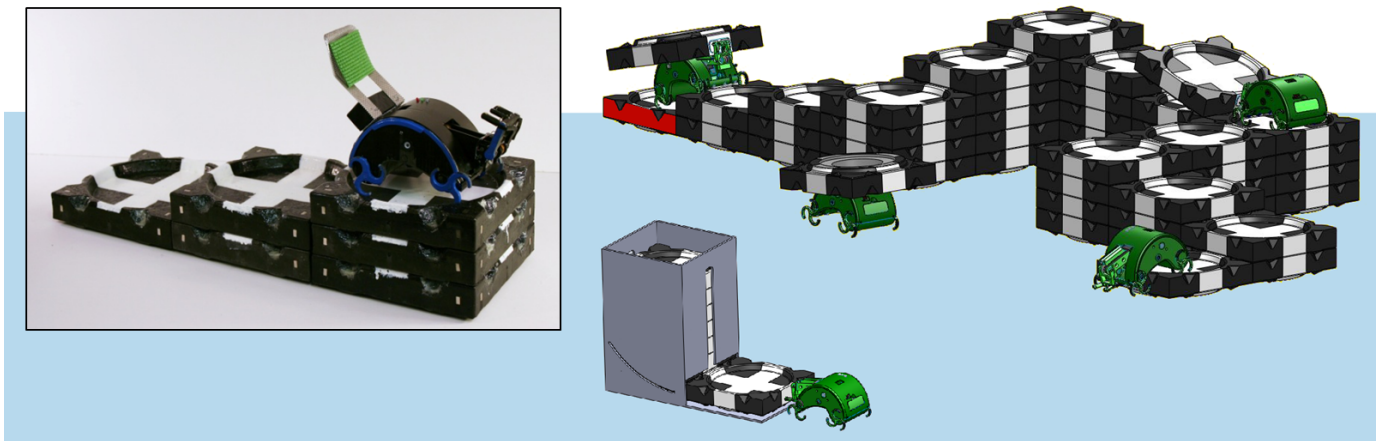
Fig. 1. System overview. Inset: one physical robot atop a six-block structure. Outer image: rendition of final goal system, where multiple robots collect free blocks from a cache and use them to build a user-specified structure. Construction begins at a marker block (red face), which provides a unique landmark.

erally for one- or two-dimensional structures but in some cases potentially applicable in three. Notable examples include robots capable of manipulating and/or maneuvering over specialized blocks [6, 10], two-dimensional systems where robots assemble structures from blocks [12] or struts [2], cooperative assembly by sophisticated robots [8, 9], and a strut-climbing robot that has been proposed as a component of a system where such robots form a structure together with passive struts [1]. Theoretical work on collective construction has also produced several algorithmic approaches to automating building with blocks or struts, in two and three dimensions [3, 4, 5, 10, 11, 12, 14].

Physical realization of such systems has been heavily limited. Algorithmic as well as hardware studies have encountered problems in reliable translation to practice, due to, e.g., assumptions about constraints on robot movement that prove difficult to realize, issues with reliability of attachment mechanisms, or gravitational difficulties for 3D systems. Most systems accordingly remain in simulation only. For 3D systems with climbing robots, previous demonstrations have not proceeded beyond a few steps and block placements.

Here we present a hardware system for autonomous construction of solid 3D structures under conditions of normal gravity, along with an example algorithm by which a multi-robot version could automatically build desired structures. This system represents a significant step beyond existing work, with several important characteristics: sensing is entirely onboard, with no global information available like GPS or overall structure state; robots are designed to act independently, under fully decentralized control; the sample structure built to illustrate system capabilities demonstrates high reliability, requiring a long sequence of successful actions performed over an extended period of autonomous operation.

III. APPROACH

Our goal is a system of autonomous robots that collect building material and use it to build a target structure in a flat, obstacle-free workspace (Fig. 1). Our approach is inspired by the decentralized, robust construction performed by termite colonies, where many simple insects simultaneously work to build large-scale structures. Robots in our system are identical, independent, and act on the basis of local information only. For the basic building material, we choose square prefabricated blocks; these are easier for robots to manipulate and to use in building rectilinear structures as are common in human construction. Blocks in structures line up as in a grid, with each elevated block supported by a stack of others directly beneath it. Robots have a smaller footprint than that of a block; they can maneuver on top of a wall one block wide, turn in place, and climb up or down the height of one block. By building staircases of blocks, robots can climb to greater heights and build complicated structures much larger than themselves.

As a first physical realization of this approach, we present a hardware system with a single robot, demonstrating the autonomous construction of a ten-block structure. This goal requires the robot to be able to carry out three kinds of tasks: (1) climbing onto a block to reach higher levels of a structure; (2) navigating on a structure, which comprises the subtasks of maneuvering safely on the structure without falling off and keeping track of its movement relative to the structure; (3) lifting, carrying, and attaching blocks to the structure. Furthermore, each task must be achieved with high reliability, so that an autonomous robot can successfully build a complex structure involving a large number of individual actions.

IV. CLIMBING

Our primary criterion in designing an effective climbing robot was for it to be able to climb blocks as tall as possible, so that a structure of a given height would require fewer blocks. We further required that the robot should be able to locomote on level surfaces both on and off the structure, and that it should not require additional hardware or complex control routines for climbing, beyond what would be needed for level locomotion.
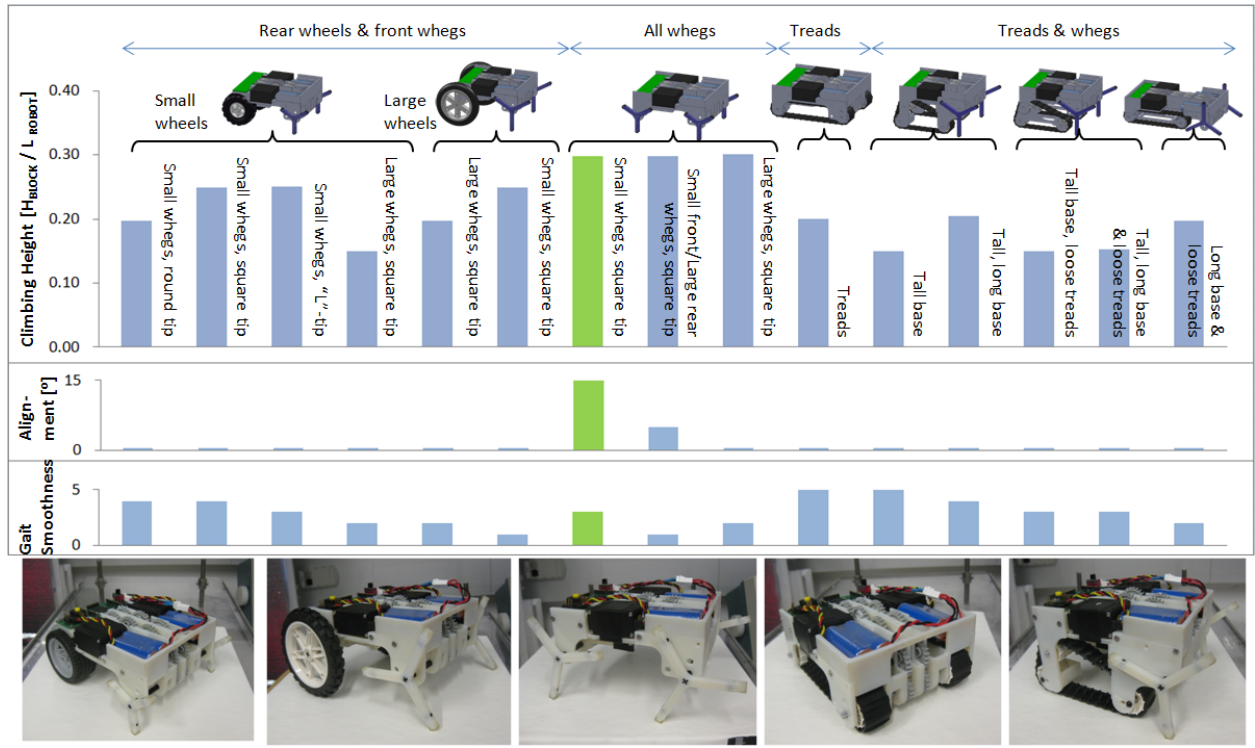
Fig. 2. Performance of various hardware configurations of test robot. Top bars show the height of the tallest block the robot was able to climb successfully in 8 out of 10 trials, relative to the robot's own length. Middle bars show the maximum angle θ for which, if the robot approached the block at θ from perpendicular, it would straighten itself out in the process of climbing. Lower bars give a qualitative rating of gait smoothness on a level surface (5=smoothest travel, 1=most lurching). Base configurations use short length and low height unless otherwise noted. "Loose" treads have an (unactuated) vertical degree of freedom at their front end. The configuration we selected is highlighted in green. Photos show several example configurations tested.

To test different possibilities, we developed a general robotic base capable of reconfiguring into many different hardware schemes, using various sizes of wheels, treads, whegs (a hybrid of wheels and legs [7]), and combinations of the three (Fig. 2). This test robot used four-wheel drive; the length of the base and its clearance above the ground could be adjusted, and three 45g weights were used to adjust its weight distribution (in general, a center of mass lower to the ground and closer to the front of the robot allowed it to climb taller blocks). Like the final robot, the test robot was untethered, with self-contained power and controller. Each of the hardware configurations was put through the following tests:

Maximum climbing height: The robot had 30 s to climb a block of given height, tested in increments of $0.05\times$ the robot's length. Fig. 2 reports the tallest height for which 8 of 10 trials were successful.

Self-alignment: The robot was misaligned by up to 15 degrees from perpendicular in its approach to the tallest block it was able to climb in the previous test. The figure reports the maximum such angle for which the robot returned to perpendicular as it climbed in 8 out of 10 trials.

Gait smoothness: The robot was set to drive straight for 30 cm on a level surface without feedback; the steadiness of the motion was visually judged on a scale from 1–5 (5 being very smooth, 1 corresponding to heavy vertical lurching). Vertical smoothness is desirable for consistency of conditions for visual feedback and block manipulation.

Based on these results (Fig. 2), we chose the configuration with four small whegs, due to its ability to climb the tallest blocks and align itself mechanically with the structure as it climbed, with acceptable smoothness on level ground. We next added angled notches to the block edges (Fig. 3), which acted as a step allowing robots to climb still taller blocks, and helped correct lateral misalignments, so a robot displaced to one side would slide back to the center of a notch. Finally, the whegs were given a curved shape (Fig. 3) to improve the robot's ability to turn in place, to help keep it from falling off the structure while turning; a slight reduction in the maximum climbing height was a side effect of this change. The final height of the blocks was 21% of their length, or 26% of the length of the robot.

The use of whegs also lets the robot locomote over rough terrain. To explore this point, we had the robot travel straight ahead open-loop over various unstructured terrains and timed how long it took to travel 1.2 m, with 5 trials for each condition: linoleum (9.5 ± 0.5 s), berber carpet (10.2 ± 0.1 s), soil with plant litter (11.0 ± 0.4 s), gravel (14 ± 1 s, on pebbles of $\sim25\pm10$mm diameter), grass (14.9 ± 0.9 s), snow (18 ± 3 s).

The final design lets the robot climb and self-align while climbing, with minimal control and hardware complexity. Moreover, the robot action—simply driving forward—is identical for climbing and for level locomotion.
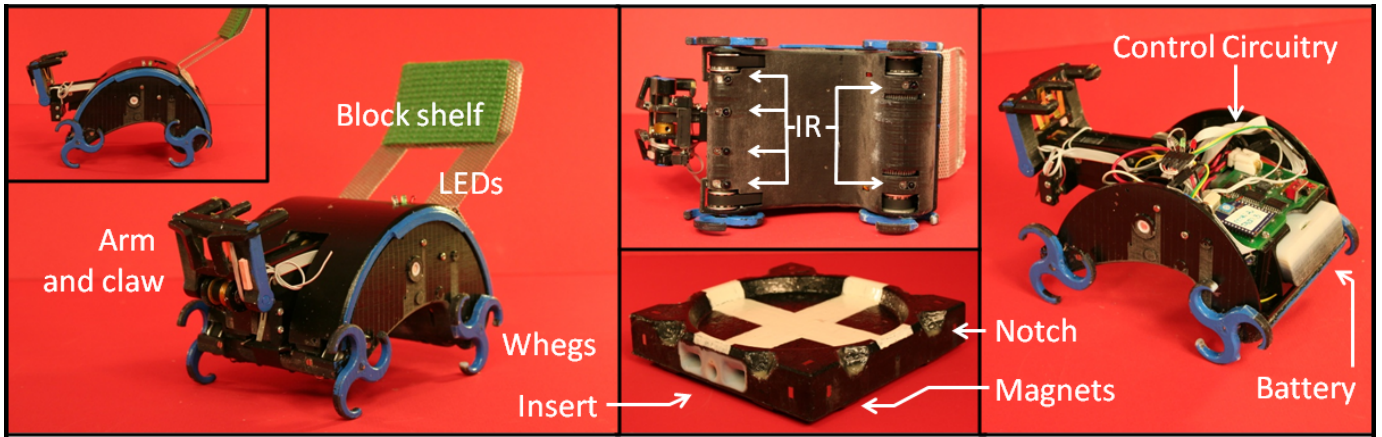
Fig. 3.   Overview of robot and block design.

V. NAVIGATION

The task of navigation requires the robot to maneuver atop a structure, including level motion, turning, and climbing up or down, while accurately keeping track of its position and orientation with respect to the structure.

To provide position feedback, 6 active infrared (IR) sensors are added to the belly of the robot, and blocks are colored black with white crosses (Fig. 3). This pattern provides the robot with information about its position and orientation relative to a block. When the robot sits in the center of a block facing one of its edges, the four IR sensors in the corners are in the four black regions, while the two front center sensors are above a white stripe. A state machine lets the robot keep track of its progress. For instance, it can tell it has moved forward one block when it reaches the end of the following sequence: (a) all corner sensors see black, (b) both rear sensors see white, (c) all front sensors see white, (d) all corner sensors see black. While climbing, the robot's rear whegs frequently slip, with the result that the outer front sensors register black before it has finished climbing to the higher block. To detect this condition and prevent the robot from prematurely thinking it has reached the next block, a mercury tilt switch gives the robot feedback about its pitch.

The robot turns using differential steering, with one motor controlling the two left whegs and a second controlling the right ones. It is necessary for the robot to be able to reliably turn in place atop a block without falling off. To correct the otherwise large drift that occurs with this four-whegs design, a circular indentation is added to the top surface of blocks. This passive mechanical feature helps keep the robot in place while it turns, while letting it easily climb out when it moves straight ahead.

Experiment: To test the robot's ability to navigate accurately and reliably, the robot was directed to repeatedly traverse the path shown in Fig. 4. It autonomously performed 40 laps (20 each while carrying a block and without), each involving two ascents, two level crossings, two descents, and a total of six
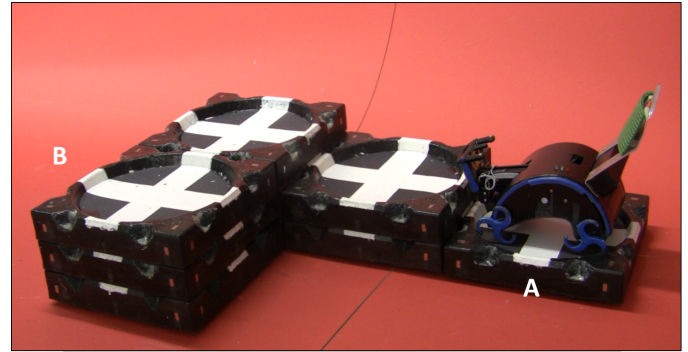
	Without block	With block
Time (sec)	112 ± 8	122 ± 12
# slips on step 1	0.5 ± 0.6	1 ± 2

Fig. 4.   Experimental setup and results for navigation tests. The robot repeatedly travels from block A to B and back. One lap thus involves two ascents, two descents, two level transitions, and six 90° turns. The table reports time required to complete a lap and number of times the rear whegs slipped while climbing the first step, over 40 laps. No errors in navigation occurred.

90° turns. No errors occurred, i.e., the robot always correctly kept track of its actual movement and never failed to move between blocks or turn as intended.

As these results demonstrate, the combination of simple feedback and passive features in the blocks allows the robot to perform autonomous navigation with high reliability. This reliability becomes very important with larger structures as the robot spends considerable time navigating.

VI. MANIPULATION

To build a structure, the robot must be able to pick up blocks, carry them while navigating on the existing structure, and attach them in desired locations. We equip the robot with an arm and gripper, and add passive mechanical features to the blocks that help the robot to grasp them and to align and secure them to the structure.
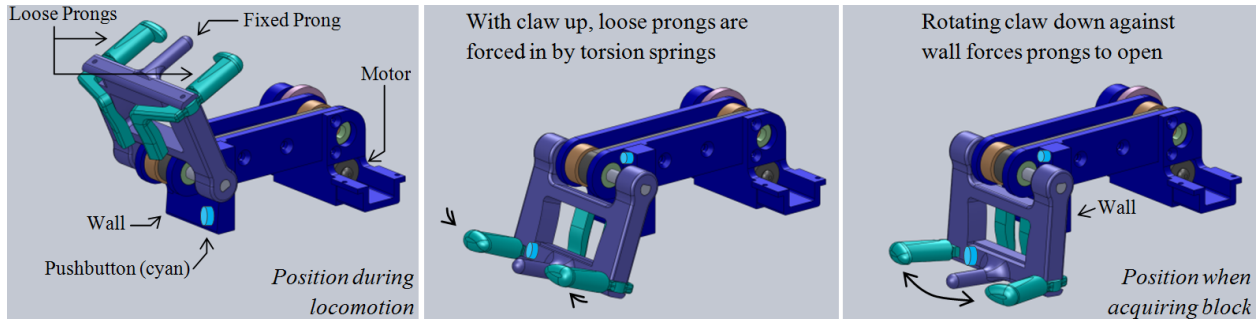
Fig. 5. SolidWorks model of the arm, demonstrating how the motion of the claw moves the prongs to release a block or hold it securely.

The arm (Fig. 5) needs to be small so that the footprint of the robot remains smaller than that of the blocks, and it must not interfere with the robot's maneuverability. In keeping with our philosophy of simplicity, the arm uses a single motor, which rotates a pronged "claw" up and down. The robot lowers the claw when it needs to pick up or attach a block, and holds it in its upper position at all other times as it navigates (whether or not it holds a block). A padded shelf on the robot's back helps support carried blocks. The claw has three prongs, two that open and close to grasp or release a block, one that remains fixed to stabilize a held block; a plastic insert matching the prongs is embedded in each block as a handle (Fig. 3). The same motor that rotates the claw also controls its second degree of freedom, the opening and closing of the gripper, through a passive mechanical design: Torsion springs normally force the movable prongs inward; when the claw is lowered, the rear side of the prongs is pressed against a "wall", forcing them to open.

As manipulation uses only one actuator, the robot controller needs to process little information when picking up a block. Three sensors are used: two pushbuttons that register when the claw is at the upper and lower ends of its range, and a third at the base of the prongs that indicates whether a block is being held.

The blocks incorporate passive mechanical features that achieve alignment and secure attachment, enabling a simple mobile robot to reliably perform these potentially difficult manipulation tasks. The ends of the prongs and sides of the matching block inserts are filleted to correct alignment errors as the robot picks up a block. Blocks are given features on their bottom faces complementary to the indentations and notches on their upper surfaces. These matching features both guide a block into exact alignment as it is lowered and help keep a stack of blocks firmly attached. Neodymium magnets on all six faces, two per vertical face and four each on top and bottom, provide one more alignment mechanism and secure adjacent blocks together.

Experiment: To test the robot's ability to lift and attach blocks reliably, the robot was directed to repeatedly take blocks from one location and attach them in another (Fig. 6). The robot acquires a block from a docking station (a specialized block without magnets to hold the one above it in
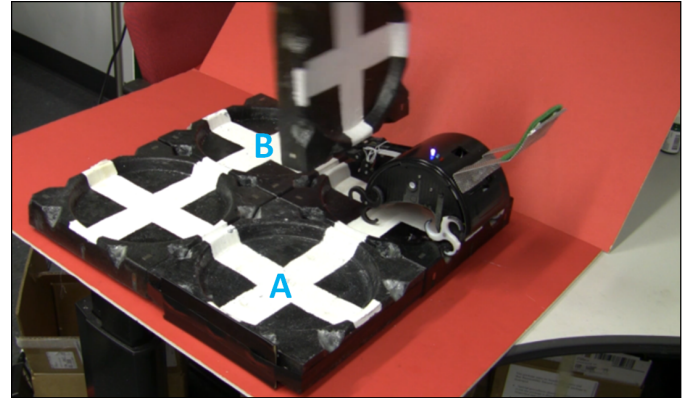
	Pick up block	Lower and attach block
Time (sec)	15 ± 5	24 ± 5

Fig. 6. Experimental setup and results for manipulation tests. The robot repeatedly picks up a block from site A and attaches it at site B. No errors occurred in 20 trials.

place), turns 90°, attaches the block, and turns back while the block is returned manually to the docking station. No errors occurred in 20 trials.

VII. CONSTRUCTION DEMONSTRATION

We demonstrate the ability of the system to autonomously build nontrivial structures by directing the robot to build a ten-block staircase (Fig. 7). This task unifies the separate robot capabilities of climbing, navigation, and manipulation discussed above. An initial line of blocks and a docking station are placed by hand; the robot is set on the first block facing the docking station; and new blocks are added manually to the docking station as needed. The construction process is otherwise entirely autonomous.

Most robot code is run on the on-board processor, with higher-level code currently running on a separate laptop for ease of debugging and communicated to the robot via Bluetooth. The robot can be programmed with high-level primitives like a Logo turtle (move forward one block, turn left or right 90°, pick up or attach a block); the program used to build the staircase is given in the caption of Fig. 7.

The reliability of the individual primitives allows the system to achieve its end goal even in a procedure of over 100 steps.
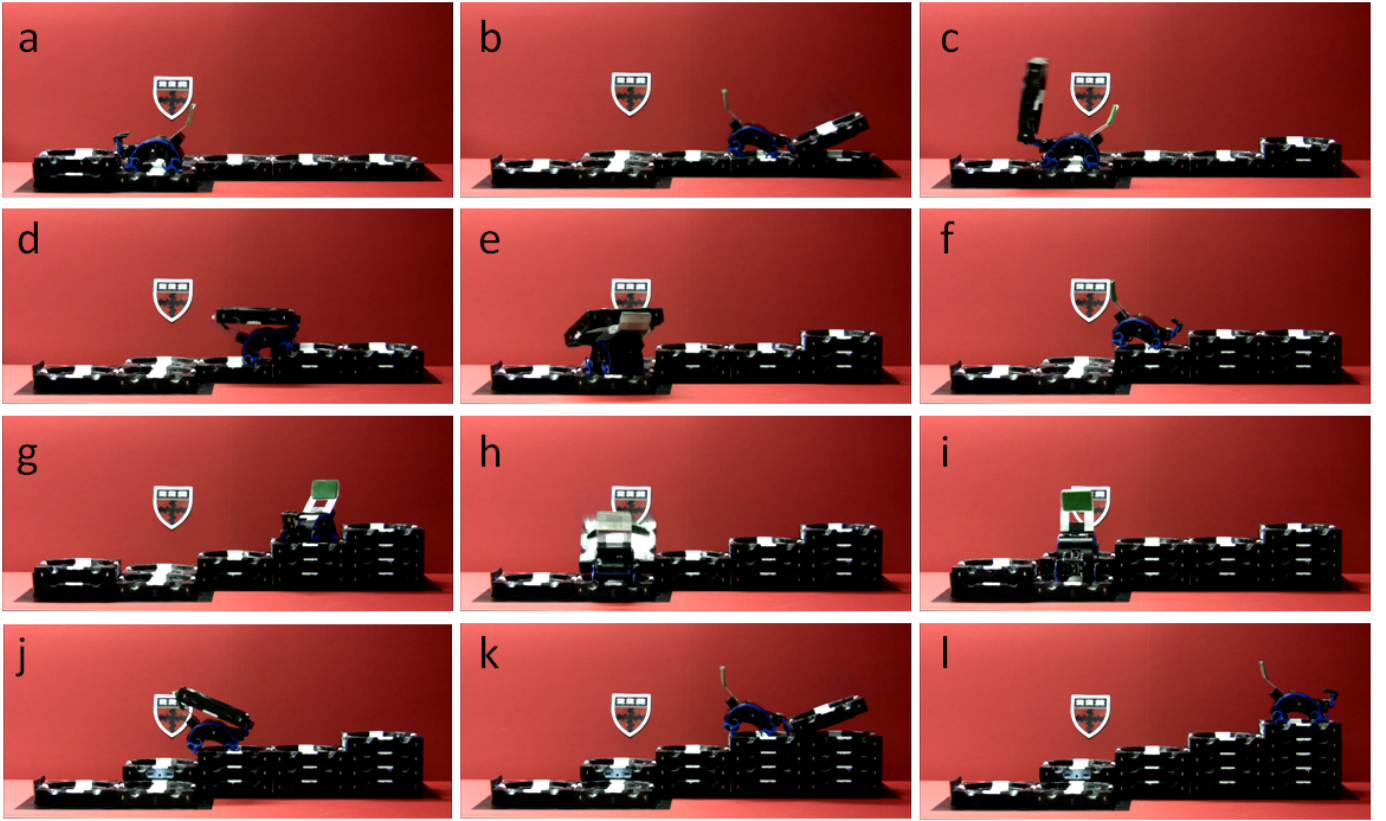
Fig. 7. Snapshots in the process of autonomously building a ten-block structure. The robot collects blocks from the docking station at left, where new blocks are added by hand as construction proceeds. The initial base layer of five blocks plus docking station was placed by hand. The robot program for building this structure is URFRFFDRRFFLFRURFRFDRRFLFRURFRFFDRRFFLFRURFRDRFRURFRFDRRFL-FRURFRFFDRRFFLFRURDLURFRDRFRURFRFDRRFLFRURFRFFDF, where the five high-level primitives are: move forward one site (F), turn left 90° (L), turn right 90° (R), pick up block from site just ahead (U), attach block at site just ahead (D). The robot automatically climbs up or down when moving forward to a site at a higher or lower level. The robot executes this 106-step program over a period of 24 minutes.

VIII. MANUFACTURING

Robots and blocks should both be easy to produce, in a swarm construction system where many such components will be required. The robot is built with a combination of off-the-shelf and 3D-printed components. Blocks are made of foam, with magnets and plastic inserts embedded, in a way intended to be mass-producible.

The robot measures 175mm (length with claw raised) × 110 mm (width) × 100 mm (height excluding block shelf), and weighs 810 g. The on/off switch, a replaceable battery pack, three LEDs for debugging, and a programming plug for interfacing with the microcontroller are all accessible from the outside. Propulsion is generated using two micro metal gear motors, with belts and timing pulleys to provide four-wheel drive; a third such motor drives the arm. All gears, bearings, belts and springs used in the robot are standard off-the-shelf components. The only hand-fabricated elements are the axles, which are machined from 3/16" aluminum rods; the block shelf, which is bent aluminum; and the pushbutton for sensing a held block, made from a folded brass strip. All other mechanical hardware in the robot, including the top and bottom halves of the shell, the arm and claw, and the whegs, are fabricated with an Objet 3D printer. Most of the contact surface of each wheg is printed with soft rubber to increase traction. The main controller is an ATmega1281, which allows easy programming and many digital and analog interfaces. Other electronics include power converters, analog filters for sensor processing, three motor drivers, IR and tilt sensors, pushbuttons for tactile sensing, and a Bluetooth module for debugging via a PC. The IR sensors are controlled by circuitry separate from the control board; their circuit both handles the output signals and turns them on one at a time to save power, without requiring control attention from the main processor. The robot runs on two Li-Ion 7.2V 750mAh battery packs, one powering the actuators, the other the controller and sensor electronics.

Blocks are fabricated from marine foam using a custom silicone mold and a two-part 4lb/ft^3 liquid foam. Inserts are printed with a 3D printer. One insert and 16 magnets (each with a magnet-to-steel-plate pull force of 1.93 lbs) are added to the mold before the foam is poured and become embedded in the block as it cures. A small layer of caulk is added to the surface of the notches to help the robot climb. The final

blocks measure $21.5 \times 21.5 \times 4.5$ cm and weigh 165–210g (depending on variations in fabrication).

IX. High-Level and Multi-Robot Control

A system capable of building staircases, like the one presented above, is similarly capable of building a wide range of complex structures. To extend to a full collective construction system, three kinds of extensions are necessary, associated with high-level user interfaces, navigation away from the structure, and multi-robot interactions.

For usability, an automated construction system should have an interface where users need specify only a desired final structure, without having to become involved in the details of robot actions. Alg. 1 gives an example of a robot routine, built on Logo-like primitives like those described in §VII above, that allows the automatic generation of structures from a large class of possibilities. A user provides as input a high-level representation, specifying only where blocks should be located in the target structure. A simple offline compiler uses depth-first search to convert this to a "structpath" representation, a directed path in the plane annotated with the height of the stack of blocks at each site in the path (Fig. 8B), if a valid structpath can be found. A valid structpath is one where (a) the heights of the stacks at successive sites differ by at most 1, and (b) no site is bordered on opposite sides by two sites that both came earlier in the directed path. The first condition ensures that robots can traverse the target structure along the given path; the second avoids situations where it would be physically difficult for a robot to attach a block. Robots then follow Alg. 1 using the static structpath representation to guide their movement along the structure. A unique marker block (shown with red face in Fig. 1) indicates where construction is to begin, and serves as the origin in a coordinate system that will be physically embodied by the square blocks as construction proceeds.

A group of robots using Alg. 1 is guaranteed to produce a requested structure irrespective of variations in factors like the number of robots and the order and timing of their actions. A companion paper ([13], in review) proves the correctness of Alg. 1, demonstrating that it will lead to completion of the target structure without problems like deadlock or untraversable intermediate states.

A structure built according to Alg. 1 grows as an elongating trail laid down along the compiled structpath, with a parade of robots starting at a fixed entry point and leaving at a variable exit point, and with a rising and falling staircase of blocks in between. This control scheme is motivated by the importance of resolving conflicts between multiple robots working in the same space. By specifying a one-dimensional path for robots to follow on and around the structure, the majority of potential conflicts can be avoided, and others resolved by giving priority to the robot further along the path (e.g., a robot approaching the structure perimeter waits for one already following the perimeter to pass; a robot traveling atop the structure waits for one in front of it to attach a block). Fig. 8 shows five robots in simulation producing an example structure.

Algorithm 1 Pseudocode robot program for collective construction.

---

loop
    get block from supply
    go to structure
    follow structure perimeter to marker block
    climb onto structure
    while on structure **do**
        follow structpath
        if holding block **then**
            if structpath specifies block at current site **and** (previous site was at higher level **or** previous site was at same level and meant to be empty according to structpath) **and** next site along structpath is at same level **then**
                move to next site along structpath
                attach block at site just vacated
interrupt: if robot detected ahead **then**
    if on structure **and** other robot ahead on structpath **then**
        pause
    else if not on structure **then**
        perform collision avoidance as appropriate

To implement this algorithm, the current hardware platform will need to be extended with the following additional hardware capabilities: (1) For interactions between robots, robots need to be able to detect the presence of other robots ahead within a short distance. For navigation away from the structure, robots need to be able to (2) travel between the structure in progress and the cache of free blocks, e.g., using long-range beacons to find the way to both sites through unstructured terrain; (3) perform wall-following upon reaching the structure, until finding the unique marker block; and (4) extend the structure's base layer by adding a block at ground level, without the benefit of the blocks' visible patterns or the physical template of an underlying block available for position adjustment. We plan to demonstrate these capabilities in future work.

X. Conclusion

In this paper we have presented a system for autonomous construction, where a mobile robot manipulates passive building blocks to construct three-dimensional structures potentially much larger than itself under conditions of gravity. Designing the robot and blocks together, and incorporating passive mechanical features, enables a simple robot to reliably perform the requisite tasks of climbing, navigation, and manipulation without sophisticated sensors or control mechanisms.

We have further presented an example algorithm for multiple simultaneously active robots to build large structures. Our next step will be to extend our hardware system to use this approach to demonstrate fully autonomous construction of large user-defined structures by multiple independent robots under decentralized control. Doing so will require the extensions to the robots' sensing and control discussed in §IX, in particular
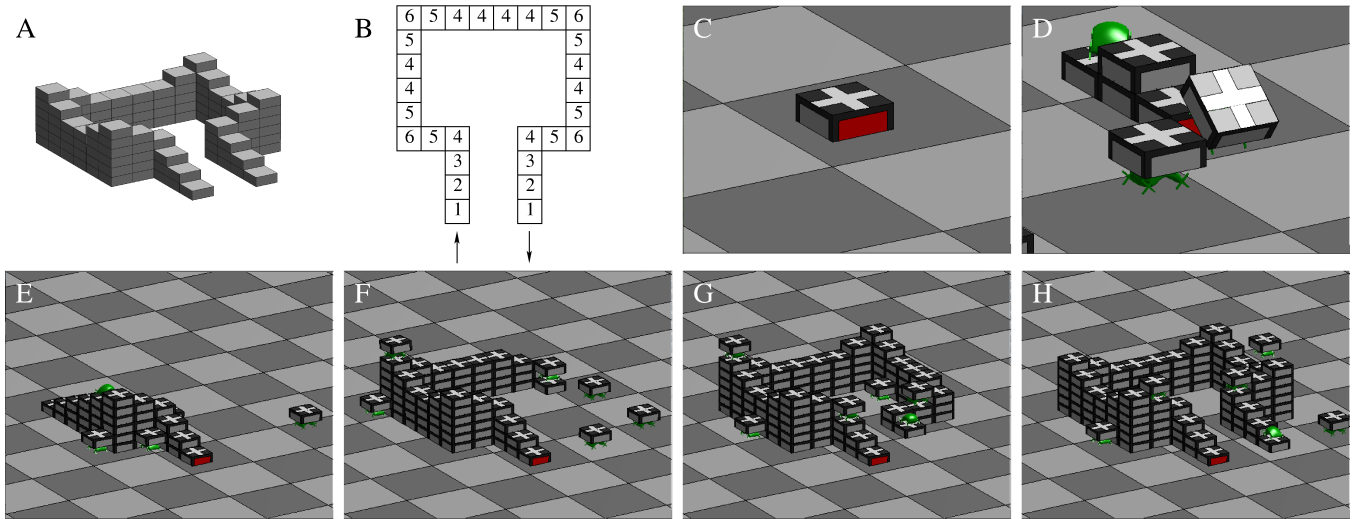
Fig. 8. Automation of collective construction. (A) The user specifies a desired final structure. (B) An offline compiler converts the input structure to a "structpath" representation, a directed path specifying stack height at each site. (C–H) Snapshots from a simulation of ten robots building the structure using the structpath representation and Alg. 1.

the abilities to sense other robots nearby and to navigate (not just locomote) away from the structure.

The hardware platform described here can potentially be used with other control algorithms to build structures from other classes of possibilities. For example, alternate algorithms can be used to build structures for which no valid structpath exists; probabilistic rules for movement and attachment could lead to structures not fully predetermined; or robots could build temporary staircases as scaffolds, removing them after a primary desired structure is complete [3, 13], thus allowing them to build taller structures with otherwise unclimbable walls.

A video showing the robot capabilities, staircase demonstration, and castle simulation can be found at http://www.eecs.harvard.edu/ssr/projects/cons/termes.html .

REFERENCES

[1] C. Detweiler, M. Vona, K. Kotay, and D. Rus. Hierarchical control for self-assembling mobile trusses with passive and active links. *Proc. ICRA 2006*, pages 1483–1490, May 2006.

[2] J. Everist, K. Mogharei, H. Suri, N. Ranasinghe, B. Khoshnevis, P. Will, and W. Shen. A system for in-space assembly. *Proc. IROS 2004*, pages 2356–2361, 2004.

[3] A. Grushin and J. Reggia. Automated design of distributed control rules for the self-assembly of prespecified artificial structures. *Robotics and Autonomous Systems*, 56(4):334–359, 2008.

[4] C. Jones and M. Matarić. Automatic synthesis of communication-based coordinated multi-robot systems. *Proc. IROS 2004*, pages 381–387, 2004.

[5] N. Napp and E. Klavins. Robust by composition: Programs for multi robot systems. *Proc. ICRA 2010*.

[6] K. Petersen. Autonomous construction of temporary human habitats. Master's thesis, University of Southern Denmark, 2008.

[7] R. Quinn, G. Nelson, R. Bachmann, D. Kingsley, J. Offi, and R. Ritzmann. Insect designs for improved robot mobility. *Proc. CLAWAR01*, pages 69–76, 2001.

[8] B. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proc. IEEE*, 94:1425–1444, 2006.

[9] A. Stroupe, A. Okon, M. Robinson, T. Huntsberger, H. Aghazarian, and E. Baumgartner. Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. *Autonomous Robots*, 20:113–123, 2006.

[10] Y. Terada and S. Murata. Automatic modular assembly system and its distributed control. *Int. J. Robotics Research*, 27(3–4):445–462, 2008.

[11] J. Werfel and R. Nagpal. Three-dimensional construction with mobile robots and modular blocks. *Int. J. Robotics Research*, 27(3–4):463–479, 2008.

[12] J. Werfel, Y. Bar-Yam, D. Rus, and R. Nagpal. Distributed construction by mobile robots with enhanced building blocks. *Proc. ICRA 2006*.

[13] J. Werfel, K. Petersen, and R. Nagpal. Distributed multi-robot algorithms for the TERMES 3D collective construction system. In review.

[14] S. Yun, M. Schwager, and D. Rus. Coordinating construction of truss structures using distributed equal-mass partitioning. *Proc. ISRR 2009*.