Robotica (2011) volume 29, pp. 103–121. © Cambridge University Press 2011
doi:10.1017/S0263574710000743

A general stiffness model for programmable matter and modular robotic structures

Paul J. White*, Shai Revzen, Chris E. Thorne and Mark Yim

Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, USA

(Received in Final Form: November 3, 2010)

SUMMARY

The fields of modular reconfigurable robotics and programmable matter study how to compose functionally useful systems from configurations of modules. In addition to the external shape of a module configuration, the internal arrangement of modules and bonds between them can greatly impact functionally relevant mechanical properties such as load bearing ability. A fast method to evaluate the mechanical property aids the search for an arrangement of modules achieving a desired mechanical property as the space of possible configurations grows combinatorially. We present a fast approximate method where the bonds between modules are represented with stiffness matrices that are general enough to represent a wide variety of systems and follows the natural modular decomposition of the system. The method includes nonlinear modeling such as anisotropic bonds and properties that vary as components flex. We show that the arrangement of two types of bonds within a programmable matter systems enables programming the apparent elasticity of the structure. We also present a method to experimentally determine the stiffness matrix for chain style reconfigurable robots. The efficacy of applying the method is demonstrated on the CKBot modular robot and two programmable matter systems: the Rubik's snake folding chain toy and a right angle tetrahedron chain called RATChET7mm. By allowing the design space to be rapidly explored we open the door to optimizing modular structures for desired mechanical properties such as enhanced load bearing and robustness.

KEYWORDS: Modular Robotics; Programmable Matter; Lumped Parameter Stiffness Model.

1. Introduction

The fields of programmable matter and modular robotics study the development of robotic systems comprising many simple units. These systems are capable of changing their form by reconfiguring the units. Reconfigurable form enables these systems to adapt to unexpected events, tolerate faults through redundancy, and extend to numerous applications.

In the context of robotics, the term "programmable matter" refers to systems composed of many active or passive units capable of being reconfigured to form different shapes. For example, Goldstein and Mowry present the idea of *telepario* where millions of identical small rearranging elements called "catoms" form shapes to make a 3D display, analogous to the telephone and television.[16] Shape forming methods include stochastic self-assembly,[4,50] external actuation,[53] self-disassembly,[15] and folding.[17,21] These systems discretely approximate a shape with a lattice of units.

Modular robots are similar to programmable matter in that they comprise many simple repeated units called modules. However, these modules typically have one or more actuated degrees of freedom (DOF) making the structure a functional robot. Modules in modular *self-reconfigurable* systems[56] can make and break bonds to neighbors and relocate on the robotic structure to form a new system.

Numerous modular robots have been developed; each can be classified as a chain, lattice, hybrid, or mobile style system.[56] Chain style systems have demonstrated numerous robot morphologies by connecting modules in series to form, for example, a snake, a legged robot, a rolling track, and a manipulator.[19,40,44,54,55] Lattice style modular robots, similar to programmable matter, comprise units occupying discrete positions in two[9,33,39] or three[27,45,47] dimensions. Hybrid modular robot systems[24,34,41] combine reduced reconfiguration complexity of the lattice style with articulation abilities of the chain style. A mobile style modular robot, one of the first modular robot implementations,[14] uses the environment to maneuver and typically forms larger groups of mobile units like trains.[6,32] This paper is primarily concerned with chain, lattice, and hybrid systems.

The work cited so far tends to focus on the physical aspects of designing and manufacturing these systems; however, planning the motion of the components to achieve a desired shape is equally important. As the number of modules in a system increases, the computational time complexity of module motion coordination typically grows exponentially.[36] Shape formation planning approaches include intermediate structures,[38] meta-modules,[5,46,48] scaffolding,[43] gradient following,[42,58] and hole motion.[11]

1.1. Structural stiffness

While current programmable matter and modular robotic research focuses on hardware and shape formation planning, little research has studied the analysis of mechanical properties of the formed structures. Depending on the application, structures formed from these systems must satisfy stiffness requirements for a given loading.

* Corresponding author. E-mail: whitepj@seas.upenn.edu

For example, research in programmable matter looks beyond the active shape representation to develop systems with programmable intrinsic properties such as elasticity. A programmable matter system comprising small units each capable of changing the stiffness of its bond to a neighbor could form a wrench with soft conforming interfaces for human hands and a rigid core to resist bending moments. Programmable matter could conform to a victim's injured limb as a temporary cast, while programmable stiffness could create flexible joints where needed while remaining rigid in critical areas to protect broken bones.

Unlike custom designed robots, a modular robot is not designed to optimally bear mechanical loads for a specific set of tasks. Different configurations of the same set of robot modules may differ substantially in their load bearing ability. An analysis method is necessary to evaluate candidate configurations or reconfiguration sequences of a modular robot bearing a load. For example, in an urban search and rescue operation, the system could squeeze into a cluttered rubble pile in a configuration optimized for locomotion to find a victim. Then a planner could determine the configuration that best supports the unstable structure while resisting compressive buckling loads and preventing collapse.

Designers and planners require a modeling and analysis methodology to find hardware solutions and configurations for these systems that suit the task. However, algorithmically determining the configuration that optimally satisfies stiffness requirements is challenging. The stiffness of a structure depends not only on the global shape, but also on the internal arrangement of modules and bonds as the modules and bonds often exhibit anisotropic stiffness characteristics. As in the shape formation problem, the configuration search space grows exponentially in the number of modules. Thus as the cost function routine of a configuration optimizer, a static solver must be able to quickly approximate the deformation of a configuration under load.

Additionally, the analysis method should be sufficiently expressive in order to model various current systems and to aid design of new systems that use different connection mechanisms. While numerous connection methods have been explored including magnetic,[33,45] electrostatic,[25] and pressure,[49] many systems[24,27,28,39,47] choose mechanical connections for high rigidity. The mechanics of the connection between units and a unit's structural stiffness determine the stiffness of the overall structure which can vary widely and typically form nonlinear relationships between displacement and applied loads.

Finite element analysis (FEA)[30] is a standard technique for analyzing the stiffness of a given structure and would seem to fit well with this problem as it has an inherent discretized modular approach. In contrast, the simulator presented here does not require expertise with or ownership of an FEA software package. Using the simulator developed in Matlab, one can easily express different configurations and incorporate planning and optimization routines. This paper presents a parameterized model that does not require an explicit definition of module geometry, useful early in the design stage. A parametric model provides an intuitive design tool enabling users to quickly evaluate possible solutions. In addition, it allows the experimental roboticist to characterize a system by fitting data to a model with a minimal number of parameters. While FEA can achieve a desired accuracy given a fine enough mesh and sufficient run time, the computational complexity of programmable matter or modular robot stiffness optimization requires the use of a computationally efficient approximation with less focus on accuracy.

1.2. Methodology

In this paper, we introduce a structural stiffness analysis method capable of modeling programmable matter and modular robotic systems composed of mechanically connected units. The method can express arbitrary elastic connections within and between modules to first order, providing a model for approximating structure displacements under load. The model handles nonlinearities such as collision and gaps between connection components. Using a new prototype programmable matter system, RATChET7mm described in Section 2 as an example, Section 3 presents the modeling method.

The simulator takes advantage of properties exhibited by modular systems: (1) lumped compliance within bonds and/or modules and (2) repetition of a finite set of module/connector types. Because self-reconfiguration requires a module capable of attaching and detaching, the bonding method is reversible, and its significantly greater compliance typically dominates the system's behavior under static load. Other classes of modular systems such as the CKBot system,[37,40] exhibit compliance within the module. Lumping the compliance into the connector and module reduces the number of modeled degrees of freedom while maintaining the ability to predict displacements. The stiffness of the modules and the bonds can be precomputed to a desired level of detail through approximate or thorough mechanical analysis,[3] FEA, or experimentation. Once determined, the simulator can readily generate arbitrary configurations comprising copies of modules and bond types and simulate the behavior. In addition, having a reduced parameter set provides design intuition and enables experimental determination of stiffness.

We demonstrate the process and efficacy of the methodology through simulation (Section 4) and experimentation (Section 5) with three systems. Using the Rubik's Snake, a toy that has inspired several programmable matter systems, we demonstrate how to quickly approximate stiffness by mechanical analysis. With the CKBot system[37,40] we present a method for experimentally determining the stiffness parameters of a modular robot and validate the model's ability to predict displacement under load (Section 5.3).

Griffith demonstrated that a chain of right angle tetrahedrons can be folded into arbitrary shapes superimposed on a dodecahedron lattice.[17] The chain is never broken (consisting of permanent hinges) and, when folded, the faces of neighboring modules are reversibly bonded (crosslinked) to retain a shape. Thus, an important question is: how can one exploit this bond type heterogeneity to design a structure with a desired stiffness? Using the RATChET7mm system that forms shapes from a folded right angle tetrahedron chain, we show that a system with two

http://dx.doi.org/10.1017/S0263574710000743

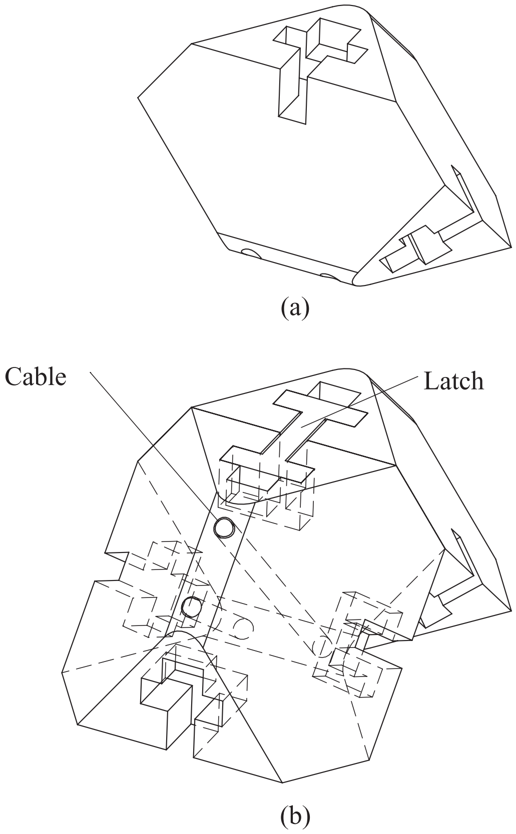


Fig. 1. Rendering of a single truncated right angle tetrahedron shaped RATChET7mm module (a) and two modules connected by an I-beam shaped latch and threaded with a pair of cables (b).

types of bond methods can exhibit programmable elasticity (Section 4.1).

2. RATChET7mm

To demonstrate the validity of the modeling method and provide illustrative examples, we present a prototype programmable matter system. The Right Angle Tetrahedron Chain Externally actuated Testbed (RATChET) is a programmable matter system composed of identical truncated right angle tetrahedrons (referred to in this paper as right angle tetrahedrons or simply tetrahedrons) that are permanently hinged together and connected with reversible bonds to hold a folded shape.

The RATChET7mm extends previous work on externally actuated chains of right angle tetrahedrons.[51,52] When configured into a lattice structure, the distance between the centroids of neighboring RACThET7 mm modules is 7 mm. While the earlier versions were larger and had automatically programmed external actuation and latching states, the RATChET7mm's main purpose is to provide an analogous passive system to study the stiffness of assembled structures and so is made to be manually reconfigurable to simplify construction. It is sufficiently simple to allow derivation and verification of an analytical model of the stiffness of a module connection.

Figure 1(a) shows a rendering of a single truncated right angle tetrahedron module. Figure 1(b) depicts two connected RATChET7mm modules. Neighboring modules in a configuration (e.g., Fig. 2(c)) connect with an I-beam shaped latch that is press fit into the 'T' shaped slot on the truncated face.

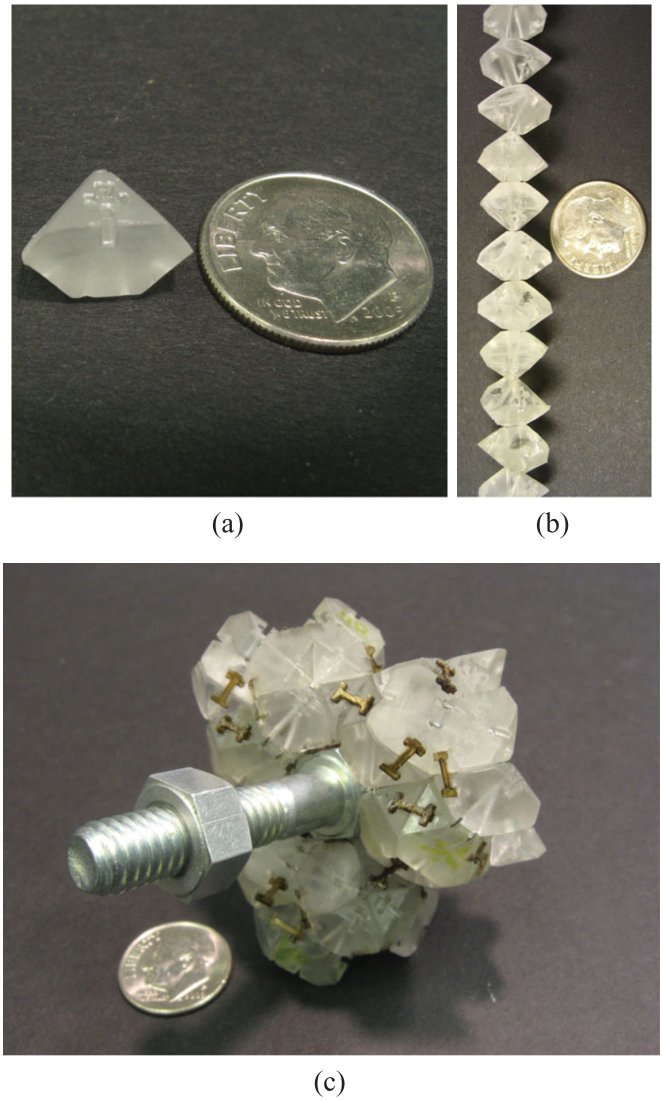


Fig. 2. (a) Single RATChET7mm module. (b) Unfolded chain of RATChET7mm modules and (c) 54 RATChET7mm modules folded and latched to form a 3/8" socket wrench.

Two circular holes through the module allow steel cables to route through. A chain is composed of a set of these tetrahedrons strung along the cable as shown in Fig. 2(b). The cable acts as the hinge and facilitates the folding process by allowing the modules to sequentially slide into place.

One hundred modules were manufactured from a stereolithography (SLA) process using polycarbonate-like material. The latches are manufactured from laser cut polycarbonate and the cable is steel rope.

The system begins as a chain such as the one in Fig. 2(b). Each tetrahedron is attached to a neighboring tetrahedron by a one degree-of-freedom hinge and the adjacent hinges are perpendicular. A fold is defined as bending at this hinge until two tetrahedrons make contact and can latch. The geometry of the tetrahedron is such that when it is folded, six tetrahedrons form a hexahedron, and four of those hexahedrons form a rhombic dodecahedron.

http://dx.doi.org/10.1017/S0263574710000743

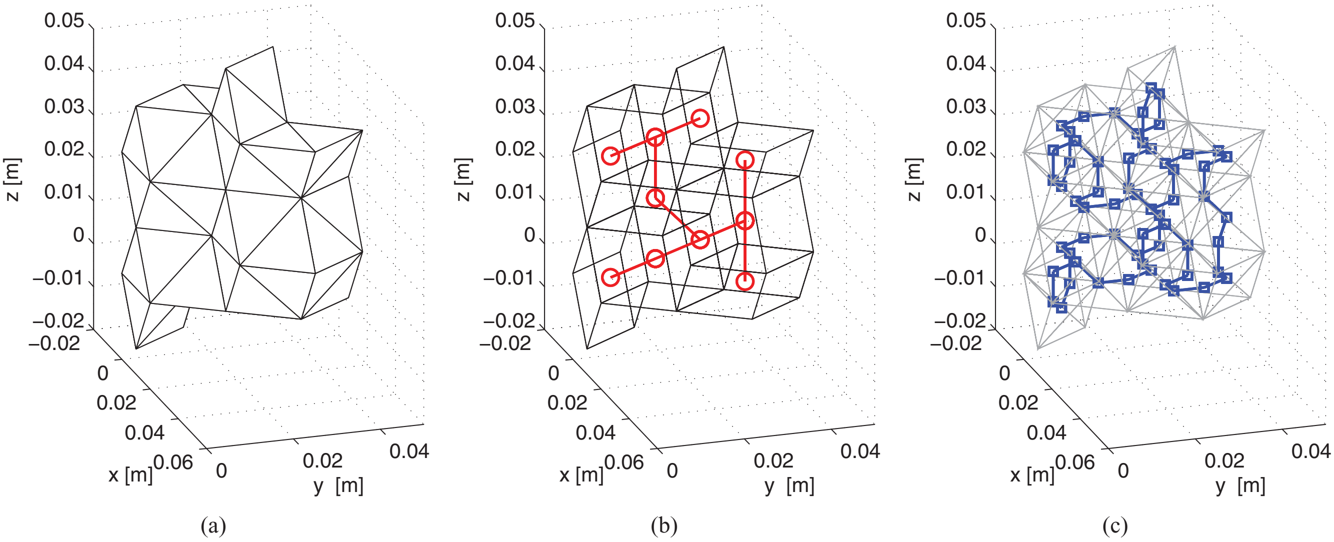


Fig. 3. Panel (a) shows a beam approximated by 60 tetrahedrons. Panel (b) shows the hexahedrons for each group of six tetrahedrons; lines and circular nodes indicate hexahedron spanning tree graph. Lines and square nodes in panel (c) show the Hamiltonian path graph defined by the hexahedron graph in (b).

Figure 3(a) shows an example of a beam that is composed of 60 tetrahedrons, and Fig. 3(b) shows groupings of six tetrahedrons into hexahedrons. The rhombic dodecahedron is well known as a zonohedron that can tile 3D space. It can be considered the Voronoi tessellation of a face-centered cubic lattice.[10]

Configurations are formed by manually folding the chain. As each module is folded to its neighbor, latches are pressed into place connecting the folded module to neighboring modules in the structure.

A routine written in MATLAB automatically voxelates the desired shape and generates the folding plan[51] using an algorithm by Griffith *et al.*[17,18] The algorithm constructs a Hamiltonian path (i.e., chain path) $E_C \subset E_T$ as a subset of the tetrahedron graph edge set E_T by first subdividing the volume into hexahedrons, then constructing a spanning tree of the hexahedral lattice, and finally wrapping E_C around the spanning tree. Although we are not aware of a proof that this method can generate all Hamiltonian paths for a shape voxelated by hexahedrons, we use foldings generated by this algorithm in our study of RATChET7mm structures.

Figure 3(b), shows a spanning tree for a hexahedron graph. This uniquely defines the Hamiltonian path shown in Fig. 3(c). The module configuration starts as a chain and each module folds in one of two directions according to the Hamiltonian path. The Hamiltonian path effectively defines the path the cable makes through the configuration. Figure 2(c) depicts an example folded configuration of 54 RATChET7mm modules.

3. Model

3.1. 6 DOF stiffness

We seek to model structures comprising modules interconnected by arbitrary mechanical links. From a strictly mathematical standpoint, the configuration of one rigid body in space relative to another is a 6D quantity consisting of both translation and rotation. Thus, to first order, any forces tying two rigid bodies together can be modeled by a linear function of these six configuration variables. Being linear, such a function can be represented by a matrix, referred to as the *stiffness matrix* that provides (generalized) forces, known as *wrenches*, as a function of (generalized) positions, known as *twists*.[35] A wrench is a generalized force that combines the three force components with the three torque components in a single 6D vector. Likewise, the twist is a 6D vector combining the three linear and three rotational displacement components. The inverse of the stiffness matrix expresses the twist as a function of applied wrench, and is known as the *compliance matrix*. For brevity, we will refer to rigid bodies connected by a linear stiffness as being connected by a *6 DOF spring*, which is a generalization of both linear and torsional springs. While a 6 DOF stiffness matrix can be composed from conventional linear and torsional springs, Huang and Schimmels[23] showed that they cannot model an arbitrary 6 DOF spring.

A 6 DOF spring can model arbitrary connection methods to first order. Fasse *et al.*[13,59] furthered the methods of Caccavale *et al.*[7,8] to model elastically coupled rigid bodies. The model presented in this section uses the quaternion-based potential function in Zhang *et al.*[59]

We use the notation of Fasse *et al.*[12] where p_b^a is the displacement of frame b relative to frame a written in the coordinates of a. Frames a and b are rigidly attached to bodies A and B, respectively. The columns of R_b^a are the basis vectors of frame b written in the coordinates of a. If no superscript is given, frame a is the global frame. The configuration of frame b relative to frame a is given by the homogeneous transformation matrix:

$$H_b^a = \begin{bmatrix} R_b^a & p_b^a \\ 0 & 1 \end{bmatrix}. \qquad (1)$$

The skew symmetric cross product matrix of p is written as $\tilde{p}$ and satisfies $\tilde{p}v = p \times v$.

http://dx.doi.org/10.1017/S0263574710000743

The 6 DOF spring defined by a 6 × 6 stiffness matrix K maps the change in configuration of frame b relative to frame a to a wrench that acts equally and oppositely on bodies B and A. An infinitesimal configuration change can be represented by an infinitesimal twist displacement $\delta T_b^a = [(\delta p_b^a)^{\mathrm{T}}\ (\delta\theta_b^a)^{\mathrm{T}}]^{\mathrm{T}}$ comprising translational (δp_b^a) and rotational ($\delta\theta_b^a$) 3D displacement vectors. For an infinitesimal twist displacement, the wrench w_b^a that body B applies to the elastic connection between A and B is given to the first order by

$$w_b^a \approx K\delta T_b^a \to \begin{bmatrix} f_b^a \\ \tau_b^a \end{bmatrix} \approx \begin{bmatrix} K_t & K_c \\ K_c^{\mathrm{T}} & K_o \end{bmatrix} \begin{bmatrix} \delta p_b^a \\ \delta\theta_b^a \end{bmatrix}, \qquad (2)$$

where K_t is the translational stiffness matrix, K_o is the rotational stiffness matrix and K_c is the coupling stiffness matrix.

To determine the wrench for small, finite displacements, Zhang and Fasse[59] define potential functions based on the relative translational and rotational displacements of the two bodies. For the rotational potential energy, they extend the work of Caccavale *et al.*[7] by developing a quaternion-based potential function. The relative orientation of frame b with respect to frame a is expressed by a quaternion $q_b^a = [\eta_b^a\ (e_b^a)^{\mathrm{T}}]^{\mathrm{T}}$ with scalar part η_b^a and vector part e_b^a. Using the principle of virtual work, they compute the wrench that body B applies to the elastic connection with respect to the global frame as

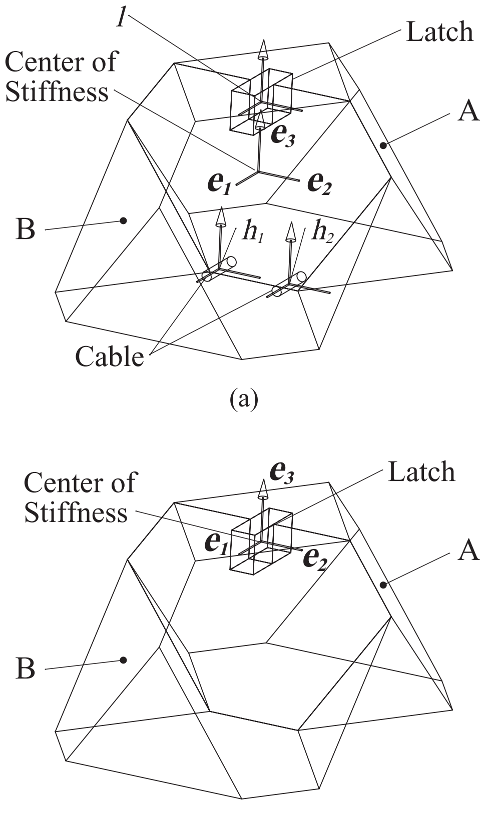
$$f_b = \frac{1}{2}R_a K_t p_b^a - \frac{1}{2}R_b K_t p_a^b + \eta_a^b(R_a + R_b)K_c e_b^a, \qquad (3a)$$

$$\begin{aligned} \tau_b = {} & \frac{1}{2}\tilde{p}_b R_a K_t p_b^a - \frac{1}{2}\tilde{p}_a R_b K_t p_a^b \\ & +2R_b\left(E_a^b\right)^{\mathrm{T}} K_o e_b^a \eta_a^b(\tilde{p}_b R_a + \tilde{p}_a R_b)K_c e_b^a \qquad (3b) \\ & +\frac{1}{2}R_b\left(\eta_a^b\left(E_a^b\right)^{\mathrm{T}} - e_b^a\left(e_b^a\right)^{\mathrm{T}}\right)K_c\left(I + R_a^b\right)p_b^a, \end{aligned}$$

where $E_a^b = \eta_a^b I - \tilde{e}_a^b$.

Fasse *et al.*[12] prove several useful properties of their potential function. It is sufficiently diverse and therefore can model arbitrary local stiffness. The potential functions are frame indifferent (i.e., equal and opposite wrenches are applied to the elastically connected bodies.) They are also port indifferent, meaning the choice of body A and B does not matter. The expression requires only algebraic operations and remains nonsingular for arbitrary displacements away from equilibrium. The method is valid for small displacements between the frames. In programmable matter and modular robot systems designed to be stiff in use, this assumption is valid as relative displacements between modules remain small.

In the case of the RATChET7mm system, each module is considered a rigid body that elastically connects to one or more neighboring modules. Using Eq. (3), we can compute the sum of forces and moments acting on each module due to its relative displacement from its neighbors. Given the stiffness matrix between each connected RATChET7mm module and external loading and fixed module boundary conditions, we can determine the static equilibrium of the entire structure.



(a)

(b)

Fig. 4. Model of two connected tetrahedron modules, A and B. Modules can either be connected by (a) a latch and a hinge or (b) a latch only. Latch l and hinge h_1, h_2 frames are labeled in (a).

We seek a means for determining the stiffness matrix K for the various links found in the systems we examine. Several approaches suggest themselves: (1) analytic derivation based on the the theory of elasticity; (2) experimental derivation through loading a link with a known wrench and measuring the resultant twist; and (3) use of approximations for rapid analysis. The following section presents the analytical method. Section 4.2 presents an approximation method and Section 5.3 presents an experimental method.

3.2. *Analytical method to determine K*

The stiffness matrix between two modules in a RATChET7mm configuration is a function of the mechanisms that connect them. There are three types of intermodule connections. Each pair of adjacent modules in a RATChET7mm configuration is connected by a plastic latch. A pair of chain neighbor modules is connected by a pair of cables running through them. When two modules are loaded such that they tend to collide, they are also connected by a mechanism that resists compression. In a RATChET7mm configuration, modules are connected in one of two ways (as shown in Fig. 4): (a) by a latch and a pair of cables or (b) by a latch only. These connections are defined by corresponding stiffness matrices K_{lh} and K_l, respecitively. When two modules collide, the stiffness matrix of their connection incorporates the collision stiffness as discussed in Section 3.3.

http://dx.doi.org/10.1017/S0263574710000743

For the RATChET7mm system, connections between modules are modeled as elastic beams.[3] The stiffness matrices for each connection type are derived by considering a fixed-free beam using the theory of elasticity. Appendix 7 presents the analytical models for the (1) latch stiffness K_l, (2) cable (hinge) stiffness K_h, and (3) face collision stiffness K_f. Analytical models provide a parameterization of the stiffness of the connection method. Section 5 validates the model experimentally. For complex connection methods, FEA[60] or experiment can determine the stiffness matrix.

The stiffness matrix K between two modules must be defined with respect to a unique coordinate system. For a pair of elastically coupled bodies, the *center of stiffness* is a unique point which maximally decouples K.[12] K_t and K_o are symmetric and thus have principal stiffness axes. A displacement along a principal translational stiffness axis applies a pure force to the elastically coupled bodies. Likewise a displacement about a principal rotational stiffness axis causes a pure moment. There may not exist a point between elastically coupled bodies where the coupling K_c vanishes. However, Lončarić[29] showed that the center of stiffness is located at the unique point that makes the coupling stiffness K_c symmetric provided $\mathrm{tr}(K_t)$ is not an eigenvalue of K_t.

The location of the center of stiffness between neighboring RATChET7mm modules depends on whether they connect with a latch only or with a latch and a pair of cables. Adjacent modules that are not chain neighbors connect with a latch only and have stiffness matrix K_l and center of stiffness at the centroid of the latch as shown in Fig. 4(b).

The stiffness between modules connected by a latch and a hinge K_{lh} is given by the sum of the stiffnesses of each beam shown in Fig. 4(a). In order to sum the stiffnesses, they are transformed[12] to the center of stiffness frame a using

$$\begin{aligned} K_{lh} = {} & Ad^{\mathrm{T}}_{H_a^l} K_l Ad_{H_a^l} + Ad^{\mathrm{T}}_{H_a^{h1}} K_{h1} Ad_{H_a^{h1}} \\ & + Ad^{\mathrm{T}}_{H_a^{h2}} K_{h2} Ad_{H_a^{h2}}, \end{aligned} \tag{4}$$

where the superscript in each transformation H of each adjoint Ad represents the center of stiffness frame of the corresponding beam. The center of stiffness between modules connected by a latch and a hinge is the position of the frame a that makes the coupling stiffness of K_{lh} symmetric.[60]

3.3. Collision

The nonlinear equation solver used in the simulation requires the function it solves to be continuous. We approximate the transition in stiffness components from collision-free value to collision value using a continuous analytical function that smoothly transitions between stiffnesses over a small distance. Using a sigmoid function of the form

$$y(x) = \frac{1}{\mathrm{e}^{m(d-x)} + 1}, \tag{5}$$

the parameters m and d can be tuned to capture the effect of a gap between the modules. Figure 5 plots the sigmoid curve used in the RATChET7mm model. The parameter x is defined by the penetration depth. The coefficient d shifts the curve along x to define the gap length, and m defines the slope of the curve as it passes through $y(x) = 0.5$. The values used to tune the simulation to experiment are $d = 0.5$ mm and $m = 8000$. Notice that the sigmoid in Fig. 5 completely switches to collision stiffness after the penetration depth x reaches 1 mm which is consistent with the approximate gap between two modules. In the simulator, when a collision is detected between two modules, the face collision stiffness K_f (defined in Appendix 7) is added to the nominal collision stiffness (K_l or K_{lh}) using the sigmoid function

$$K_{lwc} = K_l + K_f y(x), \tag{6a}$$

$$K_{lhwc} = K_{lh} + K_f y(x), \tag{6b}$$

where K_{lwc} and K_{lhwc} are the *with collision (wc)* modified stiffness matrices for latch only and latch and hinge connection types, respectively.

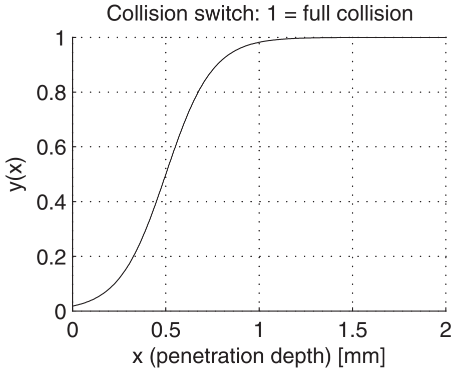


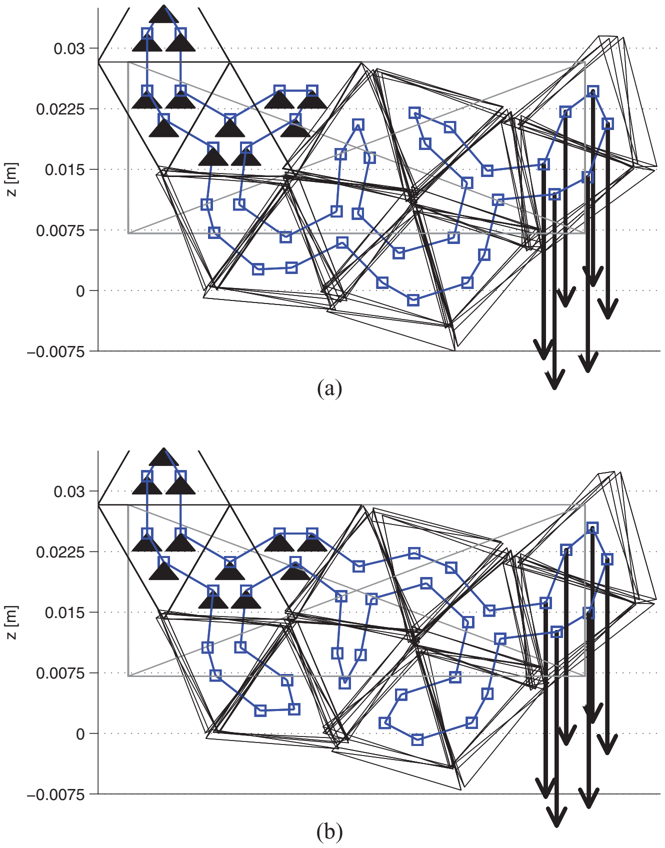
Fig. 5. Collision sigmoid switch function.

4. Simulation

Using the stiffness model described in the previous section, a mechanical simulator for a modular lattice structure should be able to determine its behavior given specified boundary conditions. This simulator, written in MATLAB, finds the equilibrium position of a statically loaded modular robot or programmable matter structure loaded with arbitrary constant wrenches. The rigid bodies can be either modules or components of modules.

The model of a modular system is represented as a graph with modules or module components as nodes. Each edge of the graph links physically connected modules and has an associated stiffness matrix defined by the type of bond between the modules. For example, for each pair of adjacent tetrahedron modules defined by edge $e_{T_i} = \{x_i, x_j\}$ ($e_{T_i} \in E_T$), the stiffness matrix between x_i and x_j is K_{lh} if $e_{T_i} \in E_C$ (i.e., if x_i and x_j are chain neighbors), otherwise the stiffness is K_l.

Our algorithm, outlined in Algorithm 1, uses MATLAB's nonlinear equation solver `fsolve` to determine the equilibrium position and orientation of each tetrahedron given loading and fixed boundary conditions. We must use a nonlinear solver due to the finite rotations found in Eq. (3). Each iteration of the solver computes the force and moment equilibrium for each module according to Eq. (3). For each fixed module, the six equilibrium expressions are replaced



(a)

(b)

Fig. 6. Deformation for two types of Hamiltonian paths in a beam (indicated by wireframe rectangle) approximated with 42 tetrahedrons. Black triangles indicate fixed modules; arrows indicate applied load. In (a) the cable passes through the lower portion of the structure under compression and allows 8.6% larger deflection at the free end than (b) where the cable passes through the portion under tension.

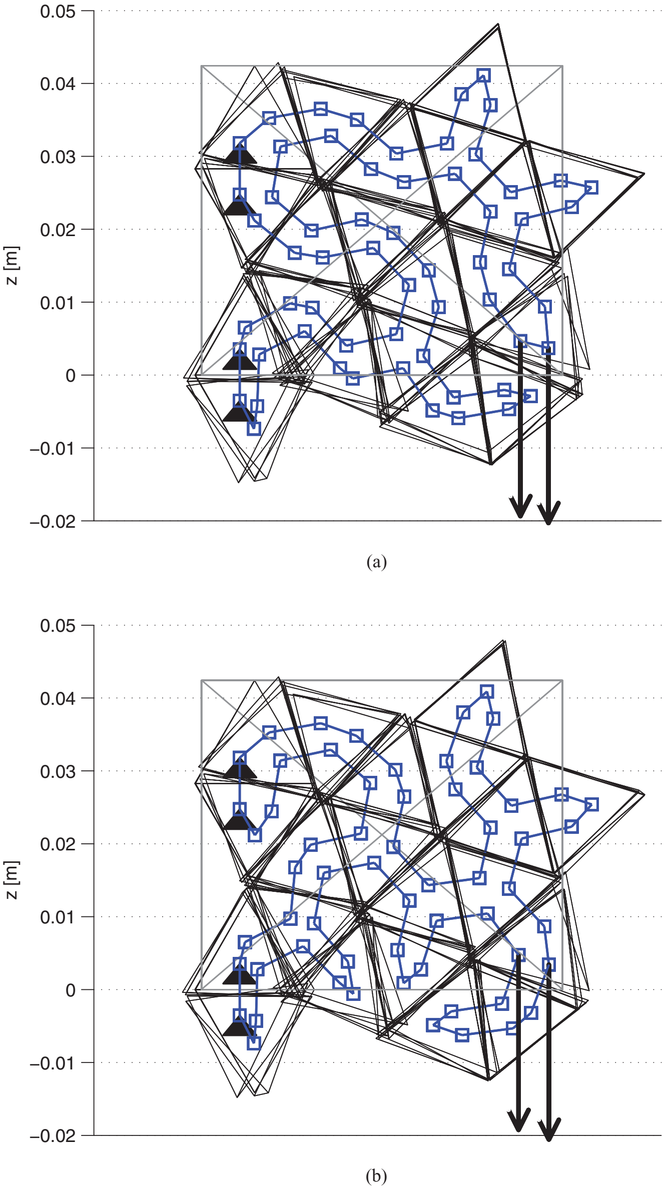
by six constraint equations on the module's position and orientation.

Like all gradient search algorithms, `fsolve` is sensitive to the initial guess. To minimize residual forces and moments, our simulator uses "load stepping" in cases of large loads – it first solves for small loads, and uses the solutions as initial guesses for larger loads. The simulator checks the residual, and, if it is too large, the simulator increases the number load steps by some factor. This method reduces the residual for cases with large displacements or a large number of modules.

The simulator detects collisions using a triangle-to-triangle intersection routine.[31] The sigmoid function described in Section 3.3 provides a continuous switch function to approximate the change in stiffness matrices with and without collision. Section 5 demonstrates how the parameters of the sigmoid function can be tuned to characterize the transition in stiffness due to collision.

4.1. RATChET7mm examples

Figures 6(a) and (b) show two chain paths (each indicated by Hamiltonian cycle with square markers at module centroids) through a beam like structure. Each frame shows the side view of a beam of 42 tetrahedron modules which appear as triangles in this view. Black triangles indicate fixed modules and arrows indicate the 120N applied load distributed over six modules.



(a)

(b)

Fig. 7. Side view of configuration in Fig. 3(a) showing deformation for two types of Hamiltonian paths. Black triangles indicated fixed modules; arrows indicate applied load.

Intuitively, Fig. 6(a) will have larger displacement since the less stiff bonds are under tension while Fig. 6(b) has a configuration where the cables, which are stiff in tension, will absorb most of the load. Under the loading conditions shown in Fig. 6, the method finds displacements of 11 mm and 10 mm respectively.

We can consider the "apparent stiffness" of the full structure – the stiffness relationship experienced in the direction of loading. Consider a homogeneous rectangular cantilever beam of length L, area moment of inertia I and Young's modulus E under a transverse load at the free end. Such a beam, which may undergo large deflections, can be analyzed using the Pseudo-Rigid-Body (PRB) Model.[22] This model approximates the beam as a rigid link pinned to ground with a torsion spring. The torsion spring models the beam's

http://dx.doi.org/10.1017/S0263574710000743

Algorithm 1 Static Equilibrium Solver

```
Given:
    ■ Connectivity graph of $N$ elements
    ■ Stiffness matrix between each connected pair of elements
    ■ Set of $Fixed$ elements
    ■ Applied wrench for each element in each column of the $6 \times N$ matrix $W_{applied}$
    ■ Initial twist for each element in each column of the $6 \times N$ matrix $T_0$
for $i = 1$ to $NumberOfLoadStepTries$ do
    for $j = 1$ to $NumberOfLoadSteps$ do
        $W_{loadstep} \leftarrow (j/NumberOfLoadSteps) \times W_{applied}$ {Iteratively increase load}
        Use nonlinear equation solver (e.g., MATLAB's fsolve) to determine equilibrium twist of each element (in each
        column of $6 \times N$ matrix $T$) given $Fixed$, $T_0$, and $W_{loadstep}$
        $T_0 \leftarrow T$ {Set solution as initial guess for next load step}
    end for
    if $residual > MaxError$ then
        $NumberOfLoadSteps \leftarrow NumberOfLoadSteps \times Factor$
    else {Valid solution found}
        break
    end if
end for
Report equilibrium twist of all modules $T$
```

stiffness in bending and has equivalent spring constant (or apparent stiffness) given by

$$K = \gamma K_{\Theta} \frac{EI}{L}, \qquad (7)$$

where γ and K_{Θ} are PRB model parameters that remain constant for constant load direction. Therefore, a difference in apparent stiffness of two configurations with the same shape (thus same L and I) and same loading condition (thus same γ and K_{Θ}) is due to a change in the apparent Young's modulus. In the two cases in Fig. 6, we get a Young's modulus of 75 MPa and 82 MPa, an 8.6% difference.

Figure 3(a) shows an example configuration of 60 RATChET7mm modules. From Kirchhoff's matrix tree theorem,[20] there are 135 possible spanning trees and hence 135 Hamiltonian paths that can be generated using Griffith's expansion of spanning trees technique.[17] Exhaustively testing all of these 135 possibilities shows there can be a variance in the apparent stiffness of the material if approximated as a homogeneous material. The path shown in Fig. 7(a) has a displacement of 12.9 mm which is the least displacement of any for the loading shown, and Fig. 7(b) has displacement of 13.0 mm which is the largest displacement. The apparent Young's modulus of the two paths would be 4.521 MPa and 4.484 MPa, a 0.8% change.

Thus, the examples in Figs. 6 and 7 demonstrate the ability to program the apparent Young's modulus of a programmable matter system with two types of bonding stiffness (K_l and K_{lh}). Note a system with a greater disparity in the stiffnesses of its two types of bonds would exhibit a greater range of programmable stiffnesses.

The simulator allows for analysis of structures with a large number of modules. Figure 8 depicts a beam of 120 modules under a transverse load of 20N. Note this configuration has more than billions of possible spanning trees. Using heuristics to determine a critical region of the structure and to optimize folding within that region is one possible method for handling the large search space.

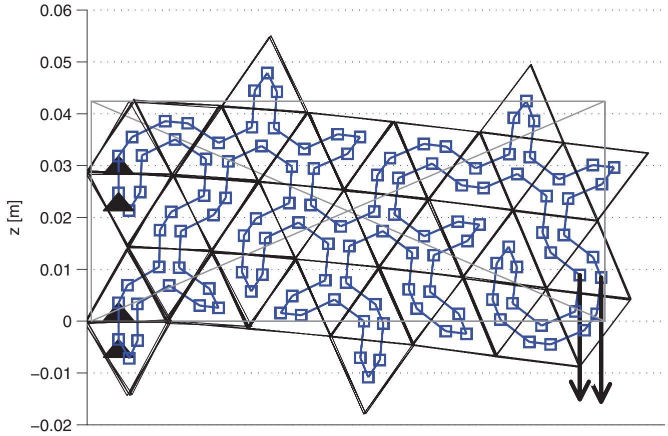


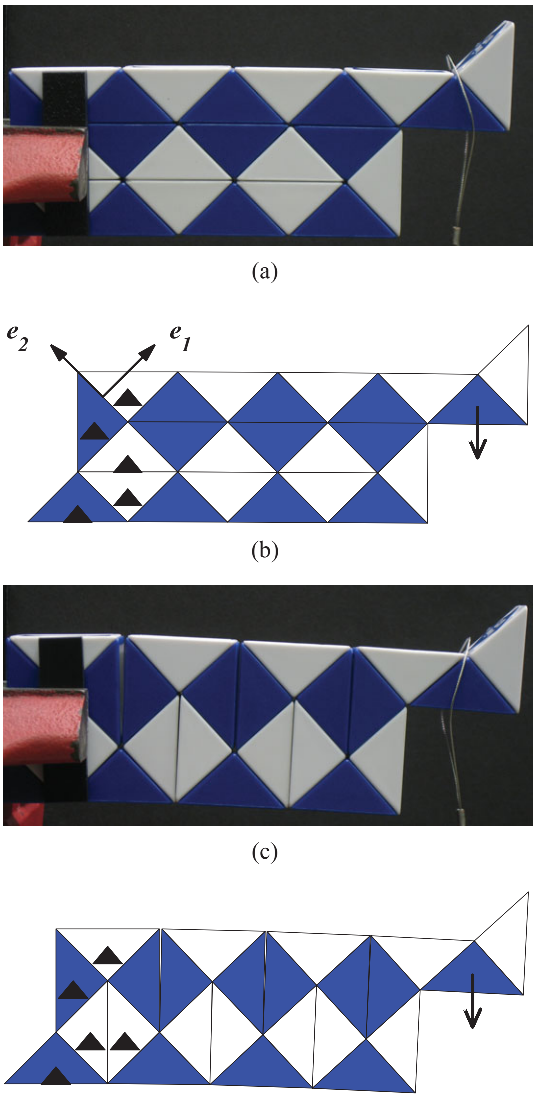
Fig. 8. Beam of 120 tetrahedrons with fixed modules (marked with black triangles), load applied to two tetrahedrons and Hamiltonian path shown.

4.2. Rubik's snake example

The simulator allows quick formulation and simulation of many different forms of programmable matter. To demonstrate, we model planar configurations of the Rubik's Snake.

The Rubik's Snake is a widely available toy with a chain of polyhedron that can be twisted into a variety of shapes that has served as an inspiration for a variety of programmable matter. The polyhedron are right angle isosceles triangular prism modules. When viewed as in Fig. 9(a), we denote the three faces of a module perpendicular to the page as the two *leg* faces and the one *hypotenuse* face. There are three module-to-module connection mechanisms: spring

http://dx.doi.org/10.1017/S0263574710000743



(a)

(b)

(c)

(d)

Fig. 9. Experimental and simulated deformations of two Rubik's Snake configurations each with 0.5N load at free end.

bolts, face protrusions, and face to face collision. Figure 9(a) shows a configuration of 24 Rubik's Snake modules. Each leg face of the module connects to a neighboring module with a spring bolt which passes through the center of the face. There is no crosslinking between hypotenuse faces. Thus, for some configurations under certain loadings (e.g., Fig. 9(c)) the hypotenuse faces along top of beam tend to separate as the configuration elastically deforms. All faces resist collision and have small interlocking protrusions and pockets which add rigidity in shear.

The center of stiffness is located in the geometric center of each face. The center of stiffness frame (such as the example frame shown in Fig. 9(b)) is oriented such that $\mathbf{e}_1$ is directed along the outward face normal of the module and $\mathbf{e}_3$ is directed out of the page. To model the system, we need to determine the leg face stiffness K_{leg}, hypotenuse face stiffness K_{hyp}, and the face collision stiffness K_{f}.

In order to quickly model displacements under load of the Rubik's Snake, we define the stiffness components using rough order of magnitude approximations. Module bonds are stiffest when modules are compressed together in collision. Thus, we approximate the face normal stiffness $K_{ft_{11}}$ to the order of magnitude of a beam in uniaxial loading with characteristic module geometry and material properties. This stiffness given by AE/L (assuming Young's modulus $E = 2 \times 10^9$ Pa, area $A = 400 \times 10^{-6}$ m^2, and length $L = 20 \times 10^{-3}$m) has order of magnitude 10^7 N/m. We observe that 10 N acting at characteristic length $L = 20 \times 10^{-3}$m away from face (giving 0.2 Nm torque) rotates the module approximately 1° (0.017 rad) and thus we model all rotational stiffness components as 10 Nm.

The collision stiffness matrix is approximated by

$$K_{ft} = \begin{bmatrix} 10^7 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{N/m}, \tag{8}$$

$$K_{fo} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \text{Nm}. \tag{9}$$

We observe the spring bolt to be significantly less rigid than the face collision stiffness, and we therefore approximate its stiffness $K_{legt_{11}}$ as a factor of 100 less than $K_{ft_{11}}$. Stiffness along the leg face ($K_{legt_{22}}$ and $K_{legt_{33}}$) appears to be stiffer than the spring bolt yet more compliant than face collision stiffness, and we approximate it as a factor of 10 less than $K_{ft_{11}}$. The stiffness matrix between the leg faces of Rubik's Snake modules ignoring collision is approximated by

$$K_{legt} = \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{bmatrix} \text{N/m}, \tag{10}$$

$$K_{lego} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \text{Nm}. \tag{11}$$

When two Rubik's Snake modules collide in loading, the collision stiffness is added to the nominal stiffness. Note that hypotenuse faces only connect when they are in collision (i.e., compression.) Thus, in collision the hypotenuse face stiffness $K_{\text{hyp}} = K_{\text{f}}$.

Figure 9 shows experimental (a) and (c) and simulated (b) and (d) deformations of two types of Rubik's Snake configurations. A vice holds the five modules shown with black triangles in Figs. 9(b) and (d) fixed. A wire rope loop supporting a mass applies a load to the configuration. In each case, the Rubik's Snake configuration has the same shape, fixed module (indicated by black triangle) boundary conditions, and 0.5 N load applied to the free end.

Note the rigidity of the configuration greatly depends on the folding pattern. In the rigid case shown in Figs. 9(a) and (b), the modules are folded into a pattern such that there is a direct path along the top of the configuration from the load to the fixed boundary. In this configuration, the spring bolts of the modules near the top are loaded in tension, and the faces of the modules near the bottom are in compression.

In contrast, the less rigid case shown in Figs. 9(c) and (d) uses a fold pattern with a circuitous path from load to fixed boundary. Vertical gaps form between modules near the top due to the lack of a crosslinking connection between the hypotenuse faces which supports tension.

Figures 9(b) and (d) demonstrate our method's ability to characterize the relative stiffnesses of two configurations using a simple inter-module connection stiffness estimation. The simulation reports deflections of 0.7 mm and 3.4 mm for the two case Figs. 9(b) and (d), respectively which would correspond to Young's modulus of homogeneous materials of 3.2 MPa and 0.7 MPa. The simulation captures the heterogeneous and anisotropic nature of the connection methods. In the early design stage or within a large search routine, such quick analysis could be used to quickly cull poor configurations to allow focus on more promising ones.

5. Experiments

We experimentally validate the results of our simulations of all three modular structures. Keeping in mind that our goal is to develop a mechanical simulation that converges rapidly enough to allow it to be used for optimizing designs by evaluating many thousands of structures, we aimed for error on the order of 10%. Error results reported here are valid within the range of the applied loads. As programmable matter and modular robot systems are typically stiff, displacements will generally be in the linear regime.

5.1. *Rubik's snake*

In the case of the Rubik's Snake, a quick check on the validity of the simulation is the estimated equivalent Young's modulus, E, of homogeneous material. Figures 9(a) and (c) show the deflections (0.8 mm and 2.9 mm, respectively) of the physical system under the same load and we can estimate E to be 2.7 MPa and 0.8 MPa, respectively, which agrees with the simulated values with approximately 15% error. This error seems reasonable as the stiffness components are quickly approximated with a rough order of magnitude. While FEA can provide a solution with higher accuracy, the simulator allows quick model development and comparison of apparent stiffness of different configurations.

5.2. *RATChET7mm*

We explore the stiffness of the RATChET7mm more deeply. Experimental measurements and analytical models of the rotational stiffnesses about the $\mathbf{e}_1$ and $\mathbf{e}_3$ axes of the center of stiffness frames (Fig. 4) are in agreement. We proceed to show that the simulator accurately models a beam configuration of modules.

The first experiment verifies the rotational stiffness of a module loaded with a moment about the $\mathbf{e}_1$ axis. Figure 10 depicts the experimental setup with two modules. One module is fixed and the other is loaded with a moment by placing a mass at the end of a moment arm. We measure the stiffness about $\mathbf{e}_1$ for each connection type: latch and cable (Fig. 10(a)) and latch only (Fig. 10(b)). An image processing routine written in MATLAB determines the angular displacement. Figure 11 shows good agreement between analytical and experimental stiffness terms in both cases.

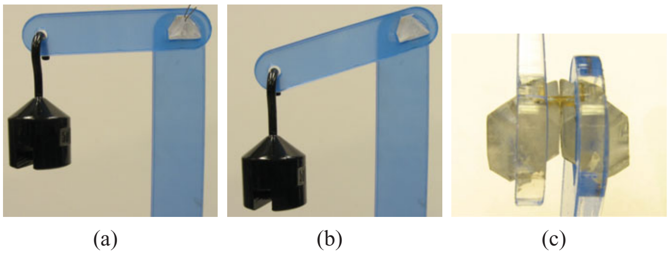


Fig. 10. Experimental setup for measuring angular displacement of connected pair of RATChET7mm modules loaded by a moment about $\mathbf{e}_1$. Each case shows displacement with one module loaded with 50 g on a 50 mm moment arm. The loaded module connected to fixed module with (a) latch and cable is noticeably stiffer than with a (b) latch only. Panel (c) shows a side view of unloaded pair of modules connected by a latch only.

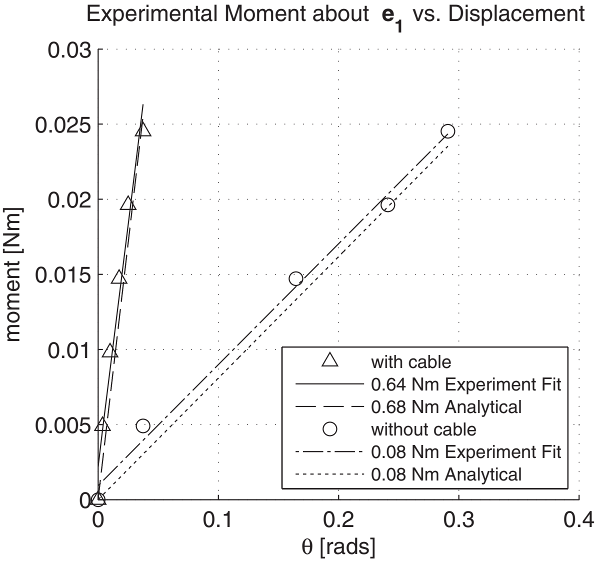
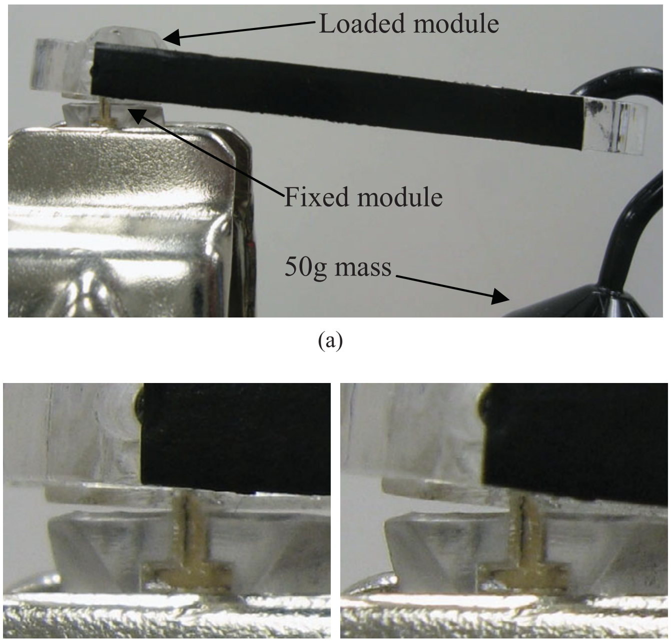


Fig. 11. Experimental data and linear fits for an $\mathbf{e}_1$ moment applied to one module connected to another. Data show a module pair connected with a cable and latch is stiffer than one without a cable by an order of magnitude.

Note the cable (Fig. 10(a)) increases the stiffness of a latch only connection (Fig. 10(b)) by an order of magnitude. The addition of the cable to the latch connection method moves the center of stiffness from the latch centroid towards the center of the module face as shown in Fig. 4. The stiffness about $\mathbf{e}_1$ increases in part due to the existence of a moment arm together with the $K_{lt_{22}}$ term of the latch translational stiffness.

The $\mathbf{e}_3$ rotational stiffness experiment verifies the $K_{lo_{33}}$ stiffness term and demonstrates the simulator's ability to capture the nonlinear effect of stiffness change due to collision. Figure 12(a) depicts the experiment consisting of a fixed module, a loaded module, and an extension arm used to apply moments about $\mathbf{e}_3$. Figure 12(b) shows a close up of the gap between the connected faces of the modules without load, and Fig. 12(c) shows the modules in collision when a 30 Nmm moment is applied to the loaded module. For moments that do not cause collision, the rotational stiffness (0.11 Nm) is due to the bending of the latch. As the loaded

http://dx.doi.org/10.1017/S0263574710000743



(a)

(b) (c)

Fig. 12. (a) Experimental setup to measure $K_{lo_{33}}$ stiffness term. In (b), the loaded module has no load and there is a visible gap between connected faces of the modules. When a 30 Nmm moment is applied to the loaded module (c), the modules collide.

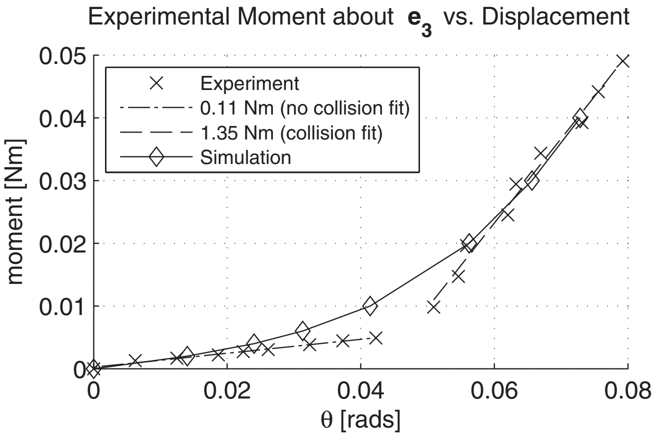
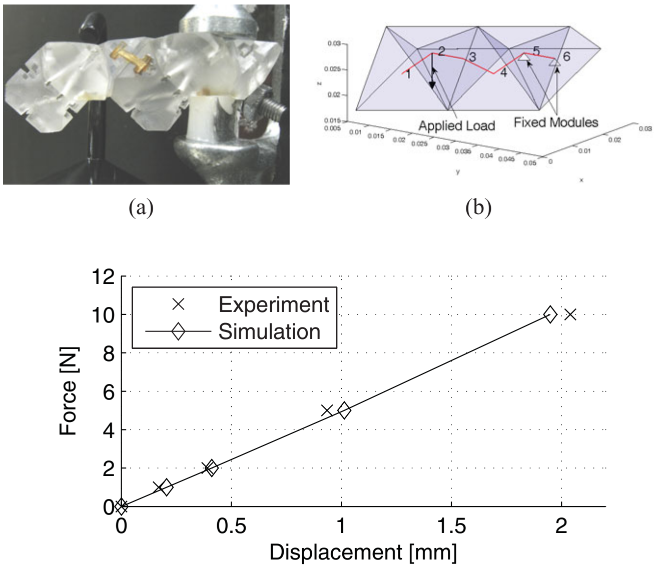


Fig. 13. Experimental data and linear fits for an $\mathbf{e}_3$ moment applied to one module connected to another. When angular displacement (θ) reaches 0.05 rad (2.9°), the face of the loaded module collides with the fixed module and the stiffness increases by an order of magnitude. Simulation curve plots angular displacement of module with tuned collision sigmoid.

module begins to collide with the fixed module, the stiffness increases to 1.35 Nm which includes the compression of the SLA polycarbonate-like material.

Figure 13 shows experimental and simulated angular displacement about the $\mathbf{e}_3$ axis for a range of applied moments about $\mathbf{e}_3$. The small gap between latched modules (shown in Fig. 12(b)) allows for some rotation without collision until θ reaches 0.05 rad (2.9°). Note that the simulation curve matches experimental data well where modules are clearly free of collision or in collision. As described in Section 3.3, we tune the sigmoid parameters such that the stiffness transitions properly from without collision to with collision. The continuous approximation of stiffness over-predicts in the region near collision. For example, at $\theta = 0.041$ rad, the approximated stiffness is four times the experimental collision free stiffness. Note that this overprediction does not exceed the collision stiffness and the continuous approximation greatly increases the stability of the simulator.



(a) (b)

(c)

Fig. 14. Beam configuration of six RATChET7mm modules under transverse loading (a) and simulation model view (b). (c) Experimental and simulated displacement of module 1 in (b).

5.2.1. *Beam simulation.*

To demonstrate the accuracy of the model, we applied a transverse load to a beam of six RATChET7mm modules. We considered two beam orientations in order apply different loading conditions at each module to module connection.

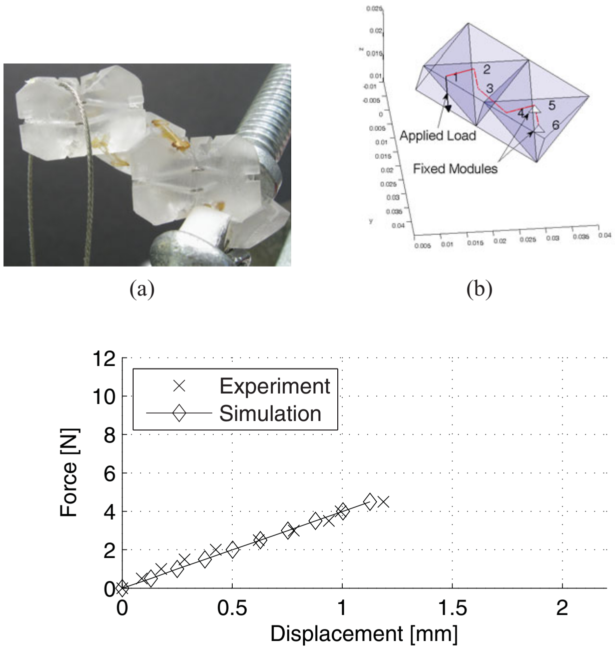
Figure 14(a) shows the experimental setup for the first loading case. The simulation view shown in Fig. 14(b) depicts the modeled boundary conditions. Two modules (5 and 6) on the right end of the beam are held fixed by a clamp. We use a set of masses to apply increasing transverse loads to module 2.

An image processing script written in MATLAB determines the displacement of module 1 at the free end. Figure 14(c) plots the vertical displacement of module 1 for experimental and simulated data. The simulation data agrees well with experimental data; the mean absolute error is 9%.

Using the same six module beam configuration, we measured its deflection under load in a second orientation as shown in Fig. 15(a). In this experiment, module 1, which is at the free end of the beam, is loaded in the direction shown in Fig. 15(b). We load module 1 by hanging increasing amounts of mass from the steel rope loop shown in Fig. 15.

Figure 15(c) compares simulation and experimental displacement of module 1. Again the simulation matches the behavior of the observed data; the mean absolute error is 13%.

http://dx.doi.org/10.1017/S0263574710000743



(a) (b)

(c)

Fig. 15. Second orientation of beam configuration of six RATChET7mm modules under transverse loading (a) and simulation model view (b). (c) Experimental and simulated displacement of module 1 in (b)

5.3. *Experimental method to determine K*

This section describes a methodology for experimentally determining stiffness components. As an example, we consider the CKBot[37,40] chain style modular robot. The modeling and experimentation method can be extended to other lattice, chain, or hybrid modular robot systems.

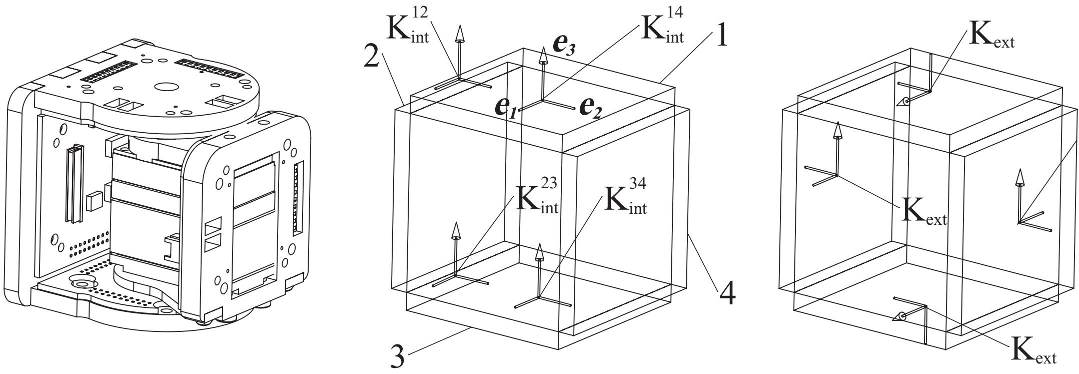
The first modeling step is to determine how to lump the compliance. In Section 3.2, we assumed RATChET7mm modules are significantly more rigid than the links. CKBot modules, however, are mated with machine screws and the observed rigidity of the module and links is of the same order. To account for internal module compliance, we model each module as four elastically connected faces. Figure 16(b) depicts the locations of the four *internal* stiffness matrices K_{int}^{ij} linking faces i and j of the same module. Figure 16(c) shows the four *external* stiffness matrices K_{ext} linking faces of neighboring modules. To minimize the model parameters, we assume due to symmetry that (1) $K_{\text{int}}^{12} = K_{\text{int}}^{23}$ and (2) all K_{ext} are the same. Also, we assume diagonal stiffness matrices yielding a total of 24 model parameters. Thus, a configuration of CKBot modules can be modeled knowing the four stiffness matrices: K_{int}^{23}, K_{int}^{34}, K_{int}^{14}, and K_{ext}.

Determining the stiffness components requires subjecting CKBot modules to different loads and measuring the displacements. We use a motion capture system (Vicon[1]) as it is a commonly available tool to roboticists and easily provides translational and rotational displacement data. However, the resolution of the system is insufficient to directly measure translational stiffness components.

Therefore, to determine the stiffness components we load a chain of six CKBot modules with a measurement frame at the end module that undergoes sufficient displacement. Figure 17(a) shows an example configuration with a mass at one of the loading points of the load mount. The load mount supports extensions with Vicon markers used to define the measurement frame $\mathbf{e}_i$. We load the six CKBot chain in several different joint angle configurations and partition the data into a fitting set and a validation set.

We observed that the two twist components Δp_1 and $\Delta\theta_3$ of the frame in Fig. 17(a) exhibit sufficiently high signal to noise ratio to accurately measure. We therefore use MATLAB's `lsqcurvefit` tool to determine the stiffness components that best matches the simulated Δp_1 and $\Delta\theta_3$ displacement of the load mount frame to the data. As the convergence time of the fitting routine is sensitive to initial guess, we set initial translational stiffness values to 10^6 N/m and rotational stiffness values to 500 Nm. These values are based on analysis in the translation case and preliminary experiments in the rotation case.

Table I reports the stiffness components determined by the fitting routine. Note that the translation stiffness components ($K_{11} - K_{33}$) for K_{int}^{12} and K_{int}^{23} are all approximately 2×10^7. This is consistent with the fact that edges of the CKBot for K_{int}^{12} and K_{int}^{23} are epoxied tightly together. Likewise, the translation components of K_{ext}, the stiffness between faces fastened with machine screws, has a relatively high value of 6×10^7 to 8×10^7. The translation components of stiffness matrices K_{int}^{34} and K_{int}^{14} have generally lower values because



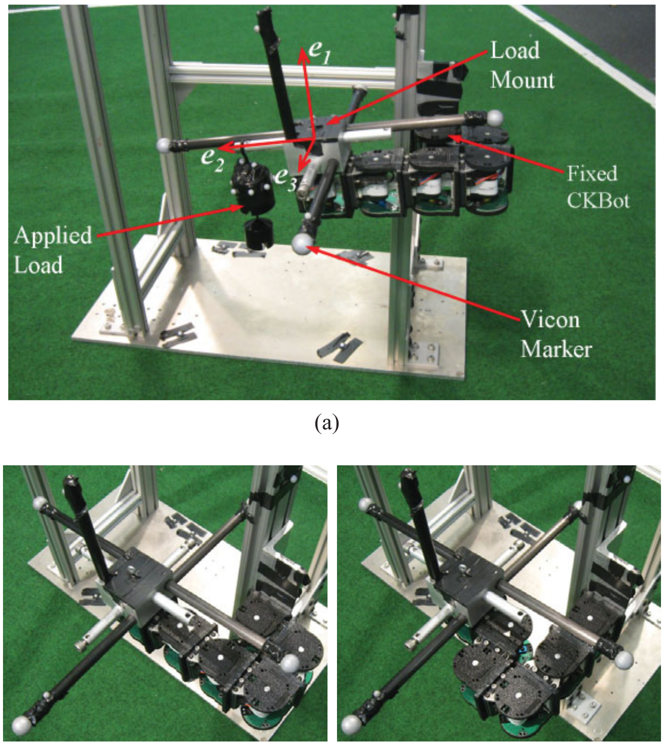
(a) (b) (c)

Fig. 16. (a) CKBot module. Four face model with (b) faces and internal stiffness matrix frames labeled and (c) external stiffness frames labeled. Superscripts denote pair of faces linked at given stiffness frame.

http://dx.doi.org/10.1017/S0263574710000743

Table I. CKBot stiffness components of frames defined in Figs. 16(b) and (c).

	$K_{\text{int}}^{12}/K_{\text{int}}^{23}$	K_{int}^{34}	K_{int}^{14}	K_{ext}	units
K_{11}	1.6E+7	4.9E+7	3.2E+4	6.1E+7	
K_{22}	2.4E+7	2.7E+5	3.1E+6	8.2E+7	N/m
K_{33}	2.1E+7	1.0E+4	5.5E+7	6.6E+7	
K_{44}	4.0E+1	1.5E+3	2.9E+4	4.3E+2	
K_{55}	6.4E+4	1.7E+1	1.1E+2	3.5E+3	Nm
K_{66}	4.4E+1	9.7E+1	3.2E+4	3.8E+4	



(a)

(b) (c)

Fig. 17. (a) shows experiment apparatus with markers for Vicon motion capture system. Two different CKBot configurations (b) and (c) (denoted C_3 and C_4, respectively) that place load mount at same position and orientation.

they represent the stiffness between the joints connecting face 4 to faces 1 and 3 (Fig. 16(b)).

Figure 18 compares experimental data with simulation results for the fitting and validation data sets. Values in parentheses report the mean absolute value of the relative error. Figures 18(a) and (b) present four typical loading conditions used to determine the 24 stiffness parameters. It shows that the fitting routine determines stiffness components that fit the simulation to the data with less than 14% error on average.

To validate the simulator predictions of Δp_1 and $\Delta\theta_3$ displacements in other joint angle configurations, we load the configurations shown in Figs. 17(b) (C_3) and 17(c) (C_4) and compare measured displacement with simulation results. Figures 18(c) and (d) report the comparison. Note that with the exception of the C_4, τ_3 case, the simulator accurately predicts the displacement with 12% error or less on average. It is important to note that though the accuracy of the simulator is approximately 15%, it correctly predicts the slope and relative magnitude of displacement of the two configurations.

One application for the simulator is to determine the modular robot configuration that displaces least while performing a desired task. As an example, consider the two CKBot configurations C_3 and C_4 shown in Figs. 17(b) and (c). Both achieve the same load mount position and orientation. However, given an applied torque load about $\mathbf{e}_3$ (τ_3) or $\mathbf{e}_1$ (τ_1) the model predicts (Fig. 18(c)) that configuration C_4 displaces (Δp_1) less in translation relative to configuration C_3. And under the same loads, Fig. 18(d) shows that configuration C_3 rotates less relative to configuration C_4. Thus, a chain style modular robot can exhibit programmable stiffness by specifying the chain path.

6. Discussion

The primary goal for this work is to develop a fast and precise method for determining the impact of varying both bond stiffness characteristics and the bond and module arrangement in a modular robot or programmable matter system. We discuss our method in terms of its *programmability, expressibility, computational cost*, and *modeling error*.

6.1. Programmability

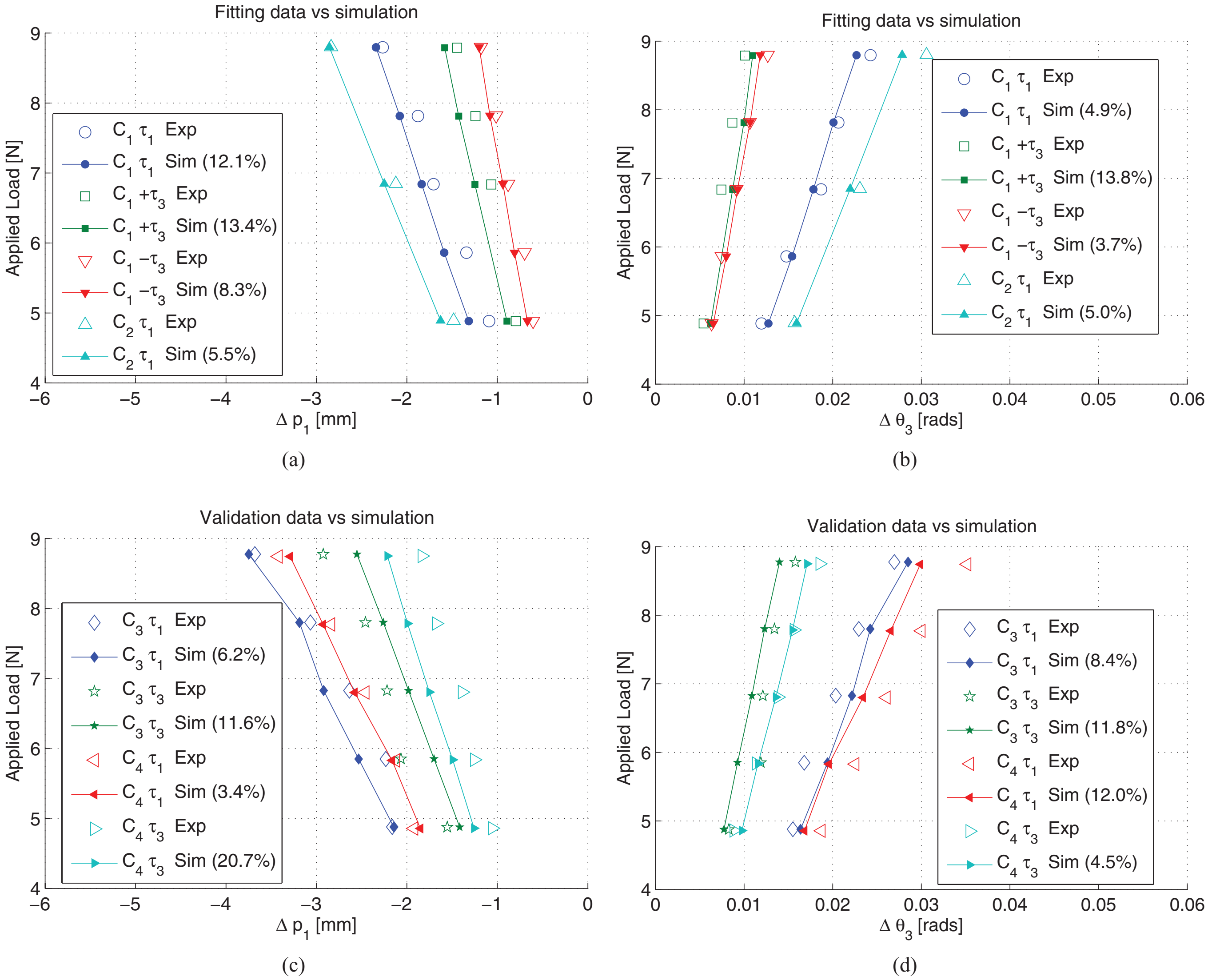
Programmability is the degree to which a method enables one to design desired properties into a system being constructed. Simulations of RATChET7mm show how the choice of Hamiltonian path influences the stiffness of the configuration. In particular, placing the cables in areas of high tension increases the apparent stiffness. For applications where maximum stiffness under a specified load is a priority, the path of the cable can be optimized to minimize the deflection of the configuration.

Note that using two types of connection methods provides the ability to tune the apparent Young's modulus. The placement of relatively compliant and rigid bonds throughout the configuration controls the stiffness at the point of loading. Though the number of modules in a configuration is finite, the number of possible configurations is very large as it is exponential in the number of modules. This allows for a broad range of achievable material properties from a system with only two inherent stiffnesses.

In the CKBot case, the choice of path of a chain of modules (Figs. 17(b) and (c)) results in different stiffness characteristics for the same end-point location and orientation. Our simulator allows us to anticipate the tunable apparent stiffness of the modular robot.

6.2. Expressibility

Expressibility is the degree to which a given model can be adapted to represent, or express, a large variety of systems. Our simulator can model modular robot and programmable matter systems where edges of the module or module component connectivity graph are assigned 6×6 stiffness matrices. This paper shows it applies with equal generality to a Rubik Snake's triangular prisms linked only by the chain backbone, to faces of CKBot modules linked by either permanent intra-module structures or module-to-module



(a)

(b)

(c)

(d)

Fig. 18. Comparison of experiment (Exp, lines with full markers) data with simulation (Sim, empty markers). Values in parentheses report mean absolute value of relative error. Legend entries indicate CKBot joint configuration C_i and applied load τ_i. y axis indicates weight applied to achieve τ_i. Figures show simulator's ability to model ((a) and (b)) and predict ((c) and (d)) slopes and relative magnitudes of displacements for CKBot configurations considered.

bonds, and to a folded chain of RATChET7mm tetrahedrons threaded on a steel cable backbone and interlinked by inserted latches.

In each case, the key to successful modeling has been to identify the dominant compliances in the structure. For the Rubik's Snake, these compliances were easy to identify – the only force between modules (other than the force preventing inter-penetration) is the force holding adjacent prism faces on the backbone close to each other, while allowing for rotation around a face-centered axis. For the RATChET7mm, compliances are found in backbone hinges and in module interlinks created with inserts.

For CKBot modules, the dominant compliances were initially hard to identify, as the "natural" choice of treating the modules themselves as rigid bodies turned out to be unreasonable. Instead, module faces became the rigid bodies of the mechanical simulation, and the dominant compliances are found between the faces that comprise a single robot modules in addition to compliances between connected modules. The CKBot system has an architecture that is similar to MTRAN II, MTRAN III, Conro, and Superbot as well as many other systems. Thus a similar modeling approach may work well for these systems.

The generalized stiffness matrix allows users to quickly express relative relationships in stiffness. For example, one could study the effect of a connection method that is stiffer in shear ($\mathbf{e}_2$–$\mathbf{e}_3$ plane) relative to axial ($\mathbf{e}_1$) loading. Such a connection method can be approximated by

$$K_t = \begin{bmatrix} k_t & 0 & 0 \\ 0 & 10k_t & 0 \\ 0 & 0 & 10k_t \end{bmatrix}, \tag{12}$$

$$K_o = \begin{bmatrix} k_o & 0 & 0 \\ 0 & k_o & 0 \\ 0 & 0 & k_o \end{bmatrix}. \tag{13}$$

The designer could test many different lattice configurations and later substitute a more accurate stiffness matrix.

http://dx.doi.org/10.1017/S0263574710000743

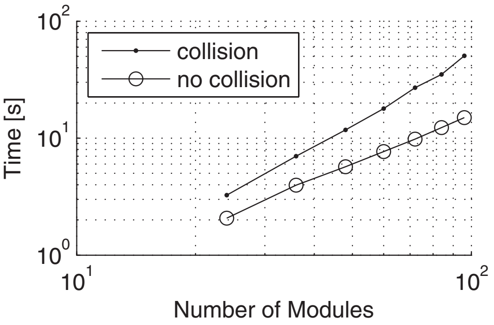


Fig. 19. Average computation time for fixed-free beam of N modules under six free end load conditions.

6.3. *Computational cost*

From its inception, our simulator was designed as a survey tool for automated assessment of modular configurations. Its *computational cost* is as low as we believe one could reasonably expect a fully general mechanical model of modular structures to have. A key benefit of the 6 DOF stiffness model is that it lumps together internal degrees of freedom without degrading its expressive ability. Its use allows the model to reduce the system to the minimal degrees of freedom necessary to predict its displacement under load.

The computational cost comes mostly from the MATLAB fsolve routine which is affected by several parameters of the configuration and the boundary conditions. The solution time is directly related to the magnitude of the load and the number of modules.

For a given acceptable residual, increasing load magnitudes requires an increasing number of load steps; this suggests that improvements in the solver are the most direct way to increase computation rates.

The computational cost is also a function of the number of modules. Figure 19 plots the computation time as a function of the number of modules in a beam configuration in seven different loading cases. In each case, the free end of a fixed-free beam of modules is loaded with 1mN of force. Plotting the curves on a log-log plot shows that the empirical computational cost is $O(N^{1.9})$ with collision detection and $O(N^{1.4})$ without.

6.4. *Modeling error*

There are several possible sources of error between experimental data and simulation results. In the RATChET7mm system, the linear elastic beam model approximates the stiffness between modules. However, it does not fully model the complexity of the module-to-module connections, (e.g., a press fit I-beam latch may slip depending on its orientation with respect to the load). The gap due to the cable slip fit through the modules and imperfect tension in the cable may lead to some additional inaccuracies.

7. Conclusion

Programmable matter and modular robots can form structures to best suit a functional requirement given short notice. In many applications, mechanical stiffness is a primary functional requirement and the system designers must provide a means to realize the required structural stiffness by choosing the arrangement of modules and the arrangement of bonds among them. One example is finding the stiffest configuration for building a wrench among the many suitable shapes or bonding paths of a RATChET7mm chain. In modular robot structures with heterogeneous and anisotropic bonding characteristics, the module configuration determines the stiffness of the structure and can be specified to tune the apparent stiffness to suit the task.

Table II. Geometric and material properties for latch and cable connection methods.

Property		Latch	Cable
length	mm	3.0	19.0
width/diameter	mm	1.0	0.24
height	mm	1.6	
E	GPa	2.3	200
ν		0.35	0.30

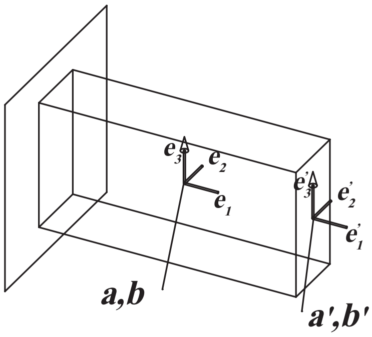


Fig. 20. Beam model with center of stiffness frame a and measured stiffness frame at free end a'. Frames a and a' are attached to module A and frames b and b' are attached to module B. At equilibrium, frames a and b coincide; likewise frames a' and b' coincide.

In this paper, we show that using a generalized 6 × 6 stiffness matrix we can express the stiffness behavior of arbitrary mechanical bonds and that this is an effective method for analyzing the apparent stiffness of programmable matter and modular robots. We offer enhancements to the 6 × 6 stiffness matrix model that handle anisotropies and nonlinearities such as increased stiffness when in collision.

We demonstrate the ability of the method to model both programmable matter with the Rubik's Snake and RATChET7mm systems and modular robots with the CKBot system. Our experiments with these physical systems validate the model. In the modular robot case, we present an experimental methodology for determining stiffness components that may be useful to similar modular robots. In the case of folded chain programmable matter, we demonstrate an ability to program apparent elasticity of a constant shape by choice of folding.

There are several potential areas for future work. The simulator can potentially guide the self-repair process of self-reconfigurable modular robot. In situations where portions of the modular robot have failed, the simulator can determine critical areas that should be repaired first to maintain sufficient rigidity. As another possible application of the simulator, one could study the effect of defects in a programmable matter configuration. The graph of stiffness matrices can easily be modified to test the effects of poor or

http://dx.doi.org/10.1017/S0263574710000743

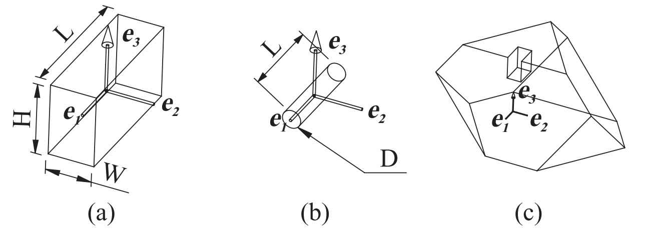


Fig. 21. Model of latch (a), cable (b) and face collision (c) connections with dimensions and coordinate frame at beam centroid. Latch and cable frames are located at their centroids; face collision frame is located at face centroid.

missing bonds between neighbors and to find configurations that are robust to localized bond failure.

In general, a simulation of a modular structure may have many sources of nonlinearities such those that arise from anisotropic stiffness and nonlinear bond force functions. The simulator can readily incorporate such arbitrary nonlinear effects. Reversible module interlinking using magnets[26] or electrostatics[25] is one important class of such nonlinearities. Magnetic interlinking of CKBot module clusters formed the basis of self-reassembly after explosion,[57] and adding such links to our simulation would allow us to anticipate the fracture loads for robots constructed of multiple clusters.

Acknowledgments

The authors acknowledge Steven Schlaefer for help with experiments. This work is funded in part by DARPA Grant W911NF-08-1-0228 (Programmable Matter).

References

1. Vicon MX Systems. http://www.vicon.com/products/viconmx.html.
2. J. R. Barber, *Elasticity* (Kluwer Academic Pub, Dordrecht, 2002).
3. F. Beer, J. Russell and J. DeWolf, *Mechanics of Materials* (McGraw-Hill, New York, NY, 2002).
4. J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp and T. Nguyen, "Programmable Parts: A Demonstration of the Grammatical Approach to Self-Organization," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton (2005) pp. 3684–3691.
5. D. Brandt and D. J. Christensen, "A New Meta-Module for Controlling Large Sheets of ATRON Modules," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007) pp. 2375–2380.
6. H. B. Brown, J. M. V. Weghe, C. A. Bererton and P. K. Khosla, "Millibot trains for enhanced mobility," *IEEE/ASME Trans. Mechatronics* **7**(4), 452–461 (2002).
7. F. Caccavale, C. Natale, B. Siciliano and L. Villani, "Six-dof impedance control based on angle/axis representations," *IEEE Trans. Robot. Autom.* **15**(2), 289–300 (1999).
8. F. Caccavale, B. Siciliano and L. Villani, "Quaternion-Based Impedance with Nondiagonal Stiffness for Robotmanipulators," *American Control Conference, 1998. Proceedings of the 1998*, Philadelphia, vol. 1 (1998).
9. G. S. Chirikjian, "Kinematics of a Metamorphic Robotic System," *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, San Diego, vol. 1 (May 1994) pp. 449–455.
10. E. Daróczy-Kiss, "On the minimum intrinsic 1-volume of voronoi cells in lattice unit sphere packings," *Periodica Mathematica Hungarica* **39**(1), 119–123 (2000).
11. M. De Rosa, S. Goldstein, P. Lee, J. Campbell and P. Pillai, "Scalable Shape Sculpting via Hole Motion: Motion Planning in Lattice-Constrained Modular Robots," *Proceedings of IEEE/RSJ International Conference Robotics and Automation*, Orlando, FL (2006) pp. 1462–1468.
12. E. D. Fasse and P. C. Breedveld, "Modeling of elastically coupled podies: Part I - General theory and geometric potential function method," *J. Dyn. Syst. Meas. Control* **120**, 496–500 (1998).
13. E. D. Fasse and P. C. Breedveld, "Modeling of elastically coupled bodies: Part II - Exponential and generalized coordinate methods," *J. Dyn. Syst. Meas. Control* **120**, 501–506 (1998).
14. T. Fukuda, S. Nakagawa, Y. Kawauchi and M. Buss, "Self Organizing Robots Based on Cell Structures - Cebot," In *Intelligent Robots, 1988., IEEE International Workshop*, Tokyo (Oct.–2 Nov. 1988) pp. 145–150.
15. K. Gilpin, A. Knaian and D. Rus, "Robot Pebbles: One Centimeter Modules for Programmable Matter Through Self-Disassembly," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage (2010).
16. S. C. Goldstein, J. D. Campbell and T. C. Mowry, "Programmable matter," *Computer* **38**(6), 99 (2005).
17. S. Griffith, Growing Machines, *PhD Thesis* (Massachusetts Institute of Technology, 2004).
18. S. Griffith, J. McBride, B. Su, B. Ren and J. M. Jacobson, "Folding any 3D shape," http://alumni.media.mit.edu/∼saul/PhD/pre_folding_s.pdf.
19. G. J. Hamlin and A. C. Sanderson, "TETROBOT Modular Robotics: Prototype and Experiments," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, vol. 2 (Nov. 1996) pp. 390–395.
20. J. M. Harris, J. L. Hirst and M. J. Mossinghoff, *Combinatorics and Graph Theory*, New York (Springer, 2008).
21. E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus and R. J. Wood, "Programmable matter by folding," *Proc. Natl. Acad. Sci.* **107**(28), 12441 (2010).
22. L. L. Howell, *Compliant Mechanisms*, New York (Wiley-Interscience, 2001).
23. S. Huang and J. M. Schimmels, "The bounds and realization of spatial stiffnesses achieved with simple springs connected in parallel," *Robot. Autom. IEEE Trans.* **14**(3), 466–475 (Jun. 1998).
24. M. W. Jorgensen, E. H. Østergaard and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Sendai, Japan (2004) pp. 2068–2073.
25. M. E. Karagozler, S. C. Goldstein and J. R. Reid, "Stress-Driven Mems Assembly + Electrostatic Forces = 1 mm Diameter Robot," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis (Oct. 2009).
26. A. Knaian, Design of Programmable Matter, *Master's Thesis* (Massachusetts Institute of Technology, 2008).
27. K. Kotay, D. Rus, M. Vona and C. McGray, "The Self-Reconfiguring Robotic Molecule," *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, vol. 1, Leuven, Belgium (1998) pp. 424–431.
28. H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo and S. Murata, "Distributed self-reconfiguration of M-TRAN III modular robotic system," *Int. J. Robot. Res.* **27**(3–4), 373–386 (2008).
29. J. Lončarić, "Normal forms of stiffness and compliance matrices," *Robot. Autom. IEEE J.* **3**(6), 567–572 (Dec. 1987).
30. S. Moaveni, *Finite Element Analysis: Theory and Application with ANSYS*, Upper Saddle River, NJ (Prentice-Hall, 1999).
31. T. Möller, "A fast triangle-triangle intersection test," *J. Graphics Tools* **2**(2), 25–30 (1997).
32. F. Mondada, A. Guignard, M. Bonani, D. Bär, M. Lauria and D. Floreano, "Swarm-bot: From Concept to Implementation," *Proceedings of the 2003 IEEE/RSJ International Conference*

http://dx.doi.org/10.1017/S0263574710000743

on Intelligent Robot and Systems, Las Vegas, Nevada, US (Oct. 27–31, 2003), pp. 1626–1631.

33. S. Murata, H. Kurokawa and S. Kokaji, "Self-Assembling Machine," *Proceedings of IEEE/RSJ IEEE International Conference on Robotics and Automation*, San Diego (1994) pp. 441–448.
34. S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatronics* **7**(4), 431 (2002).
35. R. M. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, Boca Raton (CRC, 1994).
36. A. Nguyen, L. J. Guibas and M. Yim, "Controlled module density helps reconfiguration planning," *Algorithmic and Computational Robotics: New Directions: The Fourth Workshop on the Algorithmic Foundations, (WAFR)*, AK Peters, Ltd. (2001).
37. M. Park, S. Chitta and M. Yim, "Isomorphic Gait Execution in Homogeneous Modular Robots," *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots,* Philadelphia (2006).
38. D. Rus and M. Vona, "Self-reconfiguration planning with compressible unit modules," *Proceedings of IEEE/RSJ IEEE International Conference on Robotics and Automation*, vol. 4, Detroit (1999) pp. 2513–2520.
39. D. Rus and M. Vona, "Crystalline robots: Self-reconfiguration with compressible unit modules," *Auton. Robots* **10**(1), 107–124 (2001).
40. J. Sastra, S. Chitta and M. Yim, "Dynamic rolling for a modular loop robot," *Int. J. Robot. Res.* (2007).
41. W. M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein and J. Venkatesh, "Multimode locomotion via SuperBot reconfigurable robots," *Auton. Robots* **20**(2), 165–177 (2006).
42. K. Støy, "Using cellular automata and gradients to control self-reconfiguration," *Robot. Auton. Syst.* **54**(2), 135 (2006).
43. K. Støy and R. Nagpal, "Self-Reconfiguration Using Directed Growth," *International Symposium on Distributed Autonomous Robotic Systems* (Jun. 23–25, 2004).
44. K. Støy, W. M. Shen and P. M. Will, "Using role-based control to produce locomotion in chain-type self-reconfigurable robots," *IEEE/ASME Trans. Mechatronics* **7**(4) (2002).
45. J. W. Suh, S. B. Homans and M. Yim, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," *Proceedings of IEEE/RSJ IEEE International Conference on Robotics and Automation*, vol. 4, Washington, DC (2002) pp. 4095–4101.
46. C. Ünsal and P. K. Khosla, "A Multi-Layered Planner for Self-Reconfiguration of a Uniform Group of I-Cube Modules," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* vol. 1, Maui (2001) pp. 598–605.
47. C. Ünsal, H. Kiliççöte and P. Khosla, "I(CES)-Cubes: A Modular Self-Reconfigurable Bipartite Robotic System," *SPIE Proceedings, Conference on Mobile Robots and Autonomous Systems*, vol. 3839, SPIE (Sep. 1999) pp. 258–269.
48. S. Vassilvitskii, M. Yim and J. Suh, "A complete, local and parallel reconfiguration algorithm for cube style modular robots," *Proceedings of IEEE/RSJ IEEE International Conference on Robotics and Automation*, vol. 1, Washington, DC (2002) pp. 117–125.
49. P. White, V. Zykov, J. Bongard and H. Lipson, "Three Dimensional Stochastic Reconfiguration of Modular Robots," *Robotics: Science and Systems*, Cambridge (2005) pp. 161–168.
50. P. J. White, K. Kopanski and H. Lipson, "Stochastic Self-Reconfigurable Cellular Robotics," *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, vol. 3, New Orleans, LA, USA (2004) pp. 2888–2893.
51. P. J. White, M. L. Posner and M. Yim, "Strength Analysis of Miniature Folded Right Angle Tetrahedron Chain Programmable Matter," *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, Anchorage, AK (2010) pp. 2785–2790.
52. P. J. White, C. E. Thorne and M. Yim, "Right Angle Tetrahedron Chain Externally-actuated Testbed (RATCHET): A Shape Changing System," *Proceedings of IDETC/CIE*, San Diego, CA, USA (2009).
53. P. J. White and M. Yim, "Reliable external actuation for full reachability in robotic modular self-reconfiguration," *Int. J. Robot. Res.* (2009).
54. M. Yim, Locomotion with a Unit-Modular Reconfigurable Robot *Technical Report* (Xerox PARC, 1995).
55. M. Yim, D. G. Duff and K. D. Roufas, "Polybot: A Modular Reconfigurable Robot," *Proceedings of IEEE/RSJ IEEE International Conference on Robotics and Automation*, vol. 1, San Francisco, CA, USA (2000) p. 514.
56. M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robot. Autom. Mag.* **14**(1), 43 (2007).
57. M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan and C. J. Taylor, "Towards Robotic Self-Reassembly After Explosion," *Proceedings of IEEE/RSJ IEEE International Conference on Intelligent Robots and Systems, 2007* (2007), pp. 2767–2772.
58. M. Yim, Y. Zhang, J. Lamping and E. Mao, "Distributed control for 3d metamorphosis," *Auton. Robot* **10**(1), 41 (2001).
59. S. Zhang and E. D. Fasse, "Spatial compliance modeling using a quaternion-based potential function method," *Multibody Syst. Dyn.* **4**(1), 75–101 (2000).
60. S. Zhang and E. D. Fasse, "A finite-element-based method to determine the spatial stiffness properties of a notch hinge," *J. Mech Des.* **123**, 141 (2001).

Appendix A. Analytical Stiffness Derivation of RATChET7mm Connections

A.1. Latch

We model the latch as a linear elastic rectangular cross section beam. Figure 21(a) defines the latch dimensions and body frame that is located at the centroid and aligned with the principal axes. It defines the height (1.6 mm) and width (1.0 mm) and length (3.0 mm).

The translational, rotational, and coupling stiffness matrices measured at the free end of a fixed-free beam are, respectively

$$K'_{lt} = \begin{bmatrix} \frac{A_l\,E_l}{L_l} & 0 & 0 \\ 0 & \frac{12\,C_{l2}\,E_l\,I_{l3}}{{L_l}^3} & 0 \\ 0 & 0 & \frac{12\,C_{l3}\,E_l\,I_{l2}}{{L_l}^3} \end{bmatrix}, \tag{14}$$

$$K'_{lo} = \begin{bmatrix} \frac{G_l\,H_l\,{W_l}^3\,c_2}{L_l} & 0 & 0 \\ 0 & \frac{4\,C_{l5}\,E_l\,I_{l2}}{L_l} & 0 \\ 0 & 0 & \frac{4\,C_{l6}\,E_l\,I_{l3}}{L_l} \end{bmatrix}, \tag{15}$$

$$K'_{lc} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{6\,C_{l2}\,E_l\,I_{l3}}{{L_l}^2} \\ 0 & \frac{6\,C_{l3}\,E_l\,I_{l2}}{{L_l}^2} & 0 \end{bmatrix}, \tag{16}$$

where A_l is the cross sectional area of the latch, E_l is the latch elastic modulus and I_{li} is the area moment of inertia about $\mathbf{e}_i$. The $K'_{lo_{11}}$ is determined from the expression for a fixed-free beam in torsion. The $c_2 = 0.2$ term is the correction factor for a rectangular beam in torsion.[3]

The stiffnesses due to transverse loading (along $\mathbf{e}_2$ or $\mathbf{e}_3$) are determined using the theory of elasticity.[2] Because the ratio of the length to the height of the beam is small, the Euler-Bernoulli beam equation does not apply. The correction factors C_{li} account for the effect of shear stress

$$C_{l2} = \frac{5\,{L_l}^2}{\frac{{W_l}^2\,[49\,\nu_l+48]}{4} + 5\,{L_l}^2}, \tag{17}$$

$$C_{l3} = \frac{5\,{L_l}^2}{\frac{{H_l}^2\,[49\,\nu_l+48]}{4} + 5\,{L_l}^2}, \tag{18}$$

$$C_{l5} = \frac{\frac{{W_l}^2\,[11\,\nu_l+12]}{4} + 5\,{L_l}^2}{\frac{{W_l}^2\,[17\,\nu_l+24]}{4} + 5\,{L_l}^2}, \tag{19}$$

$$C_{l6} = \frac{\frac{{H_l}^2\,[11\,\nu_l+12]}{4} + 5\,{L_l}^2}{\frac{{H_l}^2\,[17\,\nu_l+24]}{4} + 5\,{L_l}^2}, \tag{20}$$

where dimensions are defined in Fig. 21(a). Note in the limit where the length L_l is much larger than the width W_l or height H_l, the correction factors approach unity and stiffness terms are the solutions of the Euler-Bernoulli beam equation.

Fasse *et al.*[12] prove that the adjoint associated with the transformation $H_a^{a'}$ given by

$$Ad_{H_a^{a'}} = \begin{bmatrix} R_a^{a'} & \tilde{p}_a^{a'}\,R_a^{a'} \\ 0 & R_a^{a'} \end{bmatrix} \tag{21}$$

transforms a stiffness in frame a' to a by

$$K = Ad^{\mathrm{T}}_{H_a^{a'}}\,K'\,Ad_{H_a^{a'}}. \tag{22}$$

Using Eq. (22) and requiring the coupling stiffness to be symmetric we compute the position of the center of stiffness. Indeed the position of the center of stiffness is located at the centroid of the beam. Further, in the case where modules are connected by a latch only, the coupling stiffness at the center of stiffness K_{lc} is zero,

$$K_{lt} = \begin{bmatrix} \frac{A_l\,E_l}{L_l} & 0 & 0 \\ 0 & \frac{12\,C_{l2}\,E_l\,I_{l3}}{{L_l}^3} & 0 \\ 0 & 0 & \frac{12\,C_{l3}\,E_l\,I_{l2}}{{L_l}^3} \end{bmatrix}, \tag{23}$$

$$K_{lo} = \begin{bmatrix} \frac{G_l\,H_l\,{W_l}^3\,c_2}{L_l} & 0 & 0 \\ 0 & K_{lo_{22}} & 0 \\ 0 & 0 & K_{lo_{33}} \end{bmatrix}, \tag{24}$$

where

$$K_{lo_{22}} = \frac{4\,C_{l5}\,E_l\,I_{l2}}{L_l} - \frac{3\,C_{l3}\,E_l\,I_{l2}}{L_l}, \tag{25}$$

$$K_{lo_{33}} = \frac{4\,C_{l6}\,E_l\,I_{l3}}{L_l} - \frac{3\,C_{l2}\,E_l\,I_{l3}}{L_l}. \tag{26}$$

Thus the stiffness matrix of the latch at the center of stiffness is given by

$$K_l = \begin{bmatrix} K_{lt} & 0 \\ 0 & K_{lo} \end{bmatrix}. \tag{27}$$

A.2. Cable

Modeling the cable's effect on the stiffness is difficult. The cable must have a loose enough fit for manageable folding of a configuration. Because the cable is not press fit into the module, there exists a small gap that can allow the cable to move radially. The lack of a press fit implies that the cable can also slip axially with respect to the module.

Therefore, the cable is approximated by a long cylindrical beam as shown in Fig. 21(b). The length of the cable beam model is 19 mm which is the total length of cable from the point where it enters module A to the point where it exits module B. The cable depicted in Fig. 4 is not drawn to proportion; it defines the relative location of the center of stiffness of the cable.

The derivation of the stiffness matrix of the cable is similar to that of the latch. Because the length of the cable is much larger than its diameter, there is no need for the correction factors. The cable stiffness matrices in its center of stiffness frame are

$$K_{ht} = \begin{bmatrix} \frac{A_h\,E_h}{L_h} & 0 & 0 \\ 0 & \frac{12\,E_h\,I_{h3}}{{L_h}^3} & 0 \\ 0 & 0 & \frac{12\,E_h\,I_{h2}}{{L_h}^3} \end{bmatrix}, \tag{28}$$

$$K_{ho} = \begin{bmatrix} \frac{G_h\,J_h}{L_h} & 0 & 0 \\ 0 & \frac{E_h\,I_{h2}}{L_h} & 0 \\ 0 & 0 & \frac{E_h\,I_{h3}}{L_h} \end{bmatrix}, \tag{29}$$

where for clarity we use the subscript h (for hinge) to denote the cable stiffness. The complete stiffness matrix for each cable i is given by

$$K_{h_i} = \begin{bmatrix} K_{h_i t} & 0 \\ 0 & K_{h_i o} \end{bmatrix}. \tag{30}$$

A.3. Face collision

The possibility of collision of two modules under some loading directions introduces an anisotropy into the system. The connection between modules is stiffer when modules are compressed together than when they are pulled apart. When two modules collide, a face collision stiffness matrix K_f is added to the general stiffness matrix (K_l or K_{lh}).

We measure the collision stiffness in the centroid frame (shown in Fig. 21(c)) of the common face between connected modules. The collision stiffness matrix is determined from FEA. The loaded module is assumed to make a point contact with the fixed module at a point on the face furthest from axis of rotation. This assumption is valid considering the small gap between modules allows some collision free rotation until a point contact is made.

Considering small displacements from the center of stiffness frame, only displacements that cause a collision should be considered. Collision only affects the translational stiffness $K_{ft_{11}}$ normal to the face ($\mathbf{e}_1$) and the rotational stiffnesses $K_{fo_{22}}$ and $K_{fo_{33}}$ about axes lying on the face ($\mathbf{e}_2$ and $\mathbf{e}_3$). The $K_{ft_{11}}$ term is determined by loading one module along ($\mathbf{e}_1$) and recording the displacement reported by FEA. $K_{fo_{22}}$ is determined by constraining the module to rotate about $\mathbf{e}_2$ and applying a moment about this axis and recording the angular displacement. Note that in the case of the latch the direction of rotation matters. The simulator handles this asymmetry by the way it detects collision. $K_{fo_{33}}$ is determined in a similar manner.

The measured stiffness is then transformed using Eq. (22) to the center of stiffness frame between modules.

http://dx.doi.org/10.1017/S0263574710000743