SOFT ROBOTICS
Volume 00, Number 00, 2020

DOI: 10.1089/soro.2019.0142

SoRo
Soft Robotics

# Surface Actuation and Sensing of a Tensegrity Structure Using Robotic Skins

Joran W. Booth,* Olivier Cyr-Choinière,* Jennifer C. Case,* Dylan Shah,*
Michelle C. Yuen, and Rebecca Kramer-Bottiglio



## Abstract

Tensegrity robots comprising solid rods connected by tensile cables are of interest due to their flexible and robust nature, which potentially makes them suitable for uneven and unpredictable environments where traditional robots often struggle. Much progress has been made toward attaining locomotion with tensegrity robots. However, measuring the shape of a dynamic tensegrity without the use of external hardware remains a challenge. Here we show how robotic skins may be attached around the exterior of a tensegrity structure, to both control and measure its shape from its surface. The robotic skins are planar, skin-like membranes with integrated actuators and sensors, which we use to transform a passive tensegrity structure into an active tensegrity robot that performs tasks such as locomotion. In addition, sensors placed on the ends of the tensegrity rods are used to directly measure orientation relative to the ground. The hardware and algorithms presented herein thus provide a platform for surface-driven actuation and intrinsic state estimation of tensegrity structures, which we hope will enable future tensegrity robots to execute precise closed-loop motions in real-world environments.

**Keywords:** tensegrity robot, state reconstruction, robotic skins

## Objective

Soft robots offer functional advantages such as the potential for safe human/robot interactions and robustness to falls, impacts, and vibrations. One avenue of exploration in soft robotics is to leverage the advantages of tensegrity structures, a concept originally used to create lightweight architectural elements.[1,2] A tensegrity structure consists of rigid compression elements (which we call rods) that are linked by tensile elements (e.g., cables, strings). These networks form lightweight,[3] compliant, and load-bearing structures[4] due to the equilibrium between compression and tensile forces. The concept of tensegrities has also been applied to explain biological phenomena[5,6] and to create novel rolling robots.[7]

Several locomoting tensegrity robot designs have been proposed, spanning a range of rod connectivity patterns, actuation mechanisms, and locomotion strategies. Tensegrity structures are typically described by the number of rods used in the arrangement, and common arrangements include 3-, 6-, 10-, and 12-bar tensegrities. While most proposed tensegrity designs borrow from well-established arrangements, designing new arrangements of rods and tensile elements for tensegrity robots is not straightforward, leading some researchers to leverage machine-learning techniques for robot design.[8,9] In this work, we focus on the six-bar tensegrity structure, which has a rest configuration of an irregular icosahedron,[10–15] whose faces consist of 8 equilateral triangles and 12 isosceles triangles (as shown in Fig. 1).

Common locomotion strategies generally fall into two classes: rolling and vibration. In the rolling approach, adjustable rods or tension elements are used to shift the center of mass past the polygon of stability, which is also the downward face, and tip the robot from face to face.[16–25] Most of these proposed designs use cables[24] or pneumatic actuators.[10,26,27] The vibration-driven locomotion strategy exploits the tensegrities' dynamic asymmetries to produce translational motion.[16,21,28–32]. In one unique approach that does not fit neatly into these two common categories, a tensegrity was placed as a cage around a two-propeller drone, creating a robot that could hop over obstacles, roll upon landing, and right itself before hopping again.[23] Finally, we note that most of the gaits used in these proposed designs were selected by a human designer;

School of Engineering and Applied Science, Yale University, New Haven, Connecticut, USA.

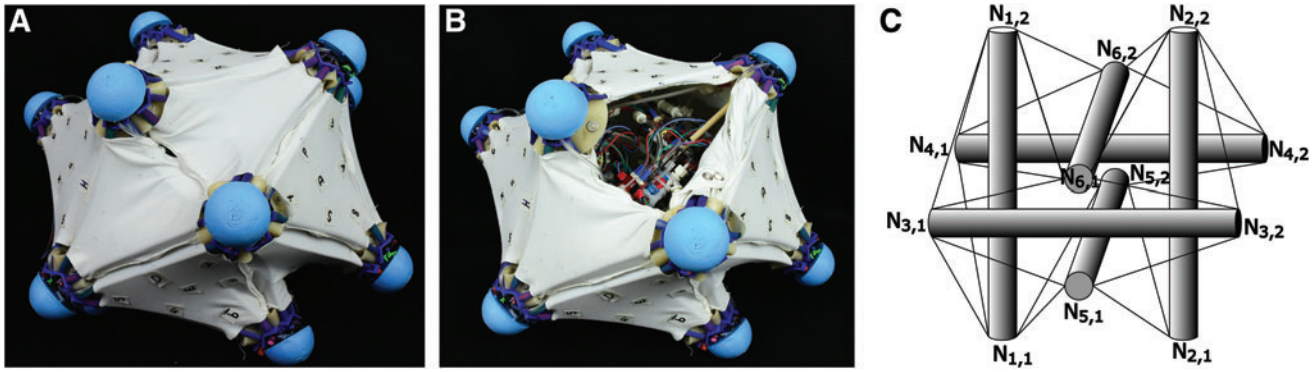


**FIG. 1.** **(A)** Six-bar tensegrity covered by robotic skins. **(B)** Interior view of the robot, showing the onboard pneumatic valves. **(C)** Node mapping for use in the state reconstruction algorithm. Sensors (*thin black lines*) measure the distances between nodes (*extremities of bars*). Color images are available online.



however, several studies demonstrated the possibility of learning locomotion strategies using machine-learning techniques, often paired with a robot simulator.[33–36]

While prior work has led to many successful designs and locomotion strategies, real-time state reconstruction of tensegrity robots has remained difficult to achieve without use of external sensors. For example, NASA's ''Super Ball Bot'' tensegrity robot is actuated by adjusting the length of its tensile elements with a motor and winch.[13,37] To accurately estimate the spatial distance between two nodes in that design, the changing length of the tensile elements between rods must be measured, and there are few sensors that can easily measure such extreme deformations. Measuring the amount the motor has turned is perhaps the most direct measurement, but does not account for stretch in the cables or deformation of the bars. ''Time of flight of light'' schemes have been proposed as another alternative; however, these approaches require external sensors, markers, or other points that can be used as base stations to improve the resolution of the system, limiting their applicability in unstructured environments.[38] Another proposed design used a single-strain sensor woven through a modular, spine-like tensegrity robot for detecting the bending states of the entire system, but no attempt was made at shape or orientation reconstruction.[39]

In this article, we present a six-bar tensegrity robot capable of rolling locomotion that is able to achieve both shape and orientation reconstruction without the use of external sensors. Our tensegrity uses robotic skins[27] to provide actuation and strain sensing capabilities between its nodes, and uses contact sensors placed at the ends of each rod to detect direct contact with the environment. In contrast to prior pneumatic tensegrity structures, we also placed the pneumatic control hardware onboard the robot (Fig. 1b). Our state reconstruction algorithms solve an optimization problem given the constraints imposed by the connectivity of our 6-bar tensegrity, shown in Figure 1c.

In the following sections, we describe the materials and methods used (Materials and Methods section), including the general principle of the robotic skins (Robotic Skins section). Next, we describe the state reconstruction model (State Reconstruction Model section) using the embedded strain sensors and node contact sensors to estimate node position and face orientation, and show validation of the model (Model Validation section). Finally, we present teleoperated rolling of the tensegrity robot on a declined plane (Rolling section).

## Materials and Methods

Our six-bar icosahedron tensegrity robot is primarily made of six wooden bars (9.5 mm diameter, 350 mm long, McMaster-Carr) and 20 triangular fabric-based robotic skins. The robotic skins include McKibben-type pneumatic actuators and capacitive strain sensors (Fig. 2a), which are positioned along the edges (Fig. 2b). While all 20 faces on the tensegrity include 3 actuators along each edge, only the 8 equilateral faces are equipped with strain sensors. We made this choice because the edges of the isosceles faces are either shared with an equilateral face or always slack and unable to yield a meaningful sensor value. In the fully assembled system, the diameter is 410 mm (node to node, e.g., between $N_{5,1}$ and $N_{6,1}$) and the entire system has a mass of 2.55 kg.

In this section, we describe each of the elements comprising the robotic skins and the rest of the robot.

### *Robotic skins*

Our previous work introduced the concept of robotic skins, which are planar substrates with embedded actuation and sensing that can be attached to the surface of a deformable body to impart controlled motion onto that body.[27] This capability enables surface-driven motion and state estimation of tensegrity robots. The robotic skins demonstrated in this article are built on a substrate of two layers of spandex fabric cut into a truncated equilateral triangle (long sides, 120 mm; short sides, 50 mm), and sewn together along the long edges to form a pocket that houses the other components of the robotic skin (Fig. 2c, d). Muslin fabric pieces are sewn into the vertices to provide a location to anchor the components and provide a place to attach the skin externally. Several anchor points are provided on the muslin fabric using snap fasteners, allowing the modular actuators and strain sensors to be exchanged. In addition, the fabric snaps allow each skin to independently attach to the wooden support bars and to each other, as discussed in the Supplementary Data. The modular actuators and sensors are attached in a triangular pattern inside the fabric skin. In this triangular arrangement, the strain sensor readings are closely coupled to the true edge length, simplifying the system modeling.

Pneumatic actuators. The pneumatic actuators in the robotic skins are McKibben-type pneumatic artificial

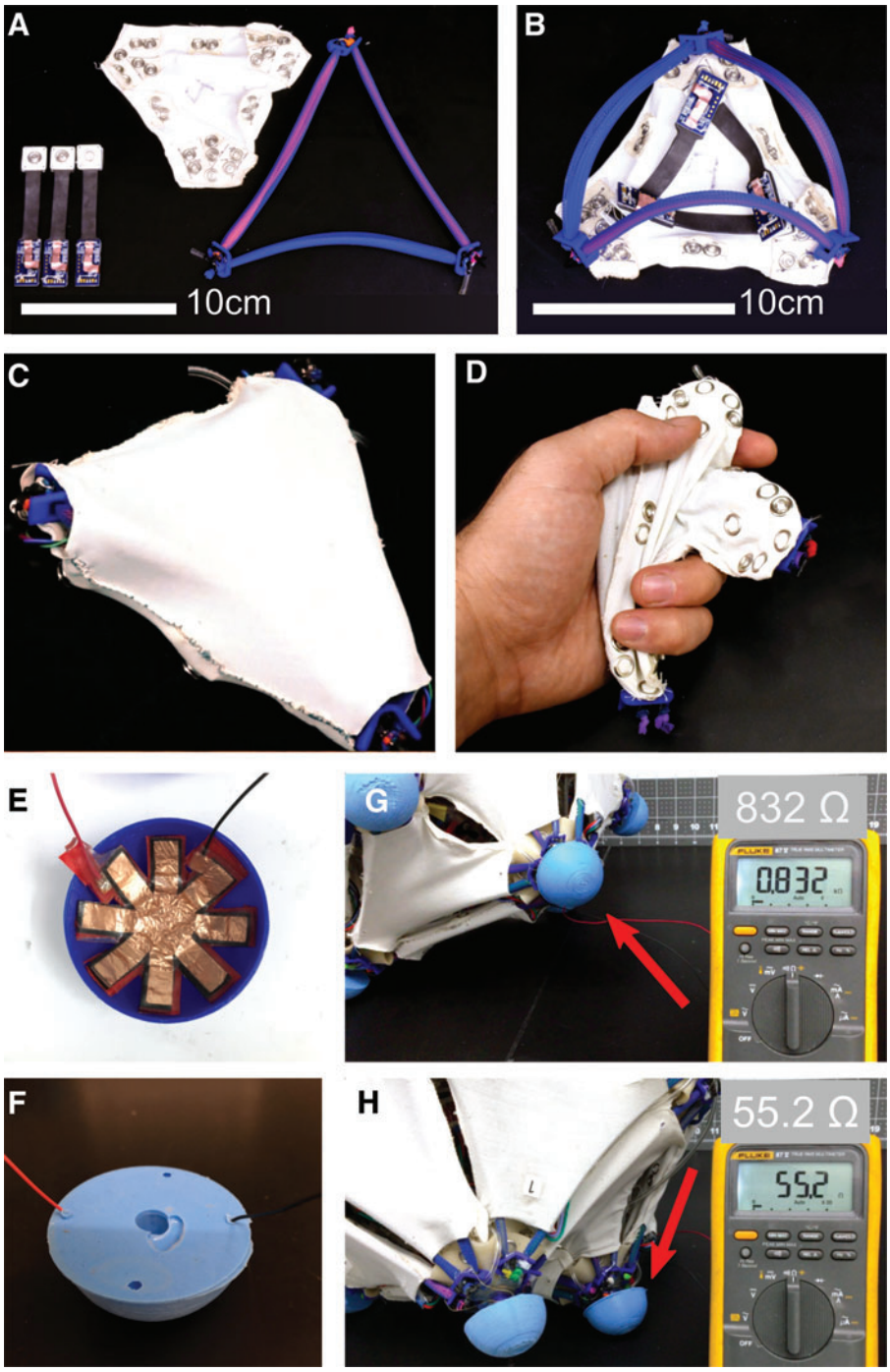


**FIG. 2.** **(A)** Main components of a robotic skin: fabric, modular sensors, and modular actuators. **(B)** Robotic skin with modular pieces attached to the fabric using fabric snaps. In this photograph, the fabric envelope is turned inside-out to expose the connection points. **(C)** Assembled robotic skin. **(D)** Hand grasping a robotic skin to demonstrate the skin's compliant and flexible nature, even with the modular components inside. **(E)** Sensing layers of the node contact sensor are placed inside the mold before casting. **(F)** Completed node contact sensor. **(G, H)** Node contact sensor (*arrow*) making contact with the ground exhibiting a lower resistance (multimeter). Color images are available online.



muscles,[40] built using similar methods as in our prior work.[27] The active length of each actuator is 150 mm. The actuators are attached to the robotic skins with a 3D-printed polylactic acid (PLA) bracket that has an integrated snap fastener. An external tether supplies air at 172 kPa (25 psi), which results in a contraction of 40 mm. Complete inflation is achieved in 0.5 s, while deflation takes 0.8 s. The longer deflation time is expected, since the flow rate during inflation is driven by a constant pressure supply, whereas the deflation is driven by a changing pressure gradient dependent on the elastic energy stored in the actuator and the corresponding compression of the gas inside the actuator.

Although the tensegrity robot used 20 robotic skins, not all edges need to be active. In its equilibrium irregular-icosahedron shape, the robot's 12 isosceles faces are those that share two nodes with the endpoints of each pair of parallel bars, with the short side of the triangle positioned between the ends of the parallel bars (e.g., the face connecting $N_{3,1}$ with $N_{6,1}$ and $N_{5,1}$, later named $F_Q$ in Table 2). In our system, the 20 robotic skins are all equilateral at rest, which creates slack between the ends of parallel bars and reduces the performance of those sensors and actuators. Therefore, out of 60 potential edges for actuator and sensor placement, 12 actuators and 12 sensors are located where they cannot be

used and only 48 can be controlled and queried. In addition, we simplify the pneumatic hardware by plumbing adjacent actuator pairs to the same pneumatic regulator. This reduces the number of required regulators from 48 to 24, without sacrificing output force. Each actuator pair was independently controlled by a pneumatic regulator circuit board,[41] which was mounted on the rods (Fig. 1b). The advantage of mounting the pressure regulators inside the robot is that we can operate the robot with a single pneumatic tether, rather than several tethers as shown in prior pneumatic tensegrity robots,[10,26] and is an important step toward creating an untethered pneumatic tensegrity robot.

Strain sensors. The robotic skins are equipped with conductive composite-based capacitive strain sensors and appropriate signal conditioning boards.[42,43] The ends of the sensors are sewn to polystyrene tabs with integrated snap buttons to facilitate attachment to the tensegrity's nodes. The sensor outputs are transmitted to a microcontroller (Arduino Uno) using the $I^2C$ protocol and passed through a digital exponential filter. Since the sensor data are only meaningful if the sensors are taut, the sensors are prestrained so that they are in tension when their associated actuator is fully contracted. The unstretched active length of the sensors is 52 mm, while the at-rest length of a sensor while on the tensegrity is roughly 90 mm. As mentioned in the Materials and Methods section, the system does not use all the pneumatic actuators to operate. The same is true for strain sensors. Sensors are only installed on the equilateral triangle robotic skins, which results in 24 active strain sensors.

### *Node contact sensors*

Twelve node contact sensors are installed on the ends of the six bars in the tensegrity robot. Each node contact sensor is made from Smooth-Sil 950 silicone (Smooth-On, Inc.) in a 3D-printed hemisphere mold with a radius of 29 mm. The radius of the hemisphere is chosen to ensure that the tensegrity will only contact the ground with the node contact sensors and not with other hardware. The elastomer provides compliance and impact absorption. The sensing element consists of a pressure-sensitive conductive polyolefin film (Velostat, Adafruit 1361) sandwiched between adhesive-backed copper foil electrodes. This sensing element is cut into the shape of a truncated octagram, sealed between two layers of double-sided VHB tape (0.5 mm thick, 3M 4905) and curled to fit the contour of the hemisphere (Fig. 2e). Connecting wires were secured to the copper electrodes. The sensing element was then embedded in the silicone during the casting process, ensuring that the wires protruded from the molded material to allow for connection (Fig. 2f).

The node contact sensors decrease electrical resistance when pressure is applied (Fig. 2g, h). A threshold resistance value is chosen after calibration to determine whether the node is in contact with something or not. Each finished node contact sensor is connected to one of two onboard microcontrollers (Arduino Pro Mini; Arduino AG) via a voltage divider.

### *Electronics*

The electronics system architecture consists of a microcontroller, strain sensor signal conditioning boards, and pneumatic regulator boards. The microcontroller communicates with the strain sensors and pneumatic regulators via the $I^2C$ protocol, and relays data to a PC via serial. Two control methods are implemented: open-loop control with a preprogrammed actuator sequence and teleoperated control of each individual actuator.

## Results

The primary result of this work is the demonstration that robotic skins can be used to actuate, sense, and control a tensegrity from its surface. The strain sensors in the robotic skins can be used to reconstruct the state of the robot, including both the spatial location of each node and the orientation of the tensegrity with respect to the ground. In this section, we introduce a state reconstruction model that is enabled by our sensor platform. We describe the validation of our state reconstruction model and demonstrate the use of the pneumatic actuators on the robotic skins to roll the entire system.

### *State reconstruction model*

Our sensor-driven state reconstruction approach is unique from prior approaches for six-bar tensegrity robots, which rely on physics models or machine vision or similar external sensors.[11,12,15,44] These physics-based models are excellent for designing a new tensegrity system and balancing the required tension forces within the system. However, these proposed physics-based models are not well suited for state reconstruction since they rely on known spring forces for system components, tension on the cables, and torques on the bars, which are difficult to measure. In contrast, our strain sensor approach takes direct measurements that can be correlated with distances between nodes, and enables state reconstruction of the tensegrity robot without any knowledge of the forces in the system.

The state of the six-bar tensegrity can be described using two mathematical objects: (1) a matrix of node positions in a locally defined coordinate system, which gives the shape of the system, and (2) a single number representing the face in contact with the ground, which gives a global orientation of the system. While this orientation system does not fully define the system in Cartesian coordinates, it does provide the system with the essential components to allow future tensegrity robots to plan their motion policies and track past motions to continue locomotion. The state reconstruction can run in real time on a computer, using the algorithm summarized in Algorithm 1. In brief, the sensor readings ($d_k$) are fed to a neural net, which estimates the face pointing down to the ground ($F_d$). The sensor readings are also fed to an algorithm that estimates the node positions ($N$) by minimizing the error between estimated positions and the sensor readings, while satisfying the constraints imposed by the rods. In addition to a neural net, we also demonstrate direct detection of the downward face using node contact sensors, which identify the downward face using a lookup table. Finally, the node positions are rotated according to the face estimation, to yield the tensegrity's state relative to the surface it is resting upon.

**Algorithm 1:** Real-time State Reconstruction

**while** *true* **do**
$\{d_k\} =$ MeasureStrainSensors();
$\{d_l\} =$ MeasureNodeContactSensors();
$F_d =$ FaceEstimation $(d_l)$;
$N =$ NodePositions $(d_k, N_{i-1})$
$N_{rotated,i} =$ RotationMatrix $(F_d, N)$;

Node position estimation. Node positions are estimated by feeding strain sensor readings to a least-squares error minimization algorithm. Let $N_{i,j} = \{x_{i,j}, y_{i,j}, z_{i,j}\}^T$ represent the node position for the $i$th bar and $j$th node on the bar (where $i = 1, 2, .., 6$ and $j = 1, 2$, see Fig. 1c) and $L$ represent the length of the rigid bars. Without loss of generality, we initially define the position of node $N_{1,1}$ as the local origin (i.e., $N_{1,1} = \{0, 0, 0\}^T$). The location of the other 11 nodes is found with respect to this node using an optimization algorithm. The node positions in subsequent iterations are calculated from the prior iteration. Our optimization seeks to minimize the error between the simulated edge lengths and the measured sensor lengths ($d_k$) and is constrained by the rigid bars, which cannot change in length.

$$\begin{aligned} \underset{N_{i,j}}{\text{minimize}} \quad & \textstyle\sum_{k=1}^{24} (\parallel N_{m_1,m_2} - N_{n_1,n_2} \parallel - d_k)^2 \\ \text{s.t.} \quad & \parallel N_{i,1} - N_{i,2} \parallel - L = 0, \; i = 1, .., 6 \end{aligned} \tag{1}$$

where the nodes $N_{m_1,m_2}$ and $N_{n_1,n_2}$ are found for each sensor identifier ($s_k$) according to Table 1. Upon initialization, the nonlinear optimization routine (sequential least-squares quadratic programming) is seeded with node positions, which approximates an equivalently sized regular icosahedron. During optimization, the prior reconstructed state acts as a seed for the next iteration.

Orientation estimation. Determination of the face in contact with the environment (ground, obstacles, *etc.*) is critical for motion planning of tensegrity structures. In this work, we use the strain sensor readings in a neural network to determine the downward face and we use node contact sensors to directly detect the downward face. While an inertial measurement unit can detect the direction of gravity, it cannot be used to also determine shape, and thus, to determine shape. Here we show how the strain sensors can be used to detect both shape and orientation with respect to ground. In our demonstration of the neural network, we pause for the tensegrity to stop moving to detect the face, which significantly slows the potential speed of the system. Therefore, we also demonstrate node contact sensors that can quickly and directly detect contact with the environment. However, direct contact sensing is susceptible to sensor failure. Regardless of the method used, the downward face $F_d$ can be described using three nodes (Fig. 1) as

$$F_d = \{N_{l_1,l_2}, N_{m_1,m_2}, N_{n_1,n_2}\} \tag{2}$$

where $F_d$ describes the face according to Table 2.

(a) *Orientation Estimation Using the Strain Sensors:* Our first approach to detecting the orientation of the tensegrity robot uses the robotic skin-embedded strain sensors and relies on a model-free simulation.[45,46] While this approach has the disadvantage of needing to be trained on every individual robot, it offers flexibility if the robot is damaged, since the model-free approach can be retrained during a mission.[47]

When all actuators are at rest, the tensegrity structure is stretched more toward the top than the bottom of the structure due to gravity. The strain sensors embedded in the robotic skins can detect the stretching and we can feed this information to supervised machine-learning algorithms, including regression and neural networks, to model the direction of gravity or even directly predict the downward face. It is beyond the scope of this work to compare different machine-learning algorithms, and we instead focus on this simple implementation to demonstrate that it is possible to use surface-based sensing for orientation estimation.

We created a simple neural network to estimate the downward face using strain sensor data only. To train the network, we collected 60s of strain sensor data for each of the 20 tensegrity faces. We then trained a neural network in MATLAB using the Low-Memory Broyden-Fletcher-Goldfarb-Shanno algorithm. The neural network is built with an input layer consisting of 24 sensor input nodes, 1 for each sensor, a single hidden layer consisting of 10 nodes, and an output layer consisting of 20 nodes, 1 for each face. The algorithm returns the face with the highest probability of being downward. One limitation of this neural network is that it is trained when the system is at rest, meaning that the tensegrity robot must come to a complete stop and relax all actuators before it can correctly estimate the downward face. With a more sophisticated network, such as a recurrent neural network, it would be possible to train the system to estimate the downward face during dynamic motions. Despite these limitations, this current pairing of a simple neural network with real-time strain sensors allows the system to estimate its orientation without any dedicated orientation sensors.

In future work, the neural network could also be improved by including the actuator pressures in the input layer. By including the actuator pressures and expanding the number of conditions for which the neural network data were

TABLE 1. MAPPING FROM SENSORS $s_k$ TO NODES $N_{i,j}$

| $s_k$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ | $s_k$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ | $s_k$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ | $s_k$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | $N_{1,1}$ | $N_{3,1}$ | $s_7$ | $N_{1,2}$ | $N_{6,1}$ | $s_{13}$ | $N_{2,2}$ | $N_{3,2}$ | $s_{19}$ | $N_{3,2}$ | $N_{5,1}$ |
| $s_2$ | $N_{1,1}$ | $N_{4,1}$ | $s_8$ | $N_{1,2}$ | $N_{6,2}$ | $s_{14}$ | $N_{2,2}$ | $N_{4,2}$ | $s_{20}$ | $N_{3,2}$ | $N_{6,1}$ |
| $s_3$ | $N_{1,1}$ | $N_{5,1}$ | $s_9$ | $N_{2,1}$ | $N_{3,2}$ | $s_{15}$ | $N_{2,2}$ | $N_{6,1}$ | $s_{21}$ | $N_{4,1}$ | $N_{5,2}$ |
| $s_4$ | $N_{1,1}$ | $N_{5,2}$ | $s_{10}$ | $N_{2,1}$ | $N_{4,2}$ | $s_{16}$ | $N_{2,2}$ | $N_{6,2}$ | $s_{22}$ | $N_{4,1}$ | $N_{6,2}$ |
| $s_5$ | $N_{1,2}$ | $N_{3,1}$ | $s_{11}$ | $N_{2,1}$ | $N_{5,1}$ | $s_{17}$ | $N_{3,1}$ | $N_{5,1}$ | $s_{23}$ | $N_{4,2}$ | $N_{5,2}$ |
| $s_6$ | $N_{1,2}$ | $N_{4,1}$ | $s_{12}$ | $N_{2,1}$ | $N_{5,2}$ | $s_{18}$ | $N_{3,1}$ | $N_{6,1}$ | $s_{24}$ | $N_{4,2}$ | $N_{6,2}$ |

TABLE 2. MAPPING FROM FACES $F_\alpha$ TO NODES $N_{i,j}$

| $F_\alpha$ | $N_{l_1,l_2}$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ | $F_\alpha$ | $N_{l_1,l_2}$ | $N_{m_1,m_2}$ | $N_{n_1,n_2}$ |
|---|---|---|---|---|---|---|---|
| $F_A$ | $N_{1,1}$ | $N_{3,1}$ | $N_{4,1}$ | $F_K$ | $N_{1,2}$ | $N_{2,2}$ | $N_{6,1}$ |
| $F_B$ | $N_{1,1}$ | $N_{3,1}$ | $N_{5,1}$ | $F_L$ | $N_{1,2}$ | $N_{2,2}$ | $N_{6,2}$ |
| $F_C$ | $N_{1,1}$ | $N_{2,1}$ | $N_{5,1}$ | $F_M$ | $N_{2,2}$ | $N_{3,2}$ | $N_{6,1}$ |
| $F_D$ | $N_{1,1}$ | $N_{4,1}$ | $N_{5,2}$ | $F_N$ | $N_{2,2}$ | $N_{4,2}$ | $N_{6,2}$ |
| $F_E$ | $N_{1,1}$ | $N_{4,1}$ | $N_{5,2}$ | $F_O$ | $N_{1,2}$ | $N_{4,1}$ | $N_{6,2}$ |
| $F_F$ | $N_{2,1}$ | $N_{5,1}$ | $N_{3,2}$ | $F_P$ | $N_{2,2}$ | $N_{3,2}$ | $N_{4,2}$ |
| $F_G$ | $N_{2,1}$ | $N_{3,2}$ | $N_{4,2}$ | $F_Q$ | $N_{3,1}$ | $N_{5,1}$ | $N_{6,1}$ |
| $F_H$ | $N_{2,1}$ | $N_{4,2}$ | $N_{5,2}$ | $F_R$ | $N_{3,2}$ | $N_{5,1}$ | $N_{6,1}$ |
| $F_I$ | $N_{1,2}$ | $N_{3,1}$ | $N_{4,1}$ | $F_S$ | $N_{4,1}$ | $N_{5,2}$ | $N_{6,2}$ |
| $F_J$ | $N_{1,2}$ | $N_{3,1}$ | $N_{6,1}$ | $F_T$ | $N_{4,2}$ | $N_{5,2}$ | $N_{6,2}$ |

trained, the future system would likely be capable of detecting the downward face even when actuators are active. If the downward face can be detected when actuators are active, that would reduce the sensors' settling time for the face detection neural network, since no waiting time would be needed for the actuators to deflate before reading reliable sensor values. However, we would expect that measurements would only be accurate after the system stopped oscillating after a roll.

(b) *Orientation Estimation Using the Node Contact Sensors:* The node contact sensors detect which tensegrity nodes are in contact with the ground, allowing for direct detection of contact without the need to pause between each locomotion step, as was the case with the current neural net that used the strain sensors. When three nodes are in contact, we infer that the corresponding face is down, and by tracking history, there is the potential for determining whether a node is touching the ground or a wall. The algorithm for detecting the downward face in this work is to return the downward face from a lookup table using the last known valid triad of nodes. In other words, if more or fewer than three nodes are in contact, the previously predicted face is retained until a new triad of nodes is found.

The node contact sensors are not able to detect the orientation of gravity directly, and so, determination of the downward face can only be inferred. On the contrary, the direct sensing of contact allows for continuous prediction of not only which face is contacting the ground but also the edge over which the tensegrity is tipping and contacts with walls. In the future, the node contact sensors and strain sensors could be used jointly to detect anomalies in terrain, such as coming into contact with a wall.

### *Model validation*

Strain sensor calibration. In the physical tensegrity robot, the distance between the nodes ($l_k$) is not the same as the active length of the sensors ($d_k$), due to the way that the sensors are attached to the structure. There are passive connections between the strain sensors and the node end caps ($d_\eta$), thus requiring calibration. To calibrate the system, we affixed one LED motion capture marker per node ($N_{i,j}$) on both extremities of a given edge ($l_k$), centered the tensegrity robot in the field of view of the motion capture system (Improv, PhaseSpace Inc.), and measured the distance and sensor values at five different actuated lengths. The above procedure was repeated for each of the 24 sensors to extract a linear relationship to map from the sensor value to the edge length between nodes ($l_k = C_k * d_k + d_\eta$), where the sensor length is multiplied by a calibration constant, $C_k$. This calibration approach does not fully model the physical considerations in the system, since an active actuator may slightly displace a sensor leading to some nonlinearity, but we determined it was sufficiently accurate for our needs.

Validation of state reconstruction. To validate the model, we simultaneously measured the position of the nodes as estimated by the state reconstruction model and the position of the nodes as measured in a motion capture system (Fig. 3). We treated the motion capture data as ground truth. To collect the motion capture data, we affixed an LED motion capture marker on each node (the same way as in the Strain Sensor Calibration section), and ran a preprogrammed actuation sequence (Fig. 3a). Since not all the markers are simultaneously visible to the motion capture system, the analysis was conducted only for the nodes that could be directly verified. Figure 3b presents the reconstructed state using data from the sensors, which are partially represented in Figure 3d, and Figure 3c presents the spatial distribution of the motion capture markers superimposed onto the position of the nodes extracted by the state reconstruction model. The Supplementary Video S1 shows in real time the deformation and state reconstruction of the tensegrity as well as the strain sensor and motion capture data (four panels of Fig. 3). The LED markers are not shown in the photographs (Fig. 3a) to reduce the number of distracting elements in the image (see Supplementary Video S2 presenting the footage of the actuated sequence with the markers on the nodes).

To determine the accuracy of the model, we compared the motion capture data and the points estimated using the strain sensors and the node position estimation algorithm. Before that could occur, however, we needed to transform the coordinate systems of the motion capture data into the coordinate system of the state reconstruction. To calculate the spatial transformation, we solved the constrained Procrustes problem for aligning the positions of known reference points on the top face of the initial data frame, while minimizing the least-squares error.[48] The result of the least-squares algorithm is a transformation matrix ($T$) that can convert the coordinates of the motion capture data ($P_{\mathrm{MC}}$) into the coordinate system of the tensegrity state reconstruction ($P_t$), by left-multiplying the three markers of the top face in motion capture coordinates as $P_{\mathrm{SR}} = T * P_{\mathrm{MC}}$.

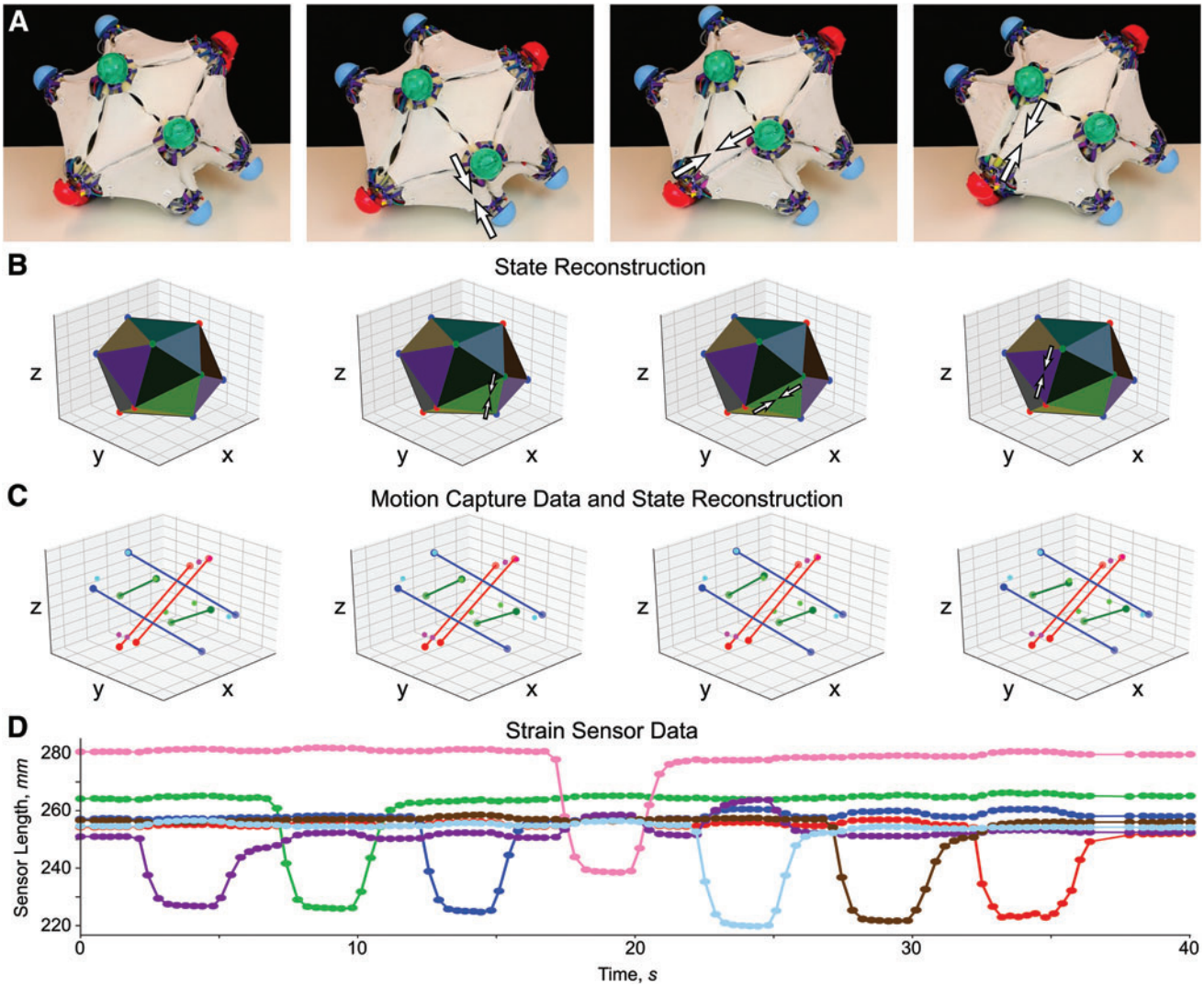


**FIG. 3.** **(A)** Tensegrity robot at different stages of a preprogrammed open-loop actuation sequence showing different segments actuated. **(B)** State reconstruction from the above sequence extracted from the strain sensors and node position estimation algorithm. **(C)** Comparison of the above state reconstruction (*darker dots* connected by *lines*) with data recorded simultaneously from a motion capture system (*lighter dots*) to validate the strain sensors' data and reconstruction model. Similar colors indicate corresponding nodes. Motion capture system was unable to simultaneously detect all the nodes, and thus, only the nodes that are visible are plotted. **(D)** Strain sensor data measured during the actuation sequence readily present the deformations of the system. For clarity, 7 of the 24 strain sensors corresponding to activated actuators are plotted. The full video sequence is available in Supplementary Video S1. Color images are available online.

We trained the rotation matrix on a single frame when the tensegrity was static instead of the entire data set to avoid overfitting, which is representative of initializing the tensegrity from an initial known configuration. Once the transformation matrix was calculated, the transformation was applied to the entire motion-capture data set and the root-mean-squared-error (RMSE) was calculated between all corresponding data points. The RMSE for the entire system is 45.8 mm, or 13.1% of the rod length. Note that a part of this error could come from the fact that not all markers were placed exactly at the center of their respective nodes in an effort to optimize an overall detection by the motion capture system.

Validation of the orientation estimation using the neural network. To validate the neural network, we measured the strain sensors of the robot in its stable state at each of the 20 faces down. The trained network has an accuracy of 99.8%, conducted on data collected in the same experimental session as the training data. As described above, this approach is limited to static unactuated states. In practice, we observe that the prediction stabilizes within a few seconds, after the system stops oscillating. To account for the 0.2% of incorrect predictions during operation, the algorithm returns the statistical mode of a moving window of the last five measurements in the data.

Validation of the face detection using the node contact sensors. To test the node contact sensors, the tensegrity was manually rotated while collection of video and sensor data (Fig. 4a, b). The test lasted $\sim$35 s and traversed 7 face

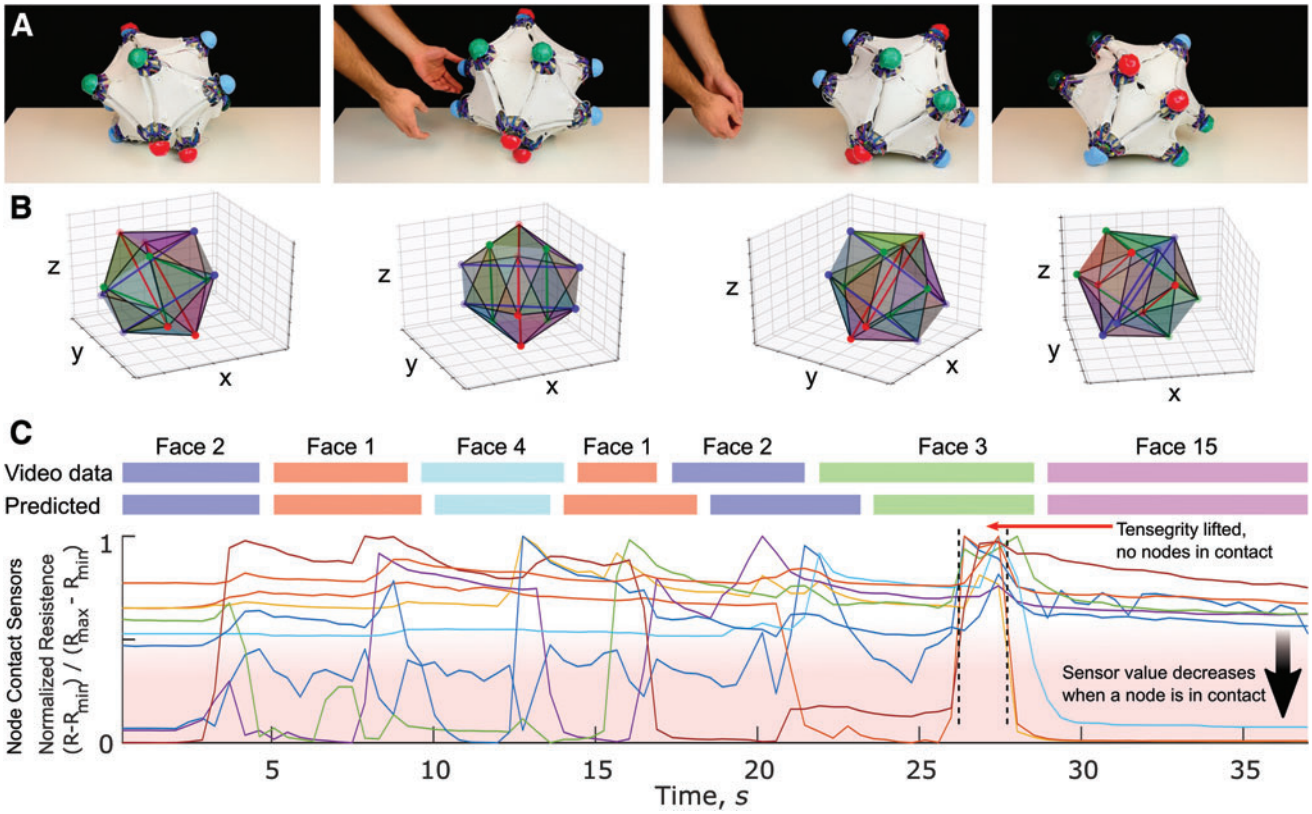


**FIG. 4.** **(A)** Sequence of manual rotations of the tensegrity robot, tipping from face to face, to validate the downward face detection algorithm. **(B)** State reconstruction from the above sequence extracted from the strain sensors, node position estimation algorithm, and node contact sensors. **(C)** Node contact sensor data (*thin lines*) measured during the rotation sequence, plotted as normalized resistance $(R - R_{min})/(R_{max} - R_{min})$. Resistance decreases when the node is in contact with the ground. Rows of colored bars on *top* show the timing interval when different faces are in contact with the ground, by video analysis and from node contact sensors. The full video sequence is available in the Supplementary Video S3. Color images are available online.

transitions. The video data were manually coded and translated into truth data using video editing software (Adobe Premier Pro) to determine the time stamps associated with individual nodes contacting or leaving the ground (Fig. 4c). The individual node contact sensors had an average accuracy of 95.0% over all time stamps. Most of the error in the node contact sensors came from either timing ambiguities between the truth data and the sensor data or from sensor malfunctions.

To efficiently remove many of the errors, we output the face with the last known valid triad of nodes and only update the face prediction when a new valid triad is sensed. In our trials, the face detection algorithm correctly identified the downward face at each transition. The overall moment-to-moment accuracy of the face detection algorithm described above—retaining a predicted face until a new valid triad of nodes is detected—is ~89.8%. Without using our algorithm, the moment-to-moment accuracy of each prediction against the truth data at each time stamp without retaining the previous valid triad yields an accuracy of 62.5%. This level of accuracy was primarily due to a single sensor behaving inconsistently, resulting in two correct sensor values, but not a valid triad of nodes and therefore a reduced accuracy.

The overall accuracy of our approach is highly dependent on the independent reliability of each node contact sensor. As a result, directly measuring the downward face using contact is very fast and reliable, but it is also sensitive to sensor failures. In future work, we will improve the manufacturing process to increase the reliability of sensors.

### *Rolling*

Locomotion of a tensegrity robot using robotic skins was demonstrated in our prior work, although without sensors or pressure control.[27] In the present work, all the control electronics and hardware were mounted in the interior of the robot to reduce the number of tethers and off-board hardware required. Furthermore, this work demonstrates a teleoperated tensegrity robot, whereas the prior work used manually operated pneumatic valves.

Our hardware implementation did, however, limit the tensegrity's locomotion capability. The inclusion of all pressure regulators onboard increased the system mass and caused the robot to sag, therefore lowering the center of gravity by a few centimeters. In addition, the actuators we implemented were only capable of 12% strain in either direction, and therefore not able to displace the center of gravity over the polygon of stability when starting from an equilateral face and rolling to any other face while on flat ground.[10–12] To enable rolling from any face, the robot was laid on a declined slope (−8.7°). Figure 5 presents snapshots,

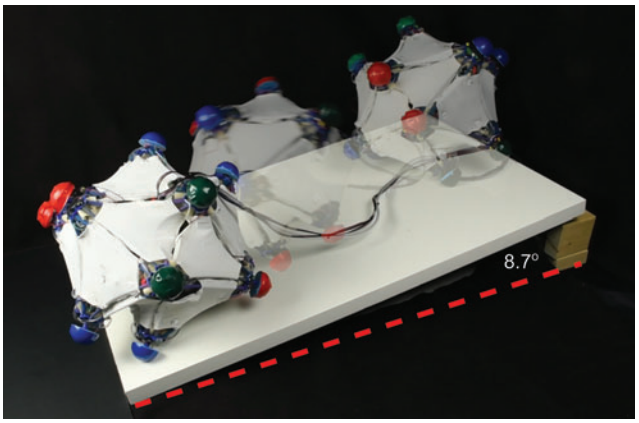


**FIG. 5.** Teleoperated rolling locomotion of the tensegrity robot (from *right* to *left*) on a downward inclined plane (−8.7° slope) shown as superposition of screenshots of the corresponding video in the Supplementary Video S3. Due to the weight of the onboard pressure regulators, the robot was only able to independently roll across declined surfaces. Color images are available online.

and Supplementary Video S4 shows the tensegrity robot actuating its segment, destabilizing its structure, and rolling down the plane in a sequence of six rolls. While the full locomotion capability of this tensegrity robot was only possible with a downward slope, in future work this could be addressed by using actuators capable of higher strains and decreasing the system mass.

The actuation policy uses two pneumatic actuators, which is the minimum required to achieve rolling in a pneumatic tensegrity.[10,26] It is noted that actuators located between two parallel rods in the six-bar tensegrity structure are never utilized. To reduce the size of the polygon of stability, one of the two pneumatic actuators must be located on the face touching the ground. Using lower friction nodes may help in future work. The second pneumatic actuator is used to shift the center of mass toward the desired locomotion direction, and should be located on a side of the tensegrity, close to the base face (Supplementary Fig. S1e). Since the structure has compliance inherently built into the system, other actuators do not need to be engaged.

## Conclusion

In this article, we demonstrate a robotic six-bar tensegrity structure, actualized by covering its surface with robotic skins, which are two-dimensional, modular, robotic membranes with embedded actuators and sensors. The robotic skin-clad tensegrity robot is able to individually activate its pneumatic actuators and reconstruct its state—position and orientation—using strain sensors, and it achieves teleoperated rolling locomotion on a declined plane. This unique approach is the first demonstration of real-time state reconstruction of a tensegrity robot using onboard proprioceptive sensors.

We assess that the high stability of the robot when its base face is an equilateral triangle is primarily caused by the limit of force and displacement that the actuators can deliver compared with the system's mass and size. Such high stability makes rolling from these orientations difficult. The next logical research target would be to improve systematic rolling from any face on any surface (flat, downhill, uphill, rugged). Potential ways to achieve this include reducing the mass of components (number and size) and also replacing the surface material of the nodes with a lower friction material.

In addition to the mechanical limitations, the current sensor-processing pipeline has limitations that make dynamic control challenging. Although the shape and face reconstruction algorithms operate quickly on our off-board PC, the communication between the tensegrity and the PC introduces lag in the state estimation, as seen in Supplementary Videos S2 and S3. However, preliminary results suggest that there are several areas for improvement. For example, in the current network topology, the complete set of 24 sensors can be sampled sequentially at $\sim 30$ Hz when actuation is disabled. However, even this is not a lower bound on the achievable system update frequency—a single capacitive strain sensor can be sampled at a rate of 119 Hz.[42] Faster communication rates could be attained, for example, by using a dedicated microcontroller for actuation, which would only receive data from the PC, and a separate dedicated microcontroller that would only send data from the sensors to the PC. This improvement could potentially operate quickly enough to use the sensor feedback for dynamic control, for instance, using insights gained from other high-dimensional, soft robotic systems such as continuum manipulators.[49] Further improvements in state estimation could be gained by taking into account sequence information, using algorithms such as long short-term memory or vanilla recurrent neural networks. Previous application of similar algorithms to sensor networks has been shown to be beneficial in other settings, including human-body motion capture garments[50] and soft robotic segments.[51,52]

Future work would also include the implementation of closed-loop control and dynamic orientation estimation. Since this tensegrity robot is a soft and compliant nonlinear system, machine learning tools would likely be the easiest path foward to realize closed-loop control. In this optic, cooperative coevolutionary algorithms are a promising approach. The accuracy of the neural network used with strain sensors to estimate orientation could also be enhanced by continuously training the network with ground truth data from contact sensors attached to the node ends. Finally, to eliminate the robot's dependence on an external tether, it could be equipped with an onboard source of compressed air, batteries, and Wi-Fi connection. An untethered robot would be particularly convenient for extensive training sessions under various conditions.

By introducing the concept of proprioceptive, surface-actuated tensegrities, we hope to enable future advances toward controlled, autonomous locomotion of compliant structures. In remote locations such as caves, mountains, and rocky extraterrestrial terrains, there is often a lack of certainty about the environment and lack of external sensors for localization and orientation estimation. Thus, with this combination of inherent compliance and the ability to sense their own state, tensegrity robots such as the one introduced in this article could allow exploration of previously inaccessible locations. Furthermore, this works serves as an additional step toward passive tensegrity structures that can be easily clad in removable and reconfigurable robotic skins to control the motions of these structures from their surface.

### Acknowledgments

We thank Jonathan Bruce and Massimo Vespignani at the NASA Ames Research Center for their assistance with tensegrity systems, and Nguyen Pham for his assistance in maintaining and improving the robot between experiments.

### Author Disclosure Statement

No competing financial interests exist.

### Funding Information

This work was supported by a NASA Early Career Faculty grant (grant numbers NNX14AO52G and 80NSSC17K0553). D.S.S. and J.C.C. were supported by NASA Space Technology Research Fellowships (grant numbers 80NSSC17K0164 and NNX15AQ75H). M.C.Y. was supported by an NSF Graduate Research Fellowship (grant number DGE-1333468).


### Supplementary Material

Supplementary Data S1
Supplementary Figure S1
Supplementary Video S1
Supplementary Video S2
Supplementary Video S3
Supplementary Video S4

### References


1. Fuller B. Buckminster Fuller Institute website. https://www.bfi.org/about-fuller (accessed August 25, 2020).
2. Snelson K. Tensegrity, Weaving and the Binary World. http://kennethsnelson.net/Tensegrity_and_Weaving.pdf (accessed August 25, 2020).
3. Skelton RE, Montuori R, Pecoraro V. Globally stable minimal mass compressive tensegrity structures. Comp Struct 2016;141:346–354.
4. Skelton RE, Adhikari R, Pinaud J-P, *et al.* An introduction to the mechanics of tensegrity structures. In: Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228), Orlando, FL, USA, December 4–7, 2001. Piscataway, NJ: IEEE, 2001, pp. 4254–4259.
5. Wang N, Butler J, Ingber D. Mechanotransduction across the cell surface and through the cytoskeleton. Science 1993; 260:1124–1127.
6. Ingber DE. The architecture of life. Sci Am 1998;278:48–57.
7. Paul C, Valero-Cuevas FJ, Lipson H. Design and control of tensegrity robots for locomotion. IEEE Trans Robot 2006; 22:944–957.
8. Rieffel J, Valero-Cuevas F, Lipson H. Automated discovery and optimization of large irregular tensegrity structures. Comput Struct 2009;87:368–379.
9. Rieffel J, Lipson H, Valero-Cuevas FJ. Growing form-filling tensegrity structures using map L-systems. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07), London, England, July 2007. New York, USA: Association for Computing Machinery. p. 1063.
10. Koizumi Y, Shibata M, Hirai S. Rolling tensegrity driven by pneumatic soft actuators. In: 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, May 14–18, 2012. Piscataway, NJ: IEEE, 2012, pp. 1988–1993.
11. Caluwaerts K, Despraz J, Işçen A, *et al*. Design and control of compliant tensegrity robots through simulation and hardware validation. J Roy Soc Interface 2014;11.
12. Kim K, Agogino AK, Moon D, *et al.* Rapid prototyping design and control of tensegrity soft robot for locomotion. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), Bali, Indonesia, Dec. 5–10, 2014. Piscataway, NJ: IEEE, 2014, pp. 7–14.
13. Sabelhaus AP, Bruce J, Caluwaerts K, *et al.* System design and locomotion of SUPERball, an untethered tensegrity robot. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May 26–30, 2015. Piscataway, NJ: IEEE, 2015, pp. 2867–2873.
14. Kim K, Agogino AK, Toghyan A. Robust learning of tensegrity robot control for locomotion through form-finding. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 Sept.–2 Oct. 2015. Piscataway, NJ: IEEE, 2015, pp. 5824–5831.
15. Chen L-H, Kim K, Tang E, *et al*. Soft spherical tensegrity robot design using rod-centered actuation and control. J Mech Robot 2017;9:025001–025009.
16. Böhm V, Jentzsch A, Kaufhold T, *et al.* An approach to compliant locomotion systems based on tensegrity structures. In: 56th International Scientific Colloquium Ilmenau University of Technology, Germany, September 12–16, 2011, Session 2.1. Ilmenau, Germany: ilmedia - TU Ilmenau, 2011, pp. 1–6.
17. Schorr P, Bohm V, Zimmermann K, *et al.* An Approach to the Estimation of the Actuation Parameters for Mobile Tensegrity Robots with Tilting Movement Sequences. In: 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR), Delft, Netherlands, June 20–22, 2018. Piscataway, NJ: IEEE, 2018, pp. 1–8.
18. Mirletz BT, Bhandal P, Adams RD, *et al.* Goal-directed CPG-based control for tensegrity spines with many degrees of freedom traversing irregular terrain. Soft Robot 2015;2; 165–176.
19. Sabelhaus AP, Ji H, Hylton P, *et al.* Mechanism design and simulation of the ULTRA spine: A tensegrity robot. In: Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 5A: 39th Mechanisms and Robotics Conference, Boston, Massachusetts, USA, August 2–5, 2015. New York, NY: The American Society of Mechanical Engineers, V05AT08A059.
20. Friesen JM, Glick P, Fanton M, *et al.* The second generation prototype of a Duct Climbing Tensegrity robot, DuCTTv2. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May 16–21, 2016. Piscataway, NJ: IEEE, 2016, pp. 2123–2128.
21. Rieffel J, Stuk R, Valero-Cuevas FJ, *et al.* Locomotion of a tensegrity robot via dynamically coupled modules. In: Proceedings of the International Conference on Morphological Computation. March 2007.
22. Böhm V, Kaufhold T, Zeidis I, *et al.* Dynamic analysis of a spherical mobile robot based on a tensegrity structure with two curved compressed members. Arch Appl Mech 2017; 87:853–864.
23. Mintchev S, Zappetti D, Willemin J, *et al.* A Soft Robot for Random Exploration of Terrestrial Environment. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, May 21–25, 2018. Piscataway, NJ: IEEE, 2018, pp. 7492–7497.

24. Vespignani M, Ercolani C, Friesen JM, *et al.* Steerable locomotion controller for six-strut icosahedral tensegrity robots. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October, 2018. Piscataway, NJ: IEEE, 2018, pp. 2886–2892.
25. Kim K, Agogino AK, Agogino AM. Rolling locomotion of cable-driven soft spherical tensegrity robots. Soft Robot 2020;7:346–361.
26. Hirai S, Koizumi Y, Shibata M, *et al.* Active shaping of a tensegrity robot via pre-pressure. In: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, July 9–12, 2013. Piscataway, NJ: IEEE, 2013, pp. 19–25.
27. Booth JW, Shah D, Case JC, *et al.* OmniSkins: Robotic skins that turn inanimate objects into multifunctional robots. Science Robotics 19 Sep 2018;3:eaat1853.
28. Schorr P, Böhm V, Zentner L, *et al.* Motion characteristics of a vibration driven mobile tensegrity structure with multiple stable equilibrium states. J Sound Vib 2018;437:198–208.
29. Schorr P, Böhm V, Zentner L, *et al.* Dynamical investigation of crawling motion system based on a multistable tensegrity structure. In: Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO, Porto, Portugal, July 29–31, 2018. Setúbal, Portugal: SCITEPRESS Digital Library, 2018, pp. 122–130.
30. Khazanov M, Humphreys B, Keat WD, *et al.* Exploiting dynamical complexity in a physical tensegrity robot to achieve locomotion. In: Artificial Life Conference Proceedings 2013, East Lansing, MI, USA, July 19–22, 2013. Cambridge, MA: MIT Press, Cambridge, 2013, pp. 965–972.
31. Böhm V, Zimmermann K. Vibration-driven mobile robots based on single actuated tensegrity structures. In: 2013 IEEE International Conference on Robotics and Automation. 2013:5475–5480.
32. Zappetti D, Mintchev S, Shintake J, *et al.* Bio-inspired Tensegrity Soft Modular Robots. In: Proceedings of the 6th International Conference, Living Machines 2017: Biomimetic and Biohybrid Systems, Stanford, CA, USA, July 26–28, 2017. Cham, Switzerland: Springer Nature Switzerland AG, 2017, pp. 497–508.
33. Khazanov M, Jocque J, Rieffel J. Evolution of locomotion on a physical tensegrity robot. In Artificial Life Conference Proceedings 2014, Manhattan, New York, USA, July 30–August 2, 2014. Cambridge, MA: MIT Press, Cambridge, 2014, pp. 232–238.
34. Khazanov M, Jocque J, Rieffel J. Developing morphological computation in tensegrity robots for controllable actuation. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO Comp '14), Vancouver, BC Canada, July, 2014. New York, NY: Association for Computing Machinery, 2014, pp. 1049–1052.
35. Rieffel J, Trimmer B, Lipson H. Mechanism as mind-what tensegrities and caterpillars can teach us about soft robotics. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, Winchester, United Kingdom, August 5–8, 2008. Cambridge, MA: MIT Press, Cambridge, 2008, pp. 506–512.
36. Rieffel JA, Valero-Cuevas FJ, Lipson H, Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. J Roy Soc Interface 2010;7:613–621.
37. Bruce J, Caluwaerts K, Iscen A, *et al.* Design and evolution of a modular tensegrity robot platform. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June, 2014. Piscataway, NJ: IEEE, 2014, pp. 3483–3489.
38. Caluwaerts K, Bruce J, Friesen JM, *et al.* State estimation for tensegrity robots. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). 2016:1860–1865.
39. Campanaro L. Sensor integration and controller design for a tensegrity-modular robot. Master's thesis, Politecnico di Torino 2018;4.
40. Tondu B. Modelling of the McKibben artificial muscle: A review. J Intel Mater Syst Struct 2012;23:225–253.
41. Booth JW, Case JC, White EL, *et al.* An addressable pneumatic regulator for distributed control of soft robots. In: 2018 IEEE International Conference on Soft Robotics (RoboSoft), Livorno, Italy, 24–28 April, 2018. Piscataway, NJ: IEEE, 2018, pp. 25–30.
42. White EL, Yuen MC, Case JC, *et al.* Low-cost, facile, and scalable manufacturing of capacitive sensors for soft systems. Adv Mater Technol 2017;2:1700072.
43. White EL, Case JC, Kramer-Bottiglio R. A soft parallel kinematic mechanism. Soft Robot 2018;5:36–53.
44. NASA Tensegrity Robotics Toolkit (NTRT). https://ti.arc.nasa.gov/tech/asr/groups/intelligent-robotics/tensegrity/NTRT (accessed August 25, 2020).
45. Mouret J-B, Doncieux S. Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evol Comput 2012;20:91–133.
46. Chatzilygeroudis K, Rama R, Kaushik R, *et al.* Black-box data-efficient policy search for robotics. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, September 24–28, 2017. Piscataway, NJ: IEEE, 2017, pp. 51–58.
47. Cully A, Clune J, Tarapore D, *et al.* Robots that can adapt like animals. Nature 2015;521:503–507.
48. Arun KS, Huang TS, Blostein SD. Least-squares fitting of two 3-d point sets. IEEE Trans Pattern Anal Mach Intel 1987;5:698–700.
49. George Thuruthel T, Ansari Y, Falotico E, *et al.* Control strategies for soft robotic manipulators: A survey. Soft Robot 2018;5:149–163.
50. Kim D, Kwon J, Han S, *et al.* Deep full-body motion network for a soft wearable motion sensing suit. IEEE/ASME Trans Mech 2018;24:56–66.
51. Thuruthel TG, Shih B, Laschi C, *et al.* Soft robot perception using embedded soft sensors and recurrent neural networks. Sci Robot 2019;4:eaav1488.
52. Soter G, Conn A, Hauser H, *et al.* Bodily aware soft robots: integration of proprioceptive and exteroceptive sensors. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May, IEEE, 2018, pp. 2448–2453.


Address correspondence to:
*Rebecca Kramer-Bottiglio*
*School of Engineering and Applied Science*
*Yale University*
*10 Hillhouse Avenue*
*New Haven, CT 06520*
*USA*

*E-mail:* rebecca.kramer@yale.edu