



3-1-2022

TrussBot: Modeling, Design, and Control of a Compliant, Helical Truss of Tetrahedral Modules

Yuhong Qin

Linda Ting

Celestina Saven

Yumika Amemiya

Michael Tanis

See next page for additional authors

Follow this and additional works at: https://repository.upenn.edu/grasp_papers

 Part of the [Robotics Commons](#)

Recommended Citation

Yuhong Qin, Linda Ting, Celestina Saven, Yumika Amemiya, Michael Tanis, Randall Kamien, and Cynthia R. Sung, "TrussBot: Modeling, Design, and Control of a Compliant, Helical Truss of Tetrahedral Modules", *IEEE Conference on Robotics and Automation (ICRA)* . March 2022.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/grasp_papers/71
For more information, please contact repository@pobox.upenn.edu.

TrussBot: Modeling, Design, and Control of a Compliant, Helical Truss of Tetrahedral Modules

Abstract

Modular and truss robots offer the potential of high reconfigurability and great functional flexibility, but common implementations relying on rigid components often lead to highly complex actuation and control requirements. This paper introduces a new type of modular, compliant robot: TrussBot. TrussBot is composed of 3D-printed tetrahedral modules connected at the corners with compliant joints. We propose a truss geometry, analyze its deformation modes, and provide a simulation framework for predicting its behavior under applied loads and actuation. The TrussBot is geometrically constrained, thus requiring compliant joints to move. The TrussBot can be actuated through a network of tendons which pinch vertices together and apply a twisting motion due to the structure's connectivity. The truss was demonstrated in a physical prototype and compared to simulation results.

Supplemental video: <https://youtu.be/bcvFMq40EzI>

Keywords

modular robot, compliant, simulation, design

Disciplines

Engineering | Robotics

Author(s)

Yuhong Qin, Linda Ting, Celestina Saven, Yumika Amemiya, Michael Tanis, Randall Kamien, and Cynthia R. Sung

TrussBot: Modeling, Design, and Control of a Compliant, Helical Truss of Tetrahedral Modules

Yuhong Qin*, Linda Ting*, Celestina Saven*, Yumika Amemiya,
Michael Tanis, Randall D. Kamien, Cynthia Sung

Abstract—Modular and truss robots offer the potential of high reconfigurability and great functional flexibility, but common implementations relying on rigid components often lead to highly complex actuation and control requirements. This paper introduces a new type of modular, compliant robot: TrussBot. TrussBot is composed of 3D-printed tetrahedral modules connected at the corners with compliant joints. We propose a truss geometry, analyze its deformation modes, and provide a simulation framework for predicting its behavior under applied loads and actuation. The TrussBot is geometrically constrained, thus requiring compliant joints to move. The TrussBot can be actuated through a network of tendons which pinch vertices together and apply a twisting motion due to the structure’s connectivity. The truss was demonstrated in a physical prototype and compared to simulation results.

I. INTRODUCTION

Modular robots are systems of coordinated modules that can be designed for a wide variety of applications because of their scalability, reconfigurability, versatility, fault-tolerance, and mass reproducibility [1]. By definition, a modular system is composed of many smaller individual elements known as modules. Parameterization of a modular system involves both the size and shape of the individual modules and the total number of modules in the system; thus, the space of possible modular configurations is exponentially large and interesting to explore. A high degree of redundancy and morphability are inherent to modular systems, as demonstrated by systems like HexaMorph [2], M-TRAN [3], ATRON [4], and Miche [5]. Typically, these modular systems make use of regular geometries like tetrahedra [2], [6], cubes [5], [7], and rhombic dodecahedra [8] because these space-filling geometries provide a foundation for the construction of solid and more robust structures.

Simple tetrahedral robots with the ability to manipulate edge length have been able to achieve both walking gaits [9] and rolling gaits [10], [11]. When combined into full reconfigurable systems where truss members can be individually controlled [12], [13] and the connectivity changed [14], these

robots achieve a range of morphological versatility rivaling other existing unit-based modular systems. For example, the Tetrobot system consists of spherical joints and rigid bar components in a modular configuration [15]. The system can be reassembled and reconfigured as arms, platforms, and walking machines under various loading conditions. However, the Tetrobot’s motion dynamics were not thoroughly explored or understood.

Module rigidity introduces significant planning complexities, leading to complicated reconfiguration plans and, in some cases, configurations that can not be achieved at all [14]. Compliance can greatly simplify control [16]. Modular systems with compliant actuators are able to reconfigure autonomously or manually [17], climb surfaces [18], travel over terrain [19]–[22], and act as manipulators [23]. Similar advantages are found in flexible truss robots [24]–[26].

We are interested in understanding the behavior of compliant, modular, tetrahedral trusses and their ability to achieve a variety of motions, including basic locomotion and manipulation, when combined into different configurations. In the examples above, in order to prevent the robot from performing undesirable motions, it is necessary to structure the robot to limit degrees of freedom and control modes of deformation. This is done by overconstraining the truss and relying on element compliance to move. We focus on a particular truss geometry, a series of spherical four-bars, which has limited degrees of freedom in its rigid form, but is able to demonstrate additional rotation and twisting when the joints are compliant. The resulting TrussBot system consists of N rigid tetrahedron modules connected at their vertices. We analyze the effect of motors and tendons, which contract or release to deform the truss along desired degrees of freedom. The contributions of this paper include:

- a robust model for TrussBot, a vertex-connected truss of tetrahedral modules,
- a deformation analysis of TrussBot identifying major degrees of freedom,
- simulations of the truss under tendon driven control, and
- experimental comparisons of simulation results to a physical hardware.

The result is a compliant truss that is able to bend and twist inside out continuously, even with only limited angle range at the joints, similarly to [27], but with greater controllability.

This paper is organized as follows. Section II introduces the model used to predict and visualize the motion of the truss. Section III describes how we use the model to simulate the TrussBot motion. Section IV describes the design, fabri-

*Equal contribution.

Y. Qin, L. Ting, C. Saven, Y. Amemiya, and C. Sung are with the General Robotics, Automation, Sensing & Perception (GRASP) Lab at the University of Pennsylvania, Philadelphia, PA, USA. M. Tanis and R. Kamien are with the Department of Physics and Astronomy, University of Pennsylvania. Corresponding author: crsung@seas.upenn.edu.

Support for this project has been provided in part by NSF Grants #1138847 and #1950720, by the Penn Center for Undergraduate Research and Fellowships, and by the Google CSResearch program. Y. Qin was supported by the Penn Benjamin Franklin Scholars program. C. Saven was supported by the Penn Rachleff Scholars program. R. Kamien was supported by the Simons Investigator Grant #291825. We also thank Jasmine Hughley, Morgan McLees, and Rose Ridder for assistance with fabrication tests.

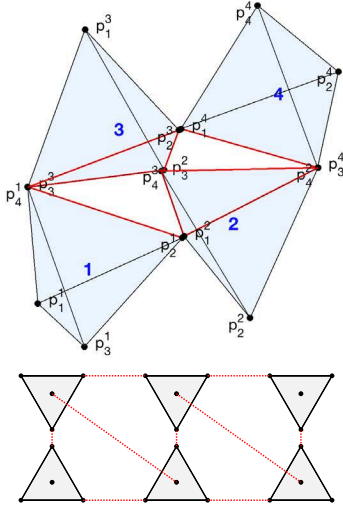


Fig. 1. Top: One unit of the TrussBot consists of four tetrahedra forming a spherical four-bar linkage. Bottom: Diagram of connections in TrussBot. Each tetrahedron is shown as a black triangle and vertex connections are drawn as red dotted lines. Here, two four-bars are shown connected in series.

cation, and hardware specifications for building the TrussBot. Section V includes the analysis of the TrussBot's degrees of freedom, deformation modes, and how the hardware implementation compares to the software simulation. We conclude with limitations and future work in Section VI.

II. TRUSSBOT MODEL

The underlying structure of the TrussBot is a chain of spherical four-bar linkages. Fig. 1 shows the basic linkage. Four modules are connected at the vertices to form the four-bar linkage with equal side lengths. Two modules are also connected diagonally opposite each other, constraining the vertices of the four-bar to move on the surface of a sphere centered on this vertex. To generate a full truss, four-bar metamodules are linked in series. The resulting truss is a helix structure with a twist of approximately one full rotation every 14 modules. The TrussBot is modular in that four-bar metamodules or individual modules can be added or removed to lengthen or shorten the truss. Additional modules can be connected in other ways to decouple the units or add extra degrees of freedom.

A. Definitions

The TrussBot is constructed out of N tetrahedral modules. Each tetrahedron $\mathbf{T}^i : i \in \{1, 2, \dots, N\}$, is a regular convex polyhedron with four triangular sides. Let the vector $\mathbf{p}_a^i = [x_a^i, y_a^i, z_a^i]^\top$ be the 3D coordinates of the a th vertex of tetrahedron $\mathbf{T}^i$, where $a \in \{1, 2, 3, 4\}$. An edge $\mathbf{e} = (\mathbf{p}_a^i, \mathbf{p}_b^i)$ connects every pair of vertices in $\mathbf{T}^i$.

To track the TrussBot's configuration, let $\mathcal{V}^i = \{\mathbf{p}_a^i : \forall a \in \{1, 2, 3, 4\}\}$ be the set of all vertices and $\mathcal{E}^i = \{(\mathbf{p}_a^i, \mathbf{p}_b^i) : a < b \in \{1, 2, 3, 4\}\}$ be the set of all edges in $\mathbf{T}^i$. The TrussBot is a graph $\mathcal{G} = (\mathcal{T}, \mathcal{C})$, where each node $\mathbf{T}^i \in \mathcal{T}$ is a module, and a connection $\mathbf{c} = (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathcal{C}$ connects vertex $\mathbf{p}_a^i$ to vertex $\mathbf{p}_b^j$. Then

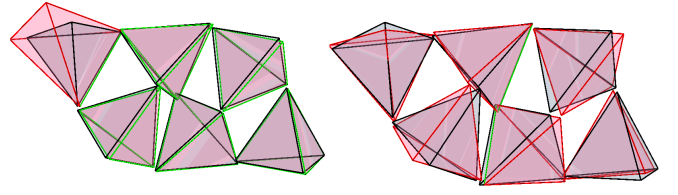


Fig. 2. Deformation modes. Black lines represent the original positions of edges, and colored lines represent new positions.

$N = |\mathcal{T}|$ is the total number of tetrahedral modules, and $M = |\mathcal{C}|$ is the total number of connections. The set $\mathcal{V} = \cup_i \mathcal{V}^i$ is the set of all vertices, and $\mathcal{E} = \cup_i \mathcal{E}^i$ is the set of all edges. The total number of vertices $|\mathcal{V}| = 4N$ and edges $|\mathcal{E}| = 6N$. Sample numbering is shown in Fig. 1.

The state of a TrussBot $\mathcal{G}$ is denoted as $\mathbf{q} = [\mathbf{p}_1^\top, \mathbf{p}_2^\top, \dots, \mathbf{p}_4^\top]^\top$, a $12N \times 1$ column vector containing the position of every vertex $\mathbf{p}_i \in \mathcal{V}$. Let $\Delta \mathbf{q}$ and $\dot{\mathbf{q}}$ represent the displacement and velocity of $\mathcal{G}$, respectively.

B. Rigidity Analysis

The rigidity matrix $\mathbf{R}$ offers insight into the degrees of freedom (DoF) when there is no deformation in the robot's geometry [28]. We construct a rigidity matrix $\mathbf{R}$ of size $(6N + 3M) \times (12N)$

$$\forall \mathbf{e}_k \in (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathcal{E} \cup \mathcal{C} : \mathbf{R}_{(k, 4i+a)} = (\mathbf{p}_a^i - \mathbf{p}_b^j)^\top \quad (1)$$

$$\mathbf{R}_{(k, 4j+b)} = (\mathbf{p}_b^j - \mathbf{p}_a^i)^\top$$

where we use the simplified notation that $\mathbf{R}_{(k, c)}$ is the k th row elements contained in the 3-column block including columns $3c$ to $3c + 2$. The k th row of $\mathbf{R}$ represents the k th edge of the truss. Each three-column block represents x , y , and z coordinate constraints on the endpoints of that edge.

If the truss is moving with velocity $\dot{\mathbf{q}}$, then $\mathbf{R}\dot{\mathbf{q}}$ represents the change in lengths of the edges. It follows that the nullspace of $\mathbf{R}$ includes velocities that maintain the rigidity of the truss, and the dimensionality of the nullspace is the number of DoF of the truss. It should be noted that the vertices in a connection $\mathbf{c} \in \mathcal{C}$ are actually coincident in a rigid truss, meaning that rows in the rigidity matrix corresponding to these connections will be near-0 and are unlikely to appear in the matrix. We therefore replace each row corresponding to a connection $\mathbf{c}$ with 3 rows

$$\forall \mathbf{c}_k = (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathcal{C} : \mathbf{R}_{k, 4i+a} = I_3 \quad (2)$$

$$\mathbf{R}_{k, 4j+b} = -I_3$$

That is, we constrain the x , y , and z velocities of the vertices forming the connection to be equal.

The rigidity analysis reveals that a TrussBot in chain configuration has 3 DoF regardless of the number of modules. As plotted in Fig. 2, two of the DoFs correspond to modules at each end rotating about the edge connected to the truss (left). The third DoF corresponds to motion in the spherical four-bar (right). Since all of the four-bars in the truss are linked to one another, the entire chain moves in sync and it can be said that the main middle section of the robot only

has 1 DoF. When the robot is in the ring configuration, the additional 3 connections fully constrain the structure, and the TrussBot is rigidly locked. These insights are promising since they mean that the physical TrussBot can be actuated using a small number of actuators.

C. Stiffness Matrix

Practically, it is complex to build rigid connections of the type modeled in Section II-B while keeping the entire structure lightweight. A TrussBot is likely to undergo deformations beyond those exposed in the rigidity analysis. As a result, the chain is not purely 1 DoF. It is therefore interesting to consider how the low DoF chain and “rigidly locked” ring structure moves. We observe from Fig. 2 that the main DoF for the chain involves coupled twisting and axial compression and that the truss resists bending along the plane of the robot. We expect similar deformation modes in the ring structure, i.e., twisting of the truss with radial compression, and very little bending out of the plane of the ring.

We construct a stiffness matrix $\mathbf{K}$ [29], which characterizes how the truss deforms under external forces. Each edge $e \in \mathcal{E}$ is modeled as a linear spring so that

$$\mathbf{K} = \mathbf{A}^\top \mathbf{C} \mathbf{A} \quad (3)$$

where $\mathbf{A}$ is the adjacency matrix representing the truss’s connectivity and $\mathbf{C}$ is a diagonal matrix of spring constants. $\mathbf{A}$ is constructed as follows:

$$\begin{aligned} \forall e_k = (\mathbf{p}_a^i, \mathbf{p}_b^i) \in \mathcal{E} : \mathbf{A}_{(i, 4i+a)} &= \frac{\mathbf{p}_a^i - \mathbf{p}_b^i}{\|\mathbf{p}_a^i - \mathbf{p}_b^i\|} \\ \mathbf{A}_{(i, 4i+b)} &= \frac{\mathbf{p}_b^i - \mathbf{p}_a^i}{\|\mathbf{p}_b^i - \mathbf{p}_a^i\|} \end{aligned} \quad (4)$$

Again we replace the rows corresponding to connections

$$\begin{aligned} \forall c_k = (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathcal{C} : \mathbf{A}_{k, 4i+a} &= \mathbf{I}_3 \\ \mathbf{A}_{k, 4j+b} &= -\mathbf{I}_3 \end{aligned} \quad (5)$$

to essentially place three springs, one along each of the coordinate axes of the connection. The adjacency matrix $\mathbf{A}$ is then a $(6N + 3M) \times 12N$ matrix.

Let $\mathbf{F} = [f_{1x}^1, f_{1y}^1, f_{1z}^1, \dots, f_{4z}^N]^\top$ be the column vector of forces applied to all vertices $\mathbf{p}_a^i \in \mathcal{V}$ along each of the three coordinate axes and $\mathbf{F}_a^i = [f_{ax}^i, f_{ay}^i, f_{az}^i]^\top$ be the force applied to vertex $\mathbf{p}_a^i$. The truss’s displacement vector $\Delta \mathbf{q}$ as a result of the force is computed as

$$\mathbf{F} = \mathbf{K} \Delta \mathbf{q} \quad (6)$$

Note that due to the truss’s regular structure, the stiffness matrix $\mathbf{K}$ contains N repetitions of a well-defined stiffness matrix block corresponding to a single four-bar. It is therefore simple to construct the stiffness matrix as N changes.

D. TrussBot’s Dominant Deformation Modes

The TrussBot’s DoF and dominant deformation modes are the eigenvectors of the stiffness matrix $\mathbf{K}$ with the lowest corresponding eigenvalues. Free DoF correspond to an eigenvalue of zero. For the chain configuration, nine such eigenvectors exist, 3 translational and 3 rotational rigid body

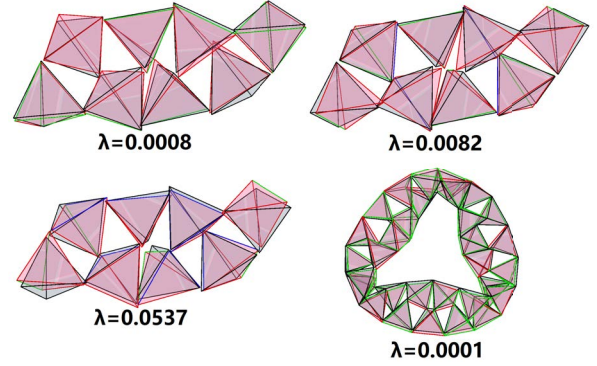


Fig. 3. Modes of greatest compliance. Red edges are longer than original, green are the same, and blue are shorter.

motions in 3D space, 2 corresponding to rotation at each end module, and 1 corresponding to global contraction and expansion. These results match the rigidity analysis.

In addition to these motions, we observe the directions in which the truss has the greatest compliance (smallest non-zero eigenvalues). Fig. 3 shows the modes of greatest compliance for the chain and the ring. In order of increasing eigenvalue, the arm performs four-bar expansion, four-bar contraction, and end rotation. Similarly, the ring experiences a twisting motion from the chain of four-bars, which causes some of the units to rotate inwards compared to the others.

III. SIMULATION

To simulate the motion of the truss under actuation, we numerically integrate over truss deformations as predicted by the stiffness matrix. A first-order forward Euler method is used. Forces are exerted on the truss by internal truss members during self-collision and with the external environment, by gravity and friction, and by tendons for actuation. All forces are modeled as point forces. To reduce accumulated numerical error in the truss edge lengths and vertex-to-vertex connections, at each time step, a gradient descent based search is used to minimize spring potential energy.

A. Tendon Actuation

Let $\mathbf{D}$ be the set of all tendons on the TrussBot. Each tendon $t_k = (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathbf{D}$ has a maximum length l_k .

$$\begin{aligned} \forall t_k = (\mathbf{p}_a^i, \mathbf{p}_b^j) \in \mathbf{D} \text{ s.t. } \|\mathbf{p}_a^i - \mathbf{p}_b^j\| &> l_k : \\ \mathbf{F}_a^i &= -\gamma_t \widehat{(\mathbf{p}_b^j - \mathbf{p}_a^i)} \\ \mathbf{F}_b^j &= \gamma_t \widehat{(\mathbf{p}_a^i - \mathbf{p}_b^j)} \end{aligned} \quad (7)$$

where $\gamma_t > 0$ is the tendon stiffness.

B. Collision Detection

Potential collisions are detected with a modified Gilbert-Johnson-Keerthi (GJK) distance algorithm [30]. GJK computes a Minkowski difference between two polyhedra and iteratively checks simplices on this difference to find the minimum distance d_{ij} between the two. The resulting simplex is used to find the point p_i on $\mathbf{T}^i$ closest to $\mathbf{T}^j$ and the point

p_j on $\mathbf{T}^j$ closest to $\mathbf{T}^i$. Let the sets of vertices closest to p_i and p_j be $\mathcal{V}_{pi} \subset \mathcal{V}^i$ and $\mathcal{V}_{pj} \subset \mathcal{V}^j$, and let p_{ci} and p_{cj} be the centroids of $\mathbf{T}^i$ and $\mathbf{T}^j$, respectively. Let f_{ij} be the deterring force applied to $\mathbf{T}^i$ and f_{ji} be the deterring force applied to $\mathbf{T}^j$.

$$f_{ij} = \begin{cases} \beta(d_{ij})(\widehat{p_{ci} - p_{cj}}) & d_{min} < d_{ij} < d_{max} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (8)$$

$$f_{ji} = -f_{ij} \quad (9)$$

where d_{min} and d_{max} are constants denoting the minimum and maximum distances between which collision forces will be applied, and the barrier function $\beta(d)$ is defined as

$$\beta(d) = \left(\frac{d - d_{min}}{d} \right)^2 - \left(\frac{d_{max} - d_{min}}{d_{max}} \right)^2 \quad (10)$$

The forces f_{ij} and f_{ji} are distributed equally to vertices in $\mathcal{V}_{pi}$ and $\mathcal{V}_{pj}$, respectively.

$$\begin{aligned} \forall \mathbf{p}_a^i \in \mathcal{V}_{pi} : \mathbf{F}_a^i &+= \gamma_\beta \frac{f_{ij}}{|\mathcal{V}_{pi}|} \\ \forall \mathbf{p}_b^j \in \mathcal{V}_{pj} : \mathbf{F}_b^j &+= \gamma_\beta \frac{f_{ji}}{|\mathcal{V}_{pj}|} \end{aligned} \quad (11)$$

where $\gamma_\beta > 0$ controls the strength of the collision forces.

C. Gravitational Forces

For a module $\mathbf{T}^i$ with mass m , the gravitational force is

$$\forall \mathbf{p}_a^i \in \mathcal{V}^i : \mathbf{F}_a^i = \frac{m}{|\mathcal{V}^i|} [0, 0, -9.81] \quad (12)$$

D. Normal Forces

Normal forces are modeled as collision avoidance forces between the truss and some environmental surface. Let d be the distance between some vertex $\mathbf{p}_a^i$ and a surface with normal $\hat{n}$. Let $\mathbf{N} = [n_{1x}^1, n_{1y}^1, n_{1z}^1, \dots, n_{4z}^N]^\top$ be the column vector of normal forces applied to all vertices and $\mathbf{N}_a^i = [n_{ax}^i, n_{ay}^i, n_{az}^i]^\top$ be the normal force applied to vertex $\mathbf{p}_a^i$.

$$\mathbf{N}_a^i = \gamma_n \beta(d) \hat{n} \quad (13)$$

$$\forall \mathbf{p}_a^i \in \mathcal{V} : \mathbf{F}_a^i = \mathbf{N}_a^i \quad (14)$$

where $\gamma_n > 0$ is a constant, and it is tuned to offset the gravitational force and ensure correct movements when simulated on the ground.

E. Friction Forces

Friction forces are directed opposite to the net applied force and perpendicular to the normal force acting on the truss. At every time step, the initial expected displacement $\Delta \mathbf{q}$ is calculated using the applied forces $\mathbf{F}$ without friction. Let the expected displacement of $\mathbf{p}_a^i$ be $\Delta \mathbf{q}_a^i$. If there is a nonzero normal force $\mathbf{N}_a^i$ acting on $\mathbf{p}_a^i$, we apply friction.

$$\begin{aligned} \forall \mathbf{p}_a^i \in \mathcal{V} : \mathbf{F}_a^i &+= \\ (\Delta \mathbf{q}_a^i - \widehat{\text{proj}_{\mathbf{N}_a^i} \Delta \mathbf{q}_a^i}) \min \left(\left\| \mathbf{F}_a^i - \text{proj}_{\mathbf{N}_a^i} \mathbf{F}_a^i \right\|, \mu_s \left\| \mathbf{N}_a^i \right\| \right) \end{aligned} \quad (15)$$

where $\mu > 0$ is the coefficient of friction.

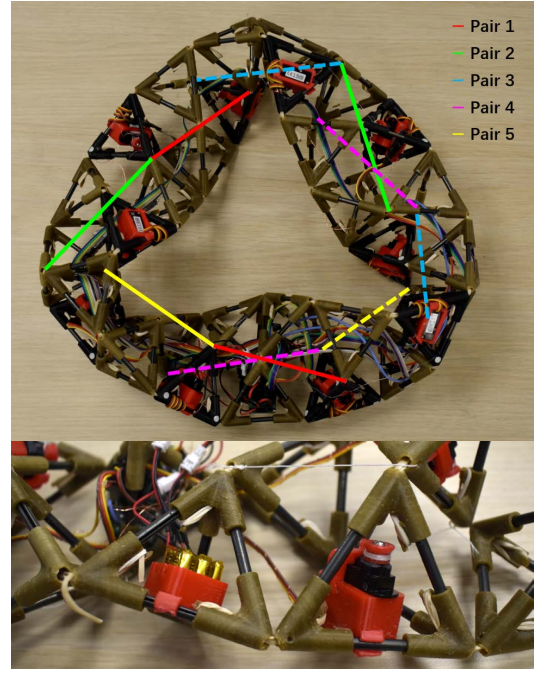


Fig. 4. Top: A constructed TrussBot consisting of 31 modules. Ten actuated modules contain motors to wind and unwind five pairs of tendons. Bottom: Close-up of a battery module (left) and an actuated module (right).

IV. HARDWARE

We built a physical prototype of the simulated TrussBot (Fig. 4). Each module is constructed from six spring steel bars connected by four 3D printed corner pieces. The spring steel bars are 1/8" in diameter and 1-3/8" long (McMaster part #98296A886). Each corner has a hole at the tip and a hook on the inside for a rubber band to be threaded through to connect to another module. This creates a compliant and flexible joint. The side length of the tetrahedron module is 57.5 mm. Fig. 4 shows a close-up of the module construction.

3D printed mounts for various electronic components attach to the passive modules. The mounts can be slotted and clipped in place without disassembling the structure. The motor mounts hold Turnigy TGY-1370A servomotors. 3D printed spools 6.3 mm in diameter are attached to the motor outputs and allow the motors to wind and unwind tendons on the truss. Each mount also has a snap on cap to hold the motor in place and direct the tendon directly into the spool. Other mounts hold batteries and a Teensy 3.2 control board.

To actuate the system, tendons (Berkley Trilene XL 0.008-inch diameter fishing line) are attached with one end fixed at a spool and the other end fixed to the other connecting modules. Position control on the servos turn the spools to contract and release the tendons to the desired lengths.

Due to the simplicity of the design, assembling a single module takes 4 hours of printing time and a few minutes of assembly. Assembling an entire truss takes an additional 1-2 hours, depending on the number of modules, and the modules can easily be connected and disconnected by hooking or unhooking the rubber bands.

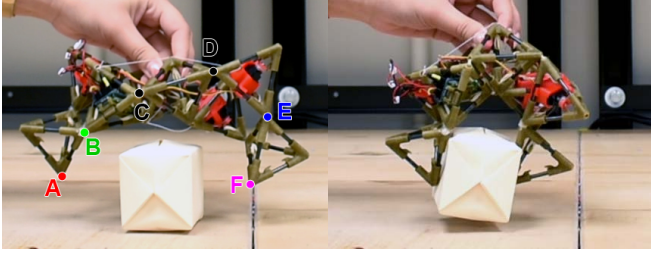


Fig. 5. Gripper with 8 modules and 2 tendons. Left: open configuration. Right: closed configuration.

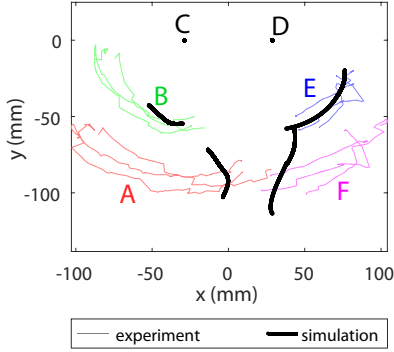


Fig. 6. Comparison of gripper opening in simulation vs experiments over 3 trials. Tracked points are labeled in Fig. 5.

V. EXPERIMENTAL RESULTS

We tested three different configurations of varying complexity. The simulation was implemented in MATLAB and run on a computer with Intel i7-6700 CPU and 16 GB memory. The following simulation parameters were used: edge stiffness $c_k = 100$, joint stiffness $c_k = 0.14$, tendon stiffness $\gamma_t = 50$, barrier strength $\gamma_\beta = 1.0$, normal force barrier strength $\gamma_n = 0.7$, and friction coefficient $\mu = 0.15$. A time step of 0.05 s was used in the numerical integration.

A. 1-DoF Gripper

The simplest demonstration is a gripper consisting of 8 modules connected in a chain. As previously discussed, the truss is theoretically 1 DoF in the center with two freely-hanging tetrahedra on either end. Two antagonistic actuators are required to open and close the gripper. A long tendon on the outside opens the gripper and a short tendon on the inside closes the gripper. The physical structure is in Fig. 5.

Figure 6 shows a comparison of the simulation and 3 experimental tests on the gripper design. For 2311 time steps, the simulation took 25 s. We tracked the locations of 6 points on the inside of the gripper via a camera oriented perpendicular to the plane of the gripper. The points from the physical test and the simulation were transformed so that the center bar C-D was kept still. As predicted, in the physical device, points B and E followed arcs centered about C and D, respectively, as the gripper open and closed. Points A and F, which are on the free modules at the end, do not match between the simulation and the physical experiments. In experiments, in the absence of any external

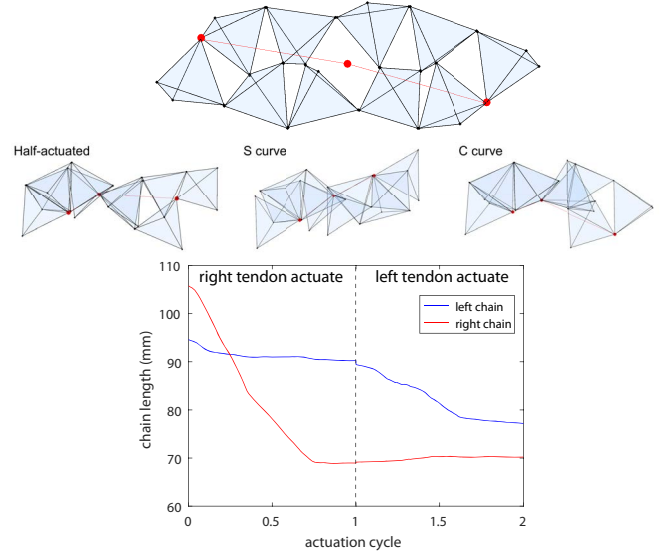


Fig. 7. Top: Diagram of truss with central skeleton highlighted. These points were used to track segment lengths over actuation cycles. Middle: Three configurations achievable with this truss. Bottom: Experimental results show the two chains are able to contract independently.

forces, these tetrahedra tended to rotate with the modules that they were connected to (resulting in points A and F following large arcs), whereas in simulation, these tetrahedra tended to remain in the same orientation since there were no external forces on them (resulting in points A and F following lines). Practically, this extra degree of freedom provided additional compliance for the gripper when picking up different objects, as shown in Fig. 5. In the future, more accurate prediction of these points could be achieved by adding a torsional stiffness to the vertex connections.

B. 2-DoF Chain

A 2-DoF chain configuration is composed of two chains of 5 modules each connected at two points. Two vertices are left free on the middle modules, decoupling the two chains. Tendons were added to both chains, allowing both to bend and contract. Figure 7 shows the truss and the various achievable configurations: half-actuated (only one tendon contracted), S-curve (chains bending in different directions), and C-curve (chains bending in the same direction).

The tendons can be contracted simultaneously or in sequence, and each chain bends independently of the other. We constructed the same chain with the physical TrussBot modules and contracted the tendons to achieve the various shapes. The bottom plot of Fig. 7 shows the lengths of the left and right chains as first the right tendon and then the left tendon is pulled by the motor. When the right tendon is actuated, only the right chain contracts, resulting in a reduction in length of 34.6% compared to the left chain's 4.6%. When the left tendon is actuated afterwards, only the left chain contracts 14.5% compared to the right chain's 1.8% expansion. Thus, the two chains are decoupled in the physical truss, matching the simulation predictions.

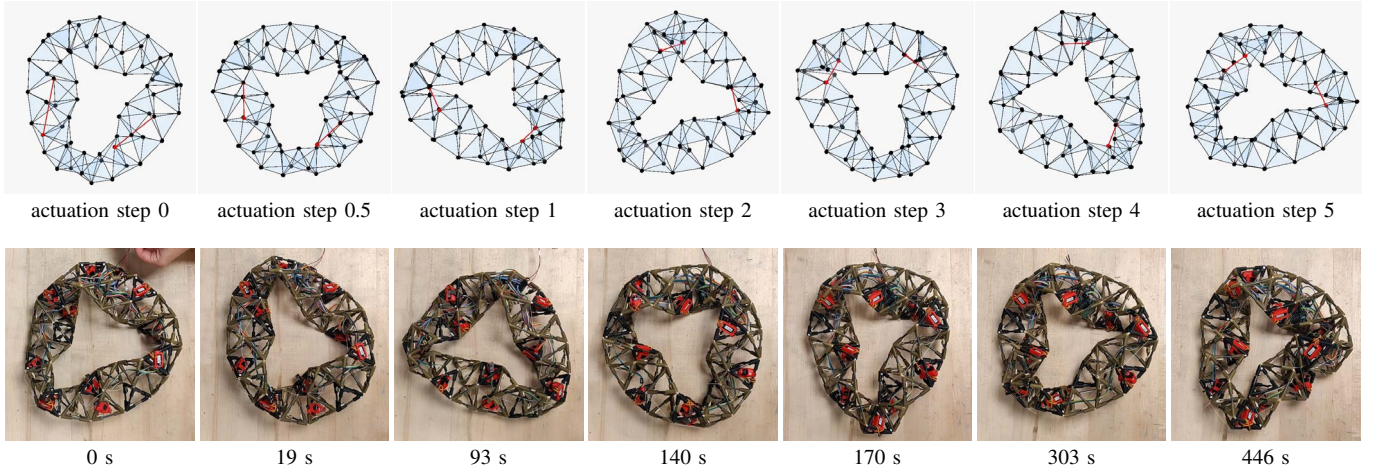


Fig. 8. Top: Snapshots of the ring simulation over an actuation cycle. Frames from actuation steps 0 to 1 show the Trussbot’s deformation under the first pair of tendon contraction, actuation steps 1, 2, 3, 4, and 5 show the start of the second, third, fourth, fifth, and first pairs of tendon contraction, respectively. Bottom: Snapshots of the TrussBot over 4 actuation cycles. Frames from 0 s to 140 s show the TrussBot’s deformation over a single cycle. Frames at 170 s, 303 s, and 446 s show the resulting TrussBot at the end of cycles 2, 3, and 4, respectively. The TrussBot follows a periodic configuration change.

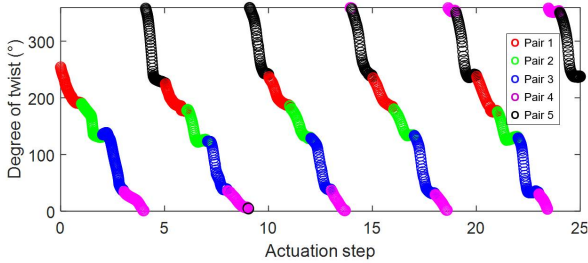


Fig. 9. The ring’s twist angle over 5 cycles of actuation. Each pair of tendons takes 1 actuation step to contract.

C. Ring

Finally, TrussBot can be configured as a ring (Fig. 4), which allows the truss to twist in on itself. The ring configuration is actuated using five pairs of actuators, as labeled in Fig. 4, which were designed by manually manipulating the simulation and physical prototype. The ring forms a triangular shape, and each tendon pair locates at the two triangular sides before contracting. The tendon pairs actuate in sequence, with each pair releasing as the next one contracts.

We simulated this design and actuation sequence and tracked it over 5 cycles of the control pattern, or 1200 time steps. The simulation took 196.7 s. Figure 8 shows snapshots of the truss over time, as compared to the physical model. The motion shows that the robot is able to twist inside out using the designed actuation pattern. Figure 9 shows the simulated twist angle of the ring, computed by tracking a single edge’s orientation with respect to the ground plane over the course of the simulation. The degree of twist per cycle is consistent, with the edge rotating a full 360 degrees per cycle. That is, the simulated TrussBot turns inside out after one cycle of the control pattern. During the course of the simulation the centroid showed no translation or net rotation on the ground. The experimental hardware matched these

predictions. It was able to replicate the twisting motion using the 5 tendon pairs, with each full actuation cycle taking 170 s.

VI. CONCLUSION

This paper describes a modular tetrahedral truss with compliant joints, TrussBot. We present an analysis of the truss’s main deformation modes and a simulation of its motion under tendon-driven control. We additionally demonstrate a hardware prototype of the TrussBot, which is easy to assemble and reconfigure into trusses of different topology. The physical platform, when compared to simulations, was able to execute the simulated control strategies and demonstrate the same predicted motions.

There are many potential directions for future work on this project. The unique geometry and movement of the TrussBot lend it to a variety of applications as a gripper, arm, crawler, and climber. The twisting motion of the ring, for example, would allow it to climb pipes without requiring continuous rotation joints at any of the hinges. Previous iterations of pole climbing robots make use of rollers [31], [32], pneumatic cylinders [18], [33], or grippers [34] to facilitate climbing. One of the few existing modular pole climbing systems is the Climbot [34], which is composed of two types of joint modules and one gripper module. The TrussBot’s actuation method is simpler compared to these existing systems. The modularity of the TrussBot also means that it can be easily reconfigured to interact with many other geometries of objects. Future work includes scaling the TrussBot, both in the number of modules per TrussBot and the sizes of modules, and testing it on a climbing task. We are also interested in discovering how varying the size and shape will affect the actuation patterns and control strategies, particularly in cases such as gripper where free unactuated degrees of freedom exist. Another area of future study is how the computational model can be used to design and optimize the actuation pattern for future applications.

REFERENCES

- [1] 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 Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.
- [2] W. Gao, K. Huo, J. S. Seehra, K. Ramani, and R. J. Cipra, "Hexamorph: A reconfigurable and foldable hexapod robot inspired by origami," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4598–4604.
- [3] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 4, pp. 431–441, 2003.
- [4] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the ATRON lattice-based self-reconfigurable robot," *Autonomous robots*, vol. 21, no. 2, pp. 165–183, 2006.
- [5] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu, "Miche: Modular shape formation by self-disassembly," *International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 345–372, 2008.
- [6] P. J. White, C. E. Thorne, and M. Yim, "Right Angle Tetrahedron Chain Externally-actuated Testbed (RATChET): A Shape Changing System," in *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, 2009.
- [7] J. W. Romanishin, J. Mamish, and D. Rus, "Decentralized control for 3d m-blocks for path following, line formation, and light gradient aggregation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4862–4868.
- [8] M. Yim, Y. Zhang, J. Lamping, and E. Mao, "Distributed control for 3D metamorphosis," *Autonomous Robots*, vol. 10, no. 1, pp. 41–56, 2001.
- [9] S. M. Motahari-Bidgoli, M. J. Mahjoob, and S. Davaria, "Simulation and analysis of a TET-walker robot motion," in *RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, 2014, pp. 914–919.
- [10] J. Wang, Y. Fei, and Z. Liu, "Locomotion modeling of a triangular closed-chain soft rolling robot," *Mechatronics*, vol. 57, pp. 150–163, 02 2019.
- [11] R. Liu, Y. Yao, and Y. Li, "Design and analysis of a deployable tetrahedron-based mobile robot constructed by Sarrus linkages," *Mechanism and Machine Theory*, vol. 152, p. 103964, 06 2020.
- [12] M. D. Rhodes and M. M. Mikulas, *Deployable controllable geometry truss beam*. National Aeronautics and Space Administration, Scientific and Technical Information Branch, 1985.
- [13] H. Furuya and K. Higashiyama, "Dynamics of closed linked variable geometry truss manipulators," *Acta astronautica*, vol. 36, no. 5, pp. 251–259, 1995.
- [14] A. Spinos, D. Carroll, T. Kientz, and M. Yim, "Variable topology truss: Design and analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2717–2722.
- [15] G. J. Hamlin and A. C. Sanderson, *Tetrobot*, ser. Springer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1998, vol. 423.
- [16] N. S. Usevitch, Z. M. Hammond, M. Schwager, A. M. Okamura, E. W. Hawkes, and S. Follmer, "An untethered isoperimetric soft robot," *Science Robotics*, vol. 5, no. 40, 2020.
- [17] M. A. Karimi, V. Alizadehyazdi, B. Busque, H. M. Jaeger, and M. Spenko, "A boundary-constrained swarm robot with granular jamming," in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 291–296.
- [18] Z. M. Ripin, B. S. Tan, A. B. Abdullah, and Z. Samad, "Development of a low-cost modular pole climbing robot," in *TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium*, vol. 1, 2000, pp. 196–200.
- [19] S. Avinash, A. Srivastava, A. Purohit, S. V. Shah, and K. M. Krishna, "A compliant multi-module robot for climbing big step-like obstacles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3397–3402.
- [20] A. Bhole, S. H. Turlapati, R. V. S., J. Dixit, S. V. Shah, and K. M. Krishna, "Design of a robust stair-climbing compliant modular robot to tackle overhang on stairs," *Robotica*, vol. 37, no. 3, p. 428–444, 2019.
- [21] N. A. Mansour, T. Jang, H. Baek, B. Shin, B. Ryu, and Y. Kim, "Compliant closed-chain rolling robot using modular unidirectional sma actuators," *Sensors and actuators. A. Physical.*, vol. 310, p. 112024, 2020.
- [22] J. Sastra, S. Chitta, and M. Yim, "Dynamic rolling for a modular loop robot," *International Journal of Robotics Research*, vol. 28, no. 6, pp. 758–773, 2009.
- [23] G. Hao and X. Kong, "Design and Modeling of a Large-Range Modular XYZ Compliant Parallel Manipulator Using Identical Spatial Modules," *Journal of Mechanisms and Robotics*, vol. 4, no. 2, 04 2012, 021009.
- [24] Y. Seguchi, M. Tanak, T. Yamaguchi, Y. Sasabe, and H. Tsuji, "Dynamic analysis of a truss-type flexible robot arm," *JSME international journal. Ser. 3, Vibration, control engineering, engineering for industry*, vol. 33, no. 2, pp. 183–190, 1990.
- [25] P. C. Hughes, W. G. Sincarsin, and K. A. Carroll, "Trussarm—a variable-geometry-truss manipulator," *Journal of Intelligent Material Systems and Structures*, vol. 2, no. 2, pp. 148–160, 1991.
- [26] K. Miura, H. Furuya, and K. Suzuki, "Variable geometry truss and its application to deployable truss and space crane arm," *Acta astronautica*, vol. 12, no. 7, pp. 599–607, 1985.
- [27] M. S. Moses, M. K. Ackerman, and G. S. Chirikjian, "Origami rotors: Imparting continuous rotation to a moving platform using compliant flexure hinges," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55942, 2013, p. V06BT07A037.
- [28] J. E. Graver, "Rigidity matroids," *SIAM Journal on Discrete Mathematics*, vol. 4, pp. 355–368, 1991.
- [29] A. Kassimali, *Matrix Analysis of Structures*. Cengage Learning, 2011.
- [30] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [31] A. Baghani, M. N. Ahmadabadi, and A. Harati, "Kinematics modeling of a wheel-based pole climbing robot (UT-PCR)," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 2099–2104.
- [32] A. Sadeghi, H. Moradi, and M. N. Ahmadabadi, "Analysis, simulation, and implementation of a human-inspired pole climbing robot," *Robotica*, vol. 30, no. 2, p. 279–287, 2012.
- [33] M. Das, A. Agrawal, A. Sonone, R. Gupta, D. Upadhyay, Y. Rao, and A. Javed, "Developing a bioinspired pole climbing robot," in *International Conference on Robotics: Current Trends and Future Challenges (RCTFC)*, 2016.
- [34] Y. Guan, L. Jiang, H. Zhu, W. Wu, X. Zhou, H. Zhang, and X. Zhang, "Climbot: A Bio-Inspired Modular Biped Climbing Robot—System Development, Climbing Gaits, and Experiments," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, 01 2016, 021026.