CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Control and locomotion of tensegrity robots through manipulation of the center of mass

Brett David Layer ⬛, Harrison Denning and Jeffrey R. Hill ⬛

Mechanical Engineering Department, Brigham Young University, Provo, UT, 84602, USA
Corresponding author: Jeffrey R. Hill; Email: dr.jeff.hill@byu.edu

Abstract

This paper provides the methodology used to simulate and control an icosahedral tensegrity structure augmented with movable masses attached to each bar to provide a means of locomotion. The center of mass of the system can be changed by moving the masses along the length of each of the bars that compose the structure. Moving the masses changes the moments created by gravitational force, allowing for the structure to roll. With this methodology in mind, a controller was created to move the masses to the desired locations to cause such a roll. As shown later in this paper, such a methodology, assuming the movable masses have the required mass, allows for full control of the system using a quasi-static controller created specifically for this system. This system has advantages over traditional tensegrity controllers because it retains its shape and is designed for high-shock scenarios.

1. Introduction

Tensegrity systems refer to a class of pin-jointed structures that consist of contiguous cables in tension and noncontiguous bars in compression. The tension in the cables stabilizes the structure in a mass efficient way since tensegrity structures can minimize the mass needed for buckling loads. With pretensioned cables, many designs are statically stable. Since the mid-twentieth century, tensegrity structures have become popular among artists and engineers alike [1]. Among the engineering community, such structures sparked broad interest due to their unique nature of combining shock absorption, a minimal mass solution to a system – especially in buckling conditions, and the potential to collapse to reduce their volumes [2–6]. These properties of tensegrity structures, as well as their ability to morph their structure, have led to the creation of highly adaptive structures as well [7, 8]. Additionally, since these structures are so malleable, they are often utilized as a replacement for a traditional soft robot [9–11].

More recently, select groups have studied these structures for various robotic applications [12, 13], especially with the intent of such structures acting as a lunar landing robot [14–18]. These robotic applications, in which using a tensegrity structure brings the significant advantages of significant shock absorption, the potential to collapse to smaller volumes, and low mass-to-stress ratios compared to other solutions, make it necessary to create a reliable, simple control system for these structures.

Modern controllers [19, 20] are tailored to rigid dynamic systems, and even controllers implemented on spherical robots [21–23] are difficult to implement on an icosahedral tensegrity system. Controllers implemented on modern tensegrity structure also will not work with this actuation design [24–28]. The proposed design utilizes movable masses to shift the robot's center of mass instead of changing the length of the strings on the structure. One solution for controlling a tensegrity structure is to use gravity

as the exclusive means of control. All that is needed for such a structure to roll is for the center of mass of the structure to be outside of the current structure's support polygon. When this occurs, the structure will rotate toward the location of the center of mass. To control the rotation of the structure and thereby the location of the structure, it is only necessary to control the location of the center of mass of the structure.

Past researchers have addressed this problem by controlling the tension or lengths of the cables and thereby the shape of the structure [29–32]. If the shape changes drastically enough, the structure will roll. This has traditionally involved attaching actuators to the ends of each bar, although there has been some progress made in housing the motor within a flexible tube [33]. When actuators are attached to the end of each bar, there is a greater chance for damage upon impact with the ground. Additionally, the dynamics presented by changing the cable lengths of the system are also more complicated and computationally expensive.

In the field of tensegrity robotics, conventional methods have primarily hinged on modifying string tension and lengths to facilitate motion. This paper diverges from this norm by introducing a novel approach using movable masses in an icosahedral tensegrity structure. Movable masses as a means of control have been explored previously in other applications [34–36] not related to tensegrity. While movable masses have been employed in previous tensegrity systems [37–39], our work distinguishes itself through its unique structural and functional innovations.

A critical drawback of the prevailing spherical tensegrity models lies in their vulnerability to impact. The curved members in these structures are predisposed to ground contact during impact, risking structural damage. Additionally, these designs lack a mechanism to reposition the masses toward the structure's core, which could otherwise offer protection in the event of an impact. The icosahedral design has specifically been used for impact absorption and addresses these limitations. By enabling the relocation of masses to the center of the structure, our model significantly enhances its impact resilience. This centralization of masses not only serves as a protective measure but also facilitates the co-location of the payload with the motors. This integration results in a more compact design, reducing the total mass and improving the system's overall efficiency.

Moreover, our approach simplifies control methodologies. The controller in our model activates at rest, guiding the masses to specific nodes to initiate desired movements. This system contrasts sharply with cable-driven models where complex algorithms are necessitated by interdependencies among multiple variables. The inherent simplicity of our model, which maintains its shape without altering properties like string length or tension, allows for an efficient, geometry-based control algorithm.

This research contributes to the understanding of mass actuation control in tensegrity systems. We demonstrate the feasibility and advantages of our approach under varied conditions, including payload dynamics and downhill stability. Our findings highlight the superiority of the icosahedral structure over spherical models in terms of control precision, impact resilience, and momentum utilization. Payload location and weight limits are discussed and illustrated in this paper, expanding upon the payload analysis in ref. [38], as well as exploring the performance capabilities on slopes.

In summary, this paper not only challenges conventional tensegrity system designs but also introduces a more robust, simpler, and efficient alternative, paving the way for future advancements in the field.

The remainder of the paper contains the following sections: Section 2 introduces the environment that was created for the model, including how the movement of masses was simulated. Section 3 explores mathematically the feasibility of the movable mass method as well as the constraints on the system to guarantee movement. Section 4 introduces the controller and explains the necessary assumptions. Section 5 contains a discussion of the results generated by the model as well as a comparison with the results of a physical system. Finally, Section 6 summarizes the current findings and discusses future work to be done.
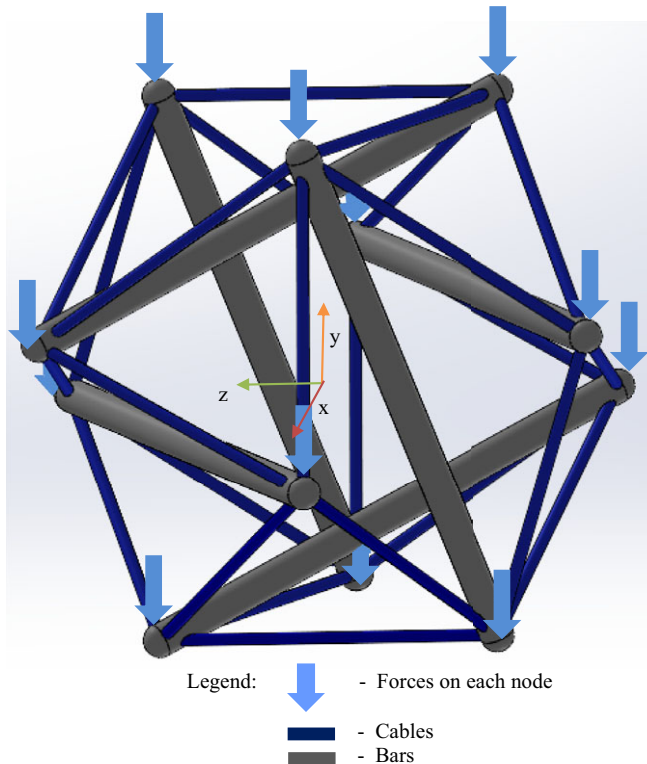
Figure 1. *The above figure shows the nominal configuration of the tensegrity structure. The gray bars are where the movable masses reside. The dark blue lines represent the tensile members of the system, and the light blue arrows represent where the forces are on the system initially. Due to the implementation of the code, the masses are simplified as forces in this simulation [40].*

2. Modeling the tensegrity environment

The tensegrity model developed for this work builds on the standard dynamic tensegrity model developed in MATLAB by MOTES [40], with the same traditional assumptions that quasi-static tensegrity structures use. These assumptions are that the system settles before a new calculation occurs, the dynamics of the system are modeled by the code library, and the movable masses are treated as point forces on this system. This model, whose shape was chosen for the convenient property that the bars create three orthonormal planes, as shown in Fig. 1, was translated into Python from the MATLAB code mentioned above and then modified to fit the needs of the robotic system. These modifications are explained below.

The initial significant modification implemented was the elimination of the fixed node requirement in the system, a prerequisite for enabling unimpeded rolling motion. The previously utilized pinned nodes were substituted with a spring and damper mechanism, designed to emulate the system's interaction with the ground. The selected values for these components were deliberately set to be considerably larger relative to the system's mass, enhancing stability and facilitating a more consistent simulation. This approach, while simplifying the ground interaction model, is aptly suited for our purposes as it focuses not on the detailed contact dynamics with the ground but rather on ensuring simulation stability. By allowing the model to roll without ground constraints, which can introduce extraneous energy into the system, this method more accurately replicates the ground reactions of the actual system.

The second change was to model the movable masses. To do this, forces were added to either node of each bar. Each mass could then be moved by changing the forces on each node of the bar, where the total force between the two nodes combined remains constant. Thus, in this model, the masses used to control the bar are simply modeled by the gravitational forces that the masses would emit.

The force approach can be used instead of modeling masses for two main reasons. First, since the system is quasi-static for the current modeling approach, the fidelity of the dynamics does not have to be particularly high. This assumes that the system has settled before the next controller iteration. The mass of the movable mass can be neglected, which greatly simplifies the dynamics of the model, while still reasonably simulating the roll characteristics for this system. Ultimately, disregarding the masses in the system would make the system move to its next position faster than it would in real life (since there would be more mass to accelerate). This behavior is acceptable for a quasi-static system since the primary concern for this paper is deliberate control in certain directions, not necessarily system speed.

Second, to reduce the initial noise of the system, a damping term is used. Again, this reduces the fidelity of the dynamics to simplify the model. By modeling the ground as a spring and the movable masses as two forces on either side of each mass' corresponding bar, it is possible to create a model that works well with quasi-static control. If a quasi-static control is not used, the model would have to be evaluated to see if such a method would suffice.

3. System roll analysis

The tensegrity system chosen for control is an icosahedron, or a 20-faced structure where each face is a triangle. This shape is composed of 12 vertices (nodes), with each node connected to the five closest nodes. Each of these nodes is located according to a permutation of the following coordinates: $(0, \pm1, \pm\phi)$, $(\pm1, \pm\phi, 0)$, $(\pm\phi, 0, \pm1)$, where ϕ is the golden ratio of approximately 1.618.

With this structure, it is possible to statically model the system as if the gravitational force of the movable masses is split equally between the two nodes at the end of each bar. This is a valid assumption for the dynamics of the model since a quasi-static controller is used.

The center of mass of the system can be modeled as

$$x_{tot}^n = \frac{1}{m_{tot}} \sum_{i=1}^{12} m_i x_i^n \tag{1}$$

where n represents each Cartesian direction, m_i is the mass of each node (i), while m_{tot} is the total mass of the system, which is the sum of m_i. For each Cartesian direction, Eq. (1) will need to be recalculated.

Assuming each of the nodes represents the same amount of mass, meaning the bars of the system have a constant mass distribution, then m_i is constant for all iterations of i, which allows it to be pulled out of the summation by multiplying it by the number of iterations. This product is equal to m_{tot}, allowing for the masses to cancel. Equation (1) can then simplify to:

$$x_{tot}^n = \sum_{i=1}^{12} x_i^n \tag{2}$$

Solving Eq. (2) reveals that the center of mass of the system is located at (0,0,0) in its initial condition. For example, in Fig. 2, the origin of the initial coordinate frame would be located at the center of the structure, at its center of mass. In this paper, the *x-z* plane will be parallel to the ground.

In any resting state, the system rests on a triangular support polygon. Due to the geometry of the regular icosahedron, each of these triangles are equilateral triangles. Due to the symmetry of the system, the center of mass of the system, excluding the movable masses, is directly above the center of the support polygon in any configuration.

There are two main configurations that describe the state of this tensegrity system, spiral and parallel. A visualization of those cases is shown in Figs. 2 and 3. In both cases, the way to induce rolling in
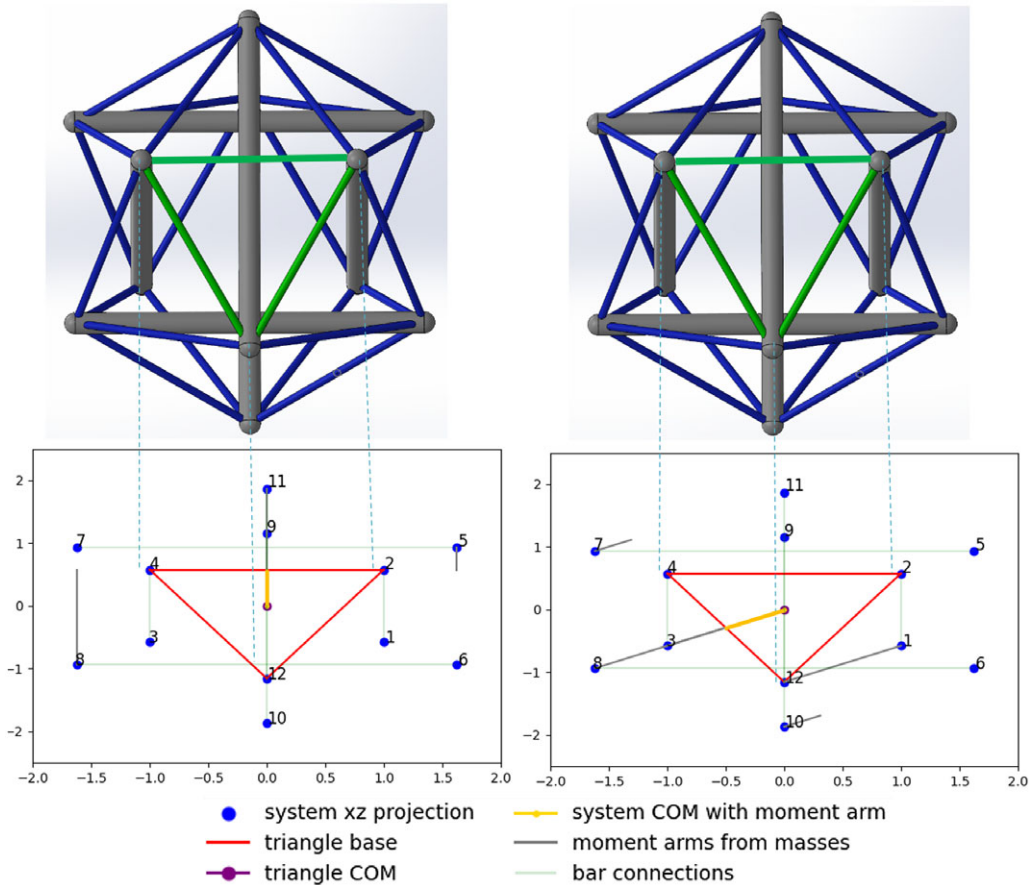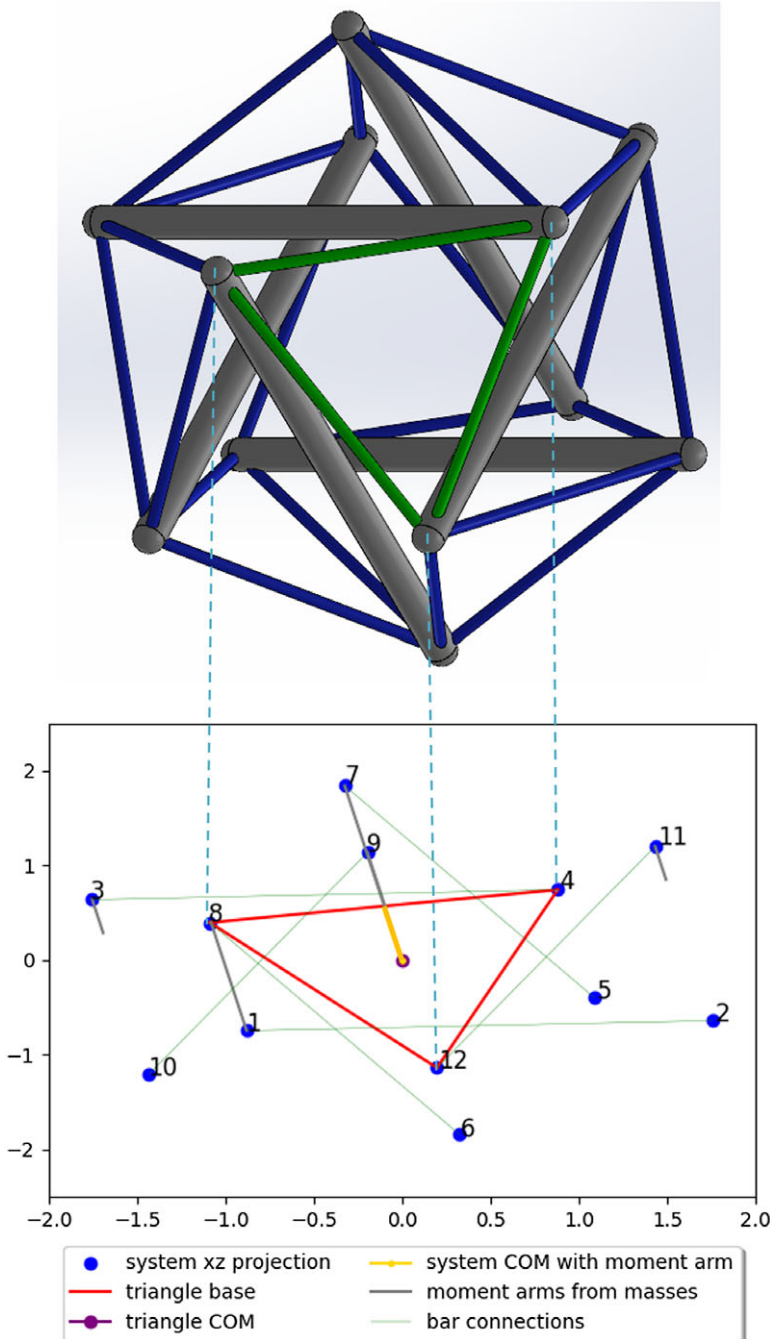
Figure 2. *The images above show the system in its parallel configuration. The images on the top show the geometry of the system looking from below the x-z plane, while the images on the bottom show the projection of the top image onto the x-z plane. The bottom images were then used to determine the ratio of the moment arms created by the movable masses to the moment arm of the mass of the system. The parallel configuration consists of two different moment arm alignments, one of which is better than the other. The edge around which rolling occurs is the edge touching the yellow moment arm.*

the structure is to move the center of mass of the entire system to the outside of the support polygon. To do this, the movable masses must overcome the system's own mass. To determine the required mass, first find the distance from the center of mass of the system to the edge of the support polygon triangle around which it will be rotated.

Next, find the equation to represent the vector about which the system will rotate. The equation for this vector is created by finding the equation of the line between the two nodes about which the system will rotate and then neglecting the component that is out of plane with the ground. This vector is the edge of the system about which the system will rotate. Once this vector is obtained, it is possible to find the moments from the movable masses of the system by finding the length of the line perpendicular to the edge on which the system will rotate. The length of this line will be the perpendicular distance from the edge about which rotation will occur to the movable masses.

Using the standard total moment equation, the summation of moments for the system is:

$$\sum M = \left(\sum_{m=1}^{6} r_m \times \left(-m_m g\hat{j} \right) \right) + r_c \times (m_c g\hat{j}) \qquad (3)$$

Figure 3. *The images above show the system in its spiral configuration. The images on the top show the geometry of the system viewed from below the x-z plane, while the images on the bottom show the projection of the top image onto the x-z plane. The bottom images were then used to determine the ratio of the moment arms created by the movable masses to the moment arm of the mass of the system. In this configuration, any way the system can rotate has the same moment arms created by the movable masses. The edge around which rolling occurs is the edge touching the yellow moment arm.*

where r_m is the moment arm for each respective movable mass and r_c is the moment arm for the center of mass, while m_m is the mass of each movable mass, and m_c is the mass of the tensegrity structure. For clarity, the $\times$ used above is the cross product. This method is sufficient since this is a quasi-static model. At the beginning of each control command, it is assumed that the robot is stationary, meaning that any dynamic effects are 0 when motion is induced.

In the model, there are two different configurations of the tensegrity system, spiral or parallel, as shown in Figs. 2 and 3. In the two configurations, the possible placements of the movable masses change the moment arms that can be used.

In each scenario, the sum of moments can be determined by summing the movable mass moment arms, represented by the sum of lengths of the gray lines in Figs. 2 and 3, and then multiplying by the mass of the movable mass. The moment of the center of mass is also included in the following equation:

$$\sum M = \left(\sum r_m \right) m_m - r_c m_c \tag{4}$$

Each of the scenarios described above has a unique r_m and r_c value because the lever arms created by the masses in each situation are unique. For simplicity, each of these values was solved numerically since the values are dependent on the size of the structure. In other words, in each of the three scenarios described above, $\sum r_m$ and r_c were solved and plugged into Eq. (4). The way that they were solved was by adding the length of the gray lines and using that as $\sum r_m$, using the length of the yellow line as r_c, and setting $\sum M = 0$. The equation was then rearranged to solve m_m as a function of m_c for each configuration. Rearranging Eq. (4) to relate the mass proportion to the lever arm when $\sum M = 0$ yields

$$\frac{m_m}{m_c} = \frac{r_c}{\sum r_m} \tag{5}$$

Solving $\frac{r_c}{\sum r_m}$ numerically for each scenario, where each scenario is shown in Figs. 2.2 and 2.3, yields the following: results

$$\frac{m_m}{m_c} > 0.8089 \tag{6}$$

$$\frac{m_m}{m_c} > 0.4058 \tag{7}$$

$$\frac{m_m}{m_c} > 0.4045 \tag{8}$$

where Eq. (6) refers to the worst-case parallel scenario, Eq. (7) refers to the best-case parallel scenario, and Eq. (8) refers to the spiral scenario.

One of the limitations to this approach is that these masses would be insufficient for actuation going up an incline. This is due to the rotation of the structure changing the moment arms available to the system. When the structure is on an incline, moving up an incline rotates the system with respect to a flat ground. This rotation changes the moment arms of the masses with respect to the axis of rotation, drastically changing the mass proportion required to move up an incline. However, this is also true in reverse. To move down an incline, much less mass is required.

Figure 4 shows how the ratio of the movable mass lever arm, $\sum r_m$, to the system mass lever arm, r_c, changes as the slope changes. Figure 5 shows the change of the moment arm available to move the system, $\sum r_m$, normalized by the length of the system's bar as a function of the slope that the structure is moving along. For clarity, the degrees shown at the bottom of the following figures represent the slope that the system is transversing. A positive value represents the system moving up a slope. For simplicity, the slope is rotated about the z axis shown in Fig. 1, meaning that the system moves up the steepest gradient.

In addition, Figs. 4–7 below are for a parallel best-case configuration in which the axis of rotation is perpendicular to the steepest gradient. The steepest gradient is the worst-case scenario for rolling up a hill since the slope of the gradient changes the movable mass proportion. It is also the worst-case
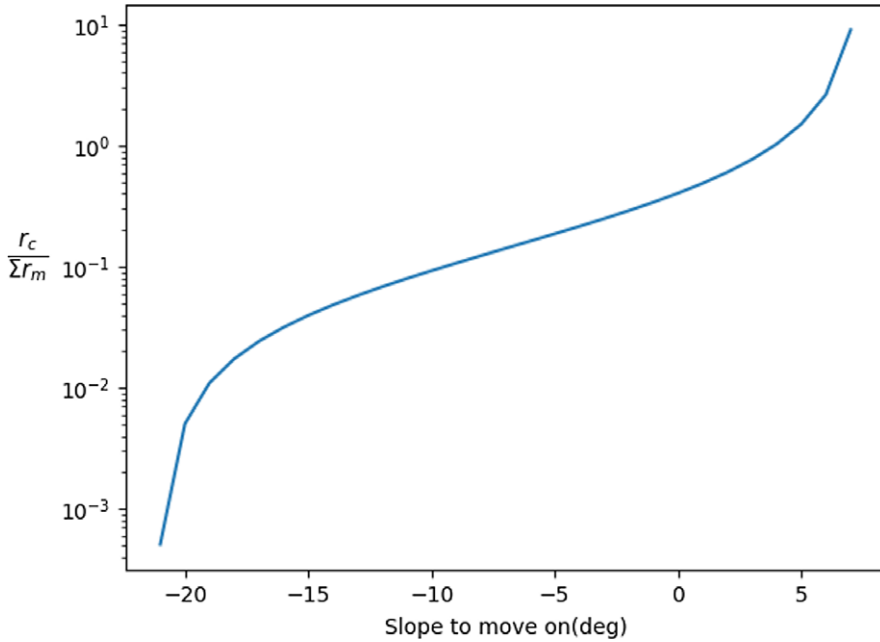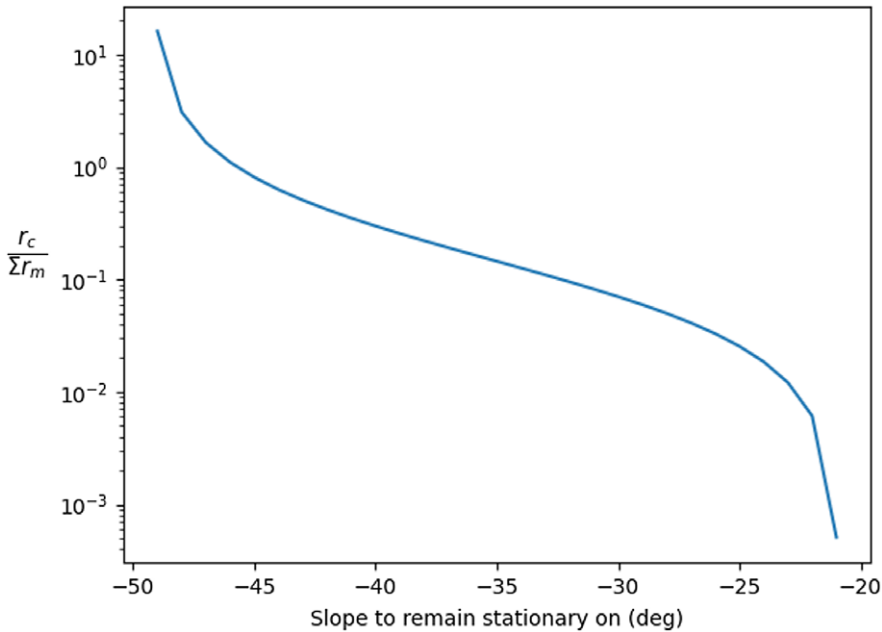
Figure 4. *This figure shows how the proportion of the movable mass to the center of mass changes to achieve locomotion on an incline. Once the proportional mass falls below 0, the system would roll off its own volition all the way down the hill simply due to the geometry of the system. The COM on the y axis represents the center of mass of the system.*
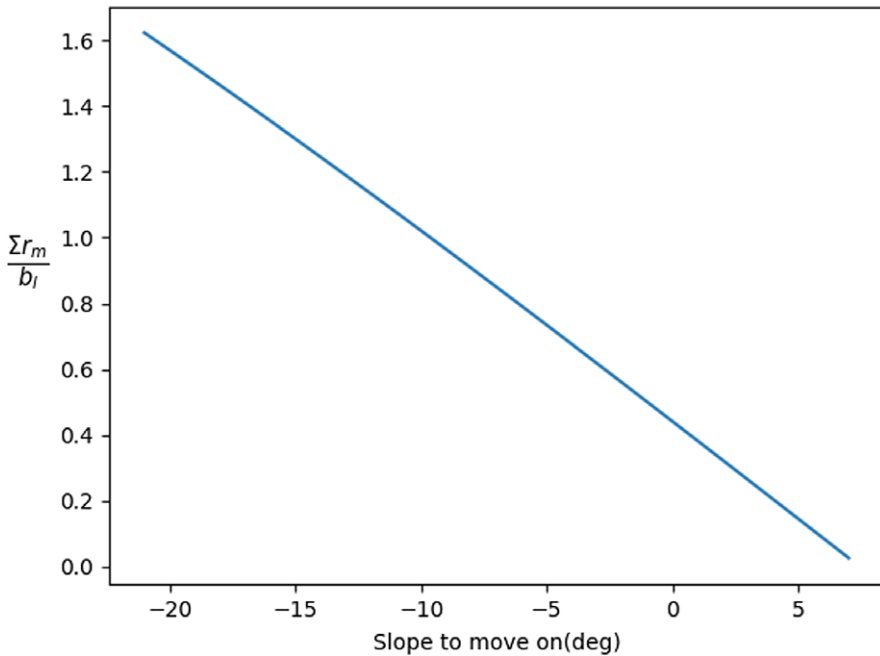
Figure 5. *This figure shows how the leverage available to the movable masses changed with the angle of the slope. This leverage arm assumes that all the masses are at the location that maximizes this arm. This graph shows how the r_m changed depending on the slope of the system used in the simulation. The b_l value is the bar length, allowing for a normalized Σr_m.*

Figure 6. *This figure shows how the proportion of the movable mass to the center of mass of the system changes to achieve rest conditions on a decline. Masses are located at the optimal locations to prevent movement. A positive angle represents the gradient that the system is moving down. The slope angle is measured about the z axis as shown in Fig. 1. The COM on the y axis represents the center of mass of the system.*

scenario for staying stationary on a downhill slope again since the structure is attempting to remain stationary against the largest gradient. The spiral configuration is one of the feasible configurations for rolling up a hill since the proportional movable masses are much lower than the worst-case parallel scenario.

Furthermore, this system can remain stationary on steeper slopes than the structure on its own could. This is because the movable masses can also be used as counterweights to prevent rotation around the desired axis. Figure 6 below shows the proportion between the payload mass and the mass of the movable masses needed to prevent rotation, as shown in Fig. 4, while Fig. 7 shows how the lever arm changes based on the angle of the system, as shown in Fig. 5. The difference between Figs. 4–5 and 6–7 is that the first set shows the mass required for movement up a slope, while the other shows the mass required to remain stationary on a slope. The stationary case is used for Figs. 6–7 since inducing motion on the slope would be trivial since any movement of the movable masses to the front of the structure would include motion.

This is a significant result because it shows that this system can stay at rest on a steep incline with no change in the mass of the movable masses. Assuming that the system is designed to the worst-case scenario above, where the mass of the movable mass is roughly 0.81, this ideal system would be capable of moving up a slope of approximately 3 degrees and have control moving down a slope up to 45 degrees. Interestingly, to keep this system static on a 45-degree slope, the mass proportion is equivalent to the mass needed in the worst-case scenario. Improvements to the climbing capability of this robot could be made by utilizing the momentum of the movable masses to allow the system to roll.

The figures above also describe the worst-case scenario for rolling down a hill. This scenario is when an axis the system could rotate about is perpendicular to the steepest gradient. The best-case scenario is when one of the nodes of the base triangle and the centroid of the triangular base are in line with
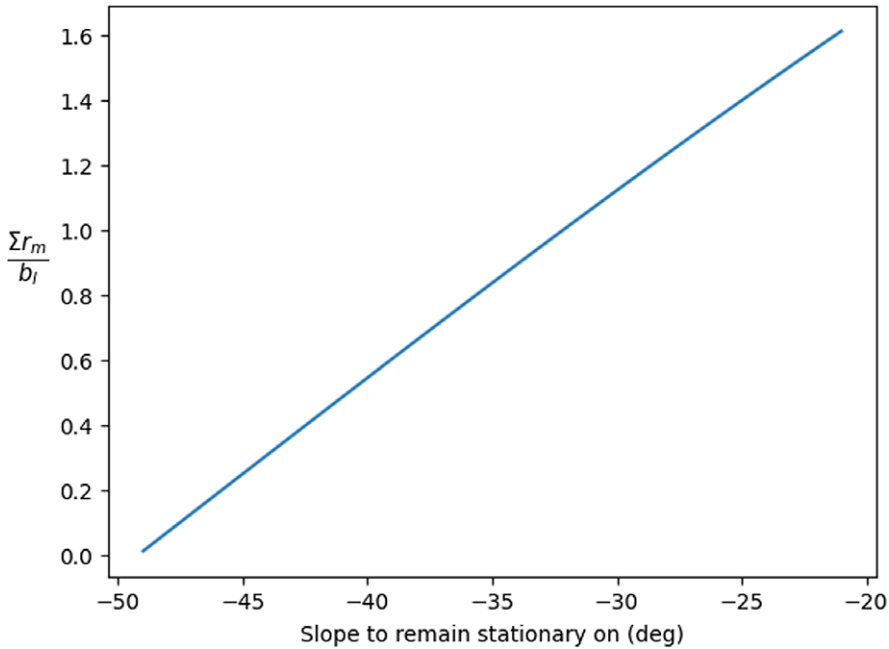
Figure 7. *This figure shows how the leverage available to the movable masses changed with the angle of the slope. This leverage arm assumes that all the masses are at the location that maximizes this arm. Like what is shown in Fig. 5, the leverage arms that the movable masses can utilize seem to decrease almost linearly based on the slope of the incline. Again, this shows how the r_m value changes with the slope that the system would rest on. The b_l value is the bar length, allowing for a normalized Σr_m.*

the steepest gradient. This configuration requires that the center of mass of the system needs to travel further to generate the needed torque to cause the system to roll, while simultaneously allowing for the movable masses to have larger moment arms to prevent movement.

Another advantage of this approach is that a payload can be added to the model with minimal variation to the approach. Assuming that the payload is added to the model's center of mass, that means

$$m_c = m_s + m_p \tag{9}$$

where m_c is the mass at the center of mass of the system, m_s is the mass of the system, and m_p is the mass of the payload of the system. In addition, this system can artificially lower the payload mass by using portions of the payload as part of the movable masses. For example, the motors and batteries themselves could comprise the movable masses, and other hardware, such as circuit boards or computer chips, could also be incorporated into the movable masses. Since this design would allow the movable masses to position themselves before impact to minimize the shock on the movable masses themselves, the sensitive hardware would be protected.

4. Controller implementation

The quasi-static controller for this system solves a few key questions that will be discussed in the following section. First, given a valid desired node, meaning a node that is initially connected to two support polygon nodes but is not part of the support polygon, how can the masses be commanded such that the system rolls to that node? Second, how is the commanded node chosen? – meaning if the model is commanded to move in a specific direction (up, down, left, right), how is the node chosen to maximize the system's movement in that direction? Finally, how does the controller determine what direction to
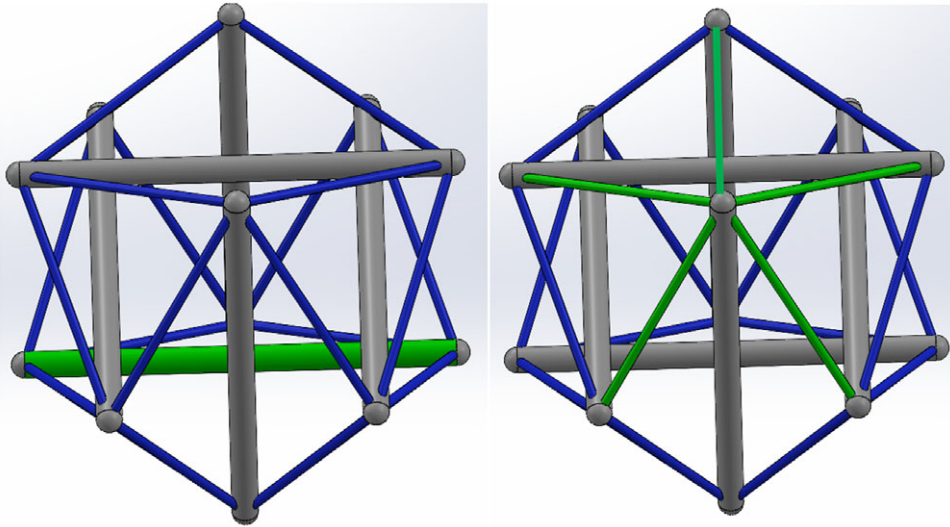
Figure 8. *The images show a node connection. The image on the left highlights the unconnected bar, while the image on the right shows the string connections coming off the desired node.*

move to reach a desired target? As shown in this section, an answer to each of these questions can be used to create a quasi-static controller.

The controller described below will operate correctly assuming that the following conditions are met. This controller assumes that the system is in a resting state. This is because the system is assumed to roll over an axis that is parallel to the ground, so for that to be true, the system needs to be in a state of rest. For this solution to converge on the target point, we also assume that the system does not over-roll, which could cause it to get stuck in a state where it is continually oscillating between two orientations. Finally, we assume that the state of the system is known in the global frame at each interval of time, as well as the location of the target we are attempting to reach.

Each node in this icosahedral tensegrity system is connected to five other nodes via cables, as illustrated in Fig. 8. Two of these nodes are on the same bar, but each of the other nodes is part of unique bars. Including the node that the model desires to move toward, the cable connections inform the model where to move four of the six masses. This is done by choosing the nodes from the string connectivity matrix that correspond to the desired node.

The location of the other masses – the mass on the bar not represented and the mass on the bar represented twice, which is shown in Fig. 8 – do not cause significantly different moments on the system no matter the node that they are placed at. This is due to the geometrical symmetry of the system, such that both nodes on those two respective bars produce nearly the same moment arm.

With that in mind, then the placement of the masses only matters if the user wants to prevent or foster momentum rolls, where the system does not stop at the next triangular polygon support and instead moves past it to a new orientation. If this is the case, the masses should go to the node that has the smaller moment arm. This will act as a damper for the system, slowing the initial roll and preventing an orientation that would cause the model to roll again once it has initially settled. Over-rolling happens quite readily in practice since a combination of the momentum from the system changing orientation and the locations of the masses essentially primes the system to continue to roll. Locating the movable masses to gain as much potential energy during a roll can help to prevent that behavior.

Figure 9 shows the system in a configuration where the masses are in the center of each bar, with the masses represented by the blue circles. The distribution of the forces created by the masses on each respective bar is shown with the blue arrows. The target node is shown as the red circle in the image, and the desired location of the other masses for movement is shown as a yellow circle. The left image
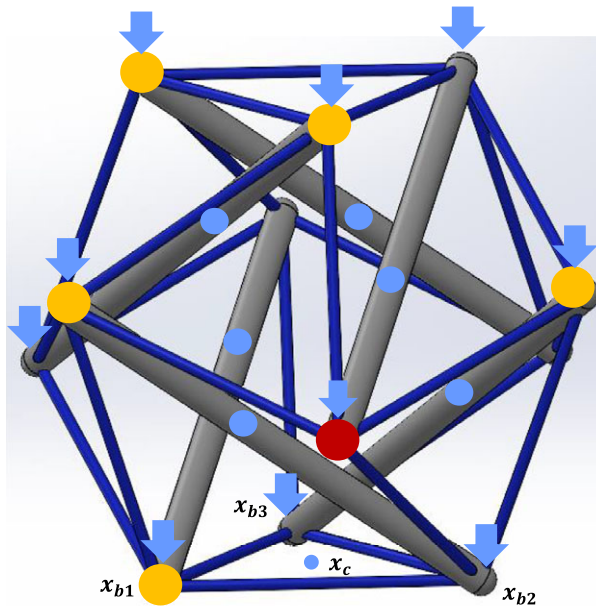
Figure 9. *This image shows the tensegrity system in its initial configuration. In this configuration, each movable mass starts in the center of their respective bars as shown by the blue circles. The red circle is the node that the system will rotate to such that at the end of the rotation event, the red circle will be on the ground. The yellow circles represent the nodes that the other movable masses must be at for the desired rotation to occur. Note that some forces were not shown for image clarity.*

in Fig. 10 shows how the forces on each node change when the masses are moved, and the image on the right shows what the system looks like after the masses have moved. For clarity, the relevant masses should move entirely to either the desired node or to one of the direct connections. This makes it so the system has the maximum moment around the edge of the support polygon, so the system rotates around the correct pivot. Masses on the bars that are not assigned a position by the controller default to the highest point on the bar.

As discussed in Section 3, there are only two unique orientations for an icosahedral tensegrity structure: the spiral configuration and the parallel configuration. Each configuration has a unique ZMP (zero moment point) condition, with the spiral configuration having an identical force requirement in every direction and the parallel configuration having an identical requirement in two directions and a larger force requirement in the other two directions. These requirements were given as the proportion of the mass of each movable mass to the mass of the entire system in Eqs. 5–7. If each of these conditions is satisfied, there is no unreachable region in this scenario. Otherwise, the only unreachable rotation would be about the worst-case scenario as described by Eq. (5). The algorithm described beyond these points assumes that there is no unreachable region and that movement occurs on flat ground.

Now that an algorithm has been established to allow the system to rotate to a feasible node, the next step is to algorithmically determine which node the system should move to. The support polygon of the icosahedral structure is composed of a triangle of cables connecting three nodes at $y = 0$ (i.e., the ground). This is the current support polygon of the system. Each support polygon node has its own x and z locations, each of which is unique to that node at that system state. Each of the string connections is vector about which rotation is possible. Thus, at any arbitrary step, there are three possible vectors for the system to rotate about.

To determine which way to rotate the system, the following process is followed. The first step is to choose the node from the support polygon that is closest to the target. If the system desires to rotate
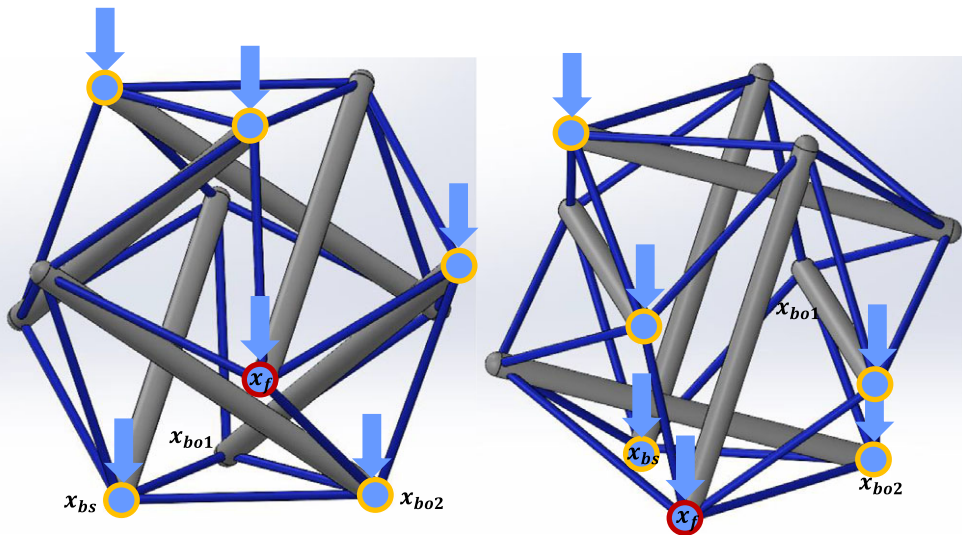
Figure 10. *This figure expands on the system shown in Fig. 9. In the left image, all the masses are in the desired location to induce rotation in the structure. As shown, there are only forces on six nodes instead of every node. After all the masses have moved, the system rotates and ends in the state as shown in the right image.*

such that its center of mass translates in the positive x direction, then it would choose the node with the largest positive x value. Choosing this node disqualifies the vector between the two other nodes as the rotation vector since rotation about that vector would cause the system to go in the direction opposite of the desired location.

Once that node is chosen, the vectors representing the directions between the chosen node and the other two nodes are created by subtracting the location of the other two nodes from the closest node. The vector that is most perpendicular to the desired direction is then chosen since rotation about that vector would lead the system to change its position in the desired direction by the largest amount. This determination is done by choosing the minimal absolute value of the dot product between the direction vector and the calculated vectors between the first node chosen in the support polygon and the other potential nodes that remain in the triangular support polygon.

Once the rotation vector has been determined, the target node can then be acquired. Since the target node is the node that is connected to both nodes that make up the rotation vector and cannot already be part of the polygonal support of the system, there is only one node that can be selected. With the target node determined, the movable masses are commanded to go to the nodes closest to the desired node. The closest nodes are the nodes connected to the desired node, as described at the beginning of this section.

Finally, to evaluate the controller to move the system to a desired target, the following methodology is used. First, a random target in *x-z* space is created for the model to move to. Then, the system determines the vector from the target point to its center of mass. That vector is then used to determine which node command would lead to the system moving the most in that direction.

The choice of direction is made using a greedy algorithm that finds the minimum distance from each node to the target in the positive and negative *x* and *z* directions. Once those minimum distances are found, it chooses the largest one and has the system roll in that direction. If all the distances are negative (meaning that its support polygon contains the required target), then the system makes it to the target, which is the success criteria of the algorithm. If further simulation is required, then another random point is chosen, and the simulation continues. The minimum negative *z* and positive z are not simply the

---

Algorithm 1: Icosahedral System Control

Variables:

> d = direction
>
> x_c = current position
>
> x_{b1} = base 1 node position
>
> x_{b2} = base 2 node position
>
> x_{b3} = base 3 node position
>
> x_b = selected base
>
> x_{bo1} = other base 1
>
> x_{bo2} = other base 2

Input: Initial Position(x_o), Target Position(x_f)

While (target not over desired location)

> **Determine** direction system must travel to reach target.
>
> $$d = x_f - x_c$$
>
> **Determine** which node that is part of the polygonal support is closest to the target.
>
> $$x_{bs} = min(x_f - x_{b1}, x_f - x_{b2}, x_f - x_{b3})$$
>
> **Determine** which vector that contains the closest node is the most perpendicular to the target**.**
>
> $$min((x_{bs} - x_{bo1}) \cdot d, (x_{bs} - x_{bo2}) \cdot d)$$
>
> **Determine** which node is the target node**.**
>
> > Select common node of b_s and the other base
>
> **Command** masses to nodes based on node the target node.
>
> **Wait** for system to roll.

End while

---

negation of the other because it is the minimum of several nodes, thus necessitating that those values be different.

The current implementation of this controller has some drawbacks that can be addressed in future work. The primary drawback is that it is possible for the system to get stuck in a loop. This loop occurs because when the system moves in the commanded direction, it also moves in another direction. This means that it is possible for the controller to command the system to go to the right, and it moves right and down. In the next step, the controller commands the system to move up, and it moves up and to the left, repeating the same step in perpetuity. New methods to address this could be to keep track of movements and then choose a suboptimal one if repetition occurs or change the controller to choose the rotation vector based on total minimum distances to the target.

Other controllers have been created for tensegrity systems that work on collision avoidance and other path planning optimizations. Such techniques could be used to refine this controller in the future [41–43]. Another application for future work could be to create a state estimator to test this controller based on noisy estimates of its actual state.

In addition, this system can be shown as a block diagram. This block diagram will follow a different logic than a typical diagram, since the controller is quasi-static, but can be represented by Fig. 11 below.

5. Results

In this section, the responses of the simulated and real systems are obtained and analyzed. Simulation testing was performed to see if the system can control specified targets and how the dynamics of the system work over many control iterations. This simulation was derived from the simulation found in the
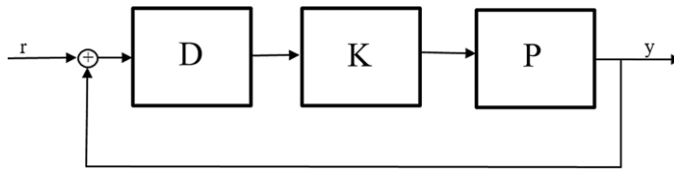
Figure 11. *The figure shows a block diagram representation of the controller and the system. This system is different from a typical block diagram since the system is quasi-static. As such, the reference r for this system is the target location, the output y is the current position of this system, and the D block is a delay block that allows the system to complete its current operation. After the delay is complete, the signal enters the controller block K. The output then commands the system, block P, to move to the next point and the process repeats.*
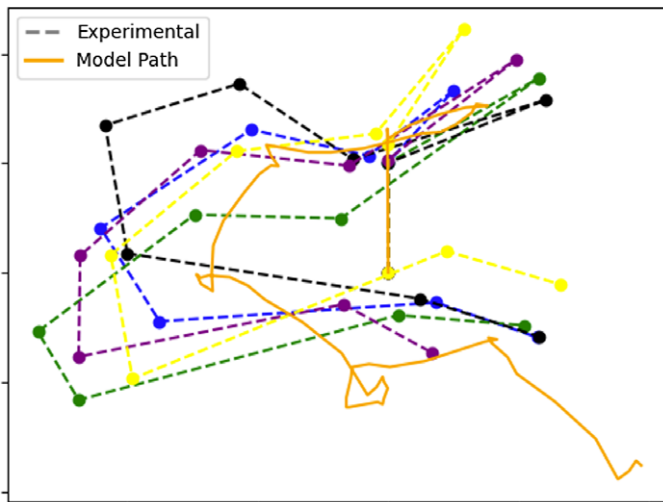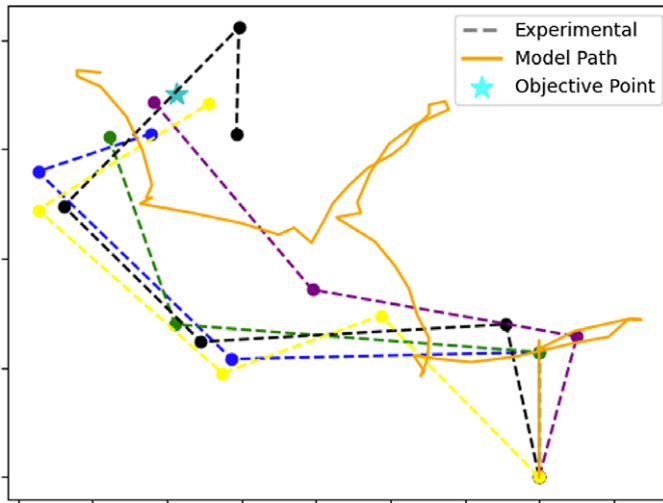

Figure 12. *The system is commanded to go up twice, left twice, down twice, and right twice, essentially commanding it to move in a square. Due to the momentum rolling of the system, the distances moved in each direction are different since the robot often moves more than one face at a time. The dashed lines are experimental results, while the solid orange line is the result found by the model.*

MOTES code [40] and modified as explained in Section 2. This simulation modeled the dynamics of the unmodified icosahedral tensegrity system, with the movable masses modeled as point forces at different nodes of the system. By changing where the forces acted, the system was capable of locomotion.

Two tests are performed on a physical system to compare experimental results to those found in the simulation. The physical system is described later in this paper. The first test is to command the system to move in an arbitrary direction, while the second test exercises the control algorithm discussed above.

Figures 12 and 13 show the *x,y* path that the centroid of the robot took on these two tests. In Fig. 12, the system was arbitrarily commanded to go up twice, left twice, down twice, and right twice, essentially to move in a square. The experimental results of five different runs, shown as dashed lines, are compared to the expected path from the analytical model. Two things should be noted. First, due to the triangular shape of the polygonal support, the system will never travel in a straight line. When the system is commanded to move up, the algorithm determines which desired node will cause the system to move up the most. We see that the first command goes directly up, whereas the second command causes the system to go up and to the right. The third command is to go left, which causes the system to roll back to the previous location. Second, it is important to note that due to momentum, sometimes the system

Figure 13. *The system is commanded to go to an objective point using the control method. The dashed lines are experimental results, while the solid orange line is the result found by the model.*

will roll twice or three times, rather than stopping at a given location. This can be seen in the second and third commands, where the system rolls an extra roll to the left. This phenomenon is seen in both the analytical model and as the experimental testing. A major source of error between the experimental tests and the analytical model is the propensity of the experimental test to slide and shift during rotation, while the analytical model assumes perfect contact with the ground, causing a divergence between the paths as the tensegrity system progresses along its path.

In Fig. 13, the system is commanded to go to a randomly generated target location up and to the left of the initial location. Using the procedure also outlined in Section 4, the system was able to move as directed and was able to transverse to the random target, though the path of the experimental tests differed from those expected from the model.

To ensure the feasibility of this movable mass model, a real-world structure was created using aluminum pipes and large movable masses, as seen in Fig. 14. These masses were connected to a motor that allowed the masses to move the length of each bar. In this iteration of the design, the masses were mounted externally to the bar, allowing for easier access to the masses. Each of these masses is 60% of the mass of the total mass of the system.

The fabricated system differs from the theoretical system in several ways. The physical movable masses are controlled using stepper motors, where the motors are part of the movable mass. While the steppers can control the location of the mass along the entire bar, they are not able to move to the very tip of each bar due to the size of the motors. Each movable mass can reach to approximately within 3 cm of the way to the end. When the masses were near the tip of the bar, they could also contact the ground outside of the support polygon, changing the ZMP of the system, as well as elevating parts of the system, effectively creating a slope that the system would need to overcome. Because of these deficiencies, rotation along the worst-case scenario was not tested since the movable masses did not have enough mass to promote movement analytically. During testing, after each roll, the system slides slightly over the ground, causing some error when compared to the theoretical model. Any change to the rotation of the system can cause large drift, and so the system could benefit from a high-fidelity state estimation, which is the subject of a future paper. Even with these inefficiencies in the physical system, the system could move as intended. Assuming that the state estimation is correct, the control system will pick the optimal solution for the next position of the robot.

In both other cases, the spiral case, and the better parallel bar case, we were able to achieve movement using the movable masses, as seen in Fig. 15. When controlled using the control algorithm described
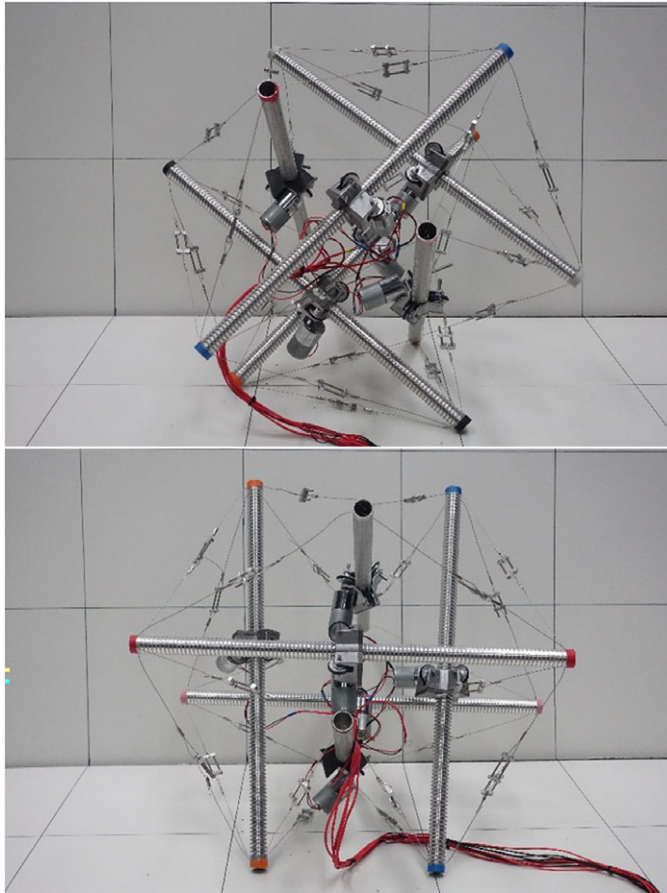
Figure 14. *These images show the physical system implemented with the movable masses. The first image shows the system in its spiral configuration where any rotation would result in a support polygon with three distinct bars touching the ground. The bottom image shows the system in its parallel configuration where rotation about the orange and blue nodes would put the opposite node of the same bar on the ground. This is the worst-case scenario for rotation since the number of moment arms available to move the system is reduced.*

in Section 4, the robot was capable of rotating to the desired node. When testing, we also saw behavior from the model in which it experiences momentum rolling as seen in the simulation. The one caveat with it is that the robot experienced more extensive momentum rolling than it was previously thought it could experience. One of the explanations for this could be that the masses and motors create a less predictable support polygon than would otherwise be established on this robot, meaning that additional momentum rolling could occur. Future work to minimize this unwanted rolling could include finding a reliable way to increase the damping of the system that does not lead to significant shape deformations.

In addition to the simple example above of how the system moved with the controller, the system was tested for a few additional iterations. The robot was tested moving in several different trajectories. These trajectories were based on an 8'X8' grid, where the robot started at one of the corners. The desired location was either one of the closest corners, the middle of an opposite edge, or the farthest corner from the robot, as seen in Fig. 16. Using the controller described in Section 4, each trajectory was attempted, as seen in Fig. 17. Success was defined as the robot straddling the desired location. In each trial, the robot was able to straddle the location using the controller logic. This model, with the movable masses too
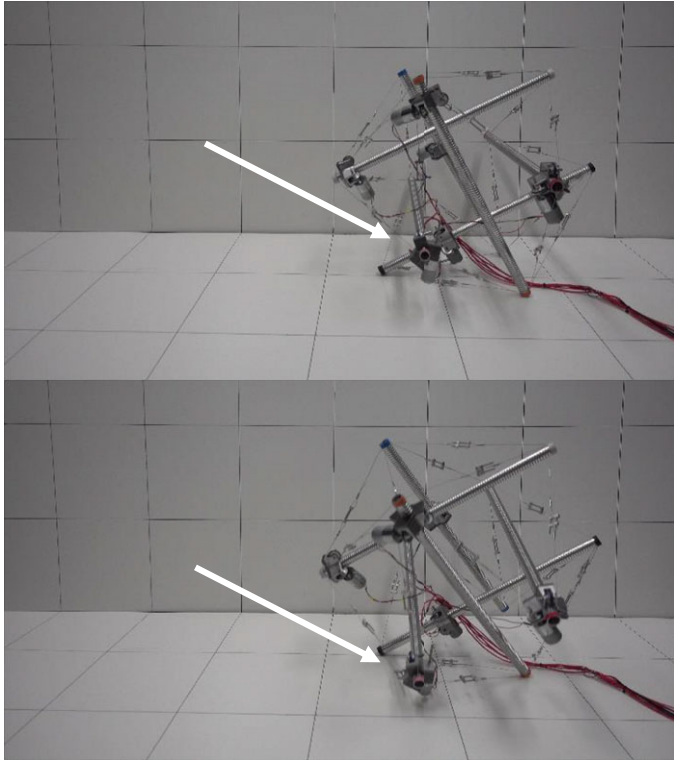
Figure 15. *This figure shows the movable masses in place to have the robot roll to the pink node (designated with arrow) in the images. The first image shows the system initiating the roll once all the masses are in place, while the second image shows the system in the middle of the rolling motion. In this image, the system is in a parallel configuration.*
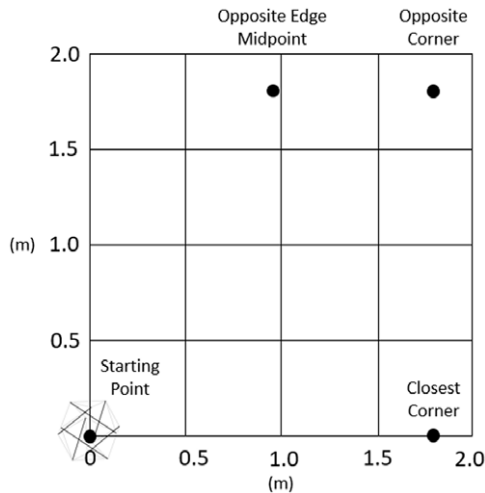
Figure 16. *This figure shows the desired locations for multiple tests. For each test, the robot starts at the origin and is commanded to reach the desired location.*

(a)   (b)   (c)
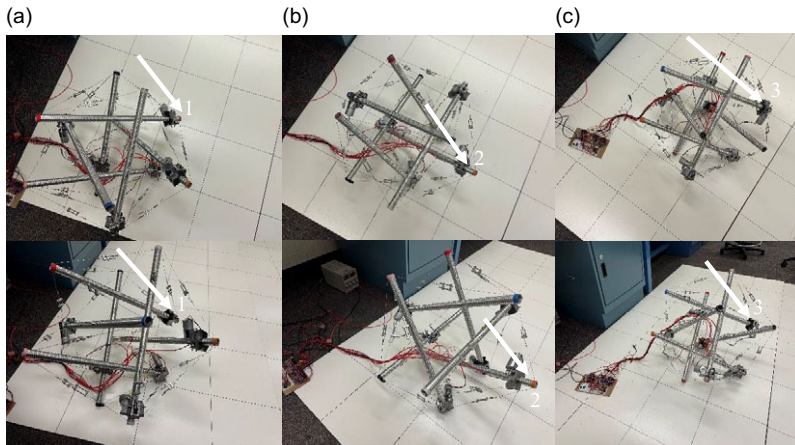
Figure 17. *These images show the control algorithm selecting nodes to control that would lead the robot to the desired location, which location is at the far corner of the board. The system is in its initial configuration in column (a). Column (b) shows the first step that the controller commanded to get to the desired location, and column (c) shows the next step. The desired node for the first step is node 1, and for the second step, it is node 2. The controller would then select node 3 as the next desired node, and the process would iterate until the system reached the desired location. For reference, the system starts in a spiral configuration, transitions to a parallel configuration, and ends in a spiral configuration for columns a, b, and c, respectively.*

light to overcome the worst-case scenario, was able to overcome this movement by using a suboptimal route and going around that configuration. At the end of the trajectory, the final robot state is measured on the grid and is compared to the desired location. Error is then calculated as the difference between the desired location and the final position of the robot. Error is acceptable if the final point is located within the final polygonal support of the structure. Data from these tests are found in Table 1.

After verifying the flat case analysis, the sloped case was considered. Since this physical system has deficiencies in how far out the movable masses can move, uphill testing was not performed. Instead, the maximum downhill stability case was focused on, with the following results.

The way we tested this scenario was by locating the masses according to the control algorithm to have the robot act as if it were traveling straight uphill. This configuration maximizes the moment arms that prevent the robot from rolling down the hill. We then incrementally increased the slope until the robot rolled down the incline.

Figure 18 shows the results of that testing. In the configuration in which the axis of rotation is perpendicular to the slope of the decline, testing validated that this system could remain stationary on a slope of 31 degrees. The reason that it is less than the predicted amount is due to the moment arms used by the masses are smaller than the moment arms of the best-case scenario. Furthermore, in the testing that was done on this system, once the slope began to become large, the structure began to slip down the slope even before it began to roll. In addition, this is also a worst-case scenario for this robot. If the axis of rotation was not facing down the steepest gradient, the robot would be able to prevent rolling down an even steeper slope, as described in Figs. 6 and 7.

Overall, the results verify that the model works and suggest that the equations have the limitation of overestimating the moment arms of the system, especially when the system cannot move the masses all the way to the extremes of the bars. Better system design could remedy this problem if external masses are designed with slots where the holes are. This could also potentially reduce momentum rolling if the movable mass hangs slightly beyond the bar, which would lead to the mass impacting the ground sooner than the bar would, reducing the overall inertia of the system when an inertial roll could begin. In essence, this testing gives a large degree of confidence to the calculated motion models.

Table 1. *The following table shows the feasibility of the controller in reaching desired points following different trajectories. These three trajectories were chosen since they can be used as an overdefined basis for any other flat trajectory. Some combination of these trajectories can be combined to create any other flat trajectory, meaning that this test is sufficient to show that this controller can be used to reach any desired point that has a flat trajectory.*

Trajectory	Final expected location (m)	Final actual location (m)	Error from actual (m)
Closest corner	(1.83, 0)	(1.71, 0.03)	0.13
		(1.85, -0.02)	0.03
		(1.87, 0.10)	0.10
		(1.92, 0.00)	0.12
		Average error	0.09
Opposite edge midpoint	(1.83, 0.91)	(1.61, 0.98)	0.23
		(1.81, 1.10)	0.19
		(1.73, 0.97)	0.12
		(1.86, 0.85)	0.09
		Average error	0.16
Opposite corner	(1.83, 1.83)	(1.84, 1.81)	0.02
		(1.89, 1.79)	0.07
		(1.97, 1.84)	0.15
		(1.80, 1.80)	0.06
		Average error	0.08
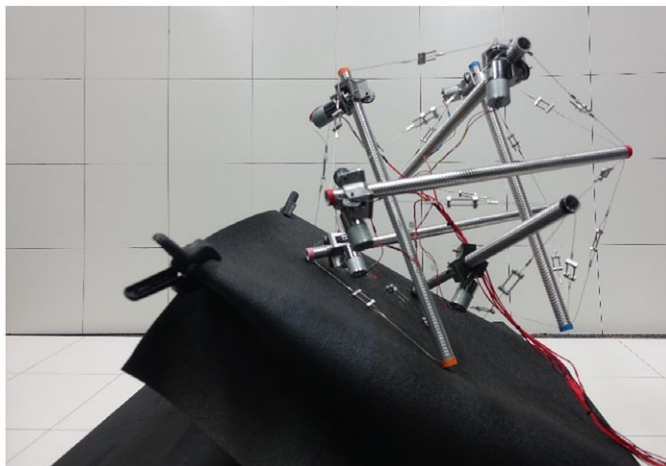
Figure 18. *The image above captures the testing results when the tensegrity system was tested on a slope. The slope shown above is a 31-degree decline that this tensegrity system could remain stationary on.*

6. Summary and final remarks

Traditional approaches for controlling tensegrity structures typically require the actuation of the tensegrity tensile members to allow for the movement of the system. This approach leads to complicated dynamics. The method presented in this paper circumvents that challenge by using only six movable masses within the bars of the system as the means of locomotion, which means that the dynamics of the system do not change over each iteration of the algorithm. In addition, this strategy also provides a

robust way of accounting for a payload, knowledge of the maximum slopes the system can account for, and a simple controller for controlling the robot.

Movable masses are a valid strategy as a means of system locomotion without the masses of the movable masses being significantly higher than that of the system. This was validated in this paper through mathematical analysis of the moments required to move the system, as well as through a physical model of the system being controlled using the control methodology created in Section 4. In future work, the mass of the movable masses could be reduced further if the momentum of the masses is utilized to move the system or if proper system design uses as much of the payload mass as possible as part of the movable masses.

In essence, this method can cause the system to rotate, move in a commanded direction, and control the system to a desired location. It can achieve this with simple and computationally cheap control algorithms. This simplicity is desired because the algorithm can entirely be implemented on a microcontroller local to the system, without needing to ship the data to an offsite computer.

This model also has a straightforward way of accounting for the mass of a payload in the movement of the structure. The mass of the payload can be distributed out to the movable masses, while still protecting the system's most critical components from shock, allowing the system to operate under harsher mechanical conditions.

Finally, this system also has very predictable behavior on slopes. This knowledge is very useful in path planning algorithms if the gradient of the landscape is known so that an achievable path can be planned. This is especially useful if the robot is known to only need to roll downhill since this robot can handle going down steep inclines well.

Future work will include making a physical system that can move the masses in the requisite manner to allow for real-time control. This involves designing a new movable mass-bar system and determining how a controller can use the desired measurements (such as velocity) low enough to prevent momentum rolling. Additionally, a new controller will have to be designed to work with the limited inputs that the physical system will have. Another reality that this controller will need to be capable of dealing with is that in practice, there are six edges along which the system will not be able to rotate. Additional work could include mass momentum methods to allow the system to rotate in a more controlled manner when desired for more optimal control. As described above, this system could also benefit from the introduction of an accelerometer, which could be combined with knowledge about the system's kinematics and geometry within a Bayesian filter to allow the system to better understand its pose. This concept can be expanded upon further by changing allowed rotations depending on the slope that the robot is on. For example, on a steep slope, it would be possible for a robot to rotate around the nominal worst-case scenario.

In addition to the ideas stated above, this type of system would be a good candidate to use reinforcement learning for path planning and actuation of masses since such an approach would treat the terrain as a maze and it would be possible to learn on that map [43].

Finally optimizing the system to be impact resistant would be a good way to utilize the advantages inherent in this system. The movable mass solution, along with initial payload location optimization, would allow for this system to be optimized very well for high-shock environments.

In conclusion, the movable mass solution described in this paper has several unique advantages that can be leveraged when creating a robot that is highly impact resistant. Much of the locomotion mechanics of this system is readily understood through the system's geometry and has unique properties that can be leveraged depending on the mission it undertakes.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S0263574724001139.

References

[1] C. Sultan, "Tensegrity structures: Sixty years of art, science, and engineering," *Adv. Appl. Mech.* **43**, 69–145 (2009). doi: 10.1016/S0065-2156(09)43002-3.

[2] K. Nagase and R. E. Skelton, "Minimal Mass Design of Tensegrity Structures," **In:** *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems* (International Society for Optics and Photonics, 2014).

[3] R. Goyal, E. A. Peraza Hernandez and R. E. Skelton, "Analytical study of tensegrity lattices for mass-efficient mechanical energy absorption," *Int. J. Space Struct.* **34**(1-2), 1–19 (2019). doi: 10.1177/0956059919845330.

[4] K. Garanger, I. del Valle, M. Rath, M. Krajewski, U. Raheja, M. Pavone and J. Rimoli, "Soft tensegrity systems for planetary landing and exploration," *Earth and Space 2021* (2020) pp. 841–854.

[5] J. J. Rimoli, "A reduced-order model for the dynamic and post-buckling behavior of tensegrity structures," *Mech. Mater. IUTAM Symp. Dyn. Instabilities Solids* **116**, 146–157 (2018). doi: 10.1016/j.mechmat.2017.01.009.

[6] P. Wenger and D. Chablat, "Kinetostatic analysis and solution classification of a class of planar tensegrity mechanisms," *Robotica* **37**(7), 1214–1224 (2019). doi: 10.1017/S026357471800070X.

[7] C. Paul, F. J. Valero-Cuevas and H. Lipson, "Design and control of tensegrity robots for locomotion," *IEEE Trans. Robot.* **22**(5), 944–957 (2006). doi: 10.1109/TRO.2006.878980.

[8] Y. Tian, Y.-A. Yao, W. Ding and Z. Xun, "Design and locomotion analysis of a novel deformable mobile robot with worm-like, self-crossing and rolling motion," *Robotica* **34**(9), 1961–1978 (2016).

[9] N. Kastor, R. Mukherjee, E. Cohen, V. Vikas, B. Trimmer and R. White, "Design and manufacturing of tendon-driven soft foam robots," *Robotica* **38**(1), 88–105 (2020).

[10] C. Sorensen and M. D. Killpackl, "Soft Robot Shape Estimation: A Load-Agnostic Geometric Method," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023).

[11] H. P. Sanders and M. D. Killpackl, "Dynamically Feasible Trajectory Generation for Soft Robots," *2023 IEEE International Conference on Soft Robotics (RoboSoft)* (IEEE, 2023).

[12] K. Doney, A. Petridou, J. Karaul, A. Khan, G. Liu and J. Rieffel, "Behavioral Repertoires for Soft Tensegrity Robots," *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* (2020) pp. 2265–2271. doi: 10.1109/SSCI47803.2020.9308218.

[13] A. Zhakatayev, B. Abdikadirova, S. Sarmonov and H. Varol, "Dynamics of tensegrity robots with negative stiffness elements," *IEEE Access* **8**, 187114–187125 (2020). doi: 10.1109/ACCESS.2020.3031279.

[14] K. Kim, L.-H. Chen, B. Cera, M. Daly, E. Zhu, J. Despois, A. K. Agogino, V. SunSpiral and A. M. Agogino, "Hopping and Rolling Locomotion with Spherical Tensegrity Robots," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016) pp. 4369–4376. doi: 10.1109/IROS.2016.7759643.

[15] X. Wang, A. Luo and H. Liu, "Design and analysis of a double-helix tensegrity spherical lander," *Mech. Res. Commun.* **129**, 104091 (2023). doi: 10.1016/j.mechrescom.2023.104091.

[16] K. Caluwaerts, J. Despraz, A. Işçen, A. P. Sabelhaus, J. Bruce, B. Schrauwen and V. SunSpiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *J. R. Soc. Interface* **11**(98), 20140520 (2014).

[17] A. K. Agogino, V. SunSpiral and D. Atkinson, Super Ball Bot-structures for planetary landing and exploration. No. HQ-E-DAA-TN63111 (2018).

[18] D. Surovik, K. Wang, M. Vespignani, J. Bruce and K. E. Bekris, "Adaptive tensegrity locomotion: Controlling a compliant icosahedron with symmetry-reduced reinforcement learning," *Int. J. Rob. Res.* **40**(1), 375–396 (2021).

[19] B. Varga, J. K. Tar and R. Horváth, "Fractional order inspired iterative adaptive control," *Robotica* **42**(2), 482–509 (2024). doi: 10.1017/S0263574723001595.

[20] J. Huo, R. Lin and M. Yang, "Design and validation of a novel adaptive motion control for a pendulum spherical robot," *Robotica* **41**(7), 2031–2049 (2023). doi: 10.1017/S0263574723000280.

[21] S. Bagherzadeh, H. Karimpour and M. Keshmiri, "Neighboring extremal nonlinear model predictive control of a rigid body on SO(3)," *Robotica* **41**(4), 1313–1334 (2023). doi: 10.1017/S0263574722001722.

[22] S. Bu, L. Yan, X. Gao, G. Wang, P. Zhao and I.-M. Chen, "Design and motion control of spherical robot with built-in four-wheel omnidirectional mobile platform," *IEEE T. Instrum. Meas.* **72**, 1–10 (2023). doi: 10.1109/TIM.2023.3267524.

[23] K. Schröder, G. Garcia, R. Chacón, G. Montenegro, A. Marroquín, G. Farias, S. Dormido-Canto and E. Fabregas, "Development and control of a real spherical robot," *Sensors* **23**(8), 3895 (2023).

[24] M. Chen, X. Bai and R. E. Skelton, "Minimal mass design of clustered tensegrity structures," *Comput. Method. Appl. Mech. Eng.* **404**, 115832 (2023). doi: 10.1016/j.cma.2022.115832.

[25] K. W. Moored and H. Bart-Smith, "Investigation of clustered actuation in tensegrity structures," *Int. J. Solids Struct.* **46**(17), 3272–3281 (2009).

[26] S. Ma, M. Chen and R. E. Skelton, "Dynamics and control of clustered tensegrity systems," *Eng. Struct.* **264**, 114391 (2022). doi: 10.1016/j.engstruct.2022.114391.

[27] Z. Kan, H. Peng, B. Chen and W. Zhong, "Nonlinear dynamic and deployment analysis of clustered tensegrity structures using a positional formulation FEM," *Compos. Struct.* **187**, 241–258 (2018). doi: 10.1016/j.compstruct.2017.12.050.

[28] H. Peng, H. Yang, F. Li, C. Yang and N. Song, "A unified framework for mechanical modeling and control of tensegrity robots," *Mech. Mach. Theory* **191**, 105498 (2024). doi: 10.1016/j.mechmachtheory.2023.105498.

[29] K. Kim, A. K. Agogino and A. M. Agogino, "Rolling locomotion of cable-driven soft spherical tensegrity robots," *Soft Robot.* **7**(3), 346–361 (2020).

[30] R. Wang, R. Goyal, S. Chakravorty and R. E. Skelton, "Model and data based approaches to the control of tensegrity robots," *IEEE Robot. Autom. Lett.* **5**(3), 3846–3853 (2020). doi: 10.1109/LRA.2020.2979891.

[31] A. Iscen, A. Agogino, V. SunSpiral and K. Tumer, "Learning to Control Complex Tensegrity Robots," *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems* (2013).

[32] T. Tsabedze, E. Hartman and J. Zhang, "A compact, compliant, and biomimetic robotic assistive glove driven by twisted string actuators," *Int. J. Intell. Robot. Appl.* **5**(3), 381–394 (2021).

[33] K. Nagase and R. E. Skelton, "Minimal Mass Design of Tensegrity Structures," *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2014, 9061:90610W* (International Society for Optics and Photonics, 2014).

[34] S. Chesi, Q. Gong and M. Romano, "Aerodynamic three-axis attitude stabilization of a spacecraft by center-of-mass shifting," *J. Guid. Control. Dyn.* **40**(7), 1613–1626 (2017).

[35] L. He, X. Chen, K. D. Kumar, T. Sheng and C. Yue, "A novel three-axis attitude stabilization method using in-plane internal mass-shifting," *Aerosp. Sci. Technol.* **92**, 489–500 (2019). doi: 10.1016/j.ast.2019.06.019.

[36] V. Böhm, T. Kaufhold, F. Schale and K. Zimmermann, "Spherical Mobile Robot Based on a Tensegrity Structure with Curved Compressed Members," *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Banff, AB, Canada (2016) pp. 1509–1514. doi: 10.1109/AIM.2016.7576984.

[37] T. Rhodes, C. Gotberg and V. Vikas, "Compact shape morphing tensegrity robots capable of locomotion," *Front. Robot. AI* **6**, 111 (2019). doi: 10.3389/frobt.2019.00111.

[38] T. Rhodes and V. Vikas, "Compact Tensegrity Robots Capable of Locomotion Through Mass-Shifting," *Proceedings of the ASME. 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 5B: 43rd Mechanisms and Robotics Conference*, Anaheim, California, USA (2019). doi: 10.1115/DETC2019-98513.

[39] T. Kaufhold, F. Schale, V. Bohm and K. Zimmermann, "Indoor Locomotion Experiments of a Spherical Mobile Robot Based on a Tensegrity Structure with Curved Compressed members," *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)* (IEEE, 2017).

[40] R. Goyal, M. Chen, M. Majji and R. Skelton, "MOTES: Modeling of tensegrity structures," *JOSS* **4**(42), 1613 (2019). doi: 10.21105/joss.01613.

[41] X. Xu, F. Sun, Y. Luo and Y. Xu, "Collision-free path planning of tensegrity structures," *J. Struct. Eng.* **140**(4), 04013084 (2014).

[42] S. H. Juan, R. E. Skelton and J. M. M. Tur, "Dynamically Stable Collision Avoidance for Tensegrity Based Robots," *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots* (IEEE, 2009).

[43] D. Surovik, K. Wang, M. Vespignani, J. Bruce and K. E. Bekris, "Adaptive tensegrity locomotion: Controlling a compliant icosahedron with symmetry-reduced reinforcement learning," *Int. J. Robot. Res.* **40**(1), a375–396 (2021).