

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

On the Locomotion of Spherical Tensegrity Robots

Permalink

<https://escholarship.org/uc/item/7p36z1x5>

Author

Kim, Kyunam

Publication Date

2016

Peer reviewed|Thesis/dissertation

On the Locomotion of Spherical Tensegrity Robots

by

Kyunam Kim

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alice M. Agogino, Chair

Professor J. Karl Hedrick

Professor Claire J. Tomlin

Fall 2016

On the Locomotion of Spherical Tensegrity Robots

Copyright 2016
by
Kyunam Kim

Abstract

On the Locomotion of Spherical Tensegrity Robots

by

Kyunam Kim

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Alice M. Agogino, Chair

This dissertation studies novel robotic systems based on tensegrity structures, with an emphasis on their locomotion capabilities. Naturally compliant tensegrity structures have several unique properties that are advantageous for co-robotic or soft robotic platforms; they are lightweight, deployable, robust, and safe. By leveraging these distinctive features of tensegrity structures, tensegrity robots are expected to overcome the barriers for today's robots. In this regard, tensegrity robots have been envisioned for a wide range of new applications that have not been explored before, including assistive and rehabilitative healthcare, search and rescue, and planetary space exploration, to name a few. In order to be actually deployed for these applications, tensegrity robots should have mobility in the first place. For this reason, two modes of locomotion are examined for spherical tensegrity robots in this research: rolling and hopping.

This research begins by presenting four hardware prototypes of spherical tensegrity robots that have been constructed at the Berkeley Emergent Space Tensegrities laboratory. Three of them (named TT-1, TT-2, and TT-3) are based on a six-rod tensegrity structure, and the last one (named T12-R) is based on a twelve-rod tensegrity structure. A six-rod tensegrity structure is the simplest three-dimensional tensegrity structure that has an outer shape similar to a sphere, and for this reason, the structure is chosen as a basis for the first three robots. However, the rolling speed of the TT-series robots is limited because their outer surfaces consist only of triangles, which forces them to move in a zig-zag way and lose their momentum as they do so. This motivated the development of T12-R whose outer surface consists of mostly rectangles. This geometry enables the robot to move in a straight line, and thus prevents the loss of momentum. The hardware designs of all four prototypes are described in details.

A spherical tensegrity robot rolls by deforming its shape and by shifting its center of mass. The study of tensegrity deformation, however, is not trivial and poses a unique problem because kinematics and statics of tensegrity structures are tightly coupled and need to be considered concurrently. This work develops two systematic ways of obtaining desirable deformations of spherical tensegrity robots for rolling. As a first step to both approaches, a condition on the center of mass of a spherical tensegrity robot that must be satisfied for the desirable deformations is stated. The first approach relies on a greedy search algorithm, and it quickly finds one deformation satisfying the condition. This algorithm was implemented

in the NASA Tensegrity Robotics Toolkit simulator and the outcome of the simulation was tested on TT-1 and TT-2. Our hardware experiments show that the robots can realize a piecewise continuous rolling motion with the deformations found in the simulation. However, it was also observed that the robots occasionally fail to roll because the algorithm did not take reliability of the rolling motion into account when searching for the desired deformations.

To overcome this drawback, the second approach that combines a dynamic relaxation technique with a multi-generation Monte Carlo is proposed. It is known in the literature that the dynamic relaxation is well suited for solving for the deformation of tensegrity structures under non-uniform internal tension distributions. This work adapts the technique to find the deformations of the hardware robots, but with an explicit description of the rod constraint forces such that the convergence property of the technique is improved. The multi-generation Monte Carlo is then used to find a set of good deformations for reliable rolling by sampling and evaluating a number of deformed shapes through the dynamic relaxation. This procedure is simulated by using a custom-written software in MATLAB, and the results of the simulation as well as their validation on TT-2 are discussed in details. Furthermore, this latter approach is not limited to six-rod tensegrity robots only, and its generalization to other spherical tensegrity robots is presented and demonstrated with T12-R.

Hopping is another viable option for the locomotion of tensegrity robots as they can survive from significant impact shocks due to their structural compliance, while protecting the payloads they are carrying. This contrasts to many other rigid robots that are likely to break if they are dropped from a large height. Hopping could be especially useful when tensegrity robots are deployed for planetary exploration missions because it allows the robots to quickly travel long distances and to be less affected by ground conditions that are potentially unknown. To enable hopping, a tensegrity robot with a cold-gas thruster system is studied and its motion is simulated in this work. The simulation study of different hopping profiles on the Moon shows that longer hops are more energy-efficient but are subject to higher impulse at landing, which may lead to damaging the robot. Hence there is a trade-off between energy-efficiency and safety of the robot. This work also presents a path planning algorithm that is based on the A-star search algorithm, and it combines hopping and rolling for economical navigation on the lunar surface. A localization method based on height measurements of surroundings is also discussed.

Another observation from the simulation study is that a thrust vectoring mechanism is necessary to increase the fuel efficiency of the thruster-based tensegrity robot. This work introduces four mechanisms that could be used for this purpose, and among them, a system with two reaction wheels is extensively studied. A spin-axis stabilization problem of the reaction wheel system is formally posed, and the controller that orients the thrust to an arbitrary direction is developed. For this, the (z, w) -parameterization is used to describe the rotational kinematics of the thruster system, but this work modifies the parameterization to explicitly include the target orientation. Based on this description of rotational kinematics, the controller is designed by using the Lyapunov's direct method in conjunction with LaSalle's invariance principle. This controller is globally and asymptotically stable and the proof is given. The controller is simulated on an example thruster system and the results are provided.

To my beloved wife, Yunwon Choi,
who completes and sustains my life with her unwavering love and trust,
just like cables hold up a tensegrity with their tension.

Contents

Contents	ii
List of Figures	iv
List of Tables	xiii
1 Introduction	1
1.1 Tensegrity Structures	1
1.2 Tensegrity Robots	3
1.3 NASA Tensegrity Robotics Toolkit	5
1.4 Research Objectives	5
1.5 Outline of Dissertation	6
2 Hardware Design of Spherical Tensegrity Robots	8
2.1 Introduction	8
2.2 Design of Six-Rod Tensegrity Robots: TT-1, TT-2 and TT-3	9
2.3 Design of a Twelve-Rod Tensegrity Robot: T12-R	18
2.4 Conclusions	22
3 Precision Rolling of Spherical Tensegrity Robots	23
3.1 Introduction	23
3.2 Actuation Policy Development for Six-Rod Tensegrity Robots with Greedy Search Method	25
3.3 Actuation Policy Development for Six-Rod Tensegrity Robots with Multi-Generation Monte Carlo Based Learning Method	35
3.4 Actuation Policy Development for General Spherical Tensegrity Robots with Multi-Generation Monte Carlo Method	53
3.5 Conclusions	65
4 Thruster-Based Hopping Tensegrity Robots	66
4.1 Introduction	66
4.2 Thruster Evaluation	68
4.3 Simulations of Thruster-Based Mobility	70

4.4	Localization and Path Planning	76
4.5	Conclusions and Future Research	78
5	Orientation Control of Thruster Systems on Tensegrity Robots	80
5.1	Introduction	80
5.2	Attitude Kinematics	85
5.3	Dynamics of a Rigid Body with Two Reaction Wheels	90
5.4	Controls for Spin-Axis Stabilization	91
5.5	Conclusions and Future Research	100
6	Conclusions and Future Research	102
6.1	Towards Dynamic Rolling of Spherical Tensegrity Robots	103
6.2	Thrust Vectoring within Tensegrity Robots	108
	Bibliography	109

List of Figures

1.1	Tensegrity structures become more spherical if more rods are used to construct them. (a) A snub cube tensegrity built with 12 rods. (b) A truncated dodecahedron tensegrity built with 30 rods. (c) A three-frequency dodecahedron built with 90 rods. Images courtesy of Anthony Pugh [12].	4
1.2	Six-rod tensegrity robots simulated in NASA Tensegrity Robotics Toolkit (NTRT) [1], [5]. NTRT provides core methods to model, simulate, and control broad types of tensegrity robots [88].	5
1.3	A six-rod tensegrity structure has an outer shape similar to an icosahedron, or more precisely, to a Jessen's orthogonal icosahedron. Twelve-rod tensegrity structures can have several different forms, but only the one that has an outer shape similar to a rhombicuboctahedron is considered in this work.	6
2.1	Rapidly prototyped tensegrity robots that are developed at the Berkeley Emergent Space Tensegrities laboratory, University of California at Berkeley. The first three robots are based on a six-rod tensegrity structure and the last one is based on a twelve-rod tensegrity structure.	10
2.2	The outer surface of a six-rod tensegrity structure consists of eight closed equilateral triangles and twelve open isosceles triangles.	11
2.3	An edge of TT-1. The edge consists of serial connections of two looped elastic cords and a linear actuator. Spring snaps are used to connect the cord loops and actuator. Hog rings are used to fix the cord loops [1].	13
2.4	Connections at the end of a TT-2 rod. The hog rings used in TT-1 are replaced by customized clamps to facilitate length change of elastic cord loops [3].	14
2.5	A six-rod tensegrity structure with imaginary octants formed by green mutually-orthogonal planes. Each octant is distinguished by signs of accelerations in X, Y, and Z directions measured by an IMU. Blue triangles are closed triangles of the structure. The red arrow depicts the gravity vector measured when the robot is stationary. Based on the X, Y, and Z signs of the gravity vector with respect to the IMU's coordinate frame, we can figure out towards which octant the gravity vector is heading. We identify the closed triangle that is included in this octant as the current base triangle (white triangle in the figure).	16

2.6	An edge of TT-3 consists of an extension spring and an unstretchable string that goes into a hollow rod and is connected to a spool actuated by a rotary motor located inside of a capsule [4].	17
2.7	At the center of each TT-3 rod is a capsule containing electronic components [4].	17
2.8	The actual paths taken by six- and twelve-rod tensegrity robots when moving in a straight line. Because a six-rod tensegrity structure has only triangles on its outer surface, a robot based on this structure moves in a zig-zag way and switches its heading direction after each and every step. This causes large momentum loss and the structure is not suitable for fast rolling. On the other hand, a twelve-rod tensegrity structure similar to a rhombicuboctahedron has rectangles on its outer surface, which enables straight rolling and prevents loss of momentum.	19
2.9	A twelve-rod tensegrity structure that T12-R is based on has its outer geometry similar to a rhombicuboctahedron. This geometry is symmetric about three mutually-orthogonal planes one of which is shown in this figure. If the deformation of the structure is symmetric about this plane, then the shape and motion of the structure can be described on this plane by projecting the members of the structure onto the plane.	20
2.10	The outer surface of T12-R consists of triangles and rectangles. There are six perpendicular and twelve diagonal rectangles.	20
2.11	An edge of T12-R consists of an extension spring and an unstretchable string that is spooled in and out by a motor.	21
3.1	A two dimensional conceptual diagram of different stages of a step. The robot deforms its structure using cable actuation, in such a way that its GCoM leaves the current supporting polygon [1].	26
3.2	Definition of the heuristic distance (about one chosen base triangle edge) used for the greedy search algorithm [1].	27
3.3	CO-step policies found in NTRT simulation. Hashed triangles and bold dotted lines represent closed base triangles and actuated cables, respectively. Node numbers used in Table 3.1 are also shown [1].	29
3.4	OC- and OO-step policies found in simulation. Hashed triangles and thick dotted lines represent open base triangles and actuated cables, respectively. Node numbers used in Table 3.1 are also shown [1].	30
3.5	Still images of TT-1 performing a COC-step [1]. The robot starts with a CO-step and the following OC-step is automatically performed. (a) Initially at rest. (b) Deformation. (c) Rotation and Strike. (d) Recovery. A full motion video is available at http://best.berkeley.edu/	31
3.6	Extended policies implemented on TT-1 and TT-2. Hashed triangles and thick dotted lines represent base triangles and actuated cables, respectively [1].	32

3.7	(a) Trajectories of GCoM and ground contacting nodes during a CO-step, tracked with a Vicon [®] system. Markers represent the positions of GCoM and nodes for every 0.1 seconds. Dashed arrows represent the motion directions of GCoM and nodes. (b) Change of heuristic distances during a COC-step. Solid and dashed lines represent experiment and simulation results, respectively. The robot starts to rotate as soon as the heuristic distance becomes zero [1].	33
3.8	Trajectories of GCoM tracked with Vicon [®] system during the robot motions. Markers represent positions of GCoM for every 0.1 seconds [1].	34
3.9	The NTRT simulated result of height changes of the robot's center of mass while performing different steps with the developed actuation policies. Solid, dash, and dot-dash lines represent CO-, OC- and OO-steps, respectively. Notice that in order to perform an OO-step, the robot has to overcome a high potential energy barrier. Actuation policies provided in Figures 3.4c and 3.6 were used to simulate the steps [1].	34
3.10	Changes of kinetic energy and sum of nodal force magnitudes over time during a single execution of the dynamic relaxation. The quantities are normalized with respect to the maximum values of each case. Notice that both values converge to zero, meaning that an equilibrium configuration is found [2], [3].	40
3.11	A diagram of edge connections of TT-1 and TT-2. Cables are connected to nodes and a linear actuator located at the center of the edge. See Figure 2.3 for an actual hardware setup [3].	40
3.12	Coordinate systems used for describing kinematics of two end nodes of a rod. A Cartesian coordinate system, with a set of right-handed orthonormal basis vectors $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$, is used to describe the position of the first node. On the other hand, a spherical coordinate system, with a set of right-handed orthonormal basis vectors $\{\mathbf{e}_R, \mathbf{e}_\phi, \mathbf{e}_\theta\}$, is used to describe the position of the other node relative to the first one. R_0 is the rod length [3].	42
3.13	Distances measured between GCoM and three triangle edges [2], [3].	44
3.14	Scores of 5,000 equilibrium configurations obtained by repeatedly running the dynamic relaxation. In the upper figure, the empty blue circles represent scores of all of the equilibrium configurations obtained, and the filled red circle represents the lowest score. The bottom figure is a histogram of the scores of all configurations obtained [3].	45
3.15	Score distributions of chosen generations. In the upper figures, empty blue circles represent scores of all of the equilibrium configurations obtained, and the filled red circles represent the lowest scores. The bottom figures are histograms of the scores of all configurations obtained in each generation. 500 equilibrium configurations were obtained in each generation [3].	47
3.16	Evolution of minimum (blue triangles) and average (red circles) scores over 30 generations [3].	48
3.17	Node numbers of the robot structure used in Table 3.3 [3].	49

3.18	Trajectories of base triangle nodes and GCoM when TT-2 is deformed with the actuation policy provided in Table 3.3. Depending on initial actuator lengths (equivalently, initial cable tensions), the robot performs either a CO-step or a COC-step. In the figures, empty black circles, blue stars and bold red dashed lines represent node positions, GCoMs and rotation axes, respectively. Thin dashed lines show moving directions of nodes and GCoM. Thin gray triangles are base triangles that the robot crosses over while performing steps. Node numbers follow Figure 3.17. A video of the robot in action can be found at http://best.berkeley.edu/best-research/best-berkeley-emergent-space-tensegrities-robotics/ [3].	50
3.19	The equilibrium configuration with the minimum score of all generations. The figures on the left side are simulation results and the figures on the right side show deformation of our physical robot, TT-2. In the simulation figures, thick black lines, thin dashed lines and red circles represent rods, cables and nodes of the robot structure. Blue stars and gray triangles are centers of mass and base triangles of the robot, respectively [3].	51
3.20	Examples of class 1 and class 2 tensegrity structures. The structures are taken from [12].	53
3.21	A diagram of edge connections of T12-R and TT-3. A spring is connected to a rigid string that is spooled by a motor. See Figure 2.11 for an actual hardware setup.	55
3.22	The outer shape of T12-R is similar to a rhombicuboctahedron that is symmetric about three mutually-orthogonal planes. In this work, we choose Plane 1 to be the sagittal plane of T12-R and develop actuation policies that make T12-R roll in X-direction. We further assume that the actuation policies are symmetric about Plane 1. That is, a pair of cables that are symmetric about this plane always have the same length in our actuation policies.	56
3.23	We consider two different cases of s . When $s = 8$, only the lengths of outer edge strings are controlled with motors. When $s = 20$, all of the edges, except the ones that are perpendicular to the sagittal plane, are actuated. The edges that are not actuated do not contribute to the shifting of CoM on the sagittal plane. Due to the symmetry of actuation policies about the sagittal plane, the actual number of edges that are actuated is $2s$. Only the actuated edges are shown with blue dotted lines in the figures.	57
3.24	The metric used for evaluation of deformed T12-R. Assuming that the robot intends to make a step in X-direction, we measure distances between GCoM and two edges of a base rectangle that are perpendicular to the desired step direction.	58

3.25	Evolution of minimum (blue triangles) and average (red circles) scores of PD-step and DP-step simulations over 30 generations with 100 samples obtained per generation. It is assumed that only outer edge strings are actuated in these simulations and $s = 8$. In both cases, the scores saturate at generation 6. While PD-step simulation finds actuation policies with negative scores, all of the samples obtained in DP-step simulation have positive scores, which implies that no successful policy could be found for DP-steps.	60
3.26	The best equilibrium configuration of T12-R obtained from PD-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a PD-step with this actuation policy.	61
3.27	The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. Because GCoM is located inside of the base rectangle in this configuration, T12-R is not expected to perform a DP-step with this actuation policy.	61
3.28	Evolution of minimum (blue triangles) and average (red circles) scores of PD-step and DP-step simulations over 30 generations with 100 samples obtained per generation. It is assumed that the robot is fully actuated in these simulations and $s = 20$. PD-step simulation finds actuation policies with negative scores even in the first generation, which is earlier than the case of $s = 8$, due to the increased actuation effort. Moreover, DP-step simulation also finds actuation policies with negative scores. The absolute values of minimum scores of PD-step simulation are greater than those of DP-step simulation, which implies that PD-steps are more reliable and easier to perform than DP-steps.	62
3.29	The best equilibrium configuration of T12-R obtained from PD-step policy simulation. Total of 30 generations are run with 100 samples per generation. The robot is assumed to be fully actuated, i.e., $s = 20$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a PD-step with this actuation policy.	63
3.30	The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. The robot is assumed to be fully actuated, i.e., $s = 20$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a DP-step with this actuation policy.	63
3.31	Evolution of minimum (blue triangles) and average (red circles) scores of DP-step simulation over 30 generations with 100 samples obtained per generation. It is assumed that only outer edge strings of T12-R are actuated in these simulations and $s = 8$. δd is set to 8 cm, and thus $d_{min} = 4$ cm and $d_{max} = 20$ cm. The simulation finds DP-step actuation policies with negative scores even in the first generation.	64

3.32	The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. δd is set to 8 cm, and thus $d_{min} = 4$ cm and $d_{max} = 20$ cm. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a DP-step with this actuation policy.	64
4.1	A six-bar tensegrity robot, TT-3, developed at the Berkeley Emergent Space Tensegrities (BEST) laboratory. The robot has distributed controllers on the rods, clearing the center space of the structure for additional payloads, e.g., sensors or a thruster system [5].	67
4.2	A simulated trajectory of a thruster-based tensegrity rover performing a single hop under a particle mass assumption. The rover starts firing from the beginning and the burning period ends at the red point. A constant thrust vector, with its magnitude of 50 N and angle of 45 degrees, is applied to simulate the trajectory [5].	70
4.3	A thruster-based tensegrity robot modeled in NTRT on hilly terrain [5]. Videos of the robot in motion are available at http://best.berkeley.edu/best-research/best-berkeley-emergent-space-tensegrities-robotics/	71
4.4	Example flight trajectories of a thruster-based tensegrity robot on hilly terrain. The nominal hopping distances per hop are (a) 1000 m, (b) 330 m, (c) 100 m, and (d) 10 m. Red star markers represent the target location located 1000 m away from the initial position of the robot (origin). In all figures, each hop is followed by landing bounces, resulting in additional small hops. In (a), although the average travel distance of the robot is close to 1000 m, the distance traveled towards the goal is not because of the lateral displacement it made due to angular noise of the thrust vector. In (b), a red dashed line indicates the 60 m height constraint. Once a hopping phase is over, the robot may move to the target by precision rolling [5].	72
4.5	Average of distances between final locations of the robot and the target after hopping phase is over. Simulations are run five times for each hopping profiles on hilly terrain assuming noise standard deviations of 0.02 N and 0.002 radians for thrust magnitude and orientation angles, respectively. The final location of the robot is closer to the target as the nominal flight distance per hop gets smaller because the hopping resolution increases and because the robot gets more chance to correct its flying direction between the hops, but at the cost of increased energy expenditure [5].	73
4.6	Average of simulated hopping distances per hop for different nominal hopping distances. Five simulations are run for each case assuming noise standard deviations of 0.02 N and 0.002 radians for thrust magnitude and orientation angles, respectively. The average hopping distances are not equal to the nominal hopping distances due to the noise [5].	73

- 4.7 A simulated trajectory of the thruster-based tensegrity robot traveling 1 km to reach the target. The green dot is the initial position of the robot and the red star is the target location which is 1 km away from the initial robot position. During the hopping phase, the robot made two 330 m hops followed by one 100 m hop and one 10 m hop. Note that the maximum height of the robot during the hopping phase is below 60 m. When the robot gets close enough to the target, it ends the hopping phase and switches to the precision rolling phase. 75
- 4.8 Evolution of the robot's belief map while localizing itself with the known lunar surface map. Red areas represent the robot's estimated positions and green dot is the robot's true location. (a) Belief map after first measurement of neighbor grid heights. (b) Belief map after performing one step of punctuated rolling. (c) Belief map after measuring new neighbor grid heights. (d) Belief map at the end of localization process. Notice that the red areas shrank into the location of the green dot. That is, the robot successfully localized itself on the lunar map. Map courtesy of USGS Astrogeology Science Center (<http://astrogeology.usgs.gov/>) [5], [117]. 77
- 4.9 The robot develops an energy-efficient path to the goal by combining hopping and precision rolling, once it finishes the localization process. (a) A lunar terrain map with height information with initial and goal locations marked with green dot and pink star, respectively. (b) A-star algorithm is used to search for the best path to the goal position. Yellow nodes are those already expanded while white nodes are those to be expanded. The shade of the yellow nodes is proportional to the cost to reach the grid cell. (c) The planned path and actions. Blue and green lines represent paths covered by punctuated rolling and hopping, respectively. Map courtesy of USGS Astrogeology Science Center (<http://astrogeology.usgs.gov/>) [5], [117]. 78
- 5.1 A top view of a hopping trajectory consisting of three hops with a nominal hopping distance of 330 m per hop. The front view of this hopping trajectory is presented in Figure 4.4b and reproduced at the top right corner for readers. Because there exists no feedback control for thrust vectoring, the robot does not necessarily hop towards the target and undesirable lateral motion occurs due to the noises included in the simulation. This off-track lateral motion wastes the propellant of which the robot carries only limited amount. 81
- 5.2 A mockup of a gimbal-enclosed thruster system located at the center of TT-3 [5]. Red and grey parts are mockups of a gimbal and cold-gas thruster, respectively. The gimbal-enclosed thruster is connected to the robot structure with springs. The thrust vectoring algorithm guarantees that the nozzle exit flow does not interfere with the gimbal and robot structure. 82

5.3	A mockup of a cold-gas thruster system is at the center of the robot through cable connections to outer tensegrity structure [5]. The thruster direction is controlled by changing the shape of the robot structure by using the shape-shifting capability of the structure.	83
5.4	Rotations and basis vectors used to describe attitude kinematics of a rigid body. The rotation $\mathbf{R}$ between the inertial frame and body-fixed frame consists of three sub-rotations. (a) The first rotation rotates $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$ to $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$. $\mathbf{t}_3$ is the target direction about which the spin axis of the body will be stabilized. (b) The second rotation rotates $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ to $\{\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\}$. (c) The third rotation is about the axis $\mathbf{e}'_3 = \mathbf{e}_3$, and it rotates $\{\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\}$ to $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. (d) The overall rotation $\mathbf{R}$ is a combination of $\mathbf{R}_1$, $\mathbf{R}_2(w)$ and $\mathbf{R}_3(z)$ [6].	86
5.5	A cubic rigid body with two identical axis-symmetric reaction wheels. The rotation axes of the wheels are aligned with $\mathbf{e}_1$ and $\mathbf{e}_2$ [6].	93
5.6	Change of attitude parameters over time under control action. w_1 and w_2 converge to zero while z increases linearly, which implies that the body rotates about its spin axis at the equilibrium [6].	95
5.7	Change of angular velocities over time under control action. ω_1 and ω_2 converge to zero while ω_3 converges to a nonzero constant value. This implies that the body rotates about its spin axis at the equilibrium [6].	95
5.8	Change of absolute angle between the body spin axis $\mathbf{e}_3$ and target axis $\mathbf{t}_3$ over time. The angle converges to zero, meaning that $\mathbf{e}_3$ and $\mathbf{t}_3$ are aligned at the equilibrium [6].	96
5.9	Change of relative angular momenta of the wheels with respect to the primary body over time. At the equilibrium, both h_1 and h_2 become sinusoidal functions of time. The total angular momentum of the system, H , is conserved [6].	96
5.10	Change of control inputs over time. Both synthetic control inputs v_1 and v_2 converge to zero. The input control torques u_1 and u_2 become sinusoidal functions of time at the equilibrium [6].	97
5.11	Change of attitude parameters over time under control action for the singular case. w_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. Unlike the first example, w_2 and z remain zero throughout. Because the initial angular momentum of the whole system is zero, the body and the wheels do not rotate at the equilibrium [6].	98
5.12	Change of angular velocities over time under control action for the singular case. ω_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. ω_2 and ω_3 remain zero throughout. Because the initial angular momentum of the whole system is zero, the body and the wheels do not rotate at the equilibrium [6].	98

- 5.13 Change of angle between the body spin axis $\mathbf{e}_3$ and the final target axis $\mathbf{t}_3 = -\mathbf{E}_3$ over time for the singular case. The angle first converges to 90° , at around $t = 6$, as $\mathbf{e}_3$ stabilizes about $\mathbf{t}'_3 = \mathbf{E}_2$. Then it converges to zero at around $t = 12$, meaning that $\mathbf{e}_3$ and $\mathbf{t}_3$ are aligned at the equilibrium [6]. 99
- 5.14 Change of relative angular momenta of the wheels with respect to the primary body over time for the singular case. h_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. h_2 remains zero throughout. The total angular momentum of the system, H , is conserved and is equal to zero [6]. 99
- 5.15 Change of control inputs over time for the singular case. Both v_1 and u_1 first converge to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. v_2 and u_2 remain zero throughout [6]. 100
- 6.1 ZMP is a point where the tipping moment due to the gravity and inertia forces is zero. (a) ZMP is equivalent to CoP when it is located inside of a supporting polygon. The tipping moment in this case is zero as the moment generated by the gravity and inertia forces is balanced out by the ground reaction force. (b) If ZMP is located outside of a supporting polygon, then it is often referred to as FZMP, IZMP, or FRI point. The tipping moment in this case is not equal to zero as the ground reaction force cannot balance out the moment generated by the gravity and inertia forces. 105
- 6.2 The procedure of further simplifying the two-dimensional model of T12-R. (a) The planar model of T12-R when the robot maintains its symmetry about the sagittal plane and the members are projected onto the plane. This model is the same as what is presented in Figure 2.9c. White circles are nodes, bold black lines are rods, and dashed lines are cables. (b) Two assumptions are made to pairs of parallel rods to further simplify the model. First, both pairs of parallel rods remain parallel during deformation. Next, the distances between both pairs of parallel rods remain constant. (c) Under these assumptions, each pair of parallel rods behave like a single rigid rod. (d) The system can be approximated as a planar two-rod tensegrity system. Controllers for dynamic rolling of T12-R may be developed based on the simplified model and tested on a full dynamic model to evaluate their performances. 107

List of Tables

2.1	Physical parameters of TT-1 and TT-2 [1]	15
2.2	Physical parameters of T12-R	21
3.1	Actuation policies found in NTRT simulations using a greedy search. The numbers are node numbers, see Figures 3.3 and 3.4 for node numbering. 2-tuples represent actuated outer cables defined by their two end nodes and 3-tuples represent triangles formed by the three vertex nodes [1].	29
3.2	Physical Parameters of TT-1 and TT-2 [3]	46
3.3	Actuation Policy ($\mathbf{d}_{29}^*$). Node numbers follow Figure 3.17 [3].	48
3.4	Edge length comparison between simulation and hardware experiment [3]	52
4.1	Parameters and thermodynamic quantities used for nitrogen thruster analysis [5].	69
4.2	Averages of total number of hops performed, simulated hopping distances per hop and final distances between the robot and target for different hopping profiles. Total of five simulations are run for each case [5].	74
4.3	Averages of total burn time and consumed propellant mass for different hopping profiles. Simulations are run five times for each case [5].	75

Acknowledgments

This dissertation would not have been possible without the support and guidance of many individuals. I would like to convey my sincere appreciation to all people who have inspired and helped me during my doctoral study.

First and foremost, I owe my deepest gratitude to my advisor Professor Alice M. Agogino. Without her perpetual encouragement and enthusiasm, this study would hardly have been completed. Her creative ideas highly motivated me and her bright input helped me to gain a deeper insight into my own research. She was always accessible and willing to help students, and she was the first person that I would approach when I faced difficulties. Indeed, I never forget that moment when she encouraged me and cheered me up with her thoughtful words when I had a personal hardship. I am indebted to her more than she knows.

Besides my advisor, I would like to thank the rest of my dissertation committee: Professor J. Karl Hedrick and Professor Claire J. Tomlin, for their insightful comments and valuable feedback which incited me to widen my research from various perspectives. I am also deeply grateful to Professor Oliver M. O'Reilly for guiding me to the exciting world of dynamics and advising me with dynamics related research questions during this study.

My sincere thanks also go to Dr. Adrian K. Agogino and Vytas SunSpiral who provided me with an opportunity to join their team as a research intern at NASA Ames Research Center during the summers of 2014 and 2015. Their brilliant suggestions not only helped me expand the area of my research, but also made my study more thorough and enriched.

I am grateful to my advisors and colleagues for granting their permission to include material from previously published sources in this dissertation. I sincerely appreciate all the time and effort my co-authors spent for the preparation of the manuscripts and I would like to take this opportunity to thank them most heartily for their contributions and cooperation.

I thank Professor Alice M. Agogino, Dr. Adrian K. Agogino, Vytas SunSpiral, Deaho Moon, Laqshya Taneja, Aliakbar Toghyan, Bornha Dehghani for their kind permission to use material from [1] in Chapters 2 and 3.

I thank Professor Alice M. Agogino and Dr. Adrian K. Agogino for their kind permission to use material from [2] in Chapter 3.

I thank Professor Alice M. Agogino, Dr. Adrian K. Agogino, Aliakbar Toghyan, Deaho Moon, Laqshya Taneja for their kind permission to use material from [3] in Chapters 2 and 3.

I thank Professor Alice M. Agogino, Dr. Adrian K. Agogino, Vytas SunSpiral, Lee-Huang Chen, Ellande Tang, Kevin Li, Richard House, Erik Jung for their kind permission to use material from [4] in Chapter 2.

I thank Professor Alice M. Agogino, Dr. Adrian K. Agogino, Vytas SunSpiral, Lee-Huang Chen, Angelo Brian Cera, Mallory Daly, Edward Liu Zhu, Julien Despois for their kind permission to use material from [5] in Chapters 4 and 5.

I thank Professor Alice M. Agogino for her kind permission to use material from [6] in Chapter 5.

I thank all my fellow labmates at the Berkeley Emergent Space Tensegrities laboratory for their support on my study, for the stimulating discussions, and for all the fun we have had together. I would like to acknowledge them for all of their support: Andrew P. Sabelhaus, ChanWoo Yang, Alexander Yusheng Lim, Azhar Khaderi, Peadar Keegan, Xiang Li, Jeffrey L. Ware, Vincent Viola, Marcel Pozo, Justino J. Calangi, Eric Cheng-yu, Yangxin Chen, Dizhou Lu, Yuejia Liu, Jae Young Bin, Kyle Zampaglione, Terence Cho, Ian Krase, Yilian Yan, Chi Lee, Jeremy Richard Cook, and all the other current and former students of the Berkeley Emergent Space Tensegrities laboratory. I have been blessed to have these brilliant friends and colleagues, and they made my graduate life a lot more enjoyable. I wish them all the best for their future endeavors.

I would like to thank the Department of Mechanical Engineering at the University of California, Berkeley for giving me the opportunity to study in their graduate program. I am also very thankful to the Samsung Scholarship for providing me with full financial support for the first five years of my graduate study. Many thanks go to the Systems Teaching Institute of the University Affiliated Research Center for providing me with the opportunity to participate in their internship program at NASA Ames Research Center during the summers of 2014 and 2015. I would also like to acknowledge the Intelligent Robotics Group and Robust Software Engineering Group at NASA Ames Research Center for hosting me during my internship.

My deepest gratitude goes to my parents, Ki Wan Kim and Beyong Ryee Jeon, and to my sister, Yoonjoo Kim, for their unconditional trust and wholehearted support. Words cannot express how grateful I am to my family for all of the sacrifices that they have made on my behalf. Their unbounded love and their prayer for me are what sustained me thus far. I would also like to show my gratitude to my parents-in-law, Hakkeun Choi and Seokhwa Kim, and to the family of my sister-in-law, Jaewon Choi and Jonggon Lee and their little angel Hyunseung Lee, for warmly embracing me as a new family member and encouraging me throughout my graduate study.

Chapter 1

Introduction

This dissertation studies novel robotic systems based on tensegrity concepts. Tensegrity robots have a great potential to overcome the barriers for many of the current robotic systems, and they are envisioned for a wide range of new applications that have not been explored before. However, research on tensegrity robots has started only recently, and not much is known about these unique robots yet. Hence, this work aims to expand the knowledge on tensegrity robots, present their proof-of-concepts, and provide a foundation for future research on these innovative robots. Specifically, this study puts a special emphasis on the locomotion of tensegrity robots.

The goal of this chapter is to introduce the readers to tensegrity systems and to present background information that will aid the future discussions in later chapters. The rest of the chapter is organized as follows. A brief overview on the history of tensegrity research is provided in Section 1.1, followed by a review on tensegrity robots in Section 1.2. Section 1.3 introduces a simulation tool that is widely used in tensegrity robotics research and is also used in this work. Research goals of this study are provided in Section 1.4, and an outline of the remaining chapters is presented in Section 1.5.

1.1 Tensegrity Structures

The term tensegrity is first coined by Buckminster Fuller [7] as a portmanteau of *tensional integrity*. Such a name is given to tensegrity structures because of the unique design principle they are based on. Tensegrity structures are constructed by connecting isolated rigid *rods* with a network of *cables*. When a tensegrity structure is loaded, both types of members bear loads only in axial directions; rods undergo pure compressive forces and cables experience pure tensile forces. Hence there is no bending or twisting moments. For this reason, rods and cables are often referred to as compressive and tensile members, respectively. Although no rod members touch each other in a tensegrity structure, the structure maintains its shape by delicately balancing its cable tension forces and rod compression forces. The overall shape of the structure is determined by the distribution of internal forces across its members.

As pointed out in [8], [9], tensegrity structures have several unique features that distinguish themselves from other structures. First, tensegrity structures are in general lightweight as they have minimal mass distribution in terms of load bearing. That is, a tensegrity structure has most of its internal volume empty and the material is placed only on its load paths. This allows efficient use of material and results in a reduced weight of the structure. Tensegrity structures are also deployable; they can be packed into small volumes due to their structural level flexibility and low use of material, and they can expand back to their full volume states by properly adjusting cable tension. Furthermore, because tensegrity structures are constructed with “simple” mechanical components (i.e., axially loaded rods and cables), their static and dynamic models can be built and used for analyses. Lastly, tensegrity structures can survive from huge external loadings or impact shocks because of their innate compliance. All of the above characteristics of tensegrity structures could be advantageous when they are used as robotic platforms and are exploited in this work.

Early works on tensegrity structures focused on how to construct proper tensegrity structures that are self-sustained and do not collapse, which is still an ongoing research problem in the field. For example, Kenneth Snelson, an early pioneer of artistic tensegrity sculptures, created a number of innovative tensegrity structures since the 1960’s [10], [11]. He described his own sculptures as “forces made visible,” which well illustrates the nature of tensegrity structures. Since then, many artists and engineers got attracted by tensegrity systems and efforts were made to find ways to systematically build the structures [12]–[14].

With the growing interest in the design and construction of tensegrity structures, studies were expanded to the problems of kinematics and statics of the structures because the shape of a tensegrity structure is determined by its internal force distributions and vice versa. A passive tensegrity structure without any actuation cannot take an arbitrary shape, and only the shapes balancing the internal forces are allowed for the structure. In this work, these feasible shapes will be referred to as *equilibrium configurations*. The problem of searching for the feasible shapes of tensegrity structures is called a *form-finding* problem. Because the shapes and internal force distributions are tightly coupled in tensegrity structures, kinematics and statics of the structures need to be considered simultaneously [15] in order to solve the form-finding problem, and special techniques are developed for this purpose. For example, several analytical methods were proposed to approach this problem [15]–[17]. However, many of these methods are suitable for only simple tensegrity structures having a small number of rods and cables, or having some useful symmetry properties. With the growth of computing power in recent years, alternative approaches based on numerical methods were proposed to overcome this challenge and to find irregular shapes of more complicated tensegrity structures with many number of rods and cables [18]–[21]. This work also takes the latter approach to find feasible shapes of our tensegrity robots, and more details on this subject will be provided in Chapter 3.

To expand the domain of application of tensegrity structures, many researchers proposed to add actuation to passive tensegrity structures to turn them into active ones [22]–[27]. Much research in this direction focuses on deployability of tensegrity structures and suggests to use them for extraterrestrial applications such as deployable masts, antennas or space

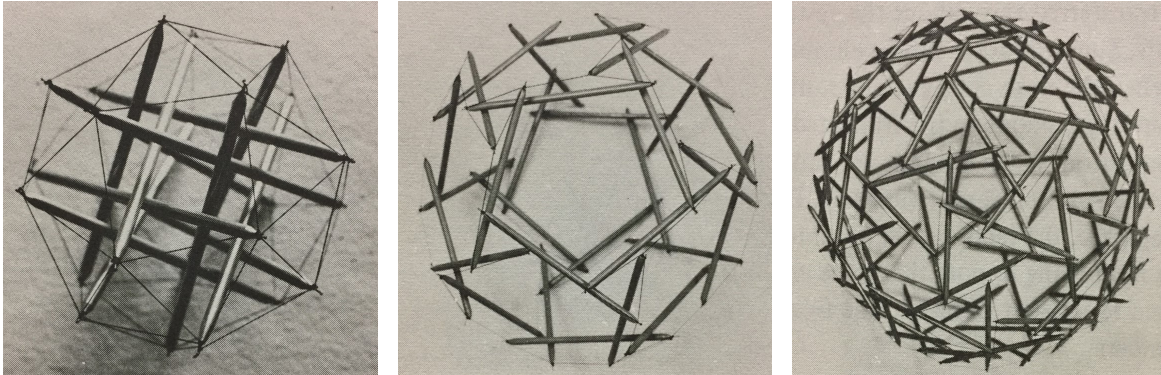
telescopes [28]–[33]. Research on the active tensegrity structures largely consists of the study of dynamics and control of tensegrity systems [34]–[45]. While the previous research showed that it is possible to build dynamic models of tensegrity structures, the models usually turn out to be quite complicated because the structures have many degrees of freedom inherent to multi-body systems and high nonlinearity arising from rods interacting with each other via flexible cables. This poses a big challenge on controller design and the motion of the current tensegrity systems is quasi-static in most cases. A recent review on tensegrity dynamics can be found in [46]. A more complete discussion on the history of tensegrity research is provided in [47].

1.2 Tensegrity Robots

Tensegrity robots constitute a special class of active tensegrity structures. The earliest research on tensegrity robots includes [48], [49], where two robots based on tensegrity prisms are presented with a control strategy for their mobility. While tensegrity structures can have asymmetric and irregular shapes, many researchers have had a special interest in the structures that are *spherical* and used them as mobile robotic platforms [50]–[54]. It is noted here that the term “spherical” does not mean that the tensegrity structures have perfectly smooth spherical outer shapes, but rather it means that their outer shapes are topologically the same as a sphere. While a spherical tensegrity structure with a small number of rods may look like a very crude approximation of a sphere, its outer shape actually converges to a sphere as more rods are used to construct the structure (Figure 1.1). Among the family of spherical tensegrity structures, a six-rod tensegrity structure is the simplest three-dimensional one and several hardware robots based on this structure have been introduced in the literature [50], [51], [53]–[60]. Most of the research on these robots studied their mobility and developed controllers for their *rolling* either from hardware experiments [55], [56], [61], [62], or in simulations [58], [63]–[67].

Strictly speaking, “rolling” is not a precise term to describe the motion of spherical tensegrity robots. For a rigid body that is rolling on the fixed surface, the material point of the body that is in contact with the surface continuously changes as the body rolls, and the velocity of the contact point is zero. However, the motion of a spherical tensegrity robot is discontinuous and involves repeated impact events between its rods and the ground¹. During the piecewise continuous motion between impacts, the rods that are in contact with the ground are *pivoted* (if the rods are assumed to be slender) and the contact points do not change. Even if the rods are assumed to have finite volumes and rounded ends such that their material points that are in contact with the ground are continuously changing and the velocities at the contact points are zero during the intermediate continuous motion, the discontinuity still exists in the overall motion and it makes the motion different from rolling. For this reason, some researchers use other terminologies, such as tumbling or tipping over, to describe the motion. However, this work will continue to use the term rolling when de-

¹A detailed description on the motion of spherical tensegrity robots is provided in Section 3.2



(a) Snub cube (b) Truncated dodecahedron (c) Three-frequency dodecahedron

Figure 1.1: Tensegrity structures become more spherical if more rods are used to construct them. (a) A snub cube tensegrity built with 12 rods. (b) A truncated dodecahedron tensegrity built with 30 rods. (c) A three-frequency dodecahedron built with 90 rods. Images courtesy of Anthony Pugh [12].

describing the motion of spherical tensegrity robots for two reasons: 1) the term is widely used and accepted in the field of tensegrity robotics, and 2) if a spherical tensegrity structure is constructed with an infinite number of rods, then its outer shape will converge to a perfect sphere and its motion will become close to rolling, as one may imagine by extrapolating the tendency shown in Figure 1.1. Often times, the motion of spherical tensegrity robots will be referred to as *punctuated rolling* to emphasize the discontinuous nature of the motion.

Researchers also studied robots that are based on other forms of tensegrity structures. Among them, tensegrity robots based on spine-like tensegrity structures have been proposed for crawling or snake-like motions and Central Pattern Generator-based controllers are developed for their locomotion [68]–[71]. Building on these works, researchers also proposed legged tensegrity robots based on the spine-like tensegrity structures [72], [73]. Moreover, tensegrity robots are proposed for duct exploration and maintenance tasks and two prototypes are built for this purpose based on multi-module tensegrity prisms [74], [75]. Some researchers showed that tensegrity structures can effectively describe anatomical structures of the human body [76], [77], and based on this concept, bio-inspired tensegrity manipulators mimicking human shoulder and elbow are developed in [78], [79].

Relatively less work has been done on sensing and state estimation of tensegrity robots. Only recently, an Unscented Kalman Filter-based method that combines inertial measurement readings with radio time-of-flight ranging measurements is presented in [80] for state estimation of a six-rod tensegrity robot. Another approach proposed in [81], [82] uses liquid metal-embedded hyperelastic strain sensors [83] to measure cable lengths of a tensegrity structure. The sensors in that work also serve as tensile elements, which shows the *redundancy* of tensegrity structures; members of tensegrity structures can simultaneously function as sensors or actuators in addition to being load-carrying members [8], [9]. Researchers also proposed to use tensegrity structures as sensors for measuring forces and torques [84], [85].

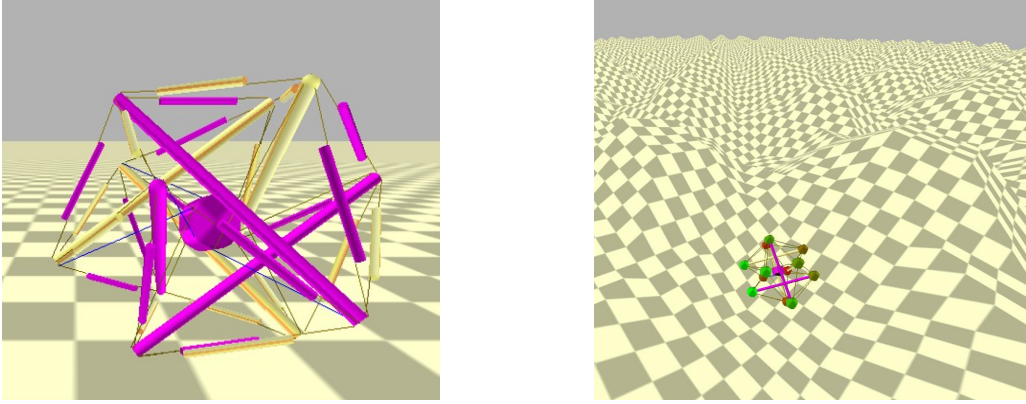


Figure 1.2: Six-rod tensegrity robots simulated in NASA Tensegrity Robotics Toolkit (NTRT) [1], [5]. NTRT provides core methods to model, simulate, and control broad types of tensegrity robots [88].

1.3 NASA Tensegrity Robotics Toolkit

Tensegrity robots are challenging to build because they are mechanically complex and their design requirements differ radically from traditional robots. For this reason, many previous studies exploited simulation prior to constructing hardware tensegrity robots to better understand their design requirements and to predict behavior and performance of the robots to be built. In most cases, custom simulators were developed and were run to study specific systems of interest [48], [63], [64], [86], [87].

Research efforts have been made at NASA Ames Research Center to develop a general-purpose simulator, named NASA Tensegrity Robotics Toolkit (NTRT), to aid and facilitate tensegrity robotics research [88]. NTRT provides all of the core methods to model, simulate, and control broad types of tensegrity robots. The performance of the simulator has been verified, for instance, in [58] where the motion of a tensegrity robot simulated in NTRT is validated with hardware experiments. NTRT is released as an open-source software and a number of researchers have started to use it as their simulation tools in recent years [67], [71], [73], [74]. Figure 1.2 shows six-rod tensegrity robots that are simulated in NTRT. This research uses both NTRT and a simulation tool that is custom developed in MATLAB.

1.4 Research Objectives

The goal of this research is three-folds:

1. Develop functional prototypes of autonomous and untethered spherical tensegrity robots and demonstrate their applicability to future robotic missions.

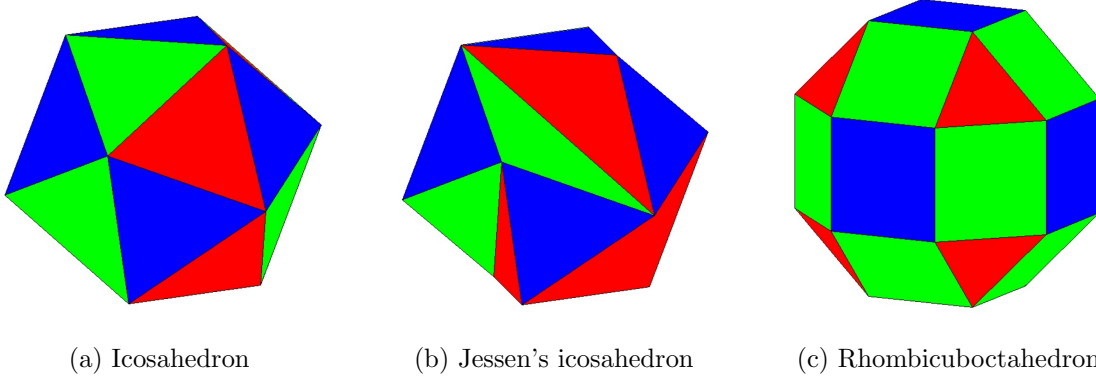


Figure 1.3: A six-rod tensegrity structure has an outer shape similar to an icosahedron, or more precisely, to a Jessen's orthogonal icosahedron. Twelve-rod tensegrity structures can have several different forms, but only the one that has an outer shape similar to a rhombicuboctahedron is considered in this work.

2. Devise systematic ways of developing control strategies for generating rolling locomotion of general spherical tensegrity robots.
3. Explore and examine hopping locomotion of spherical tensegrity robots, which will enable long distance travels of the robots on unknown terrains during, e.g., planetary exploration missions.

Among various types of tensegrity robots introduced in Section 1.2, this research focuses on two kinds of spherical tensegrity robots. One is based on a six-rod tensegrity structure that has an outer shape similar to an icosahedron, or more precisely, to a Jessen's orthogonal icosahedron [89]. The other one is based on a twelve-rod tensegrity structure that has an outer shape similar to a rhombicuboctahedron (Figure 1.3). The work aims to study two modes of locomotion for these systems – rolling and hopping. However, it will be shown that the outcome of this research is not limited to these systems only, and it can be easily generalized and applied to other spherical tensegrity robots as well.

1.5 Outline of Dissertation

The rest of the dissertation is structured as follows.

In Chapter 2, four hardware prototypes of spherical tensegrity robots are introduced. Three of them (named TT-1, TT-2 and TT-3) are based on a six-rod tensegrity structure and the last one (named T12-R) is based on a twelve-rod tensegrity structure. Their designs and motivations are described in details.

In Chapter 3, control strategies for rolling locomotion of spherical tensegrity robots are developed. Two approaches are taken for this purpose. The first approach exploits a search

method in NTRT simulation and the second approach is based on a multi-generation Monte Carlo method in conjunction with a dynamic relaxation technique. The developed control strategies are tested on the hardware robots and their successful rolling is demonstrated. It is shown that the latter approach can be applied to other general spherical tensegrity robots and the procedure is described in details.

In Chapter 4, tensegrity robots that can hop with a cold-gas thruster system are introduced. While rolling locomotion of spherical tensegrity robots has been studied in the previous research, no work has been done on hopping of the robots, to the best of the author's knowledge. Such a thruster-based tensegrity robot is envisioned for lunar missions, and its various hopping profiles on the Moon are examined in order to design safe and efficient flight trajectories for a delivery of scientific payloads. Localization and path planning of the robots on the lunar surface are also discussed.

In Chapter 5, possible mechanisms for thrust vectoring of hopping tensegrity robots are investigated. Among the available options, a system with two reaction wheels is extensively studied. A globally and asymptotically stable controller is designed by using a Lyapunov method based on the (z, w) -parameterization of rotation. Simulation results show that the controller can point the thrust to any desired direction from an arbitrary initial orientation.

In Chapter 6, conclusions and future research directions are presented. Potential research directions are suggested to achieve dynamic rolling of spherical tensegrity robots by adapting research results from biped studies. Future research problems that may arise when the reaction wheel based thrust vectoring mechanism is attached to tensegrity robots are also discussed.

Chapter 2

Hardware Design of Spherical Tensegrity Robots

2.1 Introduction

While tensegrity structures have existed more than sixty years since they were first invented and explored by early pioneers such as Buckminster Fuller and Kenneth Snelson [7], [47], it was not too long ago when they were first introduced to the field of robotics. Despite a relatively short history of using tensegrity structures as a mobile robotic platform, a number of hardware tensegrity robots with mobility have been introduced in the literature so far. The earliest hardware tensegrity robot that was proposed in the literature [48], [49] is based on a three-rod prism tensegrity structure and it crawls using body deformation. A few years later, Japanese researchers have built two tensegrity robots based on a six-rod tensegrity structure [55], [56]. Their first robot used shape memory alloy coils to deform its body and to generate locomotion, while their second robot used pneumatic actuators for the same purpose. Both robots relied on external energy sources for their operation. Another research group from Germany have built tensegrity robots based on rather unusual tensegrity structures with curved and V-shaped rods and demonstrated their locomotion capability [90], [91].

Building upon the prior work, the Dynamic Tensegrity Robotics Laboratory (DTRL) at NASA Ames Research Center – a major collaborator of the Berkeley Emergent Space Tensegrities (BEST) laboratory – have developed a tensegrity robot, named SUPERball, based on a six-rod tensegrity structure [50], [51], [53], [54], [58]–[60]. The purpose of this robot is to investigate the potential of a tensegrity robot as a next generation space exploration rover and to present a proof-of-concept. This robot was inspired by its predecessor, ReCTeR [57], which is also a six-rod tensegrity robot built at Ghent University. Both SUPERball and ReCTeR are underactuated, meaning that not all of their cables are independently actuated. Other collaborators of the DTRL have also constructed robots based on tensegrity structures with different geometries. A hardware prototype of a spine-like robot that moves similar to snakes is presented in [69], and tensegrity robots specifically developed for duct exploration

and maintenance tasks are proposed in [74], [75]. Recently, researchers at the BEST laboratory prototyped a novel quadruped robot based on a spine-like tensegrity structure [72].

As the prior work shows, tensegrity robots are either promising as a locomotion system by themselves or are expected to perform tasks that are considered to be challenging for other conventional robotic systems once they have mobility. Reliable locomotion of hardware tensegrity robots, however, is yet to be achieved and research efforts are continuously made to enable robust locomotion of tensegrity robots. As part of the efforts, this work studies two modes of locomotion – rolling and hopping – of spherical tensegrity robots and demonstrates our locomotion strategies on our own physical hardware robots.

2.1.1 Contribution of This Chapter

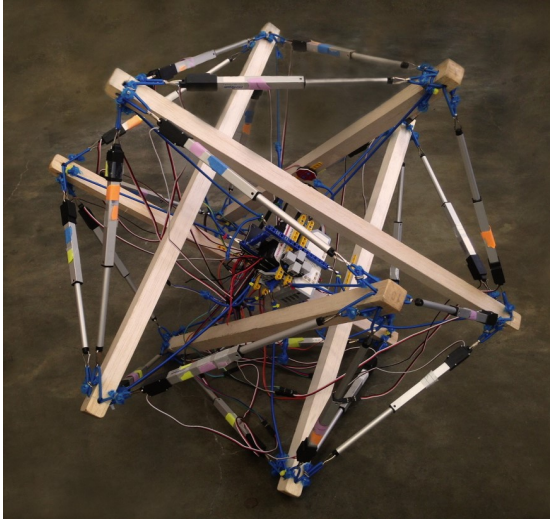
This chapter introduces four spherical tensegrity robots that are constructed at the BEST laboratory. Three of them (named TT-1, TT-2 and TT-3) are based on a six-rod tensegrity structure and the last one (named T12-R) is based on a twelve-rod structure. These robots are built for hardware demonstration of locomotion strategies to be developed in the later chapters.

All of our robots are rapidly prototyped at a low-cost with fast build time. This allowed us to quickly have functional robots at hand and focus on hardware experiments instead of spending too much time in the hardware development phase. Furthermore, rapid prototyping allowed us to go through multiple design iterations during which our robot designs were improved and optimized. We also aimed for modularity in our hardware robot design such that the design can be easily extended to other tensegrity robots that are based on different tensegrity structures with differing number of rods and cables. For instance, the hardware design of T12-R largely follows that of TT-3, and by doing so, we were able to shorten its development time. Our robots also improve upon the previously introduced six-rod tensegrity robots in two ways. First, our robots are untethered and can operate by themselves without any connection to the environment. Second, our robots are fully actuated and each cable can be actuated independently.

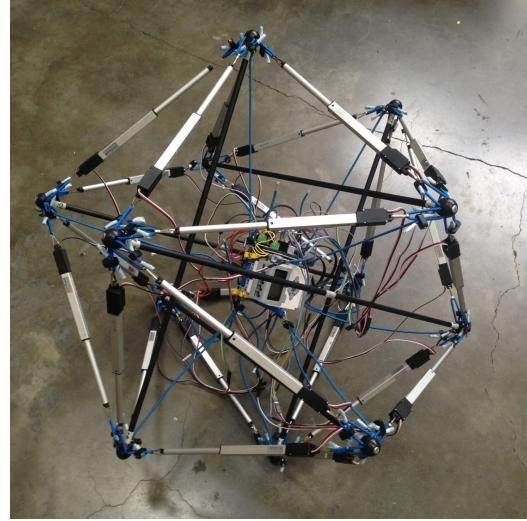
The rest of the chapter is organized as follows. In Section 2.2, we present the design of six-rod tensegrity robots – TT-1, TT-2 and TT-3 – and point out features that are added to traditional tensegrity structures to provide mobility to the robots. The design of T12-R and its motivation are discussed in Section 2.3. The chapter ends with a conclusion in Section 2.4.

2.2 Design of Six-Rod Tensegrity Robots: TT-1, TT-2 and TT-3

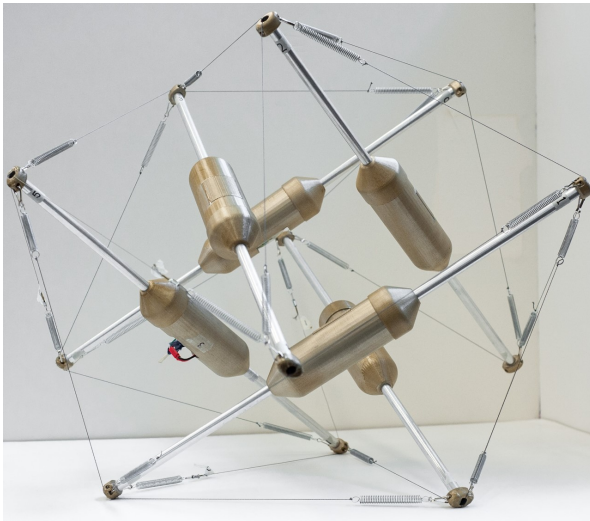
The first few robots that are built at the BEST laboratory (namely, TT-1, TT-2 and TT-3) are based on a tensegrity structure which consists of six rods and 24 cables (Figure 2.1),



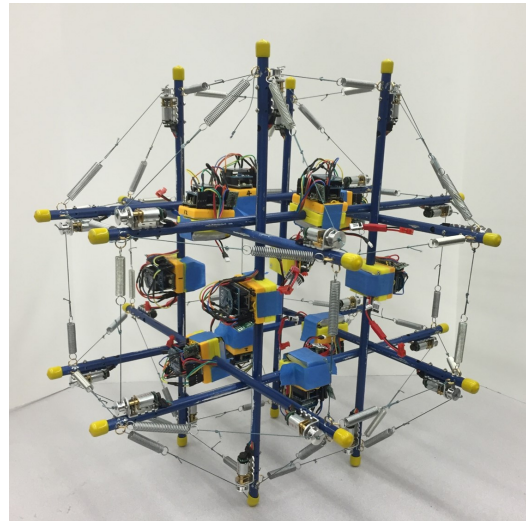
(a) TT-1 [1]



(b) TT-2 [2], [3]



(c) TT-3 [4]



(d) T12-R

Figure 2.1: Rapidly prototyped tensegrity robots that are developed at the Berkeley Emergent Space Tensegrities laboratory, University of California at Berkeley. The first three robots are based on a six-rod tensegrity structure and the last one is based on a twelve-rod tensegrity structure.

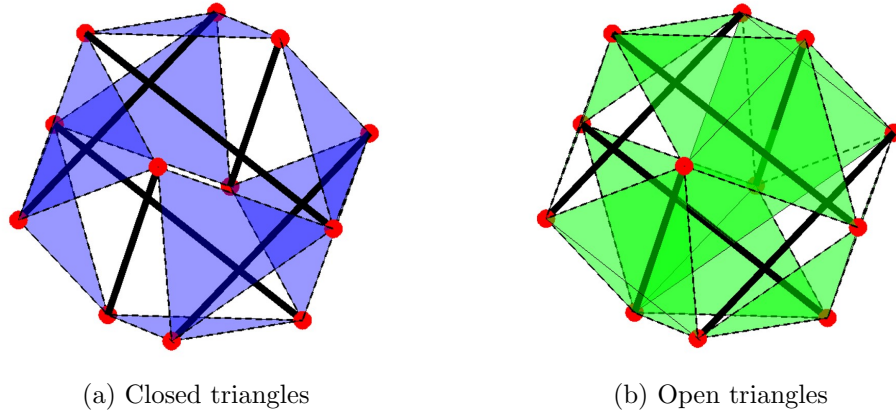


Figure 2.2: The outer surface of a six-rod tensegrity structure consists of eight closed equilateral triangles and twelve open isosceles triangles.

and hence they are structurally identical. Each rod end, or *node*¹, is connected to four cables to create a structure with eight equilateral triangles and twelve isosceles triangles formed by the positioning of the nodes. Since there is not a cable connecting every node (six are missing – one between each parallel rod ends), each of the isosceles triangles also presents itself as an *open triangle* with cables connecting two of its three edges. Meanwhile every equilateral triangle is a *closed triangle* with cables connecting all three of its edges (Figure 2.2). Throughout this work, the triangle in contact with the ground is referred to as the *base triangle*.

While a traditional tensegrity structure does not include a *payload*, which is a functional load carried by the structure, our robots are designed to have different payloads at their center space. For example, TT-1 and TT-2 carry a controller unit and supplementary electronics as their payloads, and TT-3 is designed to carry a thruster system to enable hopping. The payloads add mass to the structures and thus change the deformation properties of the structures. Whereas the cables connecting nodes in the outer structure are referred to as *outer cables*, the cables keeping the payload inside of the structure are called *inner cables*.

Structurally deforming a tensegrity robot by changing the lengths of its outer cables such that the ground projection of the robot’s center of mass (*GCoM*) escapes a supporting polygon will result in a rotation or a discrete *step* from one base triangle to another. Furthermore, we define *punctuated rolling* as the motion of the robot consisting of multiple repeated steps. We add actuators to passive tensegrity structures for our hardware robots in order to realize steps and provide them with mobility. A detailed discussion of steps and punctuated rolling is presented in Chapter 3.

Although TT-1, TT-2 and TT-3 are based on the same six-rod tensegrity structure, their hardware designs and motivations are different. In the following, we discuss the details of their hardware designs.

¹The term *node* has two definitions in this work. The first one is a point at the end of a rod, as defined here. The second one is a leaf node of a search tree, which will be discussed later in Chapter 3.

2.2.1 TT-1 and TT-2

2.2.1.1 Rod Design

Rods, together with actuators and payloads, are responsible for a significant portion of the total weights of both TT-1 and TT-2. As an effort to reduce the weights of the robots, we use lightweight balsa wood rods for TT-1 and fiber glass rods for TT-2. This choice of rod materials is the main difference between TT-1 and TT-2. The change was made for two reasons. First, fiber glass rods are thinner than balsa wood rods, hence they provide more space at the robot's center. This allows a larger payload to be installed at the robot's center space. Second, fiber glass rods are more flexible than balsa wood rods, and the robot assembled with fiber glass rods has better impact absorption capability. Although other materials such as bamboo and carbon fiber were considered, the advantages of balsa wood and fiber glass rods include low price, relatively light weight, and easy machinability. The ends of balsa wood rods are drilled with holes for cable connection and saturated with epoxy to help tolerate the tension from the connected cables. The ends of fiber glass rods are fitted with customized spherical plastic end caps with holes for cable connection.

The overall size of a six-rod tensegrity robot is determined by the length of its rod. The length of each rod for TT-1 and TT-2 is determined by the ratio of the actuator's stroke length to the length of a rod. In [66], simulations predicted that 10% actuation range relative to the rod length was sufficient to generate punctuated rolling from a six-rod tensegrity robot. Both TT-1 and TT-2 have rods with an identical length of 69 cm, and their linear actuators have a maximum stroke length of 10 cm. The ratio of actuation range to the rod length is then 14.5%. This choice of physical dimensions allows relatively large structural deformation of the robots and enables the robots to take steps by using only a partial set of actuators mounted on them. Because the rod lengths are the same for TT-1 and TT-2, their overall diameter is also the same and is approximately 1 m. This size of the robots provides enough space for payloads at their centers. Furthermore, the balsa wood and fiber glass rods weigh about the same with only a few milligrams difference. As a consequence, the overall weights of TT-1 and TT-2 are also about the same.

2.2.1.2 Cable Selection and Linear Actuator Attachment

Every edge of TT-1 and TT-2 consists of serial connections of two looped elastic cords and a linear actuator (Figure 2.3). Elastic cords with a thickness of 3.18 mm are looped and used as outer cables due to their linear elasticity and shock absorption properties. To apply forces to the elastic cord loops, a total of 24 linear actuators (Firgelli L12-R) are installed on each robot. Each linear actuator is placed between two elastic cord loops and at the middle of the edge (Figure 2.3). The linear actuators have a maximum stroke length of 10 cm, and can provide up to 45 N of force, reaching a maximum speed of 5 mm/s. In our experiments, we have observed that the maximum force provided by the linear actuators is well above the maximum force required to pull the outer cables (i.e., elastic cord loops) to their maximum stretched length during the robots' operation.

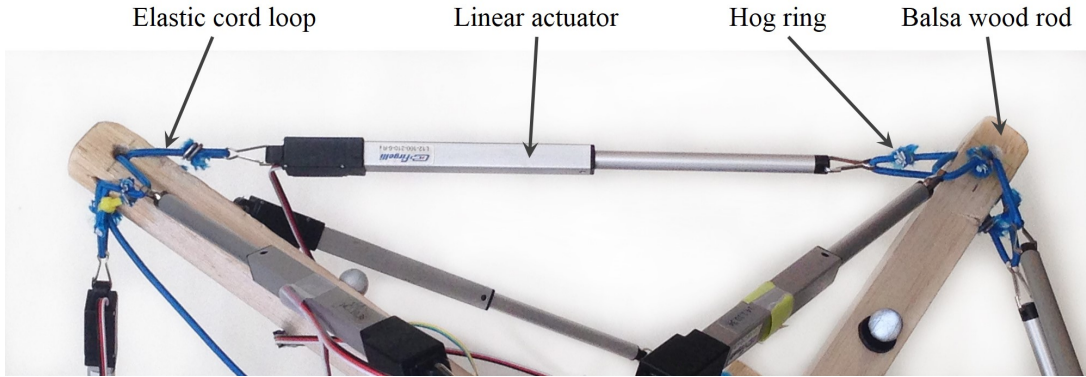


Figure 2.3: An edge of TT-1. The edge consists of serial connections of two looped elastic cords and a linear actuator. Spring snaps are used to connect the cord loops and actuator. Hog rings are used to fix the cord loops [1].

To ease the connection of the linear actuators to the elastic cord loops, spring snaps are used between them for both TT-1 and TT-2. For TT-1, the elastic cords are looped around the snaps, looped through the cross holes of the rod ends, and then fixed with hog rings in order to ensure secure connections and minimize dead lengths. Reducing dead length allows actuator retraction to focus more towards structure deformation rather than towards cable tensioning. The hog rings also help to connect all the elastic cord loops in even lengths such that their tensions are equally distributed when the robot is not deformed. There is, however, a downside of using the hog rings. The hog rings are mechanically irreversible, and once the elastic cords are looped and fixed with the hog rings, it is not possible to change the lengths of the elastic cord loops unless we replace them with new cord loops. This is undesirable because the length of the elastic cord loop has a direct relationship to the loop's effective stiffness, and we had to test elastic cord loops with several different lengths, or stiffnesses, on our robot in order to find the best working stiffness. Replacing all of the cord loops on the robot multiple times to test different stiffnesses required a lot of labor and was better avoided. To overcome this difficulty, customized clamps are used to fix the cord loops in TT-2 instead of the hog rings (Figure 2.4). With the clamps, the lengths of the cord loops are easily adjusted, without having to replace all the cord loops.

After several iterations of stiffness tests, we finalized the length of each elastic cord loop to be 3.8 cm. In the cable operation region, the effective stiffness of the cord loops is experimentally measured to be linear with a spring constant of 1,193 N/m.

2.2.1.3 Payload Attachment

TT-1 and TT-2 carry a controller and other electronic components required to control the actuators as their payloads. Specifically, the payloads are composed of a controller brick (LEGO Mindstorms EV3), Li-Ion batteries (Tenergy), and servo controllers (Mindsensors),

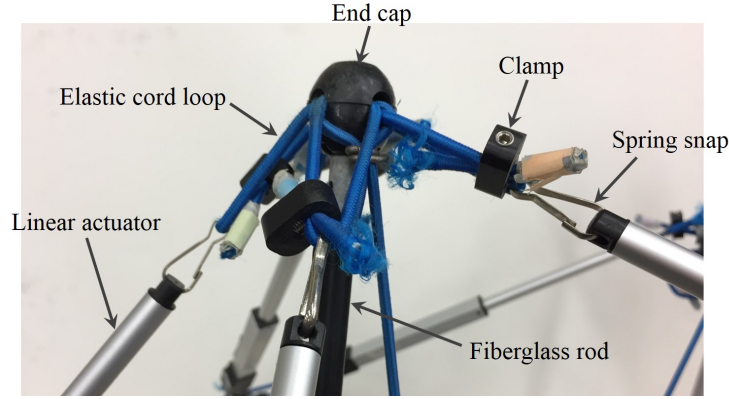


Figure 2.4: Connections at the end of a TT-2 rod. The hog rings used in TT-1 are replaced by customized clamps to facilitate length change of elastic cord loops [3].

which are used to power and control the linear actuators. If needed, additional scientific payloads may also be added. The same elastic cords used for outer cables are also used as inner cables to connect the payloads to the outer structures. A total of 12 cables are used per robot for payload connection in a way that one end of each inner cable is connected to one of the nodes, while the other end is connected to the payload by a spring snap. The rest lengths of these inner cables are set to 35 cm, which located the payloads close to the centers of the robots. Because of the presence of gravity, the tension is primarily concentrated in the inner cables connected to the nodes of the top-facing triangle above the base triangle. The payloads weigh 785 g and account for a large portion of the robots' total weight of 2.7 kg. Therefore, it is critical to place the payloads as close to the centers of the robots as possible to maintain symmetry and predictability of motion. The physical parameters of TT-1 and TT-2 are summarized in Table 2.1.

2.2.1.4 Sensors for Base Triangle Detection

TT-1 and TT-2 are equipped with an inertial measurement unit (IMU) located on their payloads. This IMU (Mindsensors) is used to determine which base triangle is contacting the ground. This information is fed back into the controller which generates and provides appropriate command signals to actuators to enable steps from the detected base triangle.

The algorithm used for the detection is based on a simple partitioning of the space². Due to the symmetry of a six-rod tensegrity structure, we can construct imaginary octants on the structure such that each closed triangle is included in each of the octants (Figure 2.5). Moreover, an IMU is attached to the central payload in a way that a 3-tuple of signs of gravity acceleration readings in X, Y, and Z directions (e.g., [+1,-1,+1] if the gravity is heading towards +X, -Y, and +Z directions with respect to the IMU's coordinate frame)

²The algorithm was originally devised by a former BEST laboratory undergraduate student, Aliakbar Toghyan, whom the author mentored.

Table 2.1: Physical parameters of TT-1 and TT-2 [1]

Total robot mass	2.7 kg
Rod length	69 cm
Rod mass	94.63 g
Outer cable rest length (looped)	3.8 cm
Inner cable rest length	35 cm
Outer cable stiffness (looped)	1,193 N/m
Payload mass	785 g
Actuator mass	56 g
Actuator stroke length	10 cm
Actuator speed	5 mm/s
Actuator body length (fully retracted, with spring snaps)	20 cm

falls into each of the octants. We assume that the robot is stationary while this detection method is running to make sure that there is no external acceleration involved other than the gravity. With this setup, by reading the signs of the gravity in X, Y, and Z directions, we can identify which octant the gravity vector belongs to, and therefore, which closed triangle is a base triangle. For example, in Figure 2.5, the gravity vector (red arrow) is heading towards one of the octants and we identify the white closed triangle, which is included in the gravity octant, as the current base triangle. While this method works the best when the ground is flat, our hardware experiments showed that the method still works on the ground with mild inclination.

An alternative approach of using force sensors for the base triangle detection was also proposed. The idea is to install force sensors at the rod ends such that the sensors touching the ground return high signals due to the weight of the robots while the others return low. Then the base triangle can be inferred from the sensor readings. We had also hoped that we would get useful information about ground reaction forces with these sensors. While the idea is simple, its implementation turned out to be tricky. During the first few design iterations, we came up with several different end cap designs using piezoresistive force sensors (Tekscan). However, we observed that these sensors tend to be noisy and inconsistent, and the end cap designs became more complicated than what we had expected as we attempted to improve the performance of the sensors. In the end, we decided to use simple binary switches instead of the force sensors and we were able to detect the base triangle successfully with a simple end cap design. However, no ground reaction force information was available with this design.

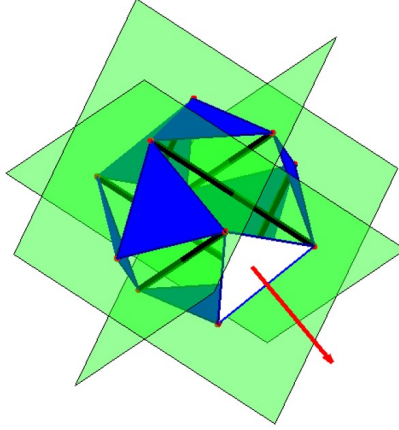


Figure 2.5: A six-rod tensegrity structure with imaginary octants formed by green mutually-orthogonal planes. Each octant is distinguished by signs of accelerations in X, Y, and Z directions measured by an IMU. Blue triangles are closed triangles of the structure. The red arrow depicts the gravity vector measured when the robot is stationary. Based on the X, Y, and Z signs of the gravity vector with respect to the IMU’s coordinate frame, we can figure out towards which octant the gravity vector is heading. We identify the closed triangle that is included in this octant as the current base triangle (white triangle in the figure).

2.2.2 TT-3

A major update in the hardware design of TT-3 from that of TT-1 and TT-2 is the clearance of the center space. As a result, TT-3 can have a larger payload than TT-1 and TT-2, or can provide some extra space between the payload and rods, which would help prevent internal collision between them. In this work, this design allowed us to install a thruster system as a payload to enable hopping of the robot. This will further be discussed in Chapter 4. In this section, we briefly introduce the design of TT-3 and point out its features that are different from TT-1 and TT-2. More details on the design of TT-3 can be found in [4], [82].

2.2.2.1 Rod Design

Unlike TT-1 and TT-2, the controllers are distributed over the rods in TT-3 in order to clear the center volume for a payload. As a result, each rod of TT-3 includes a capsule at the center, and the controllers, batteries, and other electronics are contained in the capsules (Figure 2.7). The distributed controllers communicate wirelessly using the radio chips located in the capsules. Aluminum is chosen as the rod material to sustain large impulse forces that the robot would experience from severe collisions. In fact, researchers at the BEST laboratory performed hardware drop tests of TT-3 to examine the robot’s energy absorption behavior under impact events, and the rods did not break during the experiments [4].

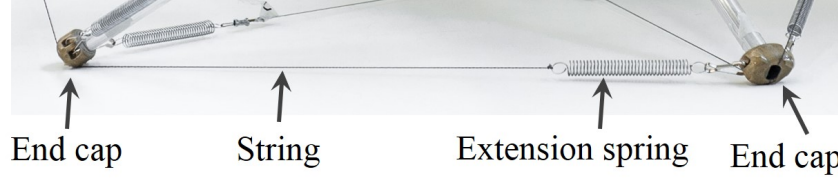


Figure 2.6: An edge of TT-3 consists of an extension spring and an unstretchable string that goes into a hollow rod and is connected to a spool actuated by a rotary motor located inside of a capsule [4].

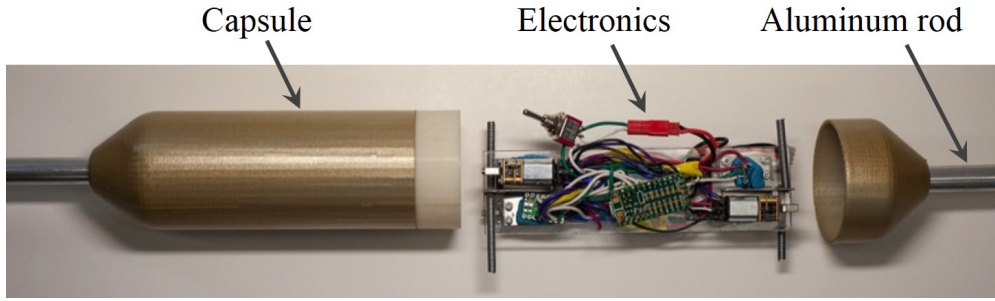


Figure 2.7: At the center of each TT-3 rod is a capsule containing electronic components [4].

2.2.2.2 Edge configuration

The edge components of TT-3 are also different from TT-1 and TT-2. An edge of TT-3 consists of an extension spring connected with an unstretchable string (Figure 2.6) that goes inside of a hollow rod and connects to a rotary motor located inside of the capsule (Figure 2.7). The edge length is controlled by winding and unwinding the string with an actuated spool. The use of extension springs enables more accurate estimation of edge tensions because their stiffness remains constant for a broader range of length change and is more consistent across the springs than the elastic cords used in TT-1 and TT-2. This edge configuration also allows easy replacement of springs.

Because TT-1 and TT-2 have linear actuators at the centers of their edges, their minimum edge length is always greater than the minimum actuator length, and this limits the amount of deformation that TT-1 and TT-2 can have. On the other hand, the edge configuration of TT-3 allows the strings to be fully spooled in to zero length, and the minimum edge length is equal to the (possibly stretched) spring length. Although we do not realize this situation on TT-3 to protect the motors from excessive loading, TT-3 can still deform greater than TT-1 and TT-2. Moreover, rotary motors of TT-3 have faster actuation speed than linear actuators of TT-1 and TT-2. As a result, TT-3 deforms faster and rolls faster than TT-1 and TT-2. Our hardware experiments showed that the rolling speed of TT-3 is approximately 5 cm/s while that of TT-1 and TT-2 is about 1 cm/s.

2.2.2.3 Sensors for Base Triangle Detection

TT-3 has a total of six IMUs, one on each rod. We intended to track the motion of the rods and estimate states of the robot using the IMU measurements, but the IMUs are mainly used for base triangle detection at the time of writing this dissertation. The base triangle detection algorithm running on TT-3 builds upon the one implemented on TT-1 and TT-2. In this algorithm, the space is partitioned into 20 subspaces, instead of octants in Figure 2.5, such that each outer surface triangle (whether it is closed or open) of a six-rod tensegrity structure is included in one of the subspaces. Then the ground contacting surface is detected by examining to which subspace the gravity vector measured by an IMU belongs. More details about the algorithm are provided in [92].

2.3 Design of a Twelve-Rod Tensegrity Robot: T12-R

The tensegrity robots TT-1, TT-2, and TT-3 are all based on a six-rod tensegrity structure. Furthermore, earlier spherical tensegrity robots that were developed from other research groups are also based on the same tensegrity structure [53], [57], [58]. This choice of structure is made because a six-rod tensegrity structure is the simplest three-dimensional tensegrity structure that is similar to a sphere and is suitable for rolling locomotion. Although the robots based on a six-rod tensegrity structure are able to perform punctuated rolling as we will show in Chapter 3, they have one significant downside for such motion. The outer surface of a six-rod tensegrity structure consists of triangles only, and the robots based on this structure cannot roll in a straight line, but rather they have to move in a zig-zag way (Figure 2.8). While this behavior is acceptable for slow speed rolling, this is not desirable for high speed rolling because the robots should change their moving direction after each and every step, which will cause large momentum loss from the robots.

The zig-zag path can be avoided and straight rolling can be achieved if the outer surface of a tensegrity structure consists of rectangles (Figure 2.8). A twelve-rod tensegrity structure similar to a rhombicuboctahedron is the simplest spherical tensegrity structure that has this outer surface property (Figure 2.9). For this reason, this structure is chosen as a base structure of T12-R. In other words, T12-R is designed to have an outer shape that is suited to fast, or *dynamic*, rolling. To the best of the author's knowledge, T12-R is the first tensegrity robot that is based on this twelve-rod tensegrity structure. There is another important advantage of having the outer geometry similar to a rhombicuboctahedron. Since the geometry is symmetric about three mutually-orthogonal planes, if control is applied such that the deformation maintains symmetry about one of these planes, then the motion of the robot can be described on the plane by projecting its members onto the plane (Figure 2.9). Specifically, a slender rod has five degrees of freedom (DOF) in a three-dimensional space, and thus a twelve-rod tensegrity robot has a total of 60-DOF, which makes any controller design on this robot highly challenging. On the other hand, if we assume the symmetric deformation, eight rods that are parallel to the symmetry plane have only three-DOF per

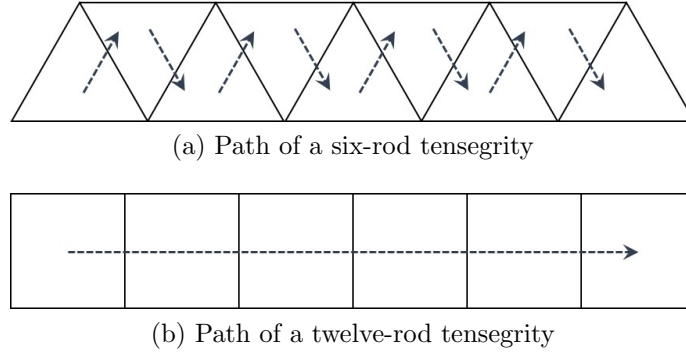


Figure 2.8: The actual paths taken by six- and twelve-rod tensegrity robots when moving in a straight line. Because a six-rod tensegrity structure has only triangles on its outer surface, a robot based on this structure moves in a zig-zag way and switches its heading direction after each and every step. This causes large momentum loss and the structure is not suitable for fast rolling. On the other hand, a twelve-rod tensegrity structure similar to a rhombicuboctahedron has rectangles on its outer surface, which enables straight rolling and prevents loss of momentum.

rod and the other four rods that are perpendicular to the plane have only two-DOF per rod. Furthermore, because the two rods that are mirrored about the plane always have the same motion, the total degrees of freedom for the eight parallel rods are halved to 12-DOF. As a result, the twelve-rod tensegrity robot exploiting the symmetry has a total of 20-DOF. This is only one third of the original system's 60-DOF and is even smaller than the total degrees of freedom of a six-rod tensegrity structure that has 30-DOF. By exploiting this reduction in DOF, dynamics of the twelve-rod tensegrity robot can be compactly written, which will facilitate the robot's locomotion controller design. The development of controllers for achieving dynamic rolling, however, remains as future research. Some potential research directions in this regard have been proposed in Chapter 6.

The outer surface of T12-R consists of two types rectangles and eight equilateral triangles. There are six *perpendicular rectangles* each of which is formed by nodes of four parallel rods and is perpendicular to the rods, and twelve *diagonal rectangles* each of which is formed by nodes of two orthogonal pairs of two parallel rods and is diagonal to the rods (Figure 2.10). The two types of rectangles are analogous to closed and open triangles of a six-rod tensegrity structure.

2.3.1 Rod Design

T12-R is built with 12 hollow aluminum rods that have length of 45 cm and outer diameter of 0.95 cm. T12-R also follows a distributed controller scheme of TT-3, hence clearing the center volume for a payload. As a result, a bundle of electronic components including controllers (Teensy 3.2), batteries (Turnigy) and wireless radios (XBee, Digi) is placed at the center of each rod. Unlike TT-3, however, motors (Pololu) are attached at the rod ends to prevent friction between edge strings and rods. For a fully actuated robot, we would need

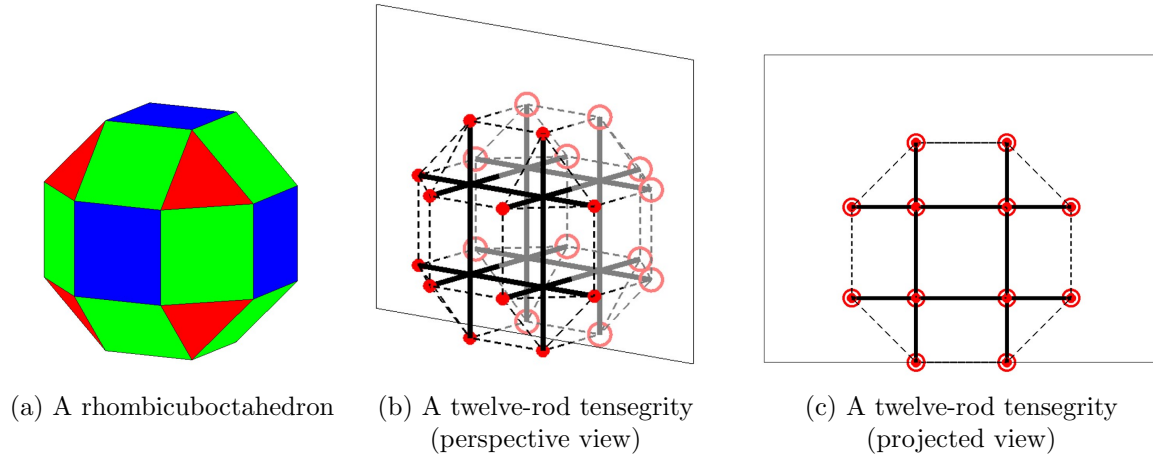


Figure 2.9: A twelve-rod tensegrity structure that T12-R is based on has its outer geometry similar to a rhombicuboctahedron. This geometry is symmetric about three mutually-orthogonal planes one of which is shown in this figure. If the deformation of the structure is symmetric about this plane, then the shape and motion of the structure can be described on this plane by projecting the members of the structure onto the plane.

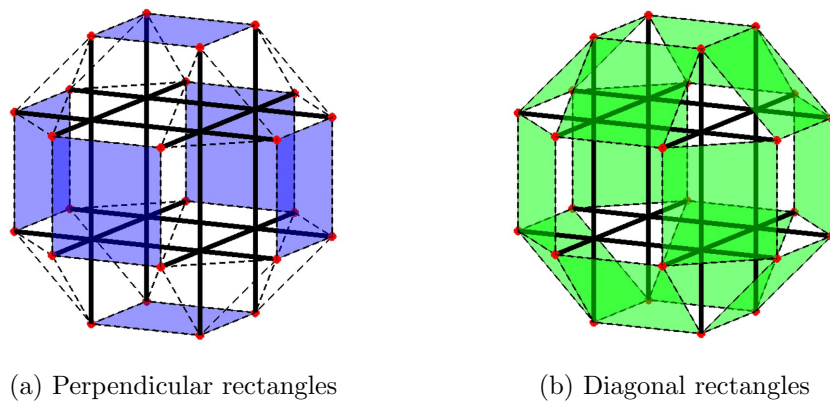


Figure 2.10: The outer surface of T12-R consists of triangles and rectangles. There are six perpendicular and twelve diagonal rectangles.

Table 2.2: Physical parameters of T12-R

Total robot mass	1.78 kg
Rod length	45 cm
Rod mass (including motors and electronics)	132.9 g
Spring rest length	4.45 cm
Spring stiffness	296 N/m
Motor free-run speed at 6V	100 rpm
Motor stall torque at 6V	0.494 Nm

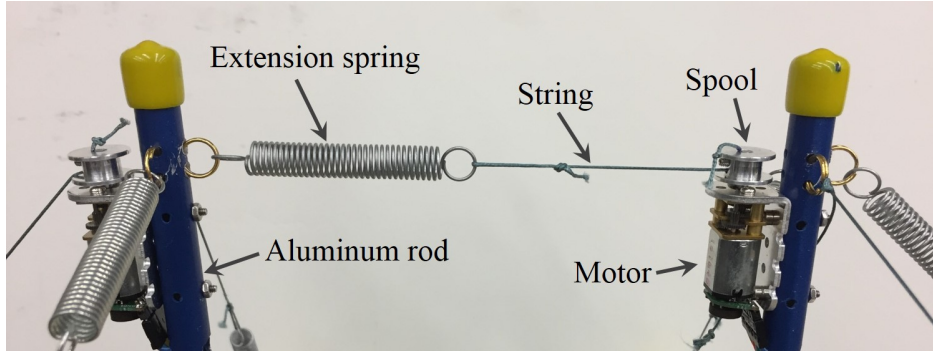


Figure 2.11: An edge of T12-R consists of an extension spring and an unstretchable string that is spooled in and out by a motor.

a total of 48 motors because T12-R has a total of 48 edges, but the current prototype only has 24 motors (two on each rod). Yet, our simulation shows that it is possible to realize locomotion with this underactuated system, as will be discussed in Section 3.4.2.3. Each rod including the motors and electronics weighs 132.9 g. The total weight of T12-R is 1.78 kg. The physical parameters of T12-R are summarized in Table 2.2.

2.3.2 Edge Configuration

The edge configuration of T12-R is the same as TT-3 (Figure 2.11). Each edge of T12-R consists of an extension spring that is connected to an unstretchable string. The length and tension of an edge is controlled by winding and unwinding the string with an actuated spool.

2.4 Conclusions

In this chapter, we have introduced several spherical tensegrity robots built at the BEST laboratory and presented detailed descriptions of their hardware designs. The first few prototypes, namely, TT-1, TT-2, and TT-3, are all based on a six-rod tensegrity structure. This choice of structure is made in the initial phase of the work because the structure is the simplest three-dimensional spherical tensegrity structure and is capable of rolling. These robots can deform their shapes by changing lengths of their edges using actuators, and this feature will be exploited in Chapter 3 where we will study deformation of the robots for the purpose of enabling their locomotion.

One major difference in the design of TT-3 from TT-1 and TT-2 is that TT-3 has distributed controllers across its rods and has a cleared center space, which allows us to place a relatively large payload at the center. In this work, we decided to place a thruster system as a payload, which will enable hopping of the robot. The hopping locomotion of tensegrity robots is discussed in details in Chapter 4.

While the robots based on a six-rod tensegrity structures are capable of rolling, it was pointed out that they are not suitable for high speed rolling because of their triangular outer surfaces. This motivated the development of T12-R that is based on a twelve-rod tensegrity structure and has an outer shape consisting of mostly rectangles. A rhombicuboctahedron-like shape allows T12-R to roll in a straight line without losing much momentum and we believe T12-R can achieve fast and dynamic rolling. The details of hardware design of T12-R was also provided in this chapter. The robots presented in this chapter are used as testbeds of this work, and the control strategies for rolling, which will be developed in the next chapter, are validated using these robots.

Chapter 3

Precision Rolling of Spherical Tensegrity Robots

3.1 Introduction

Being naturally compliant and lightweight, tensegrity robots have two large advantages for co-robotics and soft robotic applications: 1) they can be hit with significant force or fall from significant distance without sustaining serious damage, since the structure tends to deform on impact, therefore absorbing shock, and 2) accidental impact with a human causes minimal harm as the structure absorbs most of the shock. Currently, tensegrity structures are most commonly sold as toys for infants due to these safety properties.

This innate safety is a unique feature of tensegrity robots and encourages them to be operated alongside humans. Hence, tensegrity robots are envisioned for several co-robotic applications, including assistive and rehabilitative healthcare by providing hospital service or direct in-home assistance. If these uniquely deformable robots are equipped with mobility, they may be able to move supplies or medicine, concentrated in their central protected payloads, to doctors and patients around the hospital. Tensegrity robots have also been proposed for rough and unstructured terrain such as construction settings and search and rescue. NASA is exploring tensegrity robots for planetary space exploration as autonomous vehicles or co-robots working with astronauts [59], [93].

In order to be actually deployed for these applications, tensegrity robots should have mobility in the first place. At the time of TT-1 and TT-2 development, it was known that several tensegrity robots, including six-rod ones that are similar to TT-1 and TT-2, can move by using deformation of their structures [48], [49], [69], [90], [94]. For example, control strategies were developed for a six-rod tensegrity robot in simulation using an evolutionary algorithm in [65], [95], where symmetry of the structure is exploited. Japanese researchers built a physical six-rod tensegrity robot with shape memory alloy coils as actuators [52] and then improved it later with pneumatic actuators [56], [61]. They developed control strategies for the mobility of their robots from hardware experiments. An interesting approach was

introduced in [96], where a six-rod tensegrity robot was controlled to move by exciting itself with three vibration motors running at different frequencies.

Although previous research provides control strategies for six-rod tensegrity robots, we were not able to directly import them to our robots TT-1 and TT-2 for several reasons. The policies developed in simulation were based on different physical parameters and there were discrepancies between the simulated models and our hardware robots, which made it difficult to use the outcome of simulations right away. On the other hand, the policies developed from hardware experiments were specific to the physical robots used. Furthermore, both types of policies provided less physical insight on how spherical tensegrity robots locomote. Therefore, a more systematic approach is required for the development of control strategies for the ground mobility of tensegrity robots such that the developed method is general enough to be applied to any spherical tensegrity robots and is easy to test on hardware robots.

3.1.1 Contribution of This Chapter

In this chapter, we first develop control strategies that realize precision rolling of six-rod tensegrity robots, TT-1 and TT-2. We use two different methods for this purpose: a greedy search algorithm and a multi-generation Monte Carlo based learning algorithm in conjunction with a dynamic relaxation technique. The outcome of both algorithms is a set of *actuation policies*. An actuation policy contains information on what commands should be given to actuators to realize certain types of *steps* by specifying lengths of edge members that are controlled by the actuators. In this work, we only consider cable actuated robots, and only edges of tensegrity robots change their lengths for structural deformation. Rods are assumed to be rigid and do not change in their lengths. We validate the developed actuation policies with both algorithms on TT-1 and TT-2 and present the results.

The application of the multi-generation Monte Carlo based method is not limited to six-rod tensegrity robots only. We show that the method can be used to develop actuation policies for other spherical tensegrity robots and illustrate the procedure by taking T12-R as an example. This proves that our method provides a generalized and systematic way to find actuation policies for rolling locomotion of spherical tensegrity robots. We hope that our method will benefit future researchers who are studying locomotion of such robots.

The rest of the chapter is organized as follows. In Section 3.2, we develop actuation policies for six-rod tensegrity robots using a greedy search and present hardware experiment results. In Section 3.3, the form-finding problem of tensegrity robots is introduced and a dynamic relaxation technique is proposed to solve the problem. The details of how this technique can be used together with a multi-generation Monte Carlo method to develop actuation policies for six-rod tensegrity robots are also provided. This approach is then extended to more general spherical tensegrity robots in Section 3.4, and we describe the procedure with T12-R as an example. Lastly, the chapter ends with a conclusion in Section 3.5.

3.2 Actuation Policy Development for Six-Rod Tensegrity Robots with Greedy Search Method

3.2.1 Definition of Step

Precision rolling of a six-rod tensegrity robot can be broken down into a sequence of steps. For our tensegrity robots, a step consists of five different stages (Figure 3.1):

1. *At rest*: The robot is initially undeformed and at rest.
2. *Deformation*: The robot deforms until the ground projection of its center of mass (GCoM) escapes current base triangle.
3. *Rotation*: Rotation about one edge of the base triangle.
4. *Strike*: The robot lands on the next base triangle.
5. *Recovery*: The robot recovers to an undeformed state, preparing for the next step.

Due to the highly symmetric nature of a six-rod tensegrity structure, motion possibilities can be generalized for the triangle classes described in Chapter 2. Within the structure, each closed triangle is surrounded by three open triangles, while each open triangle is surrounded by two closed triangles and one open triangle. As a result, when starting on a closed base triangle, the robot could only rotate onto one of three surrounding open base triangles with a single step. On the other hand, when starting on an open base triangle, the robot could land on either one of the two neighboring closed base triangles or an open base triangle. In summary, three different types of locomotion steps are considered herein:

- *CO-step* leads the robot from a closed base triangle to an adjacent open base triangle.
- *OC-step* leads the robot from an open base triangle to an adjacent closed base triangle.
- *OO-step* leads the robot from an open base triangle to an adjacent open base triangle.

A similar classification of steps is also presented in [56], [61]. While similar, our robot extends their work by being untethered and carrying a payload, which changes the dynamics of motion. Also, while they experimentally found pairs of actuators that will make a step, we first develop actuation policies in simulation by utilizing a greedy search algorithm, and then implement and verify the controllers on the hardware. By taking this approach, we are able to quickly identify new actuation policies whenever there is a hardware change, successfully meeting the concept of rapid prototyping. Moreover, our actuation policies are not limited to include two actuators. However, we only find a single actuation policy per running of the algorithm, while [56], [61] found all possible pairs of actuators resulting in a step. To compensate this, we run the algorithm multiple times and find several actuation policies.

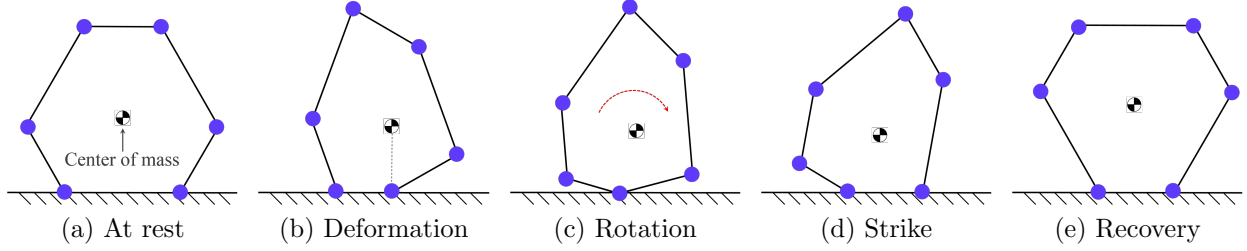


Figure 3.1: A two dimensional conceptual diagram of different stages of a step. The robot deforms its structure using cable actuation, in such a way that its GCoM leaves the current supporting polygon [1].

In the following, we discuss the approach taken to search for actuation policies for CO-, OC- and OO-steps of our robots. The outer cables of TT-1 and TT-2 are actuated with the linear actuators to deform their structures in a desired way. All the linear actuators are fully extended initially, and the actuation policies identify subsets of the actuators that will achieve one or more steps if fully retracted in phase. Furthermore, we only consider steps on a flat ground in this work, although the approach presented below can also be applied to other elevation conditions and terrains.

3.2.2 Greedy Search Method

During the initial stage of TT-1 and TT-2 development, a greedy search algorithm is used to find step-wise actuation policies that will result in one or more steps of the robots from their initial positions. We chose this algorithm because it is well-known that the greedy search can find solutions, even for problems of high complexity, in a reasonable amount of time if good heuristics are used [97]. The solutions, however, are usually not optimal and incomplete. This lack of optimality was not an issue at the time of TT-1 and TT-2 development because the first thing we wanted to check was the possibility of the robots making steps via self-deforming and our goal was to find a set of feasible, but not necessarily optimal, actuation policies. In fact, the greedy search was able to find successful actuation policies for our robots, as will be presented later.

The greedy search algorithm for our problem is implemented in the NASA Tensegrity Robotics Toolkit (NTRT) as follows. First, our heuristic function is defined as the distance between the robot's GCoM and one of the base triangle's edges serving as the rotation axis of that step (Figure 3.2). The heuristic is assumed to be positive if GCoM is within the base triangle and negative if it is outside. Because a six-rod tensegrity structure is well-balanced when not deformed, the search starts from a positive heuristic value. When the heuristic has a value equal to zero, it means that GCoM is located right on the chosen edge and the robot is about to perform a step. We do not track the heuristic beyond this point, which would have been negative if we did. A larger negative heuristic value implies that GCoM

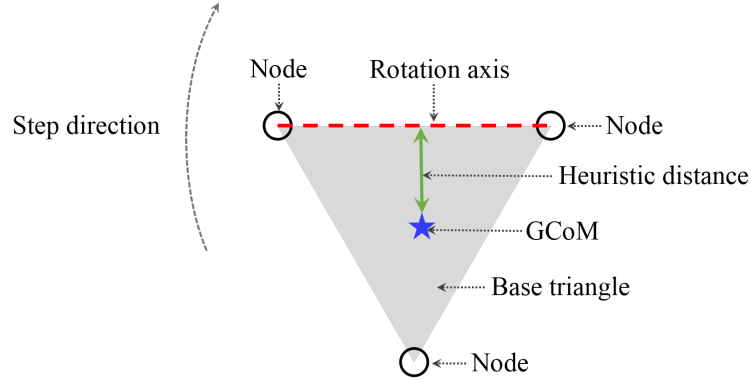


Figure 3.2: Definition of the heuristic distance (about one chosen base triangle edge) used for the greedy search algorithm [1].

is located farther away from the base triangle and the robot is more unbalanced. In other words, a negative heuristic value can be interpreted as a measure of how much *reliable* a step is. However, a proper measurement of negative heuristic values requires the base triangle to be fixed and to be remained on the ground. This is not possible with NTRT simulations because NTRT simulates the motion of the robot and the base triangle is detached from the ground as soon as the heuristic reaches zero. For this reason, we do not consider negative heuristic values in this search method. The goal of the search is then to find actuation policies that result in a heuristic value of zero. As a result, we seek actuation policies that simply realize steps, but the steps are not necessarily reliable¹.

Initially, the robot is set to stand on one of its outer surface triangles – either closed or open – in the simulator and one edge of the base triangle is chosen as a rotation axis for a step to be performed. We assume that all of the linear actuators on TT-1 and TT-2 are fully extended and the robot is not deformed initially. Our action to expand search nodes² of the search tree is to change the lengths of linear actuators. For simplicity, we only consider binary states of the actuators, i.e., fully extended state or fully retracted state, although the linear actuators can be controlled to have any length within their range of operation. This results in a relatively small search space that has 2^{24} states in total. It may seem actuation policies developed under this assumption are somewhat coarse and restricted, but it turned out that the assumption did not prevent us from finding successful policies for all types of steps. The assumption can be easily generalized if one wants finer control on actuator lengths as long as the lengths take discrete values. From the initial undeformed robot, the first expansion of the search tree is done by fully retracting each one of 24 linear

¹We incorporate negative heuristic values in our next method, a multi-generation Monte Carlo based learning algorithm. See Section 3.3.

²Here, the term *node* is used in the context of a search algorithm. Each node can be thought as an abstraction of a potential actuation policy. Recall that in Chapter 2, a node had a different definition and referred to a rod end.

actuators, generating 24 leaf nodes. Once NTRT outputs the robot's deformed structures under these actuation scenarios, we evaluate the resulting structures based on the heuristic and assign their values. The second expansion is done from the leaf node that has the least heuristic value from the first expansion, and we expand the node by fully retracting one of the remaining 23 linear actuators in addition to the first actuator that generated this node. We then evaluate newly generated leaf nodes and do the next expansion from the node that has the least heuristic value of all the nodes generated so far. Again, the expansion is done by fully retracting each one of the linear actuators that do not belong to the node to be expanded, in addition to the actuators that are already fully retracted to generate the node. This procedure is repeated in NTRT until the node with zero heuristic value is found and this goal node is taken as our actuation policy for the step. The search is done separately for different types of steps to develop respective actuation policies.

3.2.2.1 Actuation Policy Search for CO-steps

When TT-1 or TT-2 is standing on one of its closed triangles, it can make a CO-step in three different directions. Each step uses one of the base triangle's edges as a rotation axis. Because of the three-fold symmetry of a six-rod tensegrity structure, actuation policies for the three CO-steps are also expected to be symmetric. Therefore, it suffices to search for actuation policies for only one of the three CO-steps. The actuation policies for not only the other two CO-steps but also all possible CO-steps of the robot, can be easily inferred from this result, as discussed in Section 3.2.3.1.

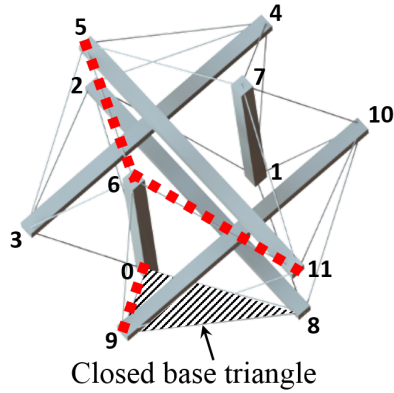
After multiple runs of the search algorithm, two successful actuation policies are found for a CO-step. Both policies include three different actuators (Table 3.1 and Figure 3.3). When these two actuation policies are forward simulated in NTRT, a CO-step is automatically followed by an OC-step, thus arriving at the next closed base triangle. In other words, the actuation policies result in a closed to open to closed base triangle step, or a *COC-step*, skipping the recovery stage of the CO-step. This happens because the applied actuation allows GCoM to additionally cross the narrow width of the next open triangle to make the OC-step. The momentum of the robot gained from the CO-step also provides some assistance.

3.2.2.2 Actuation Policy Search for OC- and OO-steps

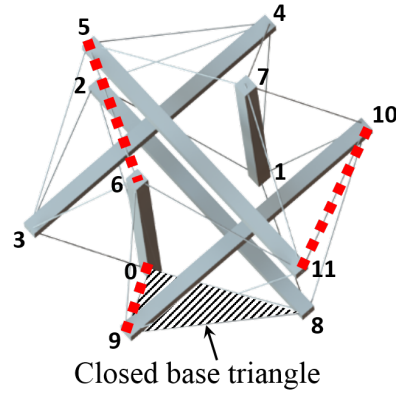
In general, a six-rod tensegrity structure is more well-balanced when it is standing on a closed base triangle than an open base triangle. This happens because open triangles have narrower widths and smaller areas than those of closed triangles due to the missing cables. Therefore, the motion of TT-1 and TT-2 is designed to land on a closed base triangle in most cases during locomotion, but there will be situations when the robots will land on an open base triangle due to instabilities or obstacles. Two different types of steps are available from this pose: OC- and OO-steps. We first consider an OC-step. Because a six-rod tensegrity structure possesses bilateral symmetry about its sagittal plane when standing on an open

Table 3.1: Actuation policies found in NTRT simulations using a greedy search. The numbers are node numbers, see Figures 3.3 and 3.4 for node numbering. 2-tuples represent actuated outer cables defined by their two end nodes and 3-tuples represent triangles formed by the three vertex nodes [1].

Type of step	Starting triangle	Landing triangle	Actuation policy
CO-step	(0,8,9)	(6,9,11)	$\{(0,9), (5,6), (6,11)\}$
CO-step	(0,8,9)	(6,9,11)	$\{(0,9), (5,6), (10,11)\}$
OC-step	(8,9,11)	(0,8,9)	$\{(6,9)\}$
OC-step	(8,9,11)	(0,8,9)	$\{(8,10)\}$
OO-step	(8,9,11)	(8,10,11)	$\{(8,9), (9,11), (7,10), (1,10), (6,11), (0,8), (1,8), (0,3)\}$



(a) CO-step policy 1



(b) CO-step policy 2

Figure 3.3: CO-step policies found in NTRT simulation. Hashed triangles and bold dotted lines represent closed base triangles and actuated cables, respectively. Node numbers used in Table 3.1 are also shown [1].

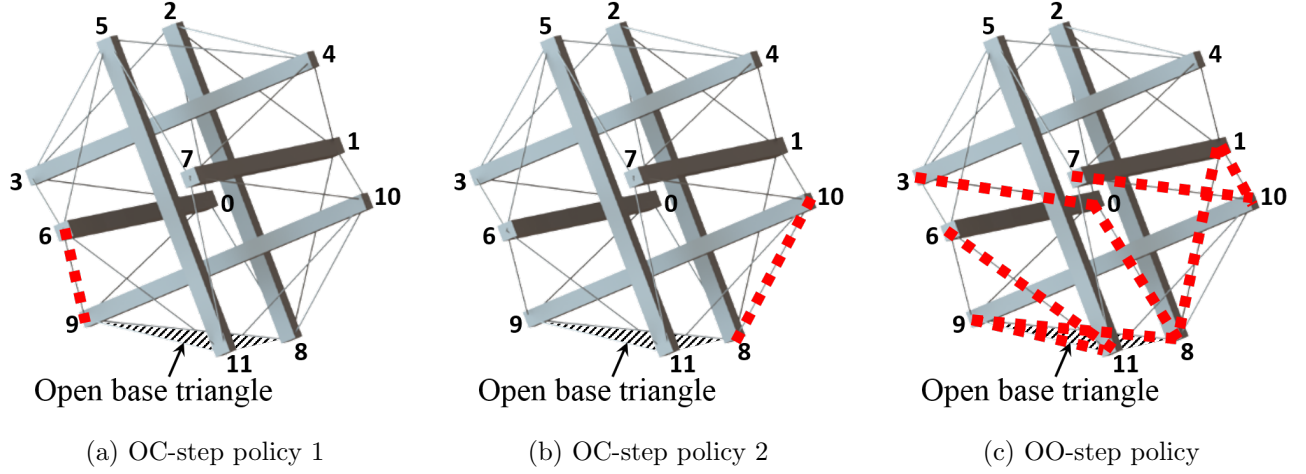


Figure 3.4: OC- and OO-step policies found in simulation. Hashed triangles and thick dotted lines represent open base triangles and actuated cables, respectively. Node numbers used in Table 3.1 are also shown [1].

base triangle, it is sufficient to search for actuation policies that would result in an OC-step towards either of the two adjacent closed base triangles.

After running the search multiple times, two successful policies are found and they are presented in Table 3.1 and Figures 3.4a and 3.4b. They both require only one actuator per step, which implies that less actuation effort is required to make an OC-step than a CO-step that required at least three actuators per step. This agrees with our previous observation that the open triangles provide less static balance compared to the closed triangles, and less deformation is sufficient to make a step from open base triangles.

For the sake of completeness, an actuation policy for an OO-step was also found (Table 3.1 and Figure 3.4c). The actuation policy involves eight actuators and is energy inefficient compared to CO- and OC-steps. For this reason, this policy has not been implemented on our hardware robots. Omitting this step does not restrict general motions of the robots, as will be shown in Section 3.2.3.2.

3.2.3 Experiments

3.2.3.1 Implementation and Extension of Actuation Policies

The actuation policies found in the NTRT simulation (Table 3.1) are implemented and tested on TT-1. First, the CO-step policies in Table 3.1 are generalized to cover all possible CO-steps from any closed base triangle. From a set of experiments, however, we observed that while some of these generalized policies are successful in making CO-steps, others are marginally unable to rotate the robot – a slight perturbation will initiate the step in some cases. This happens because our algorithm only provides information as to whether a given

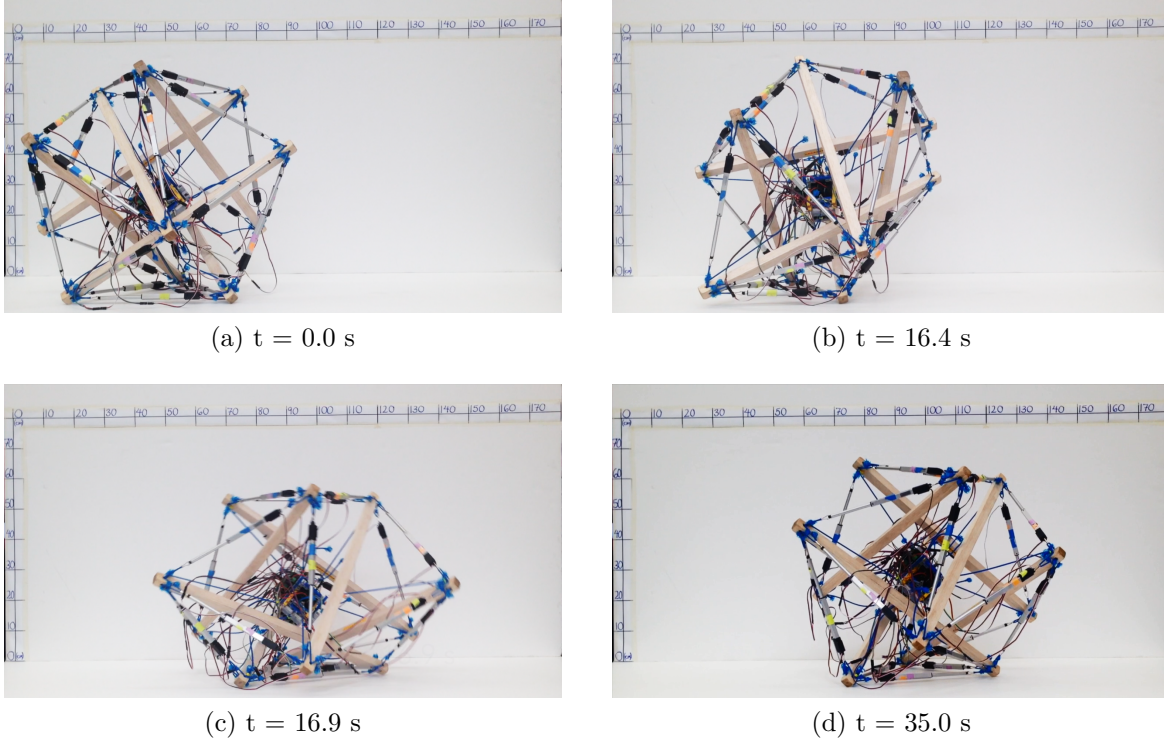


Figure 3.5: Still images of TT-1 performing a COC-step [1]. The robot starts with a CO-step and the following OC-step is automatically performed. (a) Initially at rest. (b) Deformation. (c) Rotation and Strike. (d) Recovery. A full motion video is available at <http://best.berkeley.edu/>.

policy would zero the heuristic distance and allow the robot to make a step. Because we did not consider negative heuristic values, which could be interpreted as a measure of how much unbalanced a deformed structure is, our heuristic does not tell us how much reliable the step will be. Therefore, the algorithm returns the policies that are just enough to cause a step in simulation; such policies might fail in hardware due to the stochastic sensitivity of geometries and environmental conditions. However, when the two CO-step policies found in the simulation (Table 3.1 and Figure 3.3) are combined together to create a single policy involving four actuators (two of the actuators are common in both actuation policies, see Figure 3.6a), the success rate increases. We then generalized and tested this extended CO-step policy on all closed base triangles of TT-1 and TT-2. Each possible step (three from each of the eight closed triangles) was tested three times to check for consistency. Our experiments showed that all of the extended CO-step policies are successful in making desired COC-steps (i.e., a CO-step followed by an OC-step) for all trials, as predicted by the simulation. Snapshots of TT-1 performing a COC-step are presented in Figure 3.5.

The OC-step policies developed in the NTRT simulation (Table 3.1) are also generalized

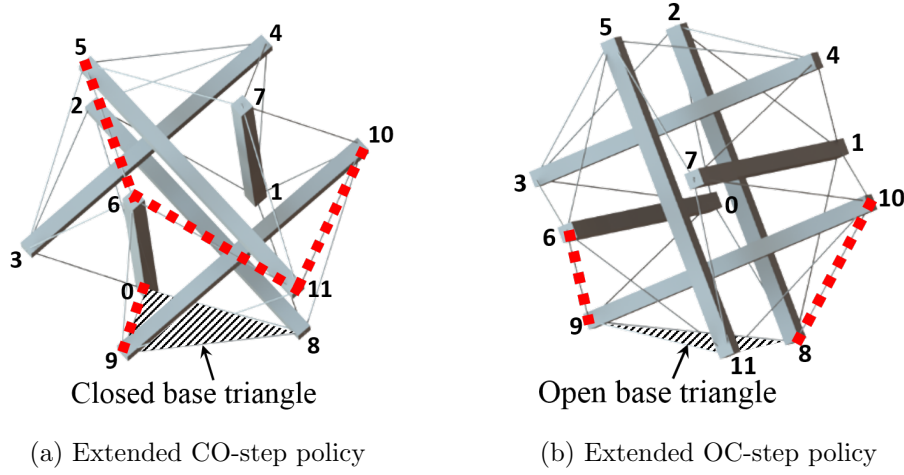


Figure 3.6: Extended policies implemented on TT-1 and TT-2. Hashed triangles and thick dotted lines represent base triangles and actuated cables, respectively [1].

to cover all possible OC-steps from any open base triangle and are tested on TT-1. As in the previous case, it was observed that some of the generalized OC-step policies are successful in making steps, but the rest are marginally unsuccessful in causing rotation. As before, the two OC-step policies found in the simulation (Table 3.1 and Figures 3.4a and 3.4b) are combined together to create a single policy involving two actuators (Figure 3.6b), and the extended OC-step policy is generalized to cover all other possible OC-steps. Then, all of the extended OC-step policies are implemented on TT-1 and TT-2. Each of the possible OC-steps (two from each of the twelve open triangles) was tested three times and all trials were successful.

To examine the validity of the applied policies, the motion of TT-1 performing a COC-step was tracked using a Vicon[®] motion tracking system. The obtained trajectories of the GCoM and base triangle nodes during this step are shown in Figure 3.7a. The markers in the figure represent the positions of the GCoM and nodes for every 0.1 seconds. Initially, the GCoM is located near the center of the base triangle. During the deformation phase, the GCoM moves towards the rotation axis of the step, reducing the heuristic distance. Node C is also moving towards Node A because the actuator between the two nodes is in action during this step. In fact, among the four actuators included in the extended CO-step policy, simulation results and hardware experiments identify this actuator as the most critical one for reducing the heuristic distance. The rotation is initiated as soon as the GCoM crosses over the rotation axis. This can be inferred from the large distance between the last two GCoM markers in Figure 3.7a. The change of the heuristic during a COC-step, measured from both simulation and hardware experiment, is shown in Figure 3.7b.

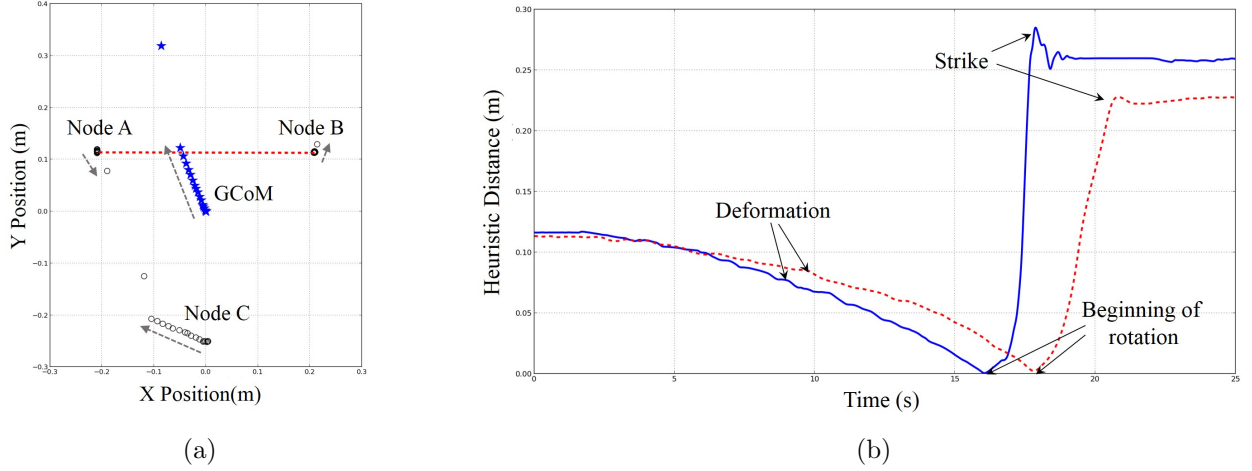


Figure 3.7: (a) Trajectories of GCoM and ground contacting nodes during a CO-step, tracked with a Vicon[®] system. Markers represent the positions of GCoM and nodes for every 0.1 seconds. Dashed arrows represent the motion directions of GCoM and nodes. (b) Change of heuristic distances during a COC-step. Solid and dashed lines represent experiment and simulation results, respectively. The robot starts to rotate as soon as the heuristic distance becomes zero [1].

3.2.3.2 Motion Primitives

More generalized motions of the robots, such as moving forward or turning left and right, have been designed by linking together appropriate COC-steps (realized by using the extended CO-step policy). These motions were tracked using a Vicon[®] motion capture system and the obtained GCoM trajectories during the motions are shown in Figure 3.8. In Figure 3.8a, the robot starts from the origin and performs five COC-steps in a zig-zag fashion about the initialized axis of interest to move forward. For the turning motions in Figures 3.8b and 3.8c, the robot starts from the origin, performs three COC-steps to move forward, and then changes its heading direction at the fourth step. The execution of turning motions shows that the robot is steerable, which is desirable when the robot needs to avoid obstacles in, e.g., co-robotic applications. These basic motions could serve as motion primitives of the robot and may be combined together to create more complex paths. Finally, we note that only CO- and OC-steps are needed to perform the motions and OO-steps are not required.

3.2.4 Energy Efficiency of Steps

In [52], it was shown that the gravitational potential energy change during the three different types of steps was identical when a non-deformable six-rod tensegrity structure was assumed and approximated as a rigid regular icosahedron and when its rods were the only mass

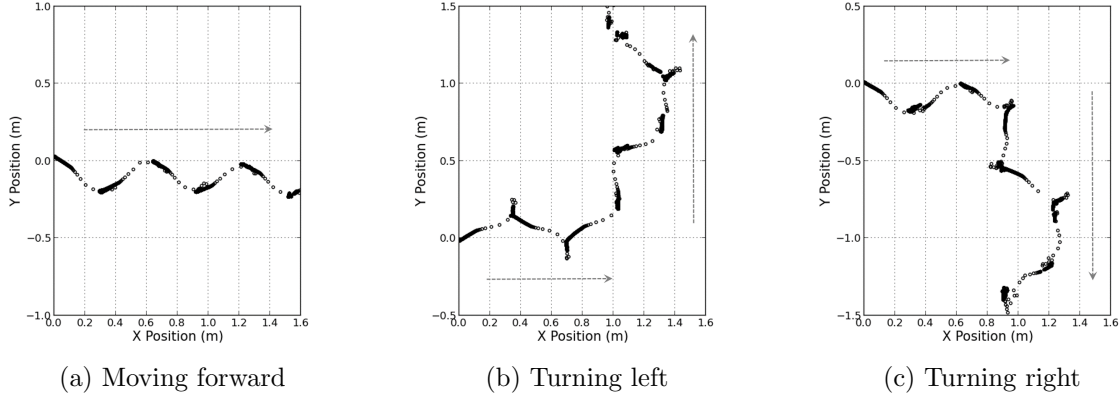


Figure 3.8: Trajectories of GCoM tracked with Vicon[®] system during the robot motions. Markers represent positions of GCoM for every 0.1 seconds [1].

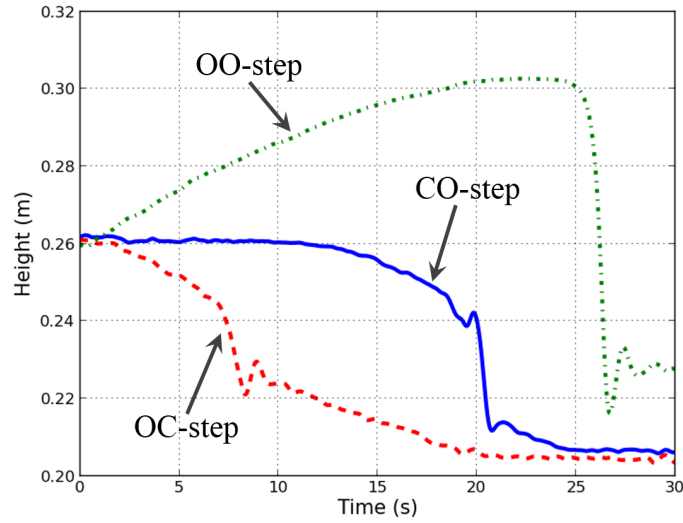


Figure 3.9: The NTRT simulated result of height changes of the robot's center of mass while performing different steps with the developed actuation policies. Solid, dash, and dot-dash lines represent CO-, OC- and OO-steps, respectively. Notice that in order to perform an OO-step, the robot has to overcome a high potential energy barrier. Actuation policies provided in Figures 3.4c and 3.6 were used to simulate the steps [1].

components. These assumptions are no longer valid in realistic robots like we have built, due to the following reasons:

- The robot’s geometry is not a regular icosahedron.
- Locomotion is generated by structural deformation.
- Rods, actuators and payloads account for a large portion of the robot’s total weight.

For a more realistic analysis, we have tracked the height change of the robot’s center of mass during CO-, OC- and OO-steps in the simulation. The steps are performed using the actuation policies presented in Figure 3.4c for an OO-step and Figure 3.6 for CO- and OC-steps. The result is shown in Figure 3.9.

Our simulations show that the maximum height difference of the robot’s center of mass during an OO-step is larger than those of CO- and OC-steps. In other words, the robot has to overcome a higher potential energy barrier when performing an OO-step. In fact, the OO-step policy requires eight actuators to overcome the barrier, while the CO-step and OC-step policies require only four and two actuators, respectively. As OO-steps turn out to be energy inefficient, only OC-steps are considered for the hardware movement when starting on an open base triangle. As a result, the robot’s most efficient locomotion is a sequence of alternating CO- and OC-steps, as we already presented in Section 3.2.3.

3.3 Actuation Policy Development for Six-Rod Tensegrity Robots with Multi-Generation Monte Carlo Based Learning Method

In the previous section, we demonstrated that the greedy search method is a quick and effective way to find actuation policies that can be easily implemented on hardware robots and successfully realize different types of steps. There are, however, a couple of drawbacks in this approach. First, we were not able to handle negative heuristic values that could be interpreted as a quality measure of steps. As discussed in Section 3.2.2, this limitation occurred from the use of NTRT as our simulation tool. Because NTRT simulates dynamics of tensegrity structures, the robot modeled in NTRT initiates a step as soon as its heuristic becomes zero and no further tracking of negative heuristic values is possible. As a result, the actuation policies obtained by the greedy search were able to realize marginal steps only. For more reliable steps, the actuation policies had to be combined together to create more robust policies. Therefore, while NTRT is a desirable tool to have for simulation of tensegrity dynamics, we concluded that the simulator is not adequate for the development of robust actuation policies with our methods. Because we investigate deformation and shapes, but not dynamics, of tensegrity structures for our actuation policy development, we would prefer to have a tool that simulates statics of tensegrity structures. In this section, we

develop a method that solves the statics problem by using a dynamic relaxation technique and implement it in MATLAB. The tensegrity structures modeled in this section, however, do not include central payloads and are not subject to gravity. These assumptions are made because we aim to study how redistribution of internal forces through actuation affects the shape of a tensegrity structure, while excluding the contribution of external forces to the shape. The removal of the gravity also results in the tensegrity deformations that are unbiased towards the gravity direction, and allows us to evaluate the shapes about all possible orientations in the space, as will be discussed in Section 3.3.2.2. This maximizes the utility of the shapes that are obtained by running the dynamic relaxation, and thus minimizes the computational effort. It turns out that the deformation of the hardware robot TT-2 is very close to the simulated model under these assumptions, and the result will be presented in Table 3.4. Furthermore, our empirical observations showed that the gravity effect on the deformations is reduced as the pretension on tensegrity structures increases. Nevertheless, the central payloads and the gravity can be easily included in our method, if needed.

The other downside of the greedy search method is a huge search space that we get when we attempt to generalize the method. Recall from the previous section that the actuators were allowed to have only two states – full extension and full retraction. For the binary state case, there are total of 2^{24} states in the search space. If the assumption is relaxed and each actuator is allowed to have n multiple states, then the total number of states increases to n^{24} , which could be a huge number if n is large. Furthermore, there exist many other spherical tensegrity structures that have more than 24 cables, and their search spaces would be even larger. Therefore, a search based method may become significantly challenging and expensive, and it is not easy to generalize the method to other tensegrity systems. To overcome this difficulty, we propose an evolutionary approach based on Monte Carlo sampling in this section. A similar approach was presented in [69], [98] where the controllers of tensegrity robots were parameterized first and the best performing sets of parameters were identified with a multi-step Monte Carlo. While these works focus on generating continuous locomotion of tensegrity robots in simulation, we seek actuation policies for discretized basic motions (i.e., steps), which can be easily implemented and tested on our hardware robots. Our method relies on the center of mass and provides a deeper insight on how spherical tensegrity robots perform steps. Again, we use MATLAB to implement and simulate the Monte Carlo based method. Hence, the software developed in this section ties dynamic relaxation and Monte Carlo methods together to find actuation policies for our hardware robots.

3.3.1 Form-Finding by Dynamic Relaxation

A tensegrity structure sustains its shape by delicately balancing forces across its members. This implies that, whenever there is a change in the internal force distribution of the structure, its shape also changes and vice versa. The close relationship between the internal force distribution and the shape of a tensegrity structure implies that statics and kinematics of tensegrity structures are tightly coupled and that they need to be solved simultaneously.

This is a unique characteristic of tensegrity structure that is not seen from other conventional mechanisms. The problem of finding shapes of tensegrity structures under given member forces is referred to as a *form-finding* problem, and special techniques are required to solve the problem. The solution of a form-finding problem is a set of *equilibrium configurations*.

Several techniques have been developed in the previous research to address this problem and their approaches are largely divided into two categories – analytical or numerical. In [15], an analytical description of a simple planar 2-DOF tensegrity mechanism is developed to solve for kinematics and statics of the mechanism. As for an example in three dimensional space, static equations of a tensegrity structure similar to a Stewart platform are developed using screw calculus in [16]. In [17], mathematical models of antiprism tensegrity structures have been developed using the principle of virtual work for static analysis of the structures subject to external loads. As can be seen from these previous works, analytical methods can be used to determine equilibrium shapes of tensegrity structures. However, the methods are mostly applied to simple tensegrity structures that are constructed with a small number of members or belong to a special class. The static equations of tensegrity structures having many members or irregular shapes are quite complicated and are difficult to solve.

To overcome the challenge, several numerical approaches have been developed as well. An adaptive force density method is proposed in [18], which improves upon a force density method originally presented in [99] for general networks. The method is based on eigenvalue analysis and spectral decomposition of an *equilibrium matrix* that represents equilibrium of nodes³ whose positions are not fixed in space. The method, however, requires the equilibrium matrix to satisfy a certain condition in order to properly include compression members of tensegrity structures in the analysis. In [19], a Monte Carlo based method is developed, which can find shapes of irregular tensegrity structures of large scale. In this method, random displacements are made to nodes iteratively until the potential energy of the system is minimized⁴.

Among the numerical methods available, we pay special attention to a well-established method called a dynamic relaxation. In [100], [101], the method is used for form-finding of wide-span cable nets and pre-stressed fabric membranes. Later on, the method is applied to tensegrity structures to find equilibrium configurations of irregular shapes [20], [21]. We build on these previous works and use the dynamic relaxation to find equilibrium configurations of our hardware robots. While most of our procedure follows the previous works, we make one improvement. In the previous research, rods and cables are distinguished by their stiffnesses, and the chance of finding proper equilibrium configurations depends on the choice of the stiffnesses. In our work, we treat rods and cables separately by developing different force equations for each of them. Moreover, we explicitly include the situation where some cables

³The term *node* here refers to a rod end.

⁴We also use a Monte Carlo based method for the actuation policy development in later sections. However, Monte Carlo is applied on a different level; we use solutions of form-finding problem as our input to Monte Carlo, while [19] uses Monte Carlo to find solutions of form-finding problem. Therefore, we use equilibrium configurations of tensegrity structures as our samples, while [19] uses nodal displacements as samples.

are slack and bear no tension in our equations.

A more extensive review on statics and form-finding of tensegrity structures can be found in [9], [102].

3.3.1.1 Dynamic Relaxation

The purpose of the dynamic relaxation is to find an equilibrium configuration of a cable net structure in an iterative way, starting from an initial configuration that does not necessarily satisfy the force balance condition at all or a subset of nodes of the structure [20], [101]. External forces applied to the structure may also be considered in the form-finding process. This section describes how the dynamic relaxation with kinetic damping is used to find an equilibrium of a six-rod tensegrity structure. This type of dynamic relaxation has been shown to be stable and convergent for systems with large local disturbances [101], which is the case of tensegrity structures.

The dynamic relaxation with kinetic damping is based on Newton's second law. First, we model a rod as two point masses located at its end nodes and assume that the masses have a rigid and massless connection between them. Consider a node i ($i = 1, \dots, 12$), with the nodal mass m_i , and assume that a force $\mathbf{F}_i(t)$ is applied to the node, where the force is a function of time. Then the motion of the node is governed by Newton's second law.

$$\mathbf{F}_i(t) = m_i \mathbf{a}_i(t). \quad (3.1)$$

In the above equation, $\mathbf{a}_i(t)$ is the acceleration of node i at time t . Using the centered finite difference form of the velocity, the acceleration can be approximated.

$$\mathbf{a}_i(t) = \dot{\mathbf{v}}_i(t) \approx \frac{\mathbf{v}_i(t + \Delta t/2) - \mathbf{v}_i(t - \Delta t/2)}{\Delta t}. \quad (3.2)$$

In (3.2), $\mathbf{v}_i(t)$ is the velocity of node i at time t and Δt is the time difference between two updates.

Substituting (3.2) into (3.1) gives an iterative form of velocity update.

$$\mathbf{v}_i(t + \Delta t/2) = \mathbf{v}_i(t - \Delta t/2) + \frac{\Delta t}{m_i} \mathbf{F}_i(t). \quad (3.3)$$

In (3.3), m_i has a fictitious value, that is, the mass may or may not be taken from an actual physical system. Usually, m_i and Δt are tuned for good convergence of the algorithm [101]. As a result, if the total force $\mathbf{F}_i(t)$ applied to node i at time t is known, the velocity of the node can be updated for the next time step using (3.3).

The position of node i at time t , denoted here as $\mathbf{r}_i(t)$, can also be updated using the updated velocity.

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t + \Delta t/2) \Delta t. \quad (3.4)$$

The dynamic relaxation is an iterative method that aims to find an equilibrium of the structure from an initial configuration which may be given arbitrarily. That is, the initial

nodal positions $\mathbf{r}_i(0)$ for all i may be chosen arbitrarily as long as they satisfy the rod length constraint. Furthermore, the initial nodal velocities are set to zero in the dynamic relaxation, that is, $\mathbf{v}_i(0) = \mathbf{0}$ for all i . Because the centered finite difference form is used for the velocity, (3.3) is slightly modified for the first velocity update.

$$\mathbf{v}_i(\Delta t/2) = \frac{\Delta t}{2m_i} \mathbf{F}_i(0). \quad (3.5)$$

In summary, if the initial configuration of the structure is known and all of the nodal forces are tracked over time, then the nodal positions, velocities and accelerations can be computed for the later time steps iteratively.

In the dynamic relaxation with kinetic damping, kinetic energy of the system is tracked over time.

$$\text{KE}(t) = \sum_{i=1}^{12} \frac{1}{2} m_i \mathbf{v}_i(t) \cdot \mathbf{v}_i(t). \quad (3.6)$$

When the peak of the kinetic energy is detected, then all of the nodal velocities and nodal forces are set to zero. This is why the method is called *kinetic damping*. By taking this step, energy is dissipated from the system, moving the system towards a local minimum energy state, or an equilibrium. The iteration restarts from the beginning with the new initial configuration defined as the latest configuration at the energy peak. Finally, the whole process is repeated until the kinetic energy as well as the sum of all nodal force magnitudes converge to zero within an error bound. At convergence, the final configuration is regarded as an equilibrium. An example plot of changes of kinetic energy and the sum of all nodal force magnitudes over time is shown in Figure 3.10.

It is noted here that, because of the way the dynamic relaxation finds an equilibrium, the intermediate states do not necessarily represent the actual physical behavior of the structure. Only the final equilibrium configuration is physically meaningful.

3.3.1.2 Nodal Forces

Because each node is connected to a rod and four cables, the nodal force $\mathbf{F}_i(t)$ consists of two different types of forces.

1. $\mathbf{F}_i^s(t)$: Spring forces applied by the cables.
2. $\mathbf{F}_i^r(t)$: A constraint force applied by the rod, or equivalently, by the rigid connection between the rod's end nodes under our point mass assumption.

The total nodal force in (3.1) is then the sum of the two forces.

$$\mathbf{F}_i(t) = \mathbf{F}_i^s(t) + \mathbf{F}_i^r(t). \quad (3.7)$$

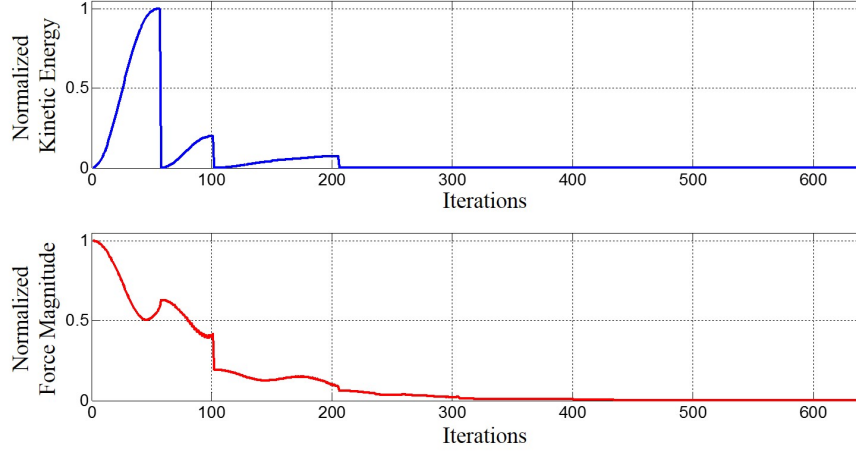


Figure 3.10: Changes of kinetic energy and sum of nodal force magnitudes over time during a single execution of the dynamic relaxation. The quantities are normalized with respect to the maximum values of each case. Notice that both values converge to zero, meaning that an equilibrium configuration is found [2], [3].

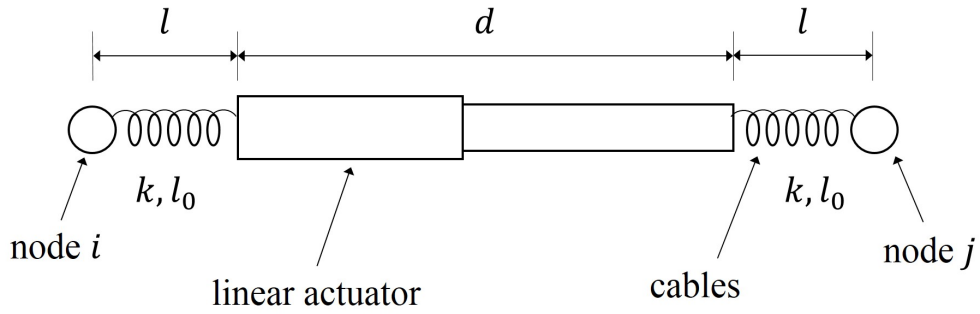


Figure 3.11: A diagram of edge connections of TT-1 and TT-2. Cables are connected to nodes and a linear actuator located at the center of the edge. See Figure 2.3 for an actual hardware setup [3].

Let $J^i = \{i_1, \dots, i_4\}$ represent a set of neighbor nodes connected to node i by cables. Then, at each time step t , the spring force $\mathbf{F}_i^s(t)$ is the sum of individual cable forces.

$$\mathbf{F}_i^s(t) = \sum_{\forall j \in J^i} \mathbf{F}_{ij}^s(t). \quad (3.8)$$

In (3.8), $\mathbf{F}_{ij}^s(t)$ represents the force exerted on node i at time t by a cable connecting nodes i and j . In order to obtain an expression for this force, consider Figure 3.11 which depicts an edge configuration of TT-1 and TT-2 (Figure 2.3). At each end of the edge are cables connected to nodes, and the two cables are actuated by a linear actuator located at the center of the edge. Because the two cables are identical, when actuated, they stretch the same in lengths. If the cables are linear with stiffness of k and have rest length of l_0 , and if the stretched lengths of the cables and the length of the actuator are denoted as l and d , respectively, then $\mathbf{F}_{ij}^s(t)$ has the following expression.

$$\mathbf{F}_{ij}^s(t) = \begin{cases} k [l(t) - l_0] \frac{\mathbf{r}_j(t) - \mathbf{r}_i(t)}{\|\mathbf{r}_j(t) - \mathbf{r}_i(t)\|} & \text{if } l > l_0 \\ \mathbf{0} & \text{if } l \leq l_0. \end{cases} \quad (3.9)$$

$$l(t) = \frac{1}{2}(\|\mathbf{r}_j(t) - \mathbf{r}_i(t)\| - d). \quad (3.10)$$

In tensegrity structures, cables can only bear tensile forces. Therefore, when the cables become shorter⁵ than their rest lengths, the cable forces are set to zero, as in (3.9). Physically, this means we have slack cables. Also, notice that we set d to be a constant and independent of time. This is done so intentionally because we will run the dynamic relaxation to find an equilibrium configuration for a given set of actuator lengths that are assumed to be constant, when we deploy a Monte Carlo based algorithm in Section 3.3.2.

In this section, the lengths of the actuators are manipulated to apply controlled forces to the robot structure, and thus, to deform it. Specifically, in Section 3.3.2, a number of 24-dimensional vectors of the actuator lengths are randomly sampled, and their resultant equilibrium configurations are found and evaluated. It is assumed that the actuators can provide large enough forces to fully operate within their stroke range, which is the case of our hardware robots as discussed in Chapter 2.

The next type of force applied to node i is a rod constraint force, $\mathbf{F}_i^r(t)$. This force does not appear in pure tensile structures and is a unique feature of tensegrity structures. It is critical to obtain the correct expression for this force, as it guarantees a constant distance between the two end nodes of the rigid rod, and thus, the integrity of the whole structure.

In order to describe the constraint force, the coordinate systems shown in Figure 3.12 are used. The position of the first node of a rod is described by a Cartesian coordinate system with a set of right-handed orthonormal basis vectors $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$. Additionally, the position of the other node is described relative to the first one by using a spherical coordinate system

⁵Here, we are measuring the straight line distances between the two end points of the cables rather than their actual physical lengths.

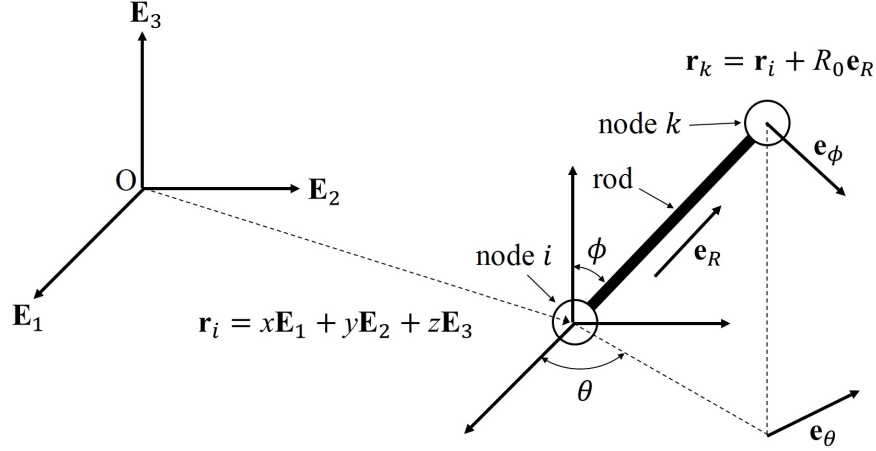


Figure 3.12: Coordinate systems used for describing kinematics of two end nodes of a rod. A Cartesian coordinate system, with a set of right-handed orthonormal basis vectors $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$, is used to describe the position of the first node. On the other hand, a spherical coordinate system, with a set of right-handed orthonormal basis vectors $\{\mathbf{e}_R, \mathbf{e}_\phi, \mathbf{e}_\theta\}$, is used to describe the position of the other node relative to the first one. R_0 is the rod length [3].

with a set of right-handed orthonormal basis vectors $\{\mathbf{e}_R, \mathbf{e}_\phi, \mathbf{e}_\theta\}$. Then, the constraint forces acting on end nodes i and k of the rod are described as follows.

$$\mathbf{F}_i^r(t) = -F_{ik}^r(t)\mathbf{e}_R(t), \quad \mathbf{F}_k^r(t) = F_{ik}^r(t)\mathbf{e}_R(t). \quad (3.11)$$

$$F_{ik}^r(t) = m_k [\ddot{x} \sin(\phi) \cos(\theta) + \ddot{y} \sin(\phi) \sin(\theta) + \ddot{z} \cos(\phi) - R_0 \dot{\phi}^2 - R_0 \dot{\theta}^2 \sin^2(\phi)] - \mathbf{F}_k^s \cdot \mathbf{e}_R. \quad (3.12)$$

$$\mathbf{e}_R(t) = \frac{\mathbf{r}_k(t) - \mathbf{r}_i(t)}{\|\mathbf{r}_k(t) - \mathbf{r}_i(t)\|}. \quad (3.13)$$

In (3.12), m_k and R_0 are mass of node k and the rod length, respectively. The coordinates x, y, z, ϕ and θ as well as the force $\mathbf{F}_k^s$ and $\mathbf{e}_R$ are functions of time, but their notations are omitted for the sake of better readability of the equation. Because the nodal positions $\mathbf{r}_i(t)$ and $\mathbf{r}_k(t)$ and the nodal velocities $\mathbf{v}_i(t)$ and $\mathbf{v}_k(t)$ are updated iteratively using (3.3) and (3.4), the angular coordinates $\phi(t)$ and $\theta(t)$ and their derivatives $\dot{\phi}(t)$ and $\dot{\theta}(t)$ can also be updated using coordinate transformations. Moreover, the acceleration of node i ,

$$\mathbf{a}_i(t) = \dot{\mathbf{v}}_i(t) = \ddot{x}(t)\mathbf{E}_1 + \ddot{y}(t)\mathbf{E}_2 + \ddot{z}(t)\mathbf{E}_3, \quad (3.14)$$

is already given by (3.2). As a result, the rod constraint force $\mathbf{F}_i^r(t) = -\mathbf{F}_k^r(t)$ can be updated over time.

Our simulations showed that, when the rod constraint forces are appropriately applied, the maximum error in rod lengths was 0.12% when running the dynamic relaxation.

3.3.2 Monte Carlo Sampling of Equilibrium Configurations

In Section 3.3.1, the dynamic relaxation with kinetic damping was used to find an equilibrium of a six-rod tensegrity structure when an initial configuration and a set of actuator lengths were given. Clearly, different sets of actuator lengths will result in different equilibrium configurations. Some of these equilibrium configurations will allow the robot to make a step from one base triangle to another, while the others will not. In order to make a step, GCoM should be placed outside of the robot structure's base triangle (Figure 3.1). The goal of this section is to find equilibrium configurations that satisfy the said condition using a single generation Monte Carlo sampling approach.

3.3.2.1 Sampling

Our cable-driven six-rod tensegrity robots, TT-1 and TT-2, are fully actuated with 24 linear actuators, allowing each cable to be independently actuated. If the lengths of all 24 actuators are known, then the equilibrium configuration associated with this length set can be found by running the dynamic relaxation.

Let us denote a vector of all 24 actuator lengths as $\mathbf{d} = [d_1, \dots, d_{24}]^T \in \mathbb{R}^{24}$, where d_i is the target length of the i -th actuator. To find desirable equilibrium configurations, a number of instances of the vector $\mathbf{d}$ are sampled by sampling each component d_i independently from a uniform distribution within a physically acceptable range. For each sampled $\mathbf{d}$, the dynamic relaxation with kinetic damping is performed from an initial configuration that is undeformed and symmetric, and the resulting equilibrium configuration is found. Each equilibrium is then scored with an evaluation function described in the next paragraph. This process is repeated over a large number of samples. Finally, the best equilibrium configuration and the sampled vector $\mathbf{d}$ that produced this equilibrium are identified.

3.3.2.2 Evaluation

As discussed in Section 3.1, in order to generate a step from a six-rod tensegrity robot, its GCoM should be placed outside of its base triangle. Assuming masses⁶ are uniformly distributed in rods and cables have negligible masses, the center of mass of the structure can be easily obtained once nodal positions at an equilibrium are known.

To evaluate each equilibrium configuration, the following evaluation function is used. First, the center of mass is projected onto the planes of the outer surface triangles of the structure. Since, in most cases, the robot steps to and from a closed triangle (Section 3.2), only this type of triangle is considered when evaluating an equilibrium. That is, the center of mass is projected onto eight different planes that define the closed triangles. Next, for each projection, the distances between the projected point and three edges of a closed triangle are measured, as shown in Figure 3.13. If the projected point crosses over an edge and

⁶Here, we are referring to an actual physical mass of a rod, not to the (fictitious) nodal masses that we used when formulating the dynamic relaxation in Section 3.3.1.1.

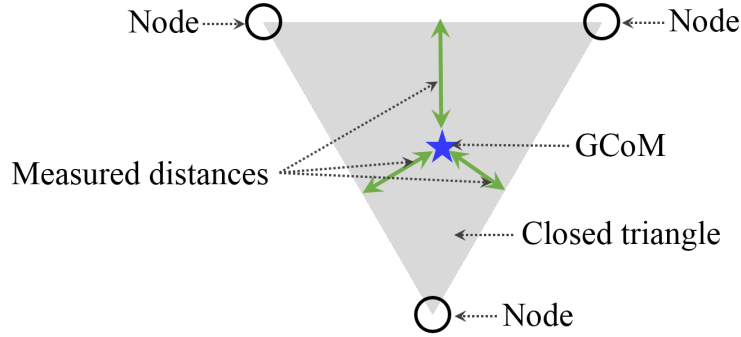


Figure 3.13: Distances measured between GCoM and three triangle edges [2], [3].

moves outside of a triangle, that distance is measured as a negative value⁷. Because the structural geometry should push GCoM as far as possible from a base triangle for a robust step, our goal is to minimize this distance towards a large negative value. For this reason, 24 distances are computed per equilibrium (three distances per triangle, eight closed triangles) and the minimum of these values is assigned as a score for that configuration. Finally, among a set of equilibrium configurations, the one with the minimum score is picked as the best configuration of the set.

3.3.2.3 Simulation Results

To demonstrate the aforementioned procedure, 5,000 samples of $\mathbf{d}$ were obtained. Their resultant equilibrium configurations were found with the dynamic relaxation and evaluated with the evaluation function. The physical parameters used in the simulation were taken from our physical robots TT-1 and TT-2, and are listed in Table 3.2⁸. The result is shown in Figure 3.14. The minimum score of all equilibrium configurations was 0.019. Because the lowest score has a positive value, it is anticipated that even this best equilibrium configuration will not allow the robot to make a step as the robot structure's GCoM will stay inside of its base triangle even after structural deformation. To resolve this problem, a learning approach is deployed in Section 3.3.3.

3.3.3 Multi-Generation Monte Carlo Based Learning Algorithm

Due to the high dimensionality of sampled vectors $\mathbf{d}$ and the wide interval of the sampling space, it is unlikely that a basic Monte Carlo will discover an adequate solution

⁷It was not possible to handle this case in the NTRT simulations because the simulated robot will start a step as soon as this distance becomes zero. See Section 3.2.2.

⁸This table is the same as Table 2.1 except that the rod length is slightly shortened to compensate for the distance between the cable connection point and the very end point of the rod. We reproduced the table for the reader's convenience.

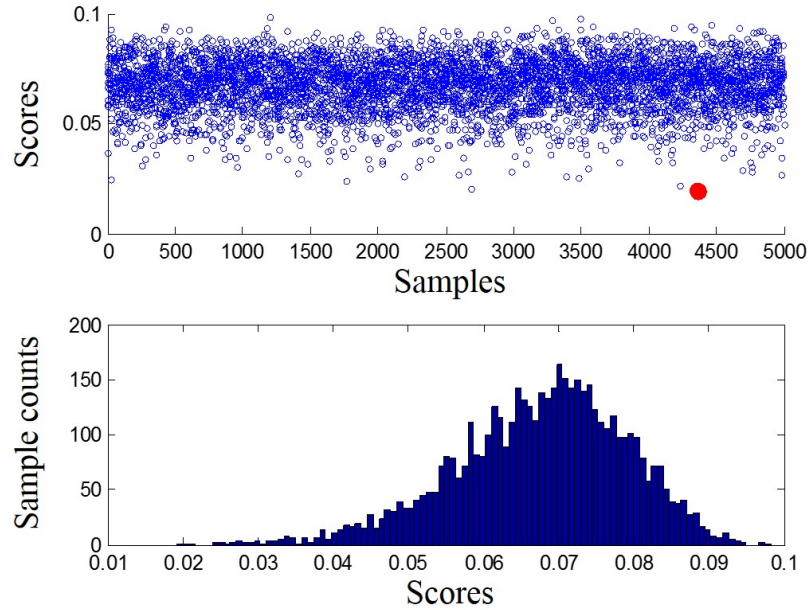


Figure 3.14: Scores of 5,000 equilibrium configurations obtained by repeatedly running the dynamic relaxation. In the upper figure, the empty blue circles represent scores of all of the equilibrium configurations obtained, and the filled red circle represents the lowest score. The bottom figure is a histogram of the scores of all configurations obtained [3].

(Section 3.3.2). Instead, we use a multi-generation learning algorithm where the highest performing samples from the previous generation are saved and new samples are generated from points that are “close” to the highest performing samples. In other words, the equilibrium configurations sampled in a new generation will look “similar” to the best equilibrium configuration from the previous generation. After enough generations, we expect most of our new samples to come from high performing regions. This learning process can be seen as a multi-step Monte Carlo or as an evolutionary algorithm where at every generation only the “winners” are taken from the previous generation.

3.3.3.1 Multi-Generation Monte Carlo

For the samples of the first generation, components of actuator length vector $\mathbf{d}_1$, where the subscript denotes its generation, are all sampled from a uniform distribution of the same range, $[d_{min}, d_{max}]$, where d_{min} and d_{max} represent minimum and maximum lengths of actuators, respectively. Once all samples of the first generation are obtained and evaluated according to the evaluation function defined in Section 3.3.2, the equilibrium configuration with minimum score, $\mathcal{C}_1^*$, as well as the actuator length vector sample that produced this equilibrium, $\mathbf{d}_1^*$, are identified.

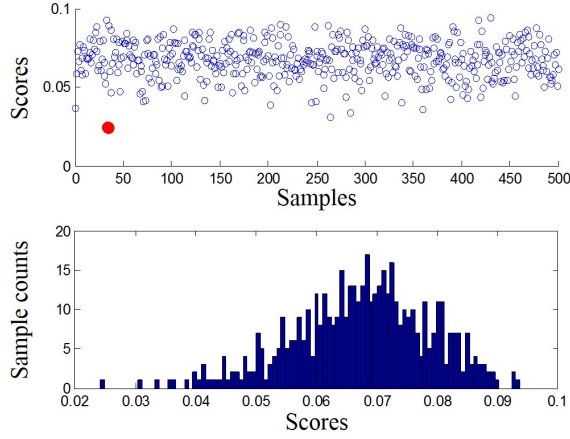
Table 3.2: Physical Parameters of TT-1 and TT-2 [3]

Parameters	Values
Effective rod length (R_0)	0.65 (m)
Cable rest length (l_0)	0.038 (m)
Cable stiffness (k)	1,193 (N/m)
Minimum actuator length (d_{min})	0.2 (m)
Maximum actuator length (d_{max})	0.3 (m)
Constant offset length (δd)	0.01 (m)

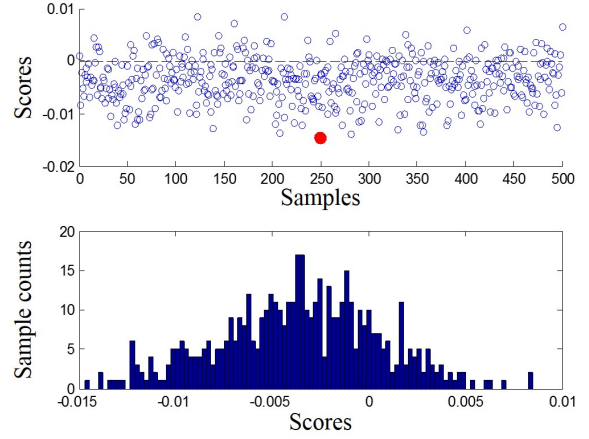
For subsequent generations j ($j = 2, 3, \dots$), actuator length vectors, $\mathbf{d}_j$, are sampled around the best sample of the previous generation, $\mathbf{d}_{j-1}^*$. That is, multiple samples of $\mathbf{d}_j$ are obtained from a uniform distribution of $[\mathbf{d}_{j-1}^* - \delta\mathbf{d}, \mathbf{d}_{j-1}^* + \delta\mathbf{d}]$, where $\delta\mathbf{d} = [\delta d, \dots, \delta d]^T$ is a 24-dimensional constant vector. Once the pre-determined number of samples are obtained and evaluated at generation j , the equilibrium configuration with minimum score, $\mathcal{C}_j^*$, as well as the sampled vector, $\mathbf{d}_j^*$, producing this equilibrium are found. In generation $(j + 1)$, $\mathbf{d}_{j+1}$ are sampled around $\mathbf{d}_j^*$ in a similar manner, and the process is repeated until termination conditions are met or the pre-defined maximum number of generations is reached.

3.3.3.2 Simulation Results

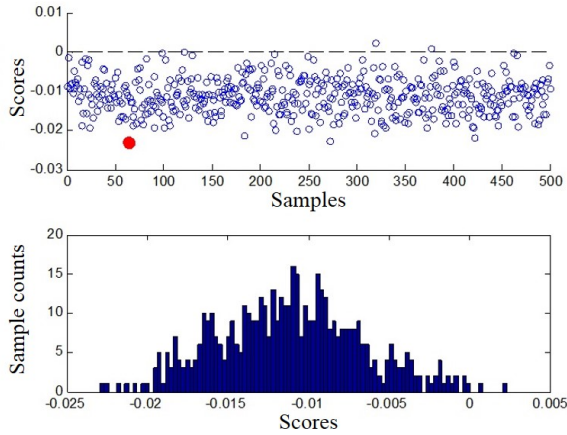
In our simulation, 30 generations were run with each generation containing 500 samples. Score distributions of chosen generations are shown in Figure 3.15. For the early generations, all of the equilibrium configurations found in a single generation had positive scores. However, as generations progressed, the number of samples having negative scores in each generation increased. Moreover, both the minimum and average scores of each generation decreased as the generations evolved (Figure 3.16). The figure shows that the first equilibrium configuration with a negative score was found in generation 5, and up to this generation, a total of 2,500 samples were obtained. This was half the number of samples obtained in Section 3.3.2, but the quality of equilibrium configurations turned out to be much better. The minimum and average scores kept decreasing in later generations and were saturated after generation 20. The best equilibrium configuration of all the samples was found in generation 29 and -0.031 was its score; this most desirable equilibrium configuration placed its GCoM 0.031 m (or 4.77% of the rod length R_0) outside of its base triangle. This configuration is depicted from different orientations in Figure 3.19. The set of actuator lengths resulted in this configuration is taken as our actuation policy and is provided in Table 3.3. The node numbers follow Figure 3.17.



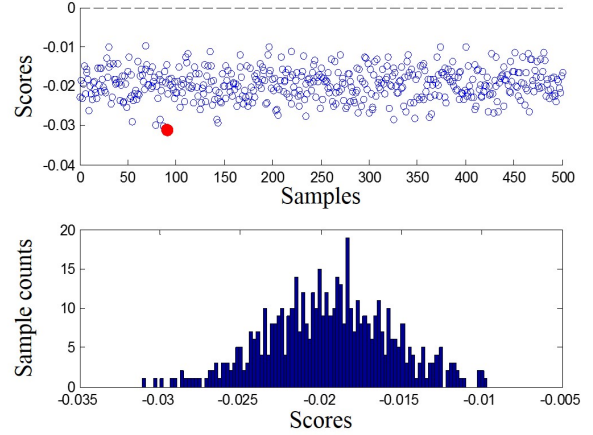
(a) Generation 1



(b) Generation 10



(c) Generation 15



(d) Generation 30

Figure 3.15: Score distributions of chosen generations. In the upper figures, empty blue circles represent scores of all of the equilibrium configurations obtained, and the filled red circles represent the lowest scores. The bottom figures are histograms of the scores of all configurations obtained in each generation. 500 equilibrium configurations were obtained in each generation [3].

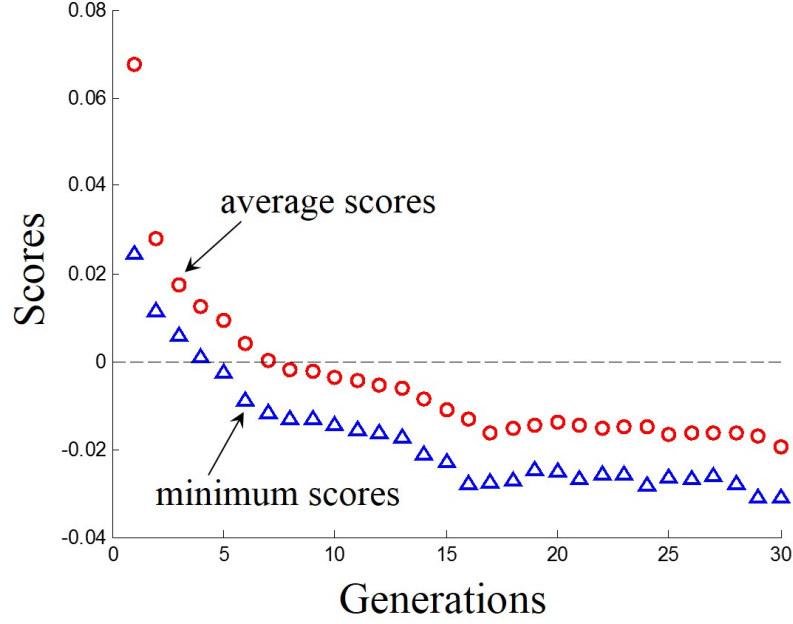


Figure 3.16: Evolution of minimum (blue triangles) and average (red circles) scores over 30 generations [3].

Table 3.3: Actuation Policy ($\mathbf{d}_{29}^*$). Node numbers follow Figure 3.17 [3].

Actuator edge node pair	(1,5)	(1,6)	(1,9)	(1,11)	(2,7)	(2,8)
Controlled actuator length (m)	0.291	0.300	0.296	0.207	0.200	0.201
Actuator edge node pair	(2,9)	(2,11)	(3,5)	(3,6)	(3,10)	(3,12)
Controlled actuator length (m)	0.294	0.298	0.261	0.251	0.297	0.214
Actuator edge node pair	(4,7)	(4,8)	(4,10)	(4,12)	(5,9)	(5,10)
Controlled actuator length (m)	0.251	0.204	0.204	0.291	0.205	0.293
Actuator edge node pair	(6,11)	(6,12)	(7,9)	(7,10)	(8,11)	(8,12)
Controlled actuator length (m)	0.285	0.259	0.294	0.299	0.243	0.203

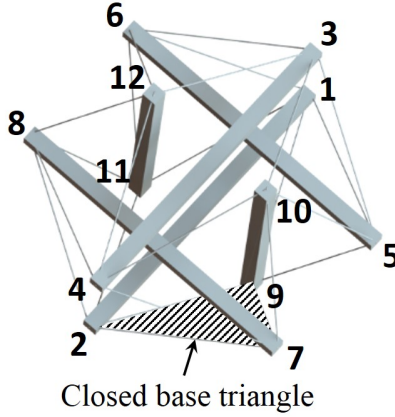


Figure 3.17: Node numbers of the robot structure used in Table 3.3 [3].

3.3.4 Experiments

A set of experiments was performed with TT-2 to test the actuation policy presented in Table 3.3. With this actuation policy, the robot was able to perform either a CO-step or a COC-step, depending on initial actuator lengths. Notice that the method in this section identifies what the most desirable geometry of the robot is in terms of making a step, but it does not provide information on what the intermediate deformations would look like while achieving this geometry from an initial configuration. This is because the dynamic relaxation that the learning algorithm is based on does not necessarily follow the deformation route of an actual physical structure in the process of finding equilibrium configurations. As a result, it is possible that deforming procedures are different for distinct initial configurations, although their final deformed geometries will be the same. For TT-2, when the actuators were starting from a fully extended state with initial lengths of d_{max} , the policy in Table 3.3 resulted in a CO-step of the robot. However, when the actuators were starting from a half extended state⁹ with initial lengths of $\frac{(d_{max}+d_{min})}{2}$, the robot performed a COC-step with the same policy. In the latter case, at the time when the robot made the first CO-step, the robot was still deforming and the width of the landing base triangle was narrow such that GCoM was able to cross over the second base triangle with remaining deformation. Consequently, the following OC-step was automatically performed, and overall a COC-step was made. In contrast, at the moment when the robot performed the first CO-step in the former case, the width of the landing base triangle had already widened and GCoM was not able to cross over the next base triangle. In Figure 3.18, the trajectories of base triangle nodes as well as GCoM on the ground plane were obtained with a Vicon[®] motion tracking system for these two initial actuator length cases.

The distance between GCoM and the rotation axis of a base triangle after completion

⁹Hence the initial cable tension was different from the previous case.

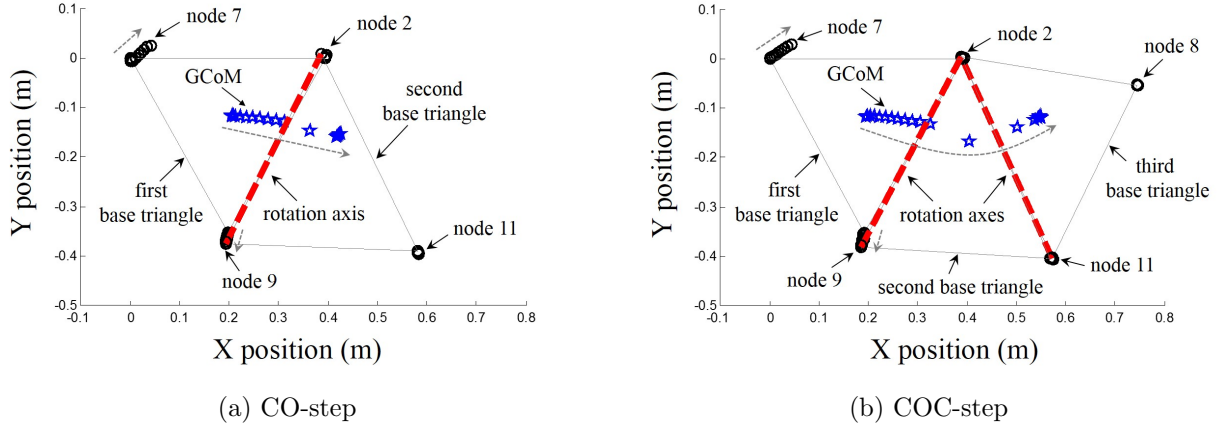


Figure 3.18: Trajectories of base triangle nodes and GCoM when TT-2 is deformed with the actuation policy provided in Table 3.3. Depending on initial actuator lengths (equivalently, initial cable tensions), the robot performs either a CO-step or a COC-step. In the figures, empty black circles, blue stars and bold red dashed lines represent node positions, GCoMs and rotation axes, respectively. Thin dashed lines show moving directions of nodes and GCoM. Thin gray triangles are base triangles that the robot crosses over while performing steps. Node numbers follow Figure 3.17. A video of the robot in action can be found at <http://best.berkeley.edu/best-research/best-berkeley-emergent-space-tensegrities-robotics/> [3].

of the deformation was measured as 0.060 m which is about twice as large as our estimated value of 0.031 m from the simulation. This difference may have caused by the existence of the gravity and the payload on the robot, which was not considered when feasible equilibrium configurations were found in the simulations. When the payload was not placed at the center of mass of the robot, it yielded asymmetry of the structure, causing the robot to favor steps in certain directions. Therefore, in order to properly test the simulated policy on the physical robot, an effort was made to reduce imbalance of the robot structure due to the payload, and the payload was placed as close to the center of mass of the robot as possible. Low resolution linear actuators also contributed to the error as they were not able to deform the robot structure to precisely match the simulated geometry. The resultant deformed structure corresponding to the actuation policy, obtained from both the simulation and physical robot, is presented in Figure 3.19 in different views. Furthermore, all of the cable lengths of the deformed structure were measured from both the simulation and hardware robot, and their comparison is given in Table 3.4. Despite the model inaccuracies, the lengths match closely in both cases, with a maximum of 5.25% error.

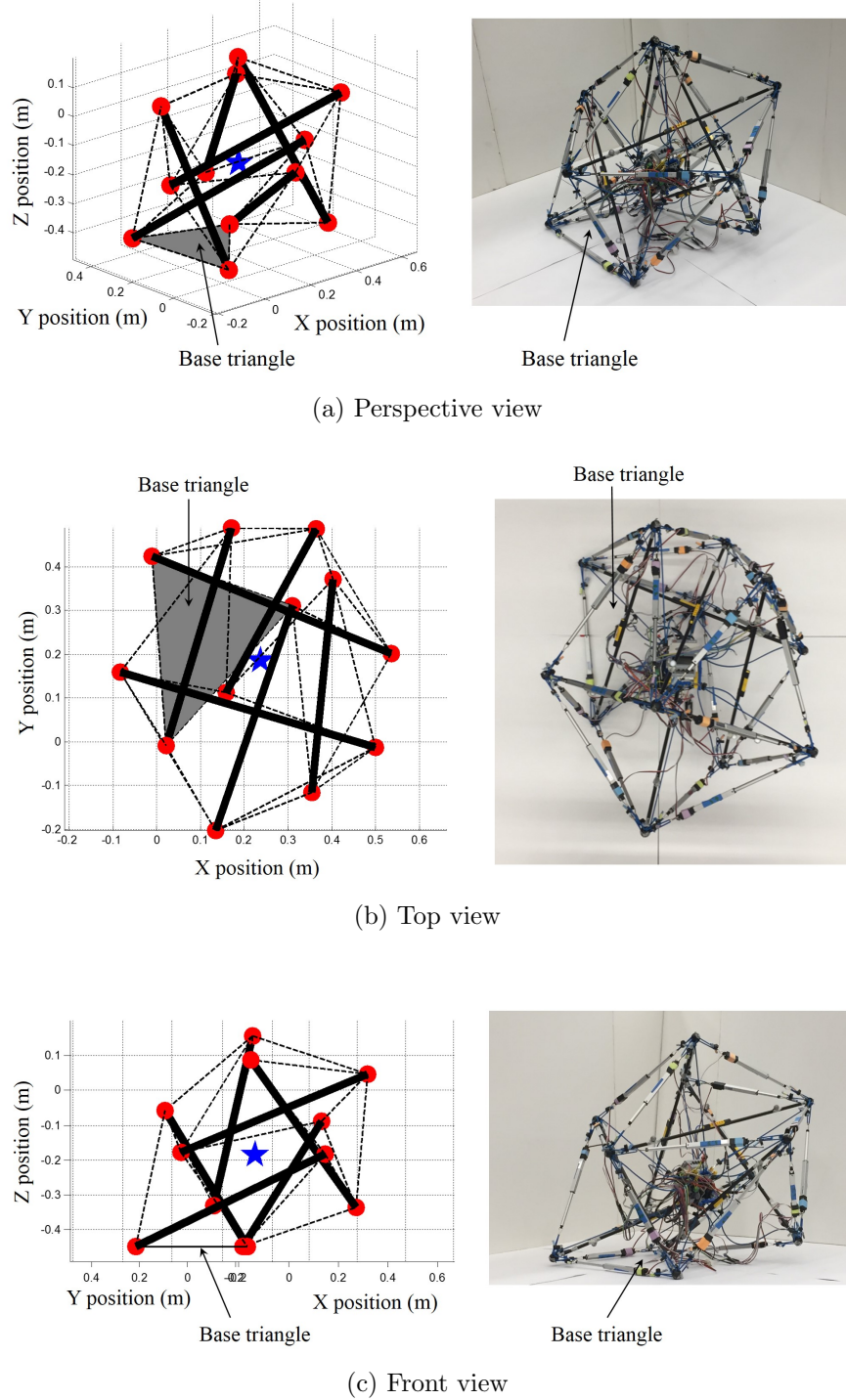


Figure 3.19: The equilibrium configuration with the minimum score of all generations. The figures on the left side are simulation results and the figures on the right side show deformation of our physical robot, TT-2. In the simulation figures, thick black lines, thin dashed lines and red circles represent rods, cables and nodes of the robot structure. Blue stars and gray triangles are centers of mass and base triangles of the robot, respectively [3].

Table 3.4: Edge length comparison between simulation and hardware experiment [3]

Edge node pairs	Simulated lengths (m)	Hardware lengths (m)	Error (%)
(1,5)	0.431	0.432	0.232
(1,6)	0.433	0.446	3.002
(1,9)	0.425	0.421	0.941
(1,11)	0.342	0.345	0.877
(2,7)	0.340	0.335	1.471
(2,8)	0.366	0.353	3.552
(2,9)	0.432	0.441	2.083
(2,11)	0.443	0.449	1.354
(3,5)	0.417	0.404	3.118
(3,6)	0.381	0.401	5.249
(3,10)	0.434	0.433	0.230
(3,12)	0.363	0.363	0.000
(4,7)	0.398	0.401	0.754
(4,8)	0.364	0.354	2.747
(4,10)	0.335	0.348	3.881
(4,12)	0.436	0.434	0.459
(5,9)	0.336	0.342	1.786
(5,10)	0.434	0.443	2.074
(6,11)	0.423	0.418	1.182
(6,12)	0.399	0.403	1.003
(7,9)	0.433	0.439	1.386
(7,10)	0.437	0.437	0.000
(8,11)	0.397	0.392	1.259
(8,12)	0.346	0.352	1.734

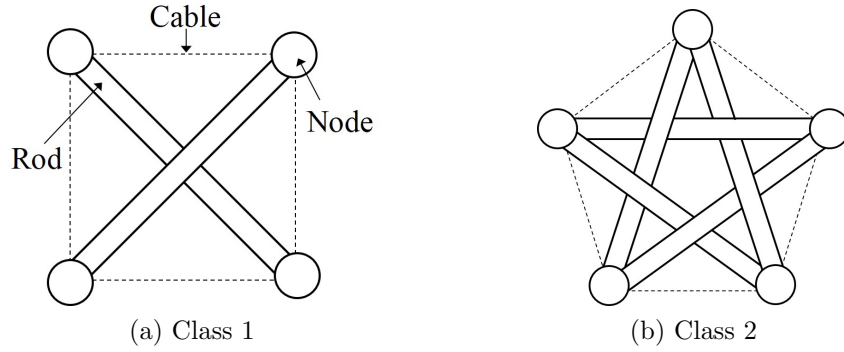


Figure 3.20: Examples of class 1 and class 2 tensegrity structures. The structures are taken from [12].

3.4 Actuation Policy Development for General Spherical Tensegrity Robots with Multi-Generation Monte Carlo Method

The application of the learning algorithm we developed in the previous section is not limited to six-rod tensegrity robots only. The method can be applied to more general spherical tensegrity robots if they are based on a *class 1* tensegrity structure. The definition of a class for tensegrity structures is given in [103] as follows:

A tensegrity configuration that has no contacts between its rigid bodies is a *class 1* tensegrity system, and a tensegrity system with as many as k rigid bodies in contact is a *class k* tensegrity system.

As depicted in Figure 1.5 of [103], the contact in the above definition is stronger than simple touch of two or more rods, and it refers to a rigid connection between the rods such that the contact points of the rods are constrained to move together (Figure 3.20).

For example, rods of a six-rod tensegrity structure do not touch each other, so this structure is classified as a class 1 tensegrity. A rhombicuboctahedron-like twelve-rod tensegrity structure that T12-R is based on is also class 1. There exists a number of spherical tensegrity structures that fall under this class 1 category, and more examples can be found in [12].

The overall framework of actuation policy development for general class 1 tensegrity structures is the same as before. We run the dynamic relaxation to find equilibrium configurations of structures for a given set of edge lengths and a multi-generation Monte Carlo based learning algorithm is deployed on top of the dynamic relaxation. However, nodal force expressions developed in Section 3.3.1.1 may need to be modified before running the dynamic relaxation to properly account for hardware design of robots under consideration. We discuss this further by taking actuation policy development for T12-R as an example.

3.4.1 Dynamic Relaxation on Class 1 Tensegrity Structures

For any class 1 tensegrity structure, a rod can be represented by two point masses at the rod ends with a constraint that the masses are a constant distance away from each other, as described in Section 3.3.1.1. Because rods do not interact in a class 1 tensegrity structure, the net force acting on each of the nodal masses is a sum of cable forces and a constraint force, see (3.7)¹⁰. Furthermore, the cable force equations – (3.8) and (3.9) – as well as the constraint force equations – (3.11) and (3.12) – are also valid for general class 1 tensegrity structures¹¹. As a result, the dynamic relaxation can be applied to general class 1 tensegrity structures in the same way as we did in Section 3.3.1.1 to find equilibrium configurations of the structures. The multi-generation Monte Carlo based learning algorithm presented in Section 3.3.3 can be run on top of the dynamic relaxation to find a good set of equilibrium configurations for steps. We illustrate the above procedure by developing actuation policies for T12-R.

3.4.1.1 Modification of Nodal Cable Force Equation

Members of an edge of T12-R and their connections are different from TT-1 and TT-2, as shown in Figure 3.21. An edge of T12-R has a spring connected to a string that is rigid and not stretchable. The length of the string, hence the tension on the edge, is controlled by a motor that winds the string with a spool. If the spring is linear and has stiffness of k , rest length of l_0 and stretched length of l , and if the length of the string is denoted by d , then the tension on this edge is computed by (3.9) with l defined as

$$l(t) = \|\mathbf{r}_j(t) - \mathbf{r}_i(t)\| - d. \quad (3.15)$$

In this equation, d is assumed to be a constant because, when we deploy a Monte Carlo based algorithm, we sample a constant vector of string lengths and use it as an input to the dynamic relaxation. The vector of string lengths plays the exact same role as the vector of actuator lengths defined in Section 3.3.2.1. We note that edge member connections of TT-3 are the same as T12-R, and (3.15) should be used instead of (3.10) when computing cable forces of TT-3.

3.4.1.2 Modification of Nodal Constraint Force Equation

While the constraint force equation (3.12) is still applicable to rods of general class 1 tensegrity structures, it suffers from singularity when computing constraint forces for vertical rods oriented in $\mathbf{E}_3$ direction because the angle θ is ill-defined for these rods. This singularity issue

¹⁰For class k tensegrity structures, there is an additional force arising from interaction of contacting rods and (3.7) is no more valid. The interested reader is referred to [42] for more details.

¹¹We assume tensile cables follow Hooke's law in this work. If a nonlinear cable is assumed, (3.9) should be modified. Moreover, (3.10) needs to be modified if the edge connection of a robot is not the same as Figure 3.11. We will demonstrate this when we develop actuation policies for T12-R.

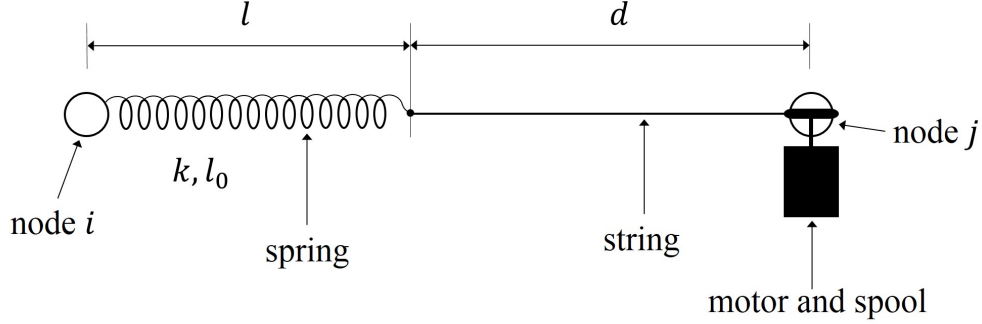


Figure 3.21: A diagram of edge connections of T12-R and TT-3. A spring is connected to a rigid string that is spooled by a motor. See Figure 2.11 for an actual hardware setup.

did not arise with a six-rod tensegrity structure because it did not have any rods oriented in the singular direction. However, in a rhombicuboctahedron-like twelve-rod tensegrity structure, there are four rods that are oriented in the singular direction. To avoid this singularity problem and improve the stability of numerical computation, we develop an alternate form of (3.12).

To begin with, we notice that the first three terms inside of the bracket of (3.12) is a dot product of the acceleration of node i , (3.14), and the unit vector

$$\mathbf{e}_R(t) = \sin(\phi(t)) \cos(\theta(t)) \mathbf{E}_1 + \sin(\phi(t)) \sin(\theta(t)) \mathbf{E}_2 + \cos(\phi(t)) \mathbf{E}_3. \quad (3.16)$$

See Figure 3.12 for notations. Since the position difference between the two end nodes is

$$\mathbf{r}_k(t) - \mathbf{r}_i(t) = R_0 \mathbf{e}_R(t), \quad (3.17)$$

the difference in velocity is

$$\begin{aligned} \mathbf{v}_k(t) - \mathbf{v}_i(t) &= R_0 \dot{\mathbf{e}}_R(t) \\ &= R_0 \left[\left(\dot{\phi} \cos(\theta) \cos(\phi) - \dot{\theta} \sin(\theta) \sin(\phi) \right) \mathbf{E}_1 \right. \\ &\quad \left. + \left(\dot{\phi} \sin(\theta) \cos(\phi) + \dot{\theta} \cos(\theta) \sin(\phi) \right) \mathbf{E}_2 - \dot{\phi} \sin(\phi) \mathbf{E}_3 \right]. \end{aligned} \quad (3.18)$$

The angular coordinates $\phi(t)$ and $\theta(t)$ and their time derivatives are functions of time, but we omitted the time dependence notations in the above equation for the sake of better readability. The dot product of the velocity difference into itself is

$$(\mathbf{v}_k(t) - \mathbf{v}_i(t)) \cdot (\mathbf{v}_k(t) - \mathbf{v}_i(t)) = R_0^2 \left(\dot{\phi}^2 - \dot{\theta}^2 \sin^2 \phi \right), \quad (3.19)$$

which shows up in the last two terms inside of the bracket of (3.12). By using the above results, we can rewrite (3.12) in another form as follows.

$$F_{ik}^r(t) = (m_k \mathbf{a}_i(t) - \mathbf{F}_k^s) \cdot \mathbf{e}_R - \frac{1}{R_0} (\mathbf{v}_k(t) - \mathbf{v}_i(t)) \cdot (\mathbf{v}_k(t) - \mathbf{v}_i(t)). \quad (3.20)$$

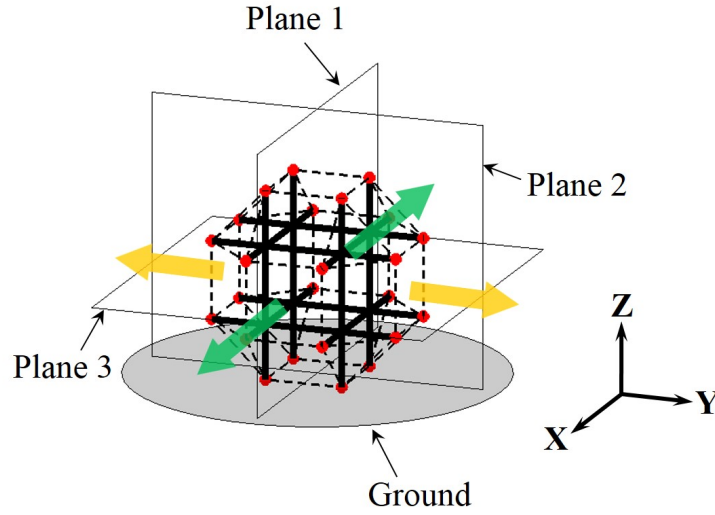


Figure 3.22: The outer shape of T12-R is similar to a rhombicuboctahedron that is symmetric about three mutually-orthogonal planes. In this work, we choose Plane 1 to be the sagittal plane of T12-R and develop actuation policies that make T12-R roll in X-direction. We further assume that the actuation policies are symmetric about Plane 1. That is, a pair of cables that are symmetric about this plane always have the same length in our actuation policies.

The advantage of (3.20) over (3.12) is that (3.20) is based on vector operations and does not include angular coordinates. As a result, it prevents singularity from happening and ensures numerical stability. With the updated nodal force equations, the dynamic relaxation can be run on any class 1 tensegrity structure.

3.4.2 Multi-Generation Monte Carlo Based Learning for T12-R

3.4.2.1 Sampling

The outer shape of T12-R is similar to a rhombicuboctahedron that is symmetric about three mutually-orthogonal planes (Figure 3.22). When T12-R is standing as shown in Figure 3.22, two of these planes are also perpendicular to the ground and T12-R can roll in a direction that is contained in either plane. We choose one of the two planes and define it as a sagittal plane of T12-R. We develop actuation policies that are symmetric about this sagittal plane, that is, a pair of cables that are symmetric about this plane are set to the same length. As a result, the deformation of T12-R is symmetric about the sagittal plane and T12-R rolls in a direction parallel to the plane. Using the structural symmetry, the actuation policies developed can easily be generalized to alternate choices of sagittal plane.

The goal of the multi-generation Monte Carlo based learning algorithm is the same as

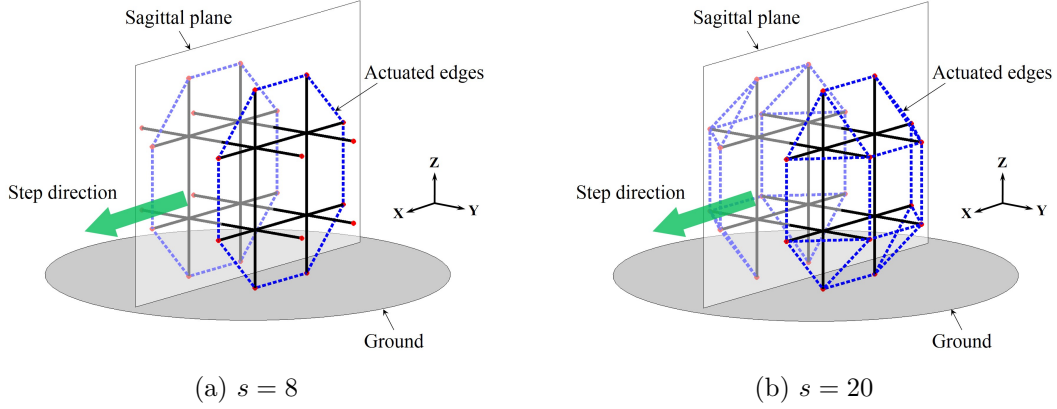


Figure 3.23: We consider two different cases of s . When $s = 8$, only the lengths of outer edge strings are controlled with motors. When $s = 20$, all of the edges, except the ones that are perpendicular to the sagittal plane, are actuated. The edges that are not actuated do not contribute to the shifting of CoM on the sagittal plane. Due to the symmetry of actuation policies about the sagittal plane, the actual number of edges that are actuated is $2s$. Only the actuated edges are shown with blue dotted lines in the figures.

before; we want to find an equilibrium configuration of T12-R that pushes its GCoM away from its supporting polygon as much as possible such that the equilibrium enables a robust step of T12-R. The input to the dynamic relaxation is a vector of edge string lengths, $\mathbf{d} = [d_1, \dots, d_s]^T \in \mathbb{R}^s$, where s is one half of the number of actuators used for deformation. We consider two cases of s as depicted in Figure 3.23: 1) $s = 8$, and 2) $s = 20$. The motivation for the first choice of s comes from our intuition that the amount of CoM shifting is largely dependent on the outer shape deformation, as one may see by projecting T12-R onto its sagittal plane. The second choice of s corresponds to a fully actuated robot. The eight actuators that are missing do not contribute to the shifting of CoM on the sagittal plane (Figure 3.23b).

3.4.2.2 Evaluation

Recall from Section 2.3 that there are two types of base polygons that T12-R stands on – perpendicular and diagonal rectangles. We do not consider the case where T12-R is standing on a triangle because this pose is not part of straight rolling depicted in Figure 2.8. T12-R can perform two different types of steps.

- *PD-step* leads the robot from a perpendicular base rectangle to an adjacent diagonal base rectangle.
- *DP-step* leads the robot from a diagonal base rectangle to an adjacent perpendicular base rectangle.

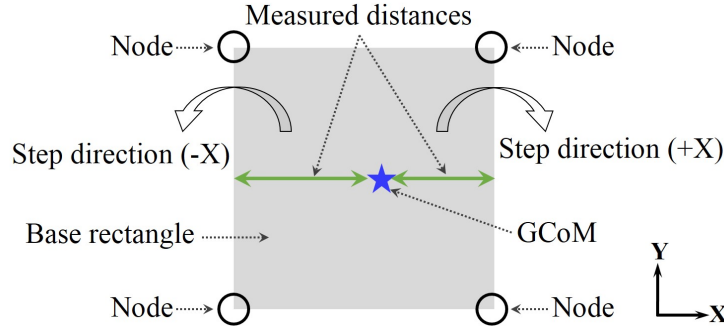


Figure 3.24: The metric used for evaluation of deformed T12-R. Assuming that the robot intends to make a step in X-direction, we measure distances between GCoM and two edges of a base rectangle that are perpendicular to the desired step direction.

For each choice of s , we run the learning algorithm twice – one for a PD-step and the other one for DP-step – and develop two actuation policies separately. To evaluate equilibrium configurations obtained by sampling, we use a similar metric as before (Figure 3.13), but this time the base polygon is a rectangle and we only measure distances between GCoM and two parallel edges that are perpendicular to the assumed moving direction (Figure 3.24) because the other two edges are irrelevant to steps from that base polygon. Obviously, this metric is applicable to both types of rectangles.

3.4.2.3 Simulation Results

The same multi-generation Monte Carlo based learning algorithm described in Section 3.3.3.1 is deployed to find actuation policies for PD- and DP-steps of T12-R. We used physical parameters measured from hardware robot T12-R, which are summarized in Table 2.2, for our simulation. In the first set of simulations, only 16 outer edge strings are actuated and s is chosen to be eight¹². Total of 30 generations are run with each generation containing 100 samples. Initially, the structure is assumed to be perfectly symmetric and all of the string lengths are set to 12 cm. We further assume that motors of T12-R can spool in (or out) the strings to the minimum (or maximum) length of $d_{min} = 8$ cm (or $d_{max} = 16$ cm). These values are chosen because we expect the motors of T12-R to be operated safely within this range without significant loading on themselves. For the samples obtained in the first generation, eight dimensional string length vectors $\mathbf{d}_1 = [d_1, \dots, d_8]^T$ are sampled from a uniform distribution of $[d_{min}, d_{max}]$. In other words, $\delta d = 4$ cm for the first generation. However, δd is set to 1 cm for all later generations such that the sampling is done within a closer neighborhood of the best sample of the previous generation.

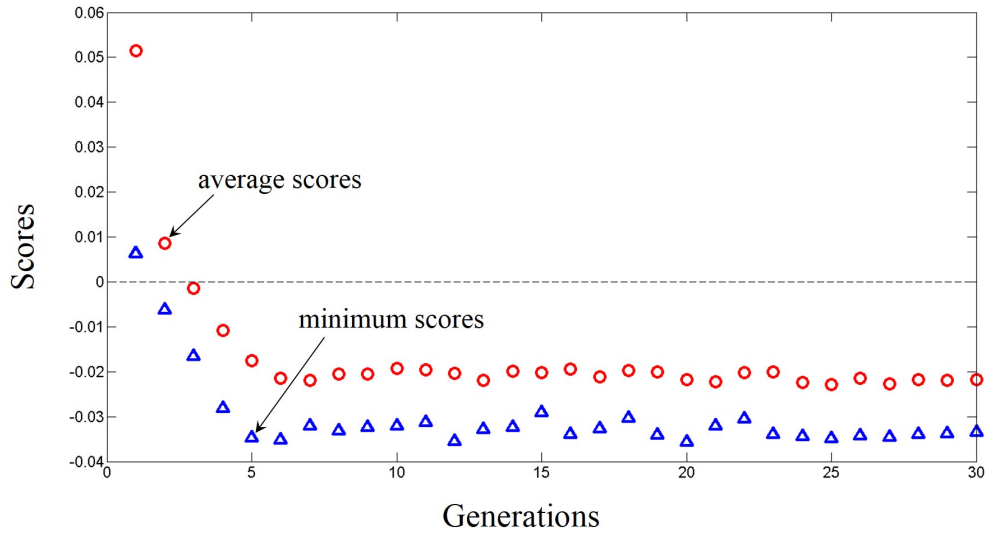
¹²Recall that, because we are exploiting symmetry of the structure about one chosen sagittal plane, the number of actuated strings is double the dimension of the string length vector sampled, s .

We run the learning algorithm twice to obtain two actuation policies, one for a PD-step and the other one for a DP-step. The evolution of average and minimum scores over generations for both cases is presented in Figure 3.25. It can be easily seen that the scores saturate after generation 6 for both cases. However, while the PD-step simulation successfully finds samples with negative scores, all of the samples obtained in the DP-step simulation have positive scores, which implies that no successful policy can be found for a DP-step. From this result, we notice that DP-steps are more difficult to make than PD-steps and more actuation effort would be required for DP-steps. This is not a surprising result; in fact, this agrees with what we observed from T12-R hardware. When we try to tip the robot over by manually pushing it, greater force is needed to make the tipping happen when starting from a diagonal rectangle. Thus, we conclude that diagonal base rectangles provide more static balance than perpendicular base rectangles. The best shapes found from each simulation is shown in Figures 3.26 and 3.27.

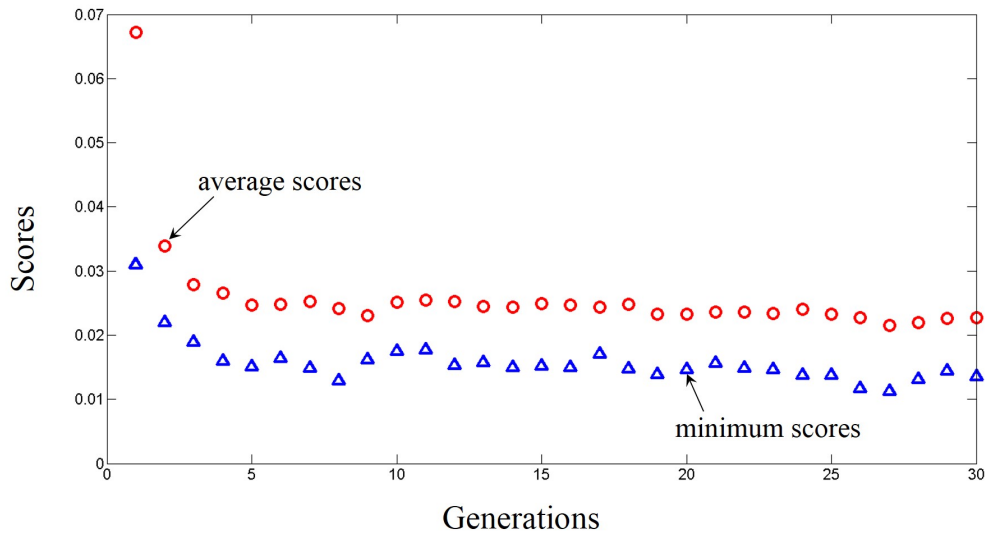
There are at least two improvements that could be made to find a successful DP-step policy: 1) use more actuators per step, and 2) increase the degree of structural deformation by widening the range of actuated string lengths. These improvements would lead to an increase in energy consumption of the system either because more actuators are powered per step or because motors with greater torque or power would be needed to increase the maximum spool length of strings due to the increased tension on edges.

Among the two options, we first investigate the case with an increased number of actuators per step. Specifically, we set $s = 20$ and develop a policy that controls lengths of 40 edge strings. Recall from Section 3.4.2.1 that this corresponds to a fully actuated robot. The evolution of average and minimum scores over generations for PD- and DP-step simulations is presented in Figure 3.28. All other parameters except s are kept the same as before. We notice that, because of the increased actuation effort, we are now able to find DP-step actuation policies with negative scores. Moreover, PD-step actuation policies with negative scores are found even in the first generation. Comparison of the minimum scores of PD-step and DP-step policies says that PD-steps are more reliable and easier to perform than DP-steps. The best equilibrium configurations found from each simulation is presented in Figures 3.29 and 3.30.

As an alternate way to find successful DP-step actuation policies, the range of actuated string lengths could be increased, which will allow greater structural deformation. To this end, we increase δd from 4 cm to 8 cm for the first generation and from 1 cm to 2 cm for the subsequent generations. Total of 30 generations are run with each generation containing 100 samples. Initially, all of the string lengths are set to 12 cm as before, but the new choice of δd alters d_{min} and d_{max} to be 4 cm and 20 cm, respectively. Only outer edge strings of T12-R are assumed to be actuated, and $s = 8$. The evolution of average and minimum scores over generations for this DP-step simulation is provided in Figure 3.31. By doubling the actuation range, we now find successful actuation policies for DP-steps even in the first generation. The best equilibrium configuration found in this simulation is shown in Figure 3.32.



(a) PD-step



(b) DP-step

Figure 3.25: Evolution of minimum (blue triangles) and average (red circles) scores of PD-step and DP-step simulations over 30 generations with 100 samples obtained per generation. It is assumed that only outer edge strings are actuated in these simulations and $s = 8$. In both cases, the scores saturate at generation 6. While PD-step simulation finds actuation policies with negative scores, all of the samples obtained in DP-step simulation have positive scores, which implies that no successful policy could be found for DP-steps.

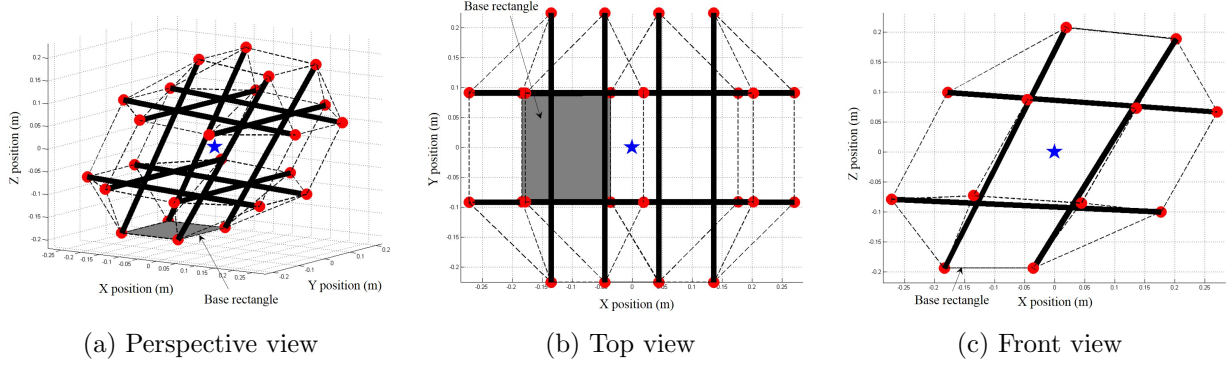


Figure 3.26: The best equilibrium configuration of T12-R obtained from PD-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a PD-step with this actuation policy.

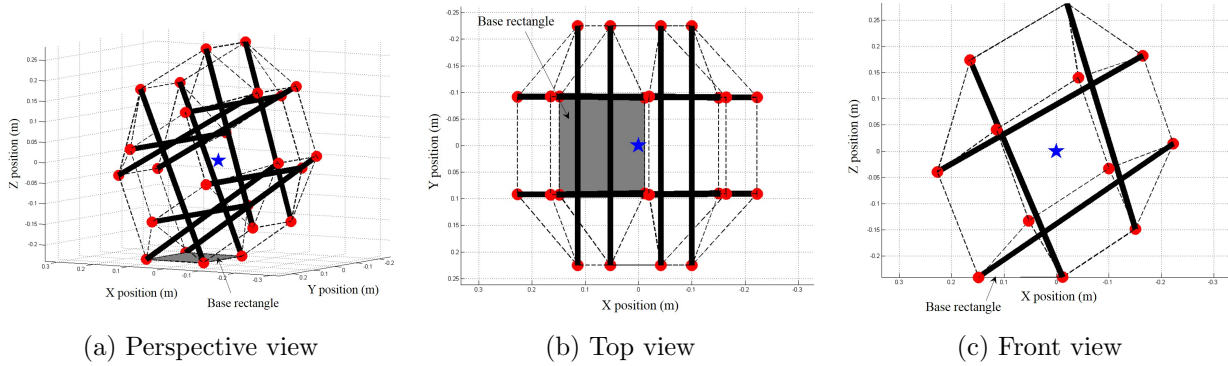
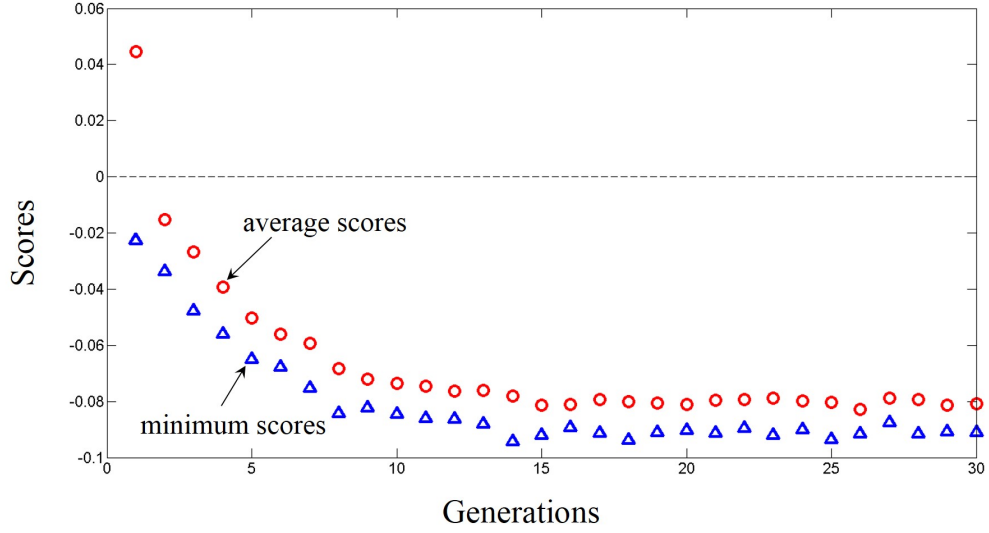
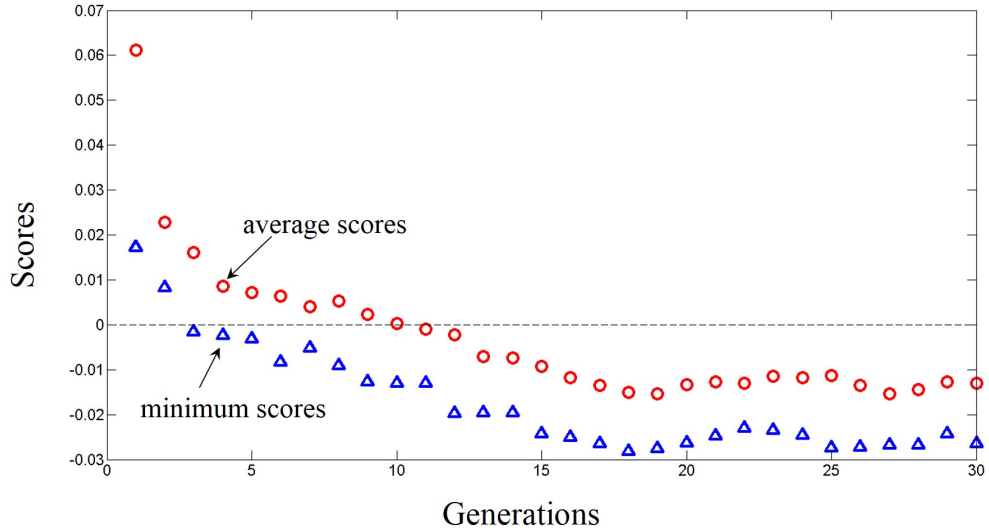


Figure 3.27: The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. Because GCoM is located inside of the base rectangle in this configuration, T12-R is not expected to perform a DP-step with this actuation policy.



(a) PD-step



(b) DP-step

Figure 3.28: Evolution of minimum (blue triangles) and average (red circles) scores of PD-step and DP-step simulations over 30 generations with 100 samples obtained per generation. It is assumed that the robot is fully actuated in these simulations and $s = 20$. PD-step simulation finds actuation policies with negative scores even in the first generation, which is earlier than the case of $s = 8$, due to the increased actuation effort. Moreover, DP-step simulation also finds actuation policies with negative scores. The absolute values of minimum scores of PD-step simulation are greater than those of DP-step simulation, which implies that PD-steps are more reliable and easier to perform than DP-steps.

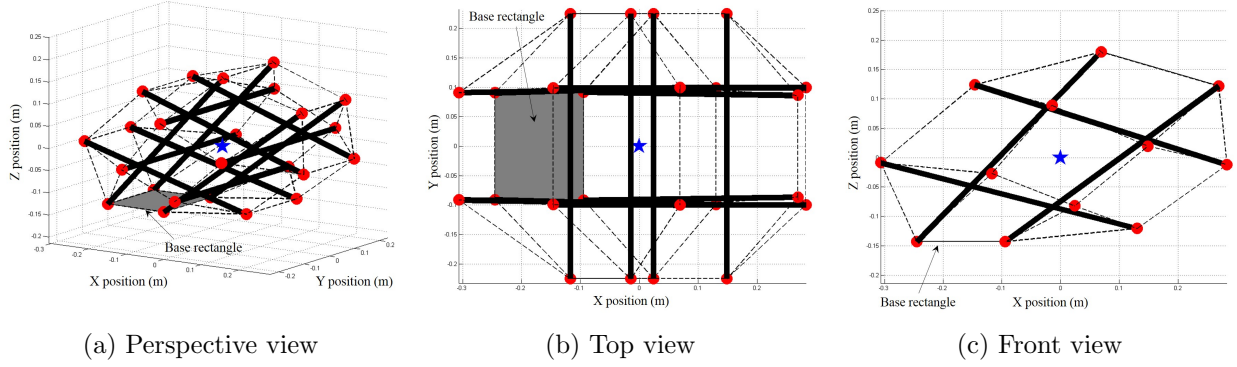


Figure 3.29: The best equilibrium configuration of T12-R obtained from PD-step policy simulation. Total of 30 generations are run with 100 samples per generation. The robot is assumed to be fully actuated, i.e., $s = 20$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a PD-step with this actuation policy.

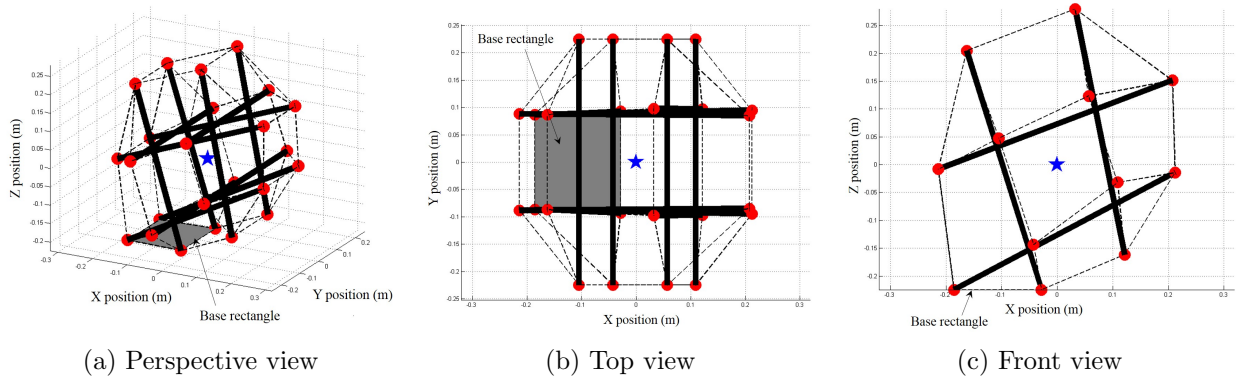
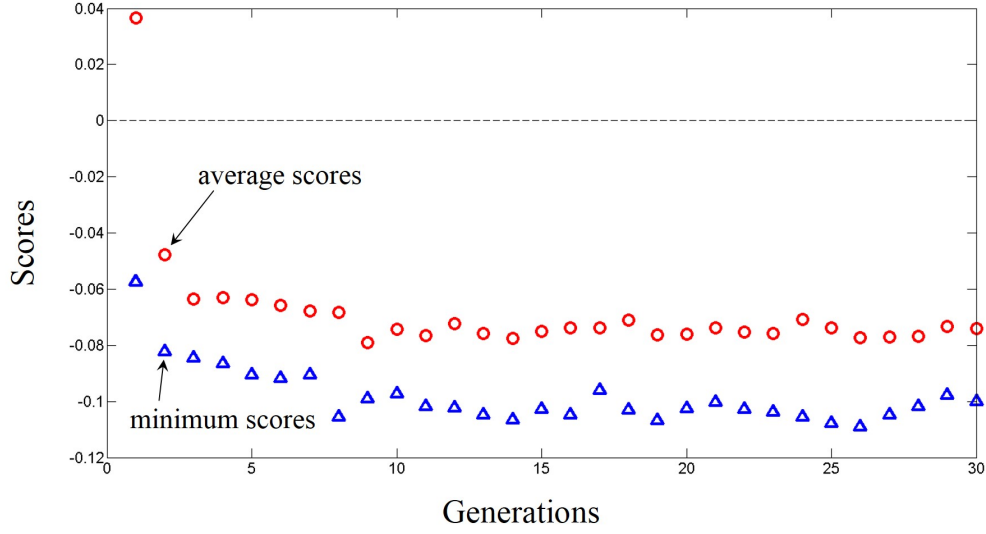


Figure 3.30: The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. The robot is assumed to be fully actuated, i.e., $s = 20$. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a DP-step with this actuation policy.



(a) DP-step

Figure 3.31: Evolution of minimum (blue triangles) and average (red circles) scores of DP-step simulation over 30 generations with 100 samples obtained per generation. It is assumed that only outer edge strings of T12-R are actuated in these simulations and $s = 8$. δd is set to 8 cm, and thus $d_{min} = 4$ cm and $d_{max} = 20$ cm. The simulation finds DP-step actuation policies with negative scores even in the first generation.

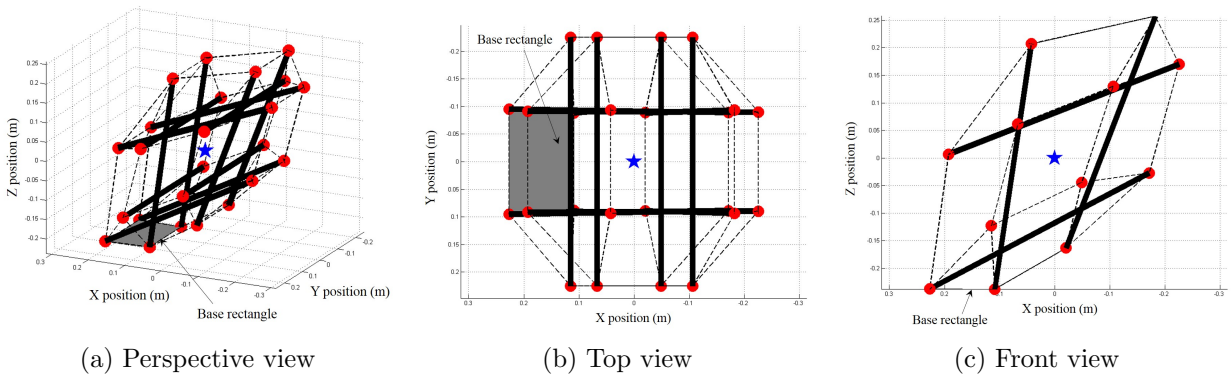


Figure 3.32: The best equilibrium configuration of T12-R obtained from DP-step policy simulation. Total of 30 generations are run with 100 samples per generation. Only outer edge strings are actuated, i.e., $s = 8$. δd is set to 8 cm, and thus $d_{min} = 4$ cm and $d_{max} = 20$ cm. Because GCoM is located outside of the base rectangle in this configuration, T12-R is expected to perform a DP-step with this actuation policy.

3.5 Conclusions

Two methods have been introduced in this chapter that are used to develop actuation policies enabling our hardware robots to perform different types of steps. The first method is based on a greedy search algorithm equipped with a heuristic function defined as the distance between GCoM and an edge of a supporting polygon serving as a rotation axis of a step. We first applied this method in NTRT simulation to find actuation policies for six-rod tensegrity robots and tested the obtained policies on our hardware robots TT-1 and TT-2. Because the heuristic function only tells us whether a policy will enable a step or not, and it does not tell us how much reliable the step is, our hardware experiments showed that TT-1 and TT-2 sometimes fail to perform a step with the obtained policies. Hence simple improvements are made to the policies to enhance the quality of steps, and with these improved policies, both TT-1 and TT-2 make successful steps from any of their base triangles. We also presented motion data of TT-1 measured with a motion tracking system and compared the energy efficiency of different types of steps.

To systematically find actuation policies for more reliable and robust steps, we proposed the second method that combines a dynamic relaxation technique with a multi-generation Monte Carlo sampling method. We presented a detailed procedure of the dynamic relaxation that finds equilibrium configurations of tensegrity structures when a set of edge member lengths is given, which we assume to be our control variable in this work. We run a multi-generation Monte Carlo, with our samples being the equilibrium configurations obtained from the dynamic relaxation, to efficiently find actuation policies that would result in reliable steps. The evaluation function used in this approach handles negative heuristic values that were not trackable with the search method. Indeed, the goal of the multi-generation Monte Carlo is to find an equilibrium configuration that minimizes the score to the largest negative value. Using this method, we developed a CO-step policy for TT-2 and tested it on our hardware robot.

We also demonstrated that the latter method is applicable to more general spherical tensegrity robots, and few modifications have been made to the method for this purpose. As an example, we developed several actuation policies for PD- and DP-steps of T12-R with this method, by varying simulation and hardware parameters until successful policies are found. We note that the set of successful parameters could serve as a guideline to the hardware design of T12-R. For instance, the total number and operation range of actuators are included in the varying parameter set, and we were able to figure out proper numbers for them that would result in successful steps of T12-R. Because T12-R is still under development at the time when this dissertation is written, hardware experiments of the developed policies remain as our future work.

Chapter 4

Thruster-Based Hopping Tensegrity Robots

4.1 Introduction

Realizing future goals in space exploration missions will require a radical change from existing architectures to allow for widespread exploration at low cost and high-reliability. This chapter presents one possible concept of low-cost surface probes that are based on tensegrity robots, which can deploy small payloads with speed, robustness, and accuracy. Achieving these goals at a low cost is difficult for classical surface rovers, which are expensive to implement and require slow navigation speeds for safe operation on a planetary surface [104], [105]. Emerging technologies, such as hopping robots, offer quick mobility but often suffer in terms of accurate positioning or payload protection [106].

These limitations may be overcome with recent advances in spherical tensegrity robots [5]. The key idea enabling the breakthrough is to add a thruster system to a tensegrity robot and allow it to hop long distances quickly without restricted by ground conditions. Tensegrity structures, composed of axially-loaded compression elements suspended in a network of tensional elements, efficiently distribute internal forces and are capable of surviving significant impact shocks [8], [12], [36], [59], [60], [76], [107]–[111]. There are no lever arms in a tensegrity structure, so forces do not magnify into joints or other common points of failure. Rather, externally applied forces distribute through the structure via multiple load paths, creating a system-level robustness and tolerance to impacts and forces applied from any direction. For this reason, when a tensegrity structure undergoes a collision event, it redistributes high impulse to its members (i.e., rods and cables) internally in a passive manner thereby preventing itself from breaking and protecting delicate payloads suspended at the center. This system-level robustness is a unique feature of tensegrity robots, which many rigid robots lack, and makes them ideally suited for operation in dynamic environments where contact forces cannot always be predicted. In [82], [112], researchers have run experiments and observed that tensegrity structures can survive from significant impacts

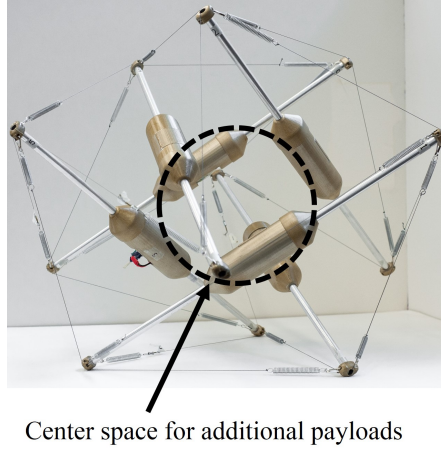


Figure 4.1: A six-bar tensegrity robot, TT-3, developed at the Berkeley Emergent Space Tensegrities (BEST) laboratory. The robot has distributed controllers on the rods, clearing the center space of the structure for additional payloads, e.g., sensors or a thruster system [5].

when they were dropped from various heights. The robustness is especially beneficial for the hopping tensegrity robot to be presented in this chapter, because it repeatedly experiences large unknown impulses at landing after hops.

Light weight is another critical feature of tensegrity structures that enables the development of hopping tensegrity robots. Being lightweight, tensegrity robots can hop with relatively small forces or *thrusts*. This allows their thruster systems to be less heavy and bulky, thereby reducing the total weight as well as the overall size of the tensegrity robots. This is in contrast with the current exploration rovers that are ponderous and fragile enough not to consider hopping as their option for mobility.

In summary, tensegrity robots have great advantages that make themselves promising for space exploration missions at lower cost and higher reliability than the current space exploration rovers.

4.1.1 Contribution of This Chapter

In this chapter, a spherical tensegrity robot, which has dramatically increased mobility with the help of a simple gas thruster located at its center space, is studied. The thruster will allow the robot to quickly cover long distances over difficult terrain using a series of hops while the tensegrity structure's inherent ability to distribute internal forces protects the payload. To realize the thruster-based hopping concept, two technology areas have been studied: 1) mobility, allowing for safe and accurate positioning of the robot using a combination of cold-gas thruster-based hopping and precision rolling, and 2) autonomy, implementing navigation algorithms that allow for efficient hopping and precise rolling, even in the event of unexpected

obstacles or failures.

The primary goals of the tensegrity robot presented in this chapter are to travel by hopping and rolling over a 1 km distance on the Moon’s surface and to safely deliver a 1 kg payload. The total weight of the robot is limited to 10 kg, including the payload, in order to reduce the mission cost. These capabilities were specified by NASA for our research program and are intended to facilitate low-cost scouting missions launched as a secondary payload on a stationary lander. However, the concept is very general and can be adapted to a large number of other missions.

The hardware robot of interest in this chapter is TT-3 shown in Figure 4.1. As presented in Chapter 2, this robot performs rolling steps by controlling cable lengths with onboard actuators to repeatedly deform, or shape-shift, its body. This cable-driven rolling design enables precise positioning of the robot. Moreover, unlike TT-1 and TT-2, TT-3 has distributed controllers located in capsules at the centers of the rods, hence clearing the center space of the structure for additional payloads¹. By adding a gas thruster at the center space, we can add the capability to quickly traverse long distances or navigate large obstacles through a series of hops. In addition, since tensegrities are lightweight (TT-3 has a mass of only 2 kg), thruster-based tensegrity robots require only a small amount of propellant for thrust when hopping. As a result, the thruster system can be of low volume and mass, and simple cold gas propellants with low specific impulse are sufficient to provide thrust for the robot to deliver a 1 kg payload over a 1 km distance on the Moon.

The rest of the chapter is organized as follows. In Section 4.2, we present a thermodynamic analysis on the choice of a cold-gas propulsion system and provide a design guideline of the system. We then examine various hopping profiles of the robot on a structured terrain using simulation and identify safe and efficient hopping trajectories in Section 4.3. Section 4.4 extends the previous section, and it studies localization and path planning algorithms for energy-efficient navigation of the robot on the lunar surface. The chapter concludes with Section 4.5.

4.2 Thruster Evaluation

A number of options are available for propulsion systems, such as solid rockets, cold-gas thrusters, monopropellant, and bipropellant propulsions. In this work, cold-gas thrusters are chosen for the following reasons: 1) they are safe to operate in a university research setting, 2) they are of low system complexity, 3) they are inexpensive and readily available, and 4) they provide sufficient low-thrust propulsion for the hopping of lightweight tensegrity robots.

Nitrogen, helium, and carbon dioxide are the most popular propellants available for cold-gas thrusters. Although helium has a high specific impulse, it also has a very low density, which necessitates a large volume for storage that is unfavorable due to size constraints. Carbon dioxide can also be problematic because it is stored in mixed gas and liquid phases,

¹More details on the design of TT-3 is provided in Chapter 2.

Table 4.1: Parameters and thermodynamic quantities used for nitrogen thruster analysis [5].

Parameters	Values	Parameters	Values
λ	0.95	a_0	204 (m/s)
P_0	2.76 (MPa)	ϵ	5
A_t	1.20e-05 (m ²)	C^*	252 (m/s)
γ	1.4	$\dot{m}$	0.132 (kg/s)
R	297 (J/kgK)	I_{sp}	46.5 (s)
T	100 (K)	F_t	50 (N)

the latter needing extra care in handling. Therefore, by elimination, we focus on nitrogen thrusters. See, for example, [113], [114] for more details on this topic.

For this concept to work, the thruster must be able to propel the tensegrity probe for a distance of 1 km over a series of hops in such a way that the payload is not damaged from excessive forces during landings. Design choices must be carefully made because large aggressive hopping could damage the payload, while small hops would be too inefficient. In this section, we analyze the feasibility of a nitrogen thruster using the design parameter values summarized in Table 4.1. In this analysis, it is assumed that an ideal one-dimensional nozzle is choked at its throat. The parameters include nozzle efficiency (λ), propellant mass flow rate ($\dot{m}$), specific impulse of nitrogen (I_{sp}), nozzle inlet pressure (P_0), nozzle's throat area (A_t), characteristic exhaust velocity (C^*), speed of sound in nitrogen (a_0), nozzle's expansion ratio (ϵ), specific gas constant of nitrogen (R), specific heat ratio of nitrogen (γ), temperature (T) and estimated thrust of (F_t). With the provided parameter values, a thrust of 50 N is expected from the nitrogen thruster. For a more thorough analysis on nozzle flows and thrust models, we redirect the readers to [115], [116].

If launched at a 45-degree angle with an initial mass of 10 kg, thrust of 50 N, and burn time (i.e., the time during which the propellant is consumed) of 9 s, the robot flies more than 1 km of lateral distance on the Moon, according to a simple particle mass dynamics model (Figure 4.2). The total amount of propellant required for this hopping motion is 1.19 kg, but notice that this weight is for the thermodynamic conditions provided in Table 4.1. The total weight of the propellant needed is affected by several factors and could further be reduced, for example, by increasing the tank pressure.

Including the propellant, tank, and tensegrity structure weight, it is estimated that the total weight of the robot is below the 10 kg assumption. Therefore, it is expected that the robot will be able to travel a distance farther than what is simulated. However, with a maximum hop height of 200 m as shown in Figure 4.2, the payload would receive a significant impact when landing. Having multiple, shorter distance hops can help reduce the landing velocity and thus reduce the landing impact. An analysis of several different multi-hop scenarios is provided in Section 4.3.

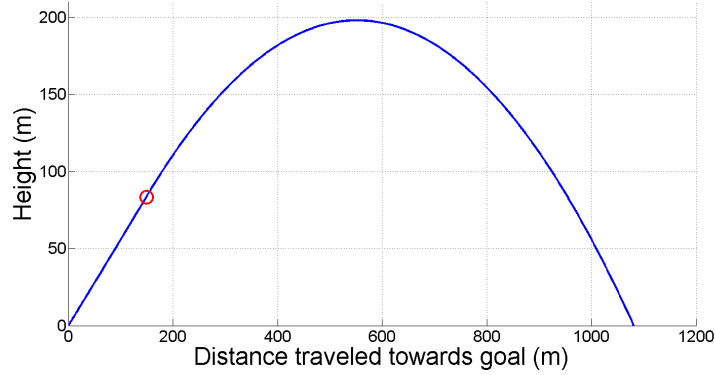


Figure 4.2: A simulated trajectory of a thruster-based tensegrity rover performing a single hop under a particle mass assumption. The rover starts firing from the beginning and the burning period ends at the red point. A constant thrust vector, with its magnitude of 50 N and angle of 45 degrees, is applied to simulate the trajectory [5].

4.3 Simulations of Thruster-Based Mobility

Simulations in this chapter are performed with the NASA Tensegrity Robotics Toolkit (NTRT) [88]. A six-rod tensegrity robot is modeled in NTRT by adopting physical parameters of TT-3 (Figure 4.3). To simulate thruster-based mobility, a thrust vector is applied to the center of the payload which is suspended at the center of the structure. To model real-world disturbances, noise is added to the magnitude and orientation of the thrust. The noise properties of the actual system will depend on the design of the system and environment where the robot operates. However, since the noise properties of the current system are not known at this stage, a simple Gaussian model is chosen. For every 0.1 ms of time step, thrust magnitude and orientation angles are contaminated with zero mean noise with standard deviations of 0.02 N and 0.002 radians, respectively. As a result, the thruster nozzle and thrust orientation vectors will not be perfectly aligned, and error between the two accumulates over time. This is a rather pessimistic open-loop control model that results in large positional errors. However, our results show that even under these assumptions it is possible to meet the design goals.

It should be noted that in a real system, the total mass of the robot will decrease as the robot performs hopping because of the propellant used. However, we do not consider this mass reduction in our simulations, that is, we simulate the worst case scenarios in terms of traveling distance. Moreover, our simulations showed that the effect of reducing weight on the distance traveled towards the goal is dominated by the effect of noises of thrust orientation angles. In the presence of noise, an initial error in thrust direction could cause off-track lateral motion of the robot, causing an increase in the positional error between the landing and target locations. As a result, in the absence of thrust orientation control, a distance towards the target after a hop may still be large even if the hop distance was

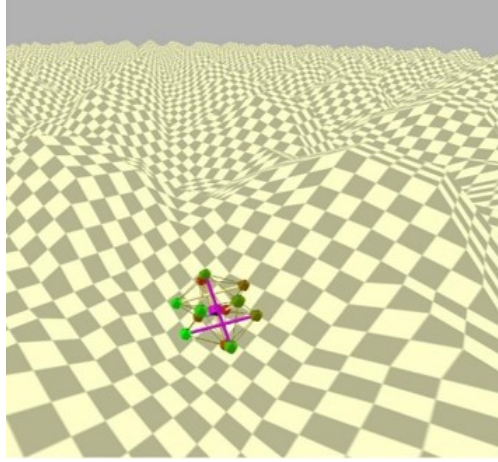


Figure 4.3: A thruster-based tensegrity robot modeled in NTRT on hilly terrain [5]. Videos of the robot in motion are available at <http://best.berkeley.edu/best-research/best-berkeley-emergent-space-tensegrities-robotics/>.

increased due to the reducing weight. For this reason, we focus on the control of thrust orientation angles in Chapter 5.

Many options are present for possible hopping trajectories. The robot may travel the whole distance of 1 km in a single hop or it may break its path into multiple hops. The choice of the hopping trajectory depends on several factors, such as terrain conditions, presence of obstacles, and energy expenditure. Here, four representative hopping profiles on hilly terrain conditions are considered. For each hopping profile, the nominal flight distance per hop is (1) 1000 m, (2) 330 m, (3) 100 m, and (4) 10 m. The required thrusts and burn times for each nominal hopping distance are first obtained with a particle mass model and then applied to the robot model in the NTRT environment. Specifically, burn times are set to 9.3 s, 5.3 s, 2.94 s, and 0.93 s for nominal hopping distances of 1000 m, 330 m, 100 m, and 10 m, respectively, with a nominal thrust of 50 N. Examples of robot flight trajectories for different hopping profiles are presented in Figure 4.4.

Figure 4.4 shows that the final location of the robot, after hopping is completed, is closer to the target as the burn time per hop (or equivalently, a nominal flight distance per hop) gets smaller because hopping resolution increases and the robot has more opportunities to correct its flying direction in between the hops. The average distances between final positions of the robot and the target location over five simulations are summarized in Figure 4.5 and Table 4.2. The averages of simulated hopping distances per hop for different nominal hopping distances are presented in Figure 4.6 and Table 4.2. The robot's initial and target locations were fixed during the simulations.

During the operation of a cold-gas thruster, the amount of propellant used is closely related to the total burn time. These are summarized in Table 4.3 for different hopping profiles. While a hopping profile with a greater burn time per hop tends to be more energy

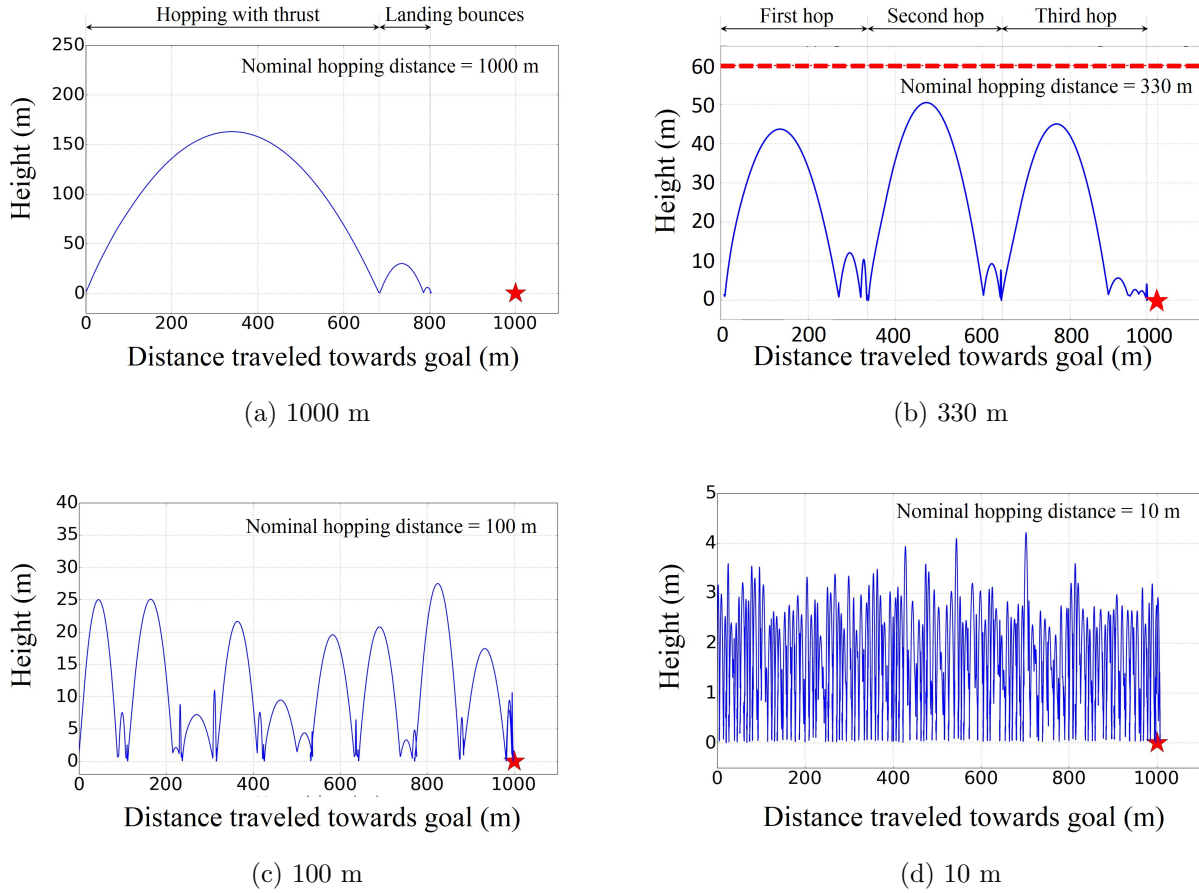


Figure 4.4: Example flight trajectories of a thruster-based tensegrity robot on hilly terrain. The nominal hopping distances per hop are (a) 1000 m, (b) 330 m, (c) 100 m, and (d) 10 m. Red star markers represent the target location located 1000 m away from the initial position of the robot (origin). In all figures, each hop is followed by landing bounces, resulting in additional small hops. In (a), although the average travel distance of the robot is close to 1000 m, the distance traveled towards the goal is not because of the lateral displacement it made due to angular noise of the thrust vector. In (b), a red dashed line indicates the 60 m height constraint. Once a hopping phase is over, the robot may move to the target by precision rolling [5].

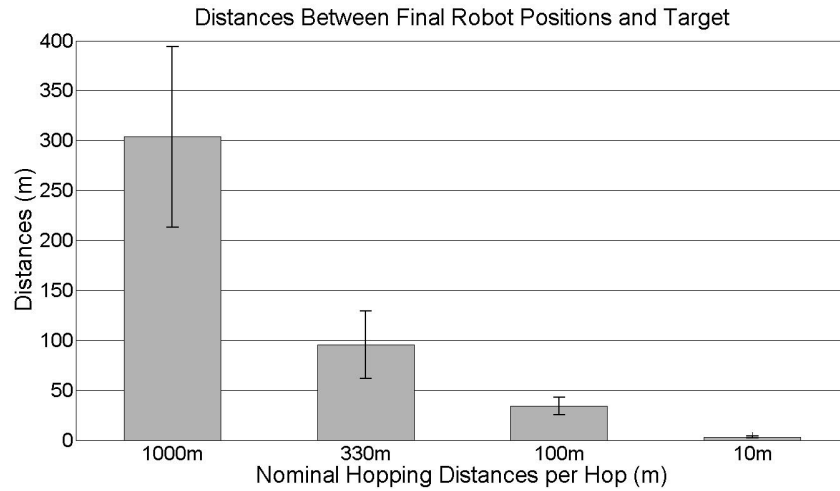


Figure 4.5: Average of distances between final locations of the robot and the target after hopping phase is over. Simulations are run five times for each hopping profiles on hilly terrain assuming noise standard deviations of 0.02 N and 0.002 radians for thrust magnitude and orientation angles, respectively. The final location of the robot is closer to the target as the nominal flight distance per hop gets smaller because the hopping resolution increases and because the robot gets more chance to correct its flying direction between the hops, but at the cost of increased energy expenditure [5].

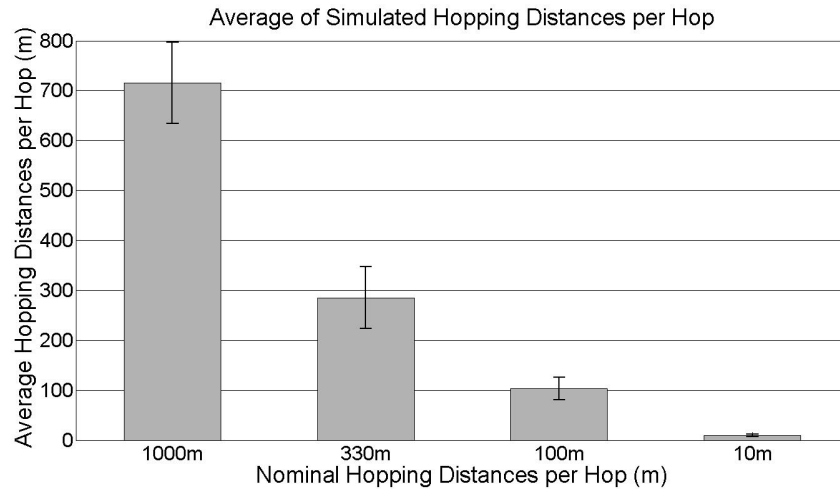


Figure 4.6: Average of simulated hopping distances per hop for different nominal hopping distances. Five simulations are run for each case assuming noise standard deviations of 0.02 N and 0.002 radians for thrust magnitude and orientation angles, respectively. The average hopping distances are not equal to the nominal hopping distances due to the noise [5].

Table 4.2: Averages of total number of hops performed, simulated hopping distances per hop and final distances between the robot and target for different hopping profiles. Total of five simulations are run for each case [5].

Nominal hopping distances (m)	1000	330	100	10
Average of total number of hops performed to the target	1	3.8	10	111.4
Average of simulated hopping distances per hop (m)	715.3	285.2	103.7	9.6
Average distances between final robot position and the target after hopping is done (m)	303.8	95.4	34.0	2.9

efficient, it is less accurate in hitting the target location (Figure 4.5). This suggests that a hopping profile that is a mix of both long and short hops will not only be more energy efficient than a single long hop strategy, but also is better able to position the robot to the target location before precision rolling begins. Although the most energy-efficient, a single long hop hopping profile, such as the case shown in Figure 4.4a, is undesirable because the robot reaches a maximum height of over 150 m with this profile, and the fall from such a height may damage the robot upon landing. In [59], [112], it was shown that a tensegrity structure can survive from a 10 m drop under the Earth’s gravity, in which case the terminal velocity of the robot just before hitting the ground is 14 m/s. Moreover, the structure was able to protect its center payload from the landing impact. Notice that most of the impulse the robot experiences during an actual landing event is caused by the change in the vertical component of velocity, and this drop test effectively mimics the situation. Under the Moon’s gravity, if a body is dropped at the height of 60 m, its final velocity just before hitting the ground is 14 m/s. Therefore, we limit the robot’s flight trajectory to stay below a maximum height of 60 m for safe landing. According to the first principle particle mass model, this corresponds to a nominal burn time of 5.3 s and a nominal flight distance of 330 m per hop with a 50 N of thrust. This is the reason why we chose to simulate 330 m hopping trajectories in the first place. Figure 4.4b shows an example of such a hopping profile consisting of three hops. The variation of the peak heights is the consequence of noises in the thrust magnitudes and orientations that was added to the simulations. With this profile, the robot, on average, was able to reach a point 95.4 m away from the target on hilly terrain. From this point, the robot can either make shorter hops or roll towards the target.

An example flight trajectory of the thruster-based robot, simulated in NTRT with the Moon’s gravity, combining hops of different nominal hopping distances and precision rolling is shown in Figure 4.7. In this simulation, flat terrain is assumed and the actual travel distances of hops are greater than the nominal hopping distances due to the secondary bounces. Hence the robot made two 330 m hops, one 100 m hop and one 10 m hop during

Table 4.3: Averages of total burn time and consumed propellant mass for different hopping profiles. Simulations are run five times for each case [5].

Nominal flight distance per hop (m)	1000	330	100	10
Total burn time to reach target (seconds)	9.3	20.1	29.4	103.6
Consumed propellant mass (kg)	1.23	2.65	3.88	13.68

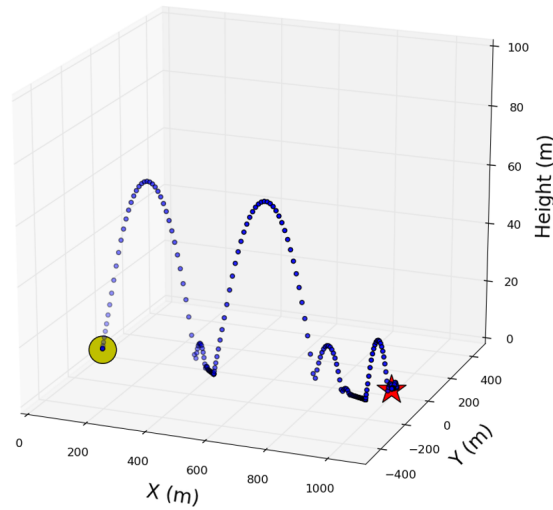


Figure 4.7: A simulated trajectory of the thruster-based tensegrity robot traveling 1 km to reach the target. The green dot is the initial position of the robot and the red star is the target location which is 1 km away from the initial robot position. During the hopping phase, the robot made two 330 m hops followed by one 100 m hop and one 10 m hop. Note that the maximum height of the robot during the hopping phase is below 60 m. When the robot gets close enough to the target, it ends the hopping phase and switches to the precision rolling phase.

its hopping phase instead of making three 330 m hops as presented in Figure 4.4b². Once the robot reaches close to the target, it ends the hopping phase and switches to the precision rolling phase.

Lastly, we note that the time required for the robot to arrive at the target location was not considered when comparing different hopping profiles because the traveling time is not a primary constraint in our mission. In general, a hopping profile with a shorter hopping distance per hop takes longer time to finish the hopping stage because the robot spends more time adjusting its orientation between hops.

²The hopping distances mentioned here refer to the nominal hopping distances. The actual hopping distances are greater than these nominal distances for all four hops.

4.4 Localization and Path Planning

Up until this point, it has been shown that a spherical tensegrity robot can utilize both rolling and hopping for its movement, but the simulations have been done on rather structured terrain. We now discuss how these separate motions may be combined for energy-efficient navigation on the actual lunar terrain³.

For a realistic analysis, we consider the map of actual lunar surface in this section instead of flat or structured hilly terrain that we considered in earlier sections. The map of the mission surface is assumed to be readily available⁴ and divided into a grid to represent the environment, with each grid cell representing a square patch with local height information. This representation allows importing height maps directly into the simulation⁵ to test the algorithm on real data. Before the robot can plan its path and actions to reach the goal, it first localizes itself on the given map. For this, the robot relies on a dynamically updated belief space that represents the probability of the robot being at a specific position. Initially, the probability is uniformly distributed across all possible positions in the map. Assuming that the robot can sense the height of the four neighbor cells – east, west, north, south – the robot updates its belief map by examining which cells on the map have their neighbors with approximately the same heights. After having its beliefs updated, the robot rolls one step in a direction and repeats the process until it is confident enough of its position (Figure 4.8). In addition to finding the robot’s initial position, this method can also be used to precisely locate itself on the map after hopping which may deteriorate the robot’s estimation of its position due to thruster noises and secondary rolling after landing.

Once the robot’s position is known, we use the A-star search algorithm [97] with the Euclidean distance to the goal as a heuristic to find an energy-efficient path (Figure 4.9). The cost function consists of several components, each of which defines the cost of different actions (e.g., rolling and hopping) based on the travel distances with the actions. For hopping motion, an additional constraint on the height difference between the robot’s initial and final positions is considered in order to prevent damaging of the robot from hard impact.

At each iteration, the robot can move to any of the eight adjacent grid cells by punctuated rolling. This movement is practical only if the height change is not too significant in the neighbor region. Otherwise, the robot might not be able to climb, e.g., a steep hill and the cost of taking this action will be expensive. On the other hand, because going downhill requires less energy than going uphill or moving on a flat terrain, the robot exploits this in planning its path. Thruster-based hopping allows a wider range of movement directions and distances. While more expensive than punctuated rolling, hopping is unavoidable in some cases, e.g., when the robot needs to escape from a crater with steep slopes. Although

³The work presented in this section is based on the work of Julien Despois [117], a former Master of Engineering student of the BEST laboratory whom the author mentored.

⁴We used a map obtained from USGS Astrogeology Science Center (<http://astrogeology.usgs.gov/>).

⁵We were not able to import the actual lunar map into NTRT at the time of this section’s work, so we developed our own software in Java. Later on, however, researchers of the BEST laboratory have completed this import and the simulation in NTRT can now be run with actual lunar terrain maps, see [118].

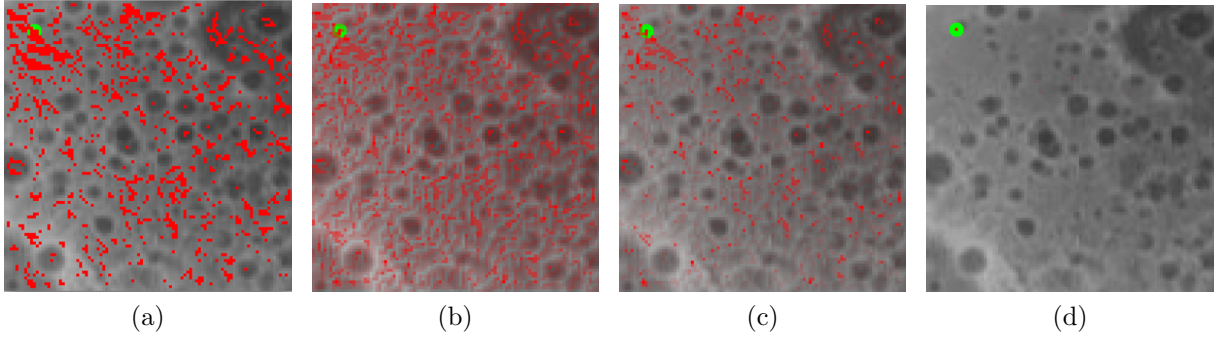


Figure 4.8: Evolution of the robot's belief map while localizing itself with the known lunar surface map. Red areas represent the robot's estimated positions and green dot is the robot's true location. (a) Belief map after first measurement of neighbor grid heights. (b) Belief map after performing one step of punctuated rolling. (c) Belief map after measuring new neighbor grid heights. (d) Belief map at the end of localization process. Notice that the red areas shrank into the location of the green dot. That is, the robot successfully localized itself on the lunar map. Map courtesy of USGS Astrogeology Science Center (<http://astrogeology.usgs.gov/>) [5], [117].

the hopping motion increases the average number of nodes⁶ reachable from each location, the A-star algorithm tries to limit the number of nodes expanded in order to reduce the computation effort.

In practice, the robot may fail when trying to execute a movement. To incorporate such failures, we define the possible outcomes of the movement along with their probabilities to happen. The robot then takes the following sequence to prevent itself from drifting too much: 1) The robot computes the best path to reach the goal from its current position. 2) The robot takes the first K actions according to its plan. Some actions may fail. 3) The robot localizes itself again after K actions. 4) The robot re-computes the best path for the remaining distance. 5) The robot lowers K to improve its accuracy near the goal. 6) The above steps are repeated until the robot arrives at the goal.

Notice that, in this approach of correcting the path of the robot, the only type of sensors we assume the robot to have is height measurement sensors that were already used for localization. If the robot has additional sensors, e.g., cameras for visual feedback, the robot will be able to locate itself more precisely on the map by sensor fusion and promptly detect its drift from the planned path, enabling more energy-efficient navigation.

⁶Here, the term *node* is used in the context of a search algorithm and refers to a leaf node of a search tree. The readers should not confuse this with rod end nodes that we defined in Chapter 2.

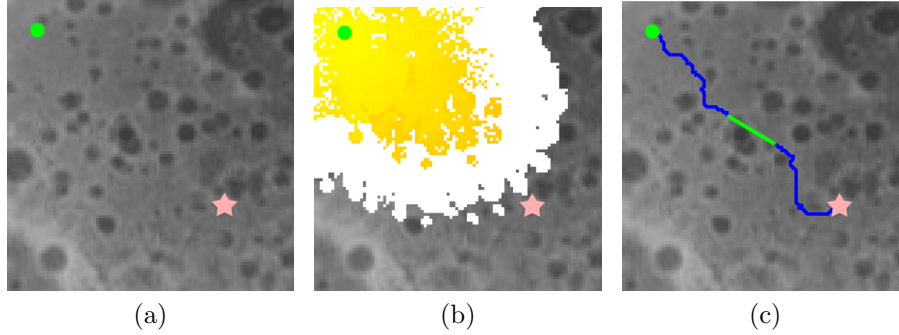


Figure 4.9: The robot develops an energy-efficient path to the goal by combining hopping and precision rolling, once it finishes the localization process. (a) A lunar terrain map with height information with initial and goal locations marked with green dot and pink star, respectively. (b) A-star algorithm is used to search for the best path to the goal position. Yellow nodes are those already expanded while white nodes are those to be expanded. The shade of the yellow nodes is proportional to the cost to reach the grid cell. (c) The planned path and actions. Blue and green lines represent paths covered by punctuated rolling and hopping, respectively. Map courtesy of USGS Astrogeology Science Center (<http://astrogeology.usgs.gov/>) [5], [117].

4.5 Conclusions and Future Research

In this chapter, we have presented a spherical tensegrity robot equipped with a thruster system and demonstrated that the robot can travel from one location to the other in an energy-efficient manner by combining hopping and rolling, focusing on applications in lunar exploration missions. Lightweight and compliant tensegrity robots are especially suitable for hopping motions in comparison to other conventional rigid body robots because they can withstand significant impacts at landing at the end of each hop without damaging itself or its payload. We have chosen nitrogen cold-gas thrusters as our propulsion system because they are inexpensive, are safe to operate, and yield low system complexity, yet still provide sufficient thrust for the hopping of lightweight tensegrity robots. Based on simulations, it has been shown that hopping with a thruster is a feasible method for long distance travel in short durations of time.

The complete mission profile of the thruster-based tensegrity robot is a combination of hopping and rolling motions. We have implemented the A-star algorithm to find a feasible path of the robot from its initial location to the goal location, which also minimizes energy consumption by appropriately combining hopping and rolling motions under the given terrain conditions. At this moment, the cost function consists of our best estimate on the energy expenditure of hopping and rolling motions. The exact costs are not well known at this point because the design of our tensegrity robots and their thruster systems is still under development and yet to be finalized. We anticipate that the future researchers at the BEST

laboratory will perform a more thorough energy analysis and further optimize traveling path of the robot. We have also shown that the robot can localize itself on a given lunar map by measuring heights of its surroundings and comparing that to its knowledge of terrain. Considering that the robot's motion always accompany uncertainties, this localization method will greatly help the robot to re-calculate right path to the target after it makes several moves by precisely correcting its current location.

Chapter 5

Orientation Control of Thruster Systems on Tensegrity Robots

5.1 Introduction

In Chapter 4, we presented a hopping tensegrity robot equipped with a cold-gas thruster. With help of the thruster, the robot is able to quickly travel long distances, while less restricted by ground conditions and the presence of obstacles. In addition, we want to steer the heading direction of the robot towards the target when the robot hops in order to reduce the energy consumption and mission time. We note that previous simulations in Chapter 4 did not use feedback control for thrust regulation. As a result, when simulating this system with a reasonable degree of trajectory noise, it was observed that the resulting trajectories were not necessarily the most energy efficient (Figure 5.1). An error in the initial thrust direction could cause the robot to be off-track, wasting part of its propellant in moving towards an undesirable direction. This behavior is also expected to occur in a real-world system, and the need for thrust vectoring arises for energy-efficient operation of the robot. Furthermore, improper thrusting of the robot could cause deformation of the outer tensegrity structure, which in turn will introduce additional disturbance to the thrust. To address these issues, the development of suitable control for thrust vectoring during hopping is crucial.

5.1.1 Related Research on Thrust Vectoring of Tensegrity Robots

As the robot makes multiple hops on its way to the target location, it has chances to correct for heading errors when it is on the ground between hops. Originally, three high level approaches for adjusting the thruster direction during hopping were investigated [5], [82].

(1) Install a two degree of freedom gimbal at the nozzle of the thruster. This will be referred to as a *gimbaled-nozzle thruster*. Gimbaled-nozzle thruster systems are well researched in rocketry and space flight for flight direction control. Since only the low-mass nozzle is controlled with the gimbal, the torque requirement is expected to be small.

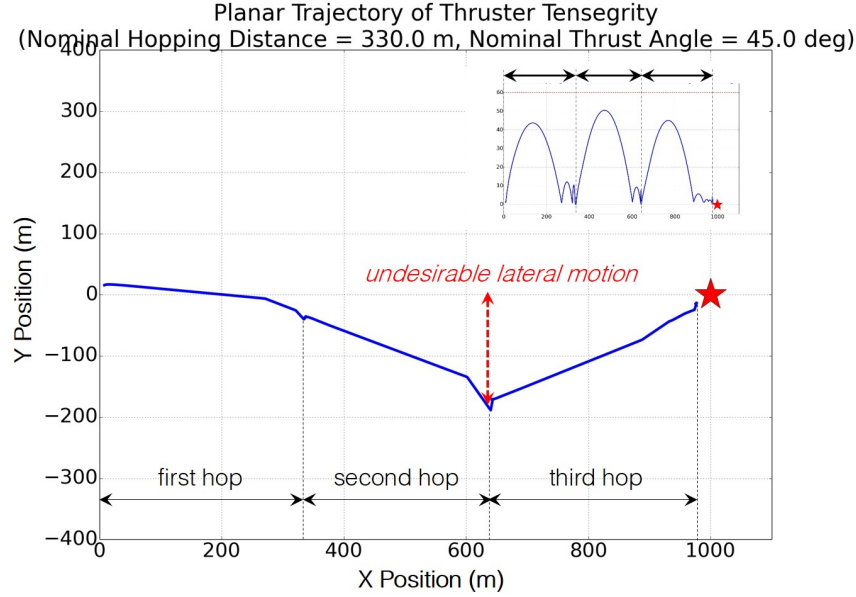


Figure 5.1: A top view of a hopping trajectory consisting of three hops with a nominal hopping distance of 330 m per hop. The front view of this hopping trajectory is presented in Figure 4.4b and reproduced at the top right corner for readers. Because there exists no feedback control for thrust vectoring, the robot does not necessarily hop towards the target and undesirable lateral motion occurs due to the noises included in the simulation. This off-track lateral motion wastes the propellant of which the robot carries only limited amount.

(2) Another concept is to enclose the tank and thruster nozzle inside of a two degree of freedom gimbal structure. This will be referred to as a *gimbal-enclosed thruster system*. The gimbal-enclosed thruster system allows the thruster a much larger range of motion in comparison to the gimbaled-nozzle thruster system. This concept was modeled and prototyped using a 3D printer as shown in Figure 5.2 to understand the large range of motion and visualize the payload size and spacing within the tensegrity robot [82]. The gimbal-enclosed thruster system can be useful when the robot has to adjust the thruster orientation for hopping but it cannot do so by moving its whole body due to the environment, e.g., when the robot is stuck in a crater. Moreover, this concept allows not only the thruster but also the payload to utilize the gimbal system. For example, this setup will enable an imaging payload, such as a camera, to rotate and sweep within the gimbal system.

(3) The third concept explored for thrust vectoring is a *cable-actuated thruster system*, which utilizes the inherent shape-shifting capability of the outer tensegrity structure [5]. This method leverages the intrinsic high-degree of freedom present in an active tensegrity robot. In the simplest form of this system, the thruster and payload would be connected to the ends of the tensegrity rods by fixed-length compliant cables as shown in Figure 5.3 and thrust

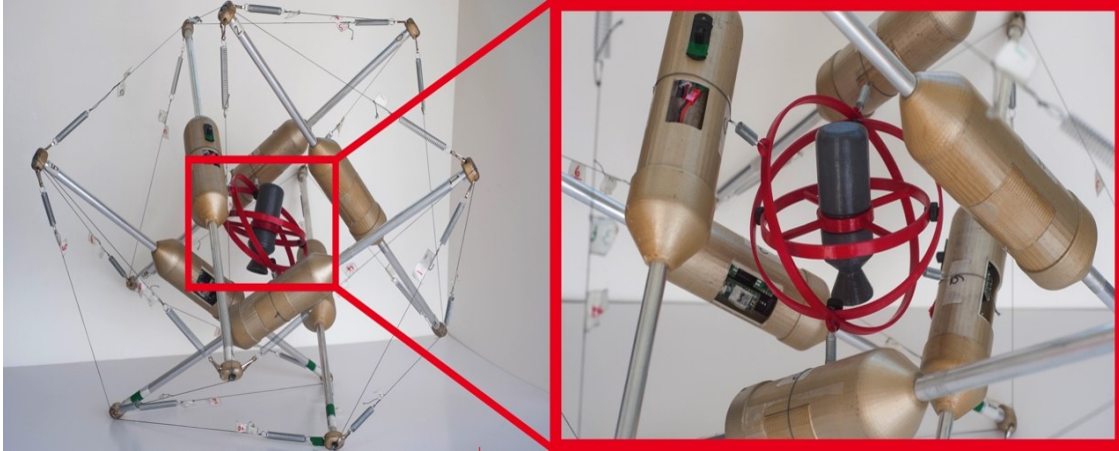


Figure 5.2: A mockup of a gimbal-enclosed thruster system located at the center of TT-3 [5]. Red and grey parts are mockups of a gimbal and cold-gas thruster, respectively. The gimbal-enclosed thruster is connected to the robot structure with springs. The thrust vectoring algorithm guarantees that the nozzle exit flow does not interfere with the gimbal and robot structure.

vectoring would be achieved by changing the shape of the structure. In a more flexible version of this concept, additional actuation would be added to the support cables of the payload to allow for more control through shape changing. Recall from Chapter 2 that the cables constituting the outer structure are referred to as outer cables, and the payload-suspension cables are referred to as inner cables. With this approach, the cable-actuated thruster can change its orientation by controlling the lengths, or equivalently tensions, of inner and/or outer cables. This approach has an advantage over the gimbal-based systems in that there is no need to add an additional gimbal system at the center of the structure and the center space can be fully used for installation of a payload. The drawback of this approach is that it is difficult to predict how the tension change will affect the thruster direction since tension is distributed in a highly coupled and nonlinear manner across tensegrity members. More elaborate study on this subject is presented in [5].

5.1.2 Contribution of This Chapter

All of the three approaches listed in the previous section have potential to achieve the goal of thrust vectoring on tensegrity robots. However, each of them also has downsides. Both gimbal-based systems (i.e., gimbaled-nozzle and gimbal-enclosed thrusters) require additional mechanical components as well as actuators and electronics that will result in a heavier robot. There is also a risk of nozzle exit flow hitting the gimbal components, which may damage the system. While it is possible to realize a cable-actuated thruster system without any additional components, it is challenging to design effective controllers for this approach because of high nonlinearity inherent to tensegrity structures.

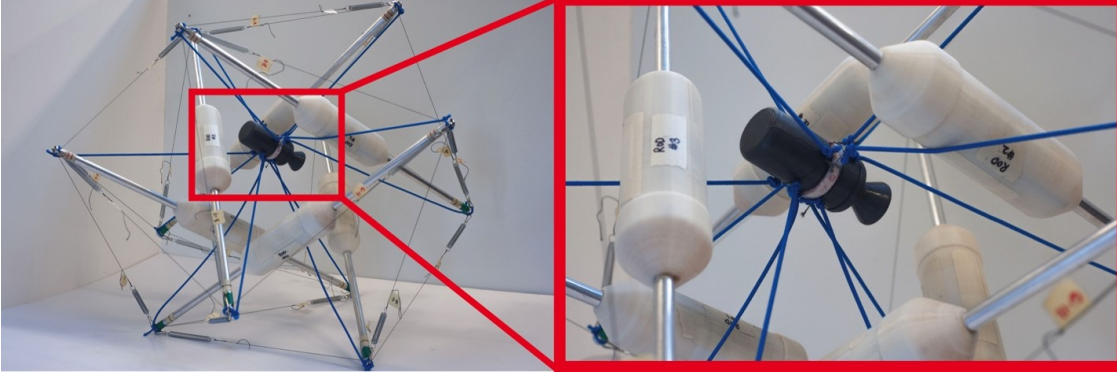


Figure 5.3: A mockup of a cold-gas thruster system is at the center of the robot through cable connections to outer tensegrity structure [5]. The thruster direction is controlled by changing the shape of the robot structure by using the shape-shifting capability of the structure.

As an alternative method to these approaches, we propose a reaction wheel based orientation control mechanism and design its controller in this chapter. For this, we assume all of the thruster components and a payload is packaged in a cubic rigid body for protection. The cubic rigid body is attached with two reaction wheels on two of its faces that are orthogonal to each other (Figure 5.5). If a thruster nozzle is installed on the face that are orthogonal to the faces with reaction wheels, then nozzle exit flow will not interfere with reaction wheels. Therefore, unlike gimbal-based systems, the risk of damaging thrust vectoring mechanism by nozzle flow does not exist with the reaction wheel system. Furthermore, it will be shown that lightweight wheels are sufficient to achieve the desired goal with the controller to be developed, and thus the total weight of the robot is only slightly increased.

With these advantages of the reaction wheel system in mind, we will mainly focus on its controller design in the rest of this chapter.

5.1.3 Prior Research on Attitude Stabilization

In space engineering, attitude stabilization of a rigid body is, in most cases, done by using gas jet thrusters or reaction wheels. If three independent torques can be supplied by either mechanism, the system is controllable and the body can be stabilized about an equilibrium [119]. Previous research has shown that, in the case of gas jet thrusters, the body can still be stabilized about an equilibrium even when less than three independent torques are supplied, under certain conditions [119], [120]. While there exists no smooth state feedback law achieving this with only two gas jet actuators [121], a time-varying continuous feedback can locally asymptotically and exponentially stabilize the attitude of the body [122].

In the case of reaction wheels, the system with three wheels is controllable if three independent torques are supplied with a certain minimum torque level [119]. When the angular momentum of a body is zero, a discontinuous state feedback controller can be developed for

the attitude stabilization of the body [123]. Recently, controllability of an underactuated satellite with two reaction wheels is analyzed and control laws are developed for the satellite's attitude stabilization for different system configurations [124], [125].

Despite the challenges of the underactuated reaction wheel systems, they are quite useful in practice due to the simplicity in their system architecture. Furthermore, they can effectively stabilize partial states of the system [120], and spin-axis stabilization using only two reaction wheels has been an important problem in this regard [126].

In this chapter, we derive a feedback control law that stabilizes a revolute motion of a rigid body about an arbitrary direction using two reaction wheels attached to it. Specifically, we aim to develop the control law that achieves two goals: 1) the spin axis of the body is oriented towards an arbitrary target direction, and 2) the body rotates about the spin axis only, if the system's angular momentum is not zero.

A similar problem was discussed in [127], where several globally asymptotically stable and globally exponentially stable control laws are developed for the purpose of spin-axis stabilization of a symmetric spacecraft using two gas jet actuators. A globally asymptotically stable control law that achieves spin-axis stabilization of a spacecraft using two reaction wheels is also presented in [126]. In both works, the final direction of the spin axis is aligned with a specified inertial axis when stabilized.

In many practical applications, however, it is desirable to orient the spin axis of the primary body towards an arbitrary axis rather than an inertial axis. Consider a spacecraft or a planetary rover that utilizes an on-board thruster system as its source of mobility and assume that the thruster nozzle axis is in the direction of the spin axis of the system. During its operation, the spacecraft or the rover is occasionally required to change the orientation of the thruster nozzle such that the nozzle axis is aligned with the desired thrust direction. In such cases, however, it is unlikely that the desired thrust direction is always parallel to an inertial axis. Our problem of controlling orientation of a thruster system used for hopping tensegrity robots is a good example for this. The desired thrust direction may change between hops and in such cases the control needs to reorient the nozzle axis to match the new target direction. Hence the need for a more generalized control law arises; the law that can stabilize the nozzle axis of the thruster system towards an arbitrary direction. In particular, use of reaction wheels is preferable than gas jet thrusters for our hopping tensegrity robots because it allows the robots to spend propellant towards hopping only and thus increases their maximum travel distance.

To this end, we improve upon the approach taken in [126], [127] by modifying the kinematics description of a body, and then developing a controller based on this description. In Section 5.2, we introduce a modified (z, w) -parameterization [128] as a way of describing the rotational kinematics of a body. We then introduce dynamics of a rigid body with two reaction wheels attached along the body's principal axes in Section 5.3. We use a feedback linearization technique in Section 5.4 to develop the desired control law and prove its stability via Lyapunov's direct method in conjunction with LaSalle's invariance principle. We also simulate the control on a thruster system of a hopping tensegrity robot and show the controller's stabilizing behavior. Lastly, conclusion is provided in Section 5.5.

5.2 Attitude Kinematics

In this section, we derive a set of kinematic equations describing the attitude of a rigid body, which will later be used for controller design in Section 5.4. The derivation closely follows [128] and is based on the (z, w) -parameterization therein. However, we modify it to include the information of the arbitrary target direction and present a new formulation extending the prior work. This modification is necessary in order to develop the controller that stabilizes the spin axis of the body about an arbitrary direction.

5.2.1 Parameterization of Rotations

Let the two sets of orthonormal right-handed basis vectors for the inertial and body-fixed reference frames be $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$ and $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, respectively. Then, there exists a rotation matrix $\mathbf{R} \in SO(3)$ that describes the relative orientation of the two frames. Using matrix notation, this relationship can be written as

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix} = \mathbf{R} \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \\ \mathbf{E}_3 \end{bmatrix}. \quad (5.1)$$

We then decompose the matrix $\mathbf{R}$ into three sub-rotations,

$$\mathbf{R} = \mathbf{R}_3(z)\mathbf{R}_2(w)\mathbf{R}_1, \quad (5.2)$$

where $\mathbf{R}_1, \mathbf{R}_2(w), \mathbf{R}_3(z) \in SO(3)$. Note that the two rotations $\mathbf{R}_2(w)$ and $\mathbf{R}_3(z)$ are parameterized by w and z , which are functions of time. $\mathbf{R}_1$ is not parameterized because we assume this matrix is constant in this work, as will be discussed later.

Furthermore, we define two additional sets of orthonormal right-handed basis vectors $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ and $\{\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\}$ to describe intermediate reference frames during the rotation $\mathbf{R}$ (Figure 5.4). The relationships between the rotation matrices and the basis vectors are given as

$$\begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix} = \mathbf{R}_1 \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \\ \mathbf{E}_3 \end{bmatrix}, \quad (5.3)$$

$$\begin{bmatrix} \mathbf{e}'_1 \\ \mathbf{e}'_2 \\ \mathbf{e}'_3 \end{bmatrix} = \mathbf{R}_2(w) \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix}, \quad (5.4)$$

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix} = \mathbf{R}_3(z) \begin{bmatrix} \mathbf{e}'_1 \\ \mathbf{e}'_2 \\ \mathbf{e}'_3 \end{bmatrix}. \quad (5.5)$$

We let $\mathbf{t}_3$ be the direction about which the spin axis of the body will be stabilized. Letting $\mathbf{e}_3$ represent the current spin axis of the rigid body, the control objective will then be to

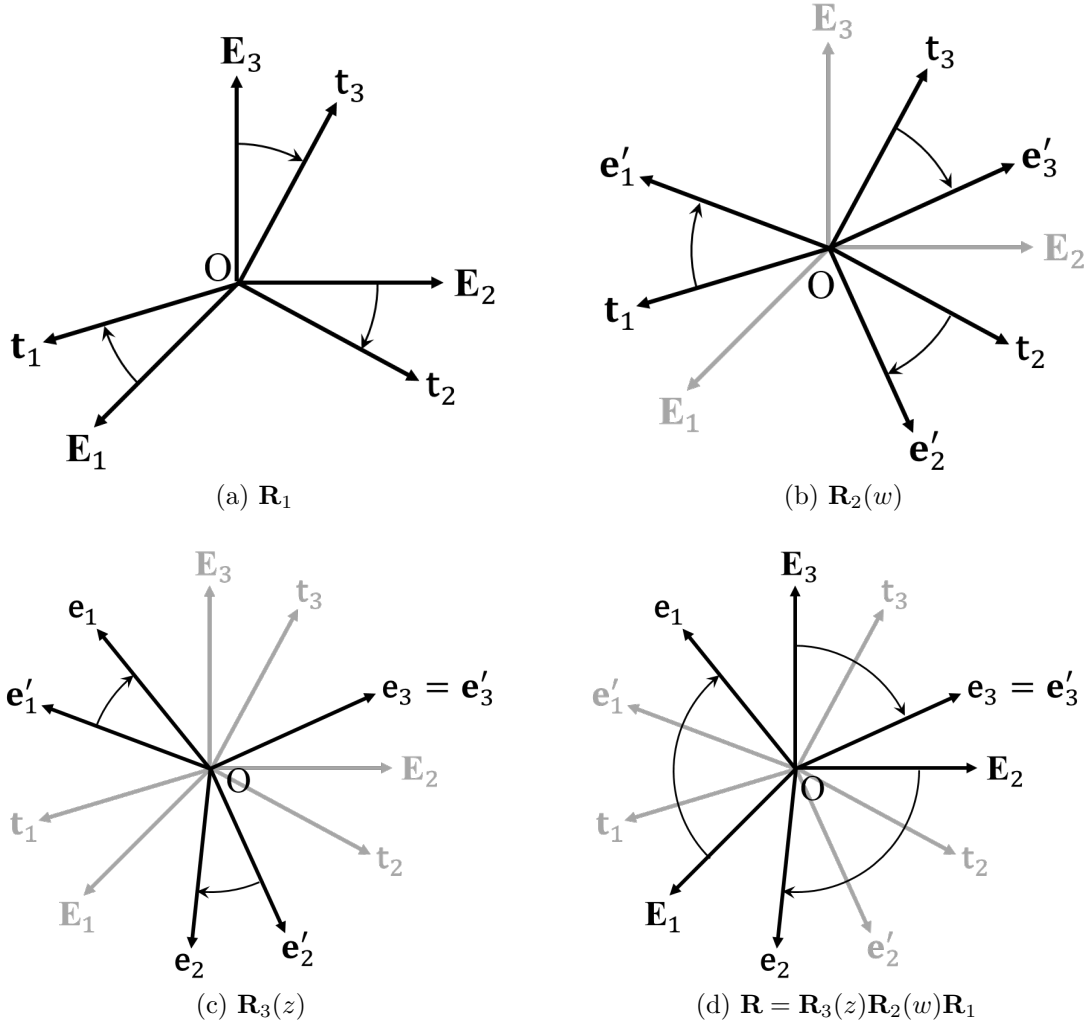


Figure 5.4: Rotations and basis vectors used to describe attitude kinematics of a rigid body. The rotation $\mathbf{R}$ between the inertial frame and body-fixed frame consists of three sub-rotations. (a) The first rotation rotates $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$ to $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$. $\mathbf{t}_3$ is the target direction about which the spin axis of the body will be stabilized. (b) The second rotation rotates $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ to $\{\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\}$. (c) The third rotation is about the axis $\mathbf{e}'_3 = \mathbf{e}_3$, and it rotates $\{\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\}$ to $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. (d) The overall rotation $\mathbf{R}$ is a combination of $\mathbf{R}_1$, $\mathbf{R}_2(w)$ and $\mathbf{R}_3(z)$ [6].

align $\mathbf{t}_3$ and $\mathbf{e}_3$. Moreover, we assume the target direction $\mathbf{t}_3$ is known a priori and time independent in this work. Then, the first rotation $\mathbf{R}_1$ is not an arbitrary rotation, but rather it is a constant rotation that rotates the third inertial axis $\mathbf{E}_3$ to $\mathbf{t}_3$. As a result, $\mathbf{R}_1$ is a constant matrix. From [128], it is known that such a rotation has the form of

$$\mathbf{R}_1 := \begin{bmatrix} k_3 + \frac{k_2^2}{1+k_3} & -\frac{k_1 k_2}{1+k_3} & k_1 \\ -\frac{k_1 k_2}{1+k_3} & k_3 + \frac{k_1^2}{1+k_3} & k_2 \\ -k_1 & -k_2 & k_3 \end{bmatrix}, \quad (5.6)$$

with the constants k_1, k_2 and k_3 satisfying the constraint

$$k_1^2 + k_2^2 + k_3^2 = 1. \quad (5.7)$$

Next, the second rotation $\mathbf{R}_2(w)$ rotates $\mathbf{t}_3$ to $\mathbf{e}'_3$, which is identical to $\mathbf{e}_3$ if we let $\mathbf{e}'_3$ to be the rotation axis of the third rotation $\mathbf{R}_3(z)$. As a result, the second rotation $\mathbf{R}_2(w)$ rotates $\mathbf{t}_3$ to $\mathbf{e}_3$. $\mathbf{R}_2(w)$ has a form similar to $\mathbf{R}_1$ and can be written as

$$\mathbf{R}_2(w) := \begin{bmatrix} c + \frac{b^2}{1+c} & -\frac{ab}{1+c} & a \\ -\frac{ab}{1+c} & c + \frac{a^2}{1+c} & b \\ -a & -b & c \end{bmatrix}, \quad (5.8)$$

with the parameters a, b and c satisfying the constraint

$$a^2 + b^2 + c^2 = 1. \quad (5.9)$$

Unlike k_1, k_2 and k_3 , notice that a, b and c are functions of time because their values change as the body rotates. By using a stereographic projection, the constraint (5.9) can be incorporated into (5.8) [128]. If we define a complex variable $w = w_1 + iw_2$ as

$$w := \frac{b - ia}{1 + c}, \quad (5.10)$$

$\mathbf{R}_2(w)$ can be re-written as

$$\mathbf{R}_2(w) = \frac{1}{1 + w_1^2 + w_2^2} \begin{bmatrix} 1 + w_1^2 - w_2^2 & 2w_1 w_2 & -2w_2 \\ 2w_1 w_2 & 1 - w_1^2 + w_2^2 & 2w_1 \\ 2w_2 & -2w_1 & 1 - w_1^2 - w_2^2 \end{bmatrix}. \quad (5.11)$$

The final rotation $\mathbf{R}_3(z)$ is the rotation about $\mathbf{e}_3 = \mathbf{e}'_3$ and is given by

$$\mathbf{R}_3(z) = \begin{bmatrix} \cos(z) & \sin(z) & 0 \\ -\sin(z) & \cos(z) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.12)$$

5.2.2 Kinematic Equations

By substituting (5.6), (5.8) and (5.12) into (5.2), we can find the expression of $\mathbf{R}$ in terms of k_1, k_2, k_3, a, b, c and z . Although we will not write the full expression of $\mathbf{R}$ here, we do note that the third row of $\mathbf{R}$ is independent of the parameter z . Consider the inverse rotation of $\mathbf{R}$ and define

$$\tilde{\mathbf{R}} := \mathbf{R}^{-1} = \mathbf{R}^T. \quad (5.13)$$

If we denote $\mathbf{R}_{(i,*)}$ as the i -th row of $\mathbf{R}$, and $\tilde{\mathbf{R}}_{(*,j)}$ as the j -th column of $\tilde{\mathbf{R}}$, we have

$$\tilde{\mathbf{R}}_{(*,3)} = [\mathbf{R}_{(3,*)}]^T = \mathbf{A} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad (5.14)$$

where we introduced a constant matrix $\mathbf{A}$ defined as

$$\mathbf{A} := \begin{bmatrix} -(k_3 + \frac{k_2^2}{1+k_3}) & \frac{k_1 k_2}{1+k_3} & -k_1 \\ \frac{k_1 k_2}{1+k_3} & -(k_3 + \frac{k_1^2}{1+k_3}) & -k_2 \\ -k_1 & -k_2 & k_3 \end{bmatrix}. \quad (5.15)$$

Notice that $\mathbf{A}$ is an involutory matrix and $\mathbf{A}^{-1} = \mathbf{A}$.

Furthermore, let us introduce two angular velocities $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$. $\boldsymbol{\omega}$ is the angular velocity vector of the body frame $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ with respect to the inertial frame $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$, whereas $\tilde{\boldsymbol{\omega}}$ is the angular velocity vector of the inertial frame $\{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$ with respect to the body frame $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. We write $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$ as

$$\begin{aligned} \boldsymbol{\omega} &= \omega_1 \mathbf{e}_1 + \omega_2 \mathbf{e}_2 + \omega_3 \mathbf{e}_3 \\ &= \omega_1^E \mathbf{E}_1 + \omega_2^E \mathbf{E}_2 + \omega_3^E \mathbf{E}_3, \end{aligned} \quad (5.16)$$

and

$$\tilde{\boldsymbol{\omega}} = \tilde{\omega}_1 \mathbf{E}_1 + \tilde{\omega}_2 \mathbf{E}_2 + \tilde{\omega}_3 \mathbf{E}_3. \quad (5.17)$$

By definition, $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$ has the relationship

$$\tilde{\boldsymbol{\omega}} = -\boldsymbol{\omega}, \quad (5.18)$$

in the inertial frame. In terms of components, this can be re-written as

$$\begin{bmatrix} \tilde{\omega}_1 \\ \tilde{\omega}_2 \\ \tilde{\omega}_3 \end{bmatrix} = - \begin{bmatrix} \omega_1^E \\ \omega_2^E \\ \omega_3^E \end{bmatrix} = -\mathbf{R}^T \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \quad (5.19)$$

With the help of skew-symmetric matrices

$$\mathbf{S}(\boldsymbol{\omega}) := \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}, \quad (5.20)$$

and

$$\mathbf{S}(\tilde{\boldsymbol{\omega}}) := \begin{bmatrix} 0 & \tilde{\omega}_3 & -\tilde{\omega}_2 \\ -\tilde{\omega}_3 & 0 & \tilde{\omega}_1 \\ \tilde{\omega}_2 & -\tilde{\omega}_1 & 0 \end{bmatrix}, \quad (5.21)$$

the time derivatives of the rotation matrices $\mathbf{R}$ and $\tilde{\mathbf{R}}$ are related to the angular velocities $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$ as

$$\dot{\mathbf{R}} = \mathbf{S}(\boldsymbol{\omega})\mathbf{R}, \quad (5.22)$$

and

$$\dot{\tilde{\mathbf{R}}} = \mathbf{S}(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{R}}. \quad (5.23)$$

Moreover, if we consider only the third column of $\dot{\tilde{\mathbf{R}}}$ in (5.23),

$$\dot{\tilde{\mathbf{R}}}_{(:,3)} = \mathbf{S}(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{R}}_{(:,3)}, \quad (5.24)$$

and by substituting (5.14) into (5.24), we obtain

$$\begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \end{bmatrix} = \mathbf{A}\mathbf{S}(\tilde{\boldsymbol{\omega}})\mathbf{A} \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \quad (5.25)$$

After some algebra, it turns out that the matrix $\mathbf{A}\mathbf{S}(\tilde{\boldsymbol{\omega}})\mathbf{A}$ is also skew-symmetric and has the form of

$$\mathbf{A}\mathbf{S}(\tilde{\boldsymbol{\omega}})\mathbf{A} = \begin{bmatrix} 0 & \tilde{\eta}_3 & -\tilde{\eta}_2 \\ -\tilde{\eta}_3 & 0 & \tilde{\eta}_1 \\ \tilde{\eta}_2 & -\tilde{\eta}_1 & 0 \end{bmatrix} =: \mathbf{S}(\tilde{\boldsymbol{\eta}}), \quad (5.26)$$

where

$$\tilde{\boldsymbol{\eta}} := \begin{bmatrix} \tilde{\eta}_1 \\ \tilde{\eta}_2 \\ \tilde{\eta}_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \tilde{\omega}_1 \\ \tilde{\omega}_2 \\ \tilde{\omega}_3 \end{bmatrix} = -\mathbf{A}\mathbf{R}^T \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \quad (5.27)$$

We used (5.19) to obtain the last equality in (5.27).

In [128], it was shown that the system of differential equations in (5.25) can be transformed into two scalar differential equations by using the complex projection variable w as

$$\dot{w}_1 = w_2\tilde{\eta}_3 + w_1w_2\tilde{\eta}_2 + \frac{\tilde{\eta}_1}{2}(1 + w_1^2 - w_2^2), \quad (5.28)$$

$$\dot{w}_2 = -w_1\tilde{\eta}_3 + w_1w_2\tilde{\eta}_1 + \frac{\tilde{\eta}_2}{2}(1 - w_1^2 + w_2^2). \quad (5.29)$$

By substituting (5.27) into (5.28) and (5.29), the above differential equations can be rewritten as

$$\dot{w}_1 = \frac{1}{2}(1 + w_1^2 + w_2^2)(\omega_1 \cos(z) - \omega_2 \sin(z)), \quad (5.30)$$

$$\dot{w}_2 = \frac{1}{2}(1 + w_1^2 + w_2^2)(\omega_1 \sin(z) + \omega_2 \cos(z)). \quad (5.31)$$

The differential equation for the z -rotation is obtained by equating the first elements of both sides of (5.22) and using (5.30) and (5.31). The result is,

$$\dot{z} = \omega_3 + (\omega_2 w_1 - \omega_1 w_2) \cos(z) + (\omega_1 w_1 + \omega_2 w_2) \sin(z). \quad (5.32)$$

The new kinematic equations (5.30)–(5.32) describe the attitude of the body with respect to the target orientation.

5.2.3 Remarks

In its original form, the (z, w) -parameterization decomposes a rotation between the inertial and body-fixed reference frames into two successive rotations, namely, z - and w -rotations, and decouples the kinematic equation of w -rotation from that of z -rotation. In this work, we have modified the parameterization to explicitly include the target direction $\mathbf{t}_3$ by adding a new rotation $\mathbf{R}_1$. Because of this added rotation, the decoupling of w - and z -rotations is removed, but the resultant kinematic equations are yet simple enough to develop the desired spin-axis stabilization control law.

When we decompose $\mathbf{R}$, we take w -rotation before z -rotation, which is the reversed order of what is presented in [128]. This order of rotations is chosen because it results in the third row of $\mathbf{R}$, or the third column of $\tilde{\mathbf{R}}$, being independent of z as provided in (5.14), and allows us to derive the differential kinematic equations (5.25), (5.30) and (5.31), which have relatively simple forms.

The representation has a singularity when $a = b = 0$ and $c = -1$, originating from the stereographic projection (5.10). In this case, $\mathbf{e}_3 = -\mathbf{t}_3$ and the spin axis of the body is pointing exactly the opposite of the target direction. This singularity case may be overcome by first attempting to stabilize $\mathbf{e}_3$ about some direction $\mathbf{t}'_3 \neq \mathbf{t}_3$. Once this is done, the control can then stabilize $\mathbf{e}_3 = \mathbf{t}'_3$ to $\mathbf{t}_3$. We provide a simulation of this singular case in Section 5.4.3.

5.3 Dynamics of a Rigid Body with Two Reaction Wheels

Let the basis vectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be chosen to be coincident with the principal axes of the primary rigid body. We assume that there are two identical axis-symmetric reaction wheels attached to this rigid body, and their rotation axes are aligned with $\mathbf{e}_1$ and $\mathbf{e}_2$. It is well-known that the dynamics of such a system are given by (see, for example, [126], [129])

$$(I_1 - J_a)\dot{\omega}_1 = (I_2 - I_3)\omega_2\omega_3 + h_2\omega_3 - u_1, \quad (5.33)$$

$$(I_2 - J_a)\dot{\omega}_2 = (I_3 - I_1)\omega_3\omega_1 - h_1\omega_3 - u_2, \quad (5.34)$$

$$I_3\dot{\omega}_3 = (I_1 - I_2)\omega_1\omega_2 - h_2\omega_1 + h_1\omega_2, \quad (5.35)$$

$$\dot{h}_1 = -J_a \dot{\omega}_1 + u_1, \quad (5.36)$$

$$\dot{h}_2 = -J_a \dot{\omega}_2 + u_2. \quad (5.37)$$

In (5.33)–(5.37), I_i is the moment of inertia of the system including the rigid body and the reaction wheels, along $\mathbf{e}_i$ direction. Specifically, if I_i^b represents the moment of inertia of the rigid body in $\mathbf{e}_i$ direction, and J_a and J_t represent the axial and transverse moments of inertia of the wheels, respectively, then

$$I_{1,2} := I_{1,2}^b + J_a + J_t, \quad I_3 := I_3^b + 2J_t. \quad (5.38)$$

Notice that $I_i - J_a > 0$ for $i = 1, 2$ in (5.33) and (5.34). ω_j is the component of the angular velocity of the rigid body in $\mathbf{e}_j$ direction, observed in the body reference frame, as given in (5.16). If we let Ω_k ($k = 1, 2$) represent the relative angular speed of the k -th wheel with respect to the rigid body, then $h_k = J_a \Omega_k$ is the relative angular momentum of the k -th wheel. Finally, u_l ($l = 1, 2$) is the input control torque to the wheel l . As a reaction to this input, $-u_l$ is also applied to the primary body.

5.4 Controls for Spin-Axis Stabilization

5.4.1 Control Objective and Equilibrium

We want the spin axis of the primary body $\mathbf{e}_3$ to be aligned with the known target direction $\mathbf{t}_3$. From (5.4) and (5.5), we find that this goal is achieved when $\mathbf{R}_2(w) = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. As a consequence, we want to converge to the state where $a^* = b^* = 0$ and $c^* = 1$, or equivalently, $w_1^* = w_2^* = 0$, where the superscript $*$ is used to denote the converged values. At convergence, $\dot{w}_1^* = \dot{w}_2^* = 0$, which implies that $\omega_1^* = \omega_2^* = 0$, and thus $\dot{\omega}_1^* = \dot{\omega}_2^* = 0$, as can be seen from (5.30) and (5.31). Additionally, (5.32) and (5.35) say that $\dot{z}^* = \omega_3^*$ and $\dot{\omega}_3^* = 0$, that is, the primary body rotates about $\mathbf{e}_3 = \mathbf{t}_3$ at a constant angular speed of ω_3^* . Summarizing, for the purpose of achieving the spin-axis stabilization, our goal for control is to make partial states w_1 , w_2 , ω_1 and ω_2 converge to zero. When the system arrives at this state, (5.33), (5.34), (5.36) and (5.37) are reduced to

$$u_1^* = \dot{h}_1^* = h_2^* \omega_3^*, \quad (5.39)$$

$$u_2^* = \dot{h}_2^* = -h_1^* \omega_3^*. \quad (5.40)$$

From (5.39) and (5.40), it follows that the relative angular momenta of the wheels and the input control torques at convergence are sinusoidal functions.

5.4.2 Controller Design

We closely follow [126] for our controller design. Consider the following control law

$$u_1 := (I_2 - I_3)\omega_2\omega_3 + h_2\omega_3 + v_1, \quad (5.41)$$

$$u_2 := (I_3 - I_1)\omega_3\omega_1 - h_1\omega_3 + v_2, \quad (5.42)$$

where the synthetic control inputs v_1 and v_2 are defined as

$$v_1 := \kappa_1 [w_1 \cos(z) + w_2 \sin(z)] + \kappa_2 \omega_1, \quad (5.43)$$

$$v_2 := \kappa_1 [w_2 \cos(z) - w_1 \sin(z)] + \kappa_2 \omega_2, \quad (5.44)$$

with positive constant gains κ_1 and κ_2 . We now show that the above control law makes the system (5.30)–(5.37) to converge to the equilibrium state discussed in Section 5.4.1:

$$w_1^* = w_2^* = \omega_1^* = \omega_2^* = 0. \quad (5.45)$$

Proof. The proof closely follows [126]. Consider the following positive definite Lyapunov candidate function from [126]:

$$V := \frac{1}{2}(I_1 - J_a)\omega_1^2 + \frac{1}{2}(I_2 - J_a)\omega_2^2 + \kappa_1 \ln(1 + w_1^2 + w_2^2). \quad (5.46)$$

By taking time derivative of V and substituting (5.30), (5.31), (5.33) and (5.34) together with the control law (5.41)–(5.44), we obtain

$$\dot{V} = -\kappa_2(\omega_1^2 + \omega_2^2), \quad (5.47)$$

which is negative semi-definite. Since V is decrescent and radially unbounded, the equilibrium (5.45) is globally stable but it is not guaranteed to be asymptotically stable [130].

Thus, the asymptotic stability of the equilibrium state (5.45) is proved by LaSalle's invariance principle [130], [131]. In (5.47), $\dot{V} = 0$ implies that $\omega_1 = \omega_2 = 0$ and thus $\dot{\omega}_1 = \dot{\omega}_2 = 0$, provided that this corresponds to the equilibrium. By substituting this, together with (5.41) and (5.42), into (5.33) and (5.34), we obtain $v_1 = v_2 = 0$. From (5.43) and (5.44), we notice that this implies $w_1 = w_2 = 0$. Therefore, the only element in the invariance set satisfying $\dot{V} = 0$ is the equilibrium (5.45), and we conclude that this equilibrium is globally and asymptotically stable. $\square$

5.4.3 Simulations

The controller is simulated on the system consisting of a cubic rigid body and two reaction wheels attached to the body, as depicted in Figure 5.5. Such a system may represent a thruster system used for hopping of tensegrity robots. In that case, the cubic rigid body would be a package of thruster system components such as compressed propellant tanks, pressure valves, thruster nozzles, etc.

In Chapter 4, we have presented a tensegrity robot that can hop using a cold-gas thruster. The goals of this robot are to travel over a 1 km distance on the Moon's surface and to safely deliver a 1 kg payload. The total weight of this robot is limited to 10 kg, including the payload, structural components and thruster system, in order to reduce the mission cost. These requirements were specified by NASA for our research program and are intended to

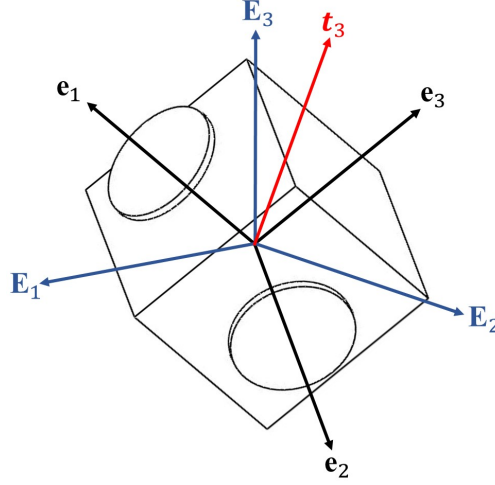


Figure 5.5: A cubic rigid body with two identical axis-symmetric reaction wheels. The rotation axes of the wheels are aligned with $\mathbf{e}_1$ and $\mathbf{e}_2$ [6].

facilitate low-cost surveillance missions launched as a secondary payload on a stationary lander.

For this robot, it is estimated that the cubic rigid body will have a width of 0.25 m and a total mass of 5 kg in order to carry sufficient amount of propellant required for the mission. Furthermore, we assume that the wheels are made of aluminum and each wheel has a radius of 0.077 m and thickness of 0.005 m such that its total mass is 0.25 kg. Under these assumptions, if the body and the wheels have a uniform mass distribution, $I_1 = I_2 = 0.053 \text{ kg m}^2$, $I_3 = 0.052 \text{ kg m}^2$ and $J_a = 0.00037 \text{ kg m}^2$.

Let us consider the case when the initial attitude of the rigid body is such that $\mathbf{e}_1 = \mathbf{E}_2$, $\mathbf{e}_2 = \mathbf{E}_3$ and $\mathbf{e}_3 = \mathbf{E}_1$. That is,

$$\mathbf{R}^0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad (5.48)$$

where the superscript 0 represents the initial condition. Furthermore, let the target thrust direction be given by $\mathbf{t}_3 = \frac{1}{\sqrt{2}}(\mathbf{E}_2 + \mathbf{E}_3)$, and therefore

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}. \quad (5.49)$$

If the $\mathbf{E}_1$ – $\mathbf{E}_2$ plane represents the ground surface, then $\mathbf{t}_3$ would make 45° angle with the ground, a desirable state for a parabolic hopping trajectory.

The initial attitude parameters $w^0 = w_1^0 + iw_2^0$ and z^0 can be obtained from

$$\mathbf{R}_3^0 \mathbf{R}_2^0 = \mathbf{R}_3(z^0) \mathbf{R}_2(w^0) = \mathbf{R}^0 \mathbf{R}_1^T. \quad (5.50)$$

From (5.8) and (5.12), notice that the third row of $\mathbf{R}_3(z^0)\mathbf{R}_2(w^0)$ is

$$[\mathbf{R}_3^0\mathbf{R}_2^0]_{(3,*)} = [-a^0 \quad -b^0 \quad c^0]. \quad (5.51)$$

Using (5.10),

$$w_1^0 = \frac{b^0}{1+c^0}, \quad w_2^0 = -\frac{a^0}{1+c^0}. \quad (5.52)$$

For our example, $a^0 = -1$ and $b^0 = c^0 = 0$, which results in $w_1^0 = 0$ and $w_2^0 = 1$. The parameter z^0 is obtained from

$$\tan(z^0) = \frac{[\mathbf{R}_3^0\mathbf{R}_2^0]_{(1,2)} - [\mathbf{R}_3^0\mathbf{R}_2^0]_{(2,1)}}{[\mathbf{R}_3^0\mathbf{R}_2^0]_{(1,1)} + [\mathbf{R}_3^0\mathbf{R}_2^0]_{(2,2)}}, \quad (5.53)$$

which gives $z^0 = 135^\circ$ in our case.

Lastly, we assume that the initial angular velocities of the body are $\omega_1^0 = 0.8$ rad/s, $\omega_2^0 = 0.5$ rad/s and $\omega_3^0 = -0.3$ rad/s, and that the initial relative angular velocities of the wheels with respect to the primary body are zero, i.e., $h_1^0 = h_2^0 = 0$. These conditions are chosen as an example motion of the hopping tensegrity robot.

The simulation results with these initial conditions and control gains of $\kappa_1 = 0.5$ and $\kappa_2 = 0.1$ are presented in Figures 5.6–5.10. In Figures 5.6 and 5.7, we see that the system arrives at the equilibrium (5.45) at around $t = 6$ s by the control action. Moreover, at this equilibrium, the body rotates about its spin axis at a constant angular speed of ω_3^* . Because of this, the angle z increases linearly, but this nonzero rotation rate around the spin axis does not affect the thrust direction if the nozzle axis is aligned with the body spin axis. Figure 5.8 shows that the absolute angle between $\mathbf{e}_3$ and $\mathbf{t}_3$ converges to zero, meaning that they are aligned together as desired at the equilibrium. As discussed in Section 5.4.1, the relative angular momenta of the wheels (h_1 and h_2) and the input control torques (u_1 and u_2) become sinusoidal functions of time at the equilibrium (Figures 5.9 and 5.10). On the other hand, both synthetic control inputs v_1 and v_2 converge to zero (Figure 5.10). Figure 5.9 shows that the total angular momentum of the system given by

$$H := \sqrt{(I_1\omega_1 + h_1)^2 + (I_2\omega_2 + h_2)^2 + (I_3\omega_3)^2}, \quad (5.54)$$

is conserved because no external torques were applied to the system.

In the above simulation, although the system started with a quite large angular displacement from the target direction and nonzero angular velocities, it quickly converged to the desired equilibrium. The required input torques are small, and thus the control can be implemented with low-torque and potentially lightweight actuators. This makes the control law particularly promising for hopping tensegrity robots where the total weight of the system is a critical factor.

An interesting case arises when both the body and the wheels are not rotating initially, that is, when the total angular momentum H of the system is initially zero. Because the

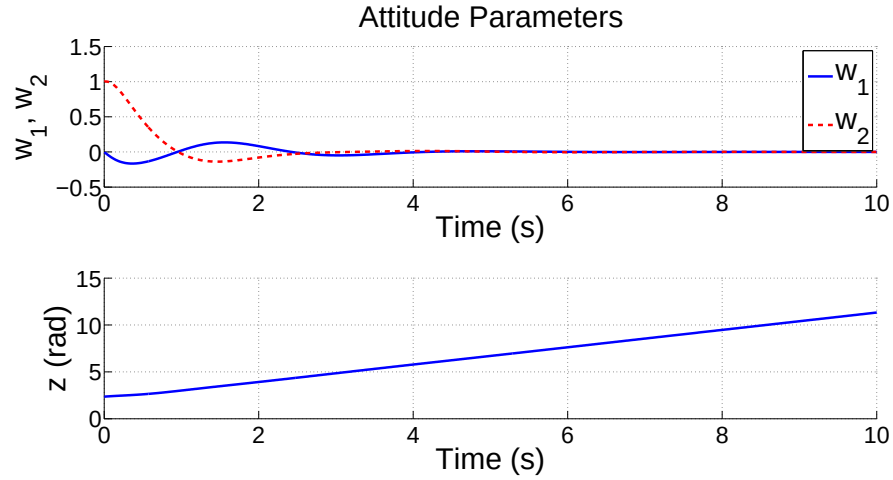


Figure 5.6: Change of attitude parameters over time under control action. w_1 and w_2 converge to zero while z increases linearly, which implies that the body rotates about its spin axis at the equilibrium [6].

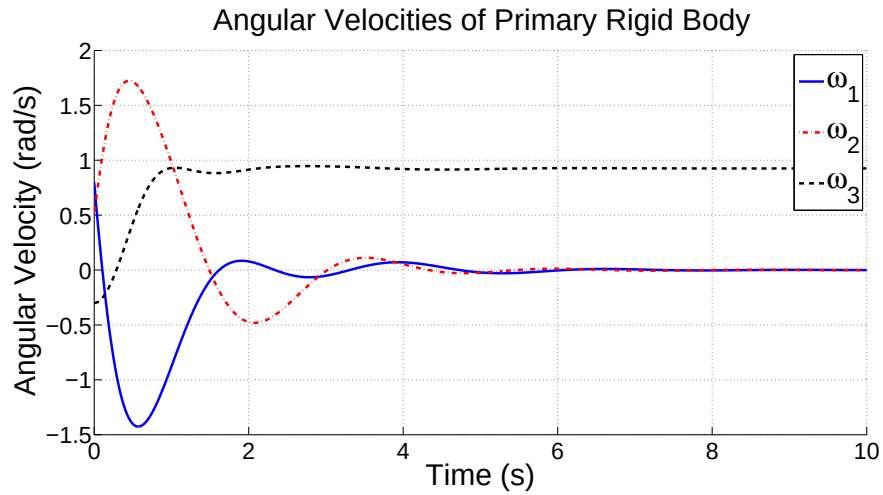


Figure 5.7: Change of angular velocities over time under control action. ω_1 and ω_2 converge to zero while ω_3 converges to a nonzero constant value. This implies that the body rotates about its spin axis at the equilibrium [6].

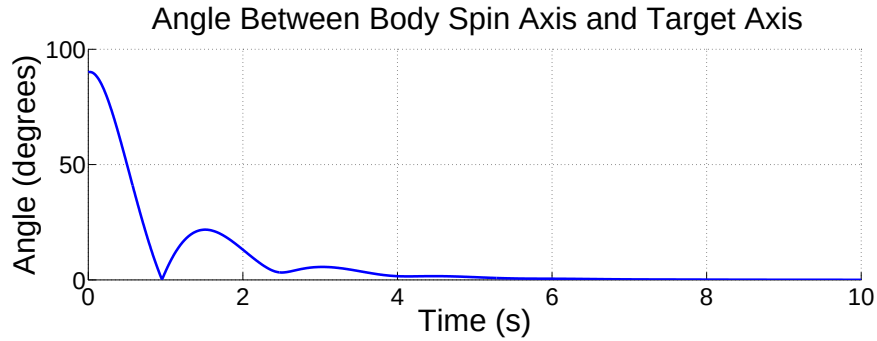


Figure 5.8: Change of absolute angle between the body spin axis $\mathbf{e}_3$ and target axis $\mathbf{t}_3$ over time. The angle converges to zero, meaning that $\mathbf{e}_3$ and $\mathbf{t}_3$ are aligned at the equilibrium [6].

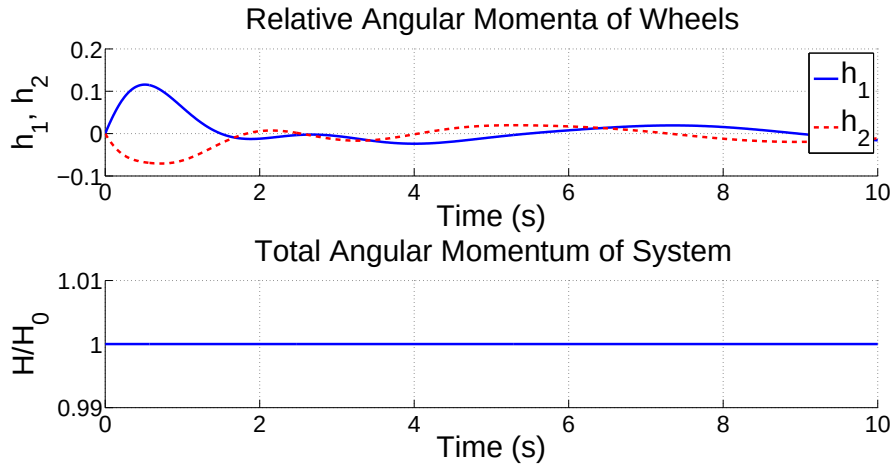


Figure 5.9: Change of relative angular momenta of the wheels with respect to the primary body over time. At the equilibrium, both h_1 and h_2 become sinusoidal functions of time. The total angular momentum of the system, H , is conserved [6].

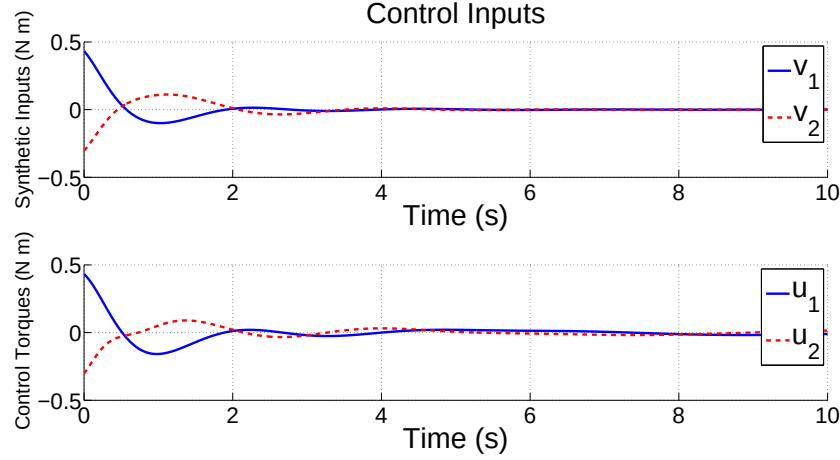


Figure 5.10: Change of control inputs over time. Both synthetic control inputs v_1 and v_2 converge to zero. The input control torques u_1 and u_2 become sinusoidal functions of time at the equilibrium [6].

total angular momentum is conserved, it should remain at zero when the equilibrium state is reached. As a result, $h_1^* = h_2^* = \omega_3^* = 0$, which implies that the body and the wheels become stationary at the equilibrium without spinning. This observation was also discussed in [120]. For a hopping tensegrity robot, this could be the case of reorienting the thruster nozzle in mid-air, assuming that the robot was launched in such a way that its angular velocities are zero and the reaction wheels do not rotate. We present a simulation of this case in the next.

Figures 5.11–5.15 show another simulation results, with the same physical parameters and control gains as in the previous simulation, where the singularity discussed in Section 5.2.3 arises. That is, the target axis is exactly at the opposite direction of the current body spin axis. Specifically, in this simulation, we assumed the principal axes of the primary body are initially aligned with the inertial axes ($\mathbf{e}_j = \mathbf{E}_j$, $j = 1, 2, 3$) and our target frame is given by $\mathbf{t}_1 = \mathbf{E}_1$, $\mathbf{t}_2 = -\mathbf{E}_2$ and $\mathbf{t}_3 = -\mathbf{E}_3$. This target frame is easily achieved by rotating the inertial frame by 180° about $\mathbf{E}_1$. However, our kinematics description results in $a^0 = b^0 = 0$ and $c^0 = -1$ and thus the target configuration is singular. To overcome this singularity, we take a two-step process as suggested in Section 5.2.3: 1) We first stabilize $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$ by rotating the body -90° about $\mathbf{E}_1$, and then 2) stabilize $\mathbf{e}_3 = \mathbf{t}'_3$ about $\mathbf{t}_3 = -\mathbf{E}_3$ by further rotating the body -90° about $\mathbf{E}_1$. Recall that we further assumed the initial angular velocities and angular momenta of the system are equal to zero in this simulation. As a consequence, the total angular momentum of the system remains zero (Figure 5.14), and the angular velocities and attitude parameters converge to zero at the equilibrium (Figures 5.11 and 5.12).

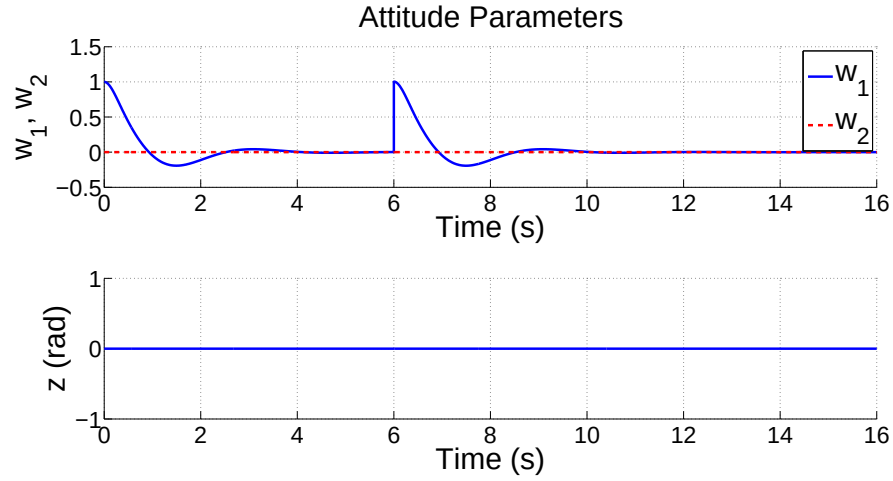


Figure 5.11: Change of attitude parameters over time under control action for the singular case. w_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. Unlike the first example, w_2 and z remain zero throughout. Because the initial angular momentum of the whole system is zero, the body and the wheels do not rotate at the equilibrium [6].

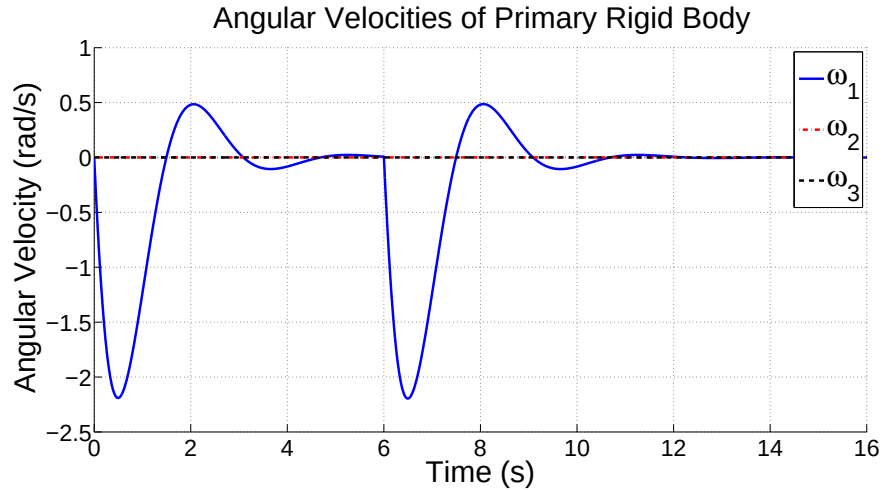


Figure 5.12: Change of angular velocities over time under control action for the singular case. ω_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. ω_2 and ω_3 remain zero throughout. Because the initial angular momentum of the whole system is zero, the body and the wheels do not rotate at the equilibrium [6].

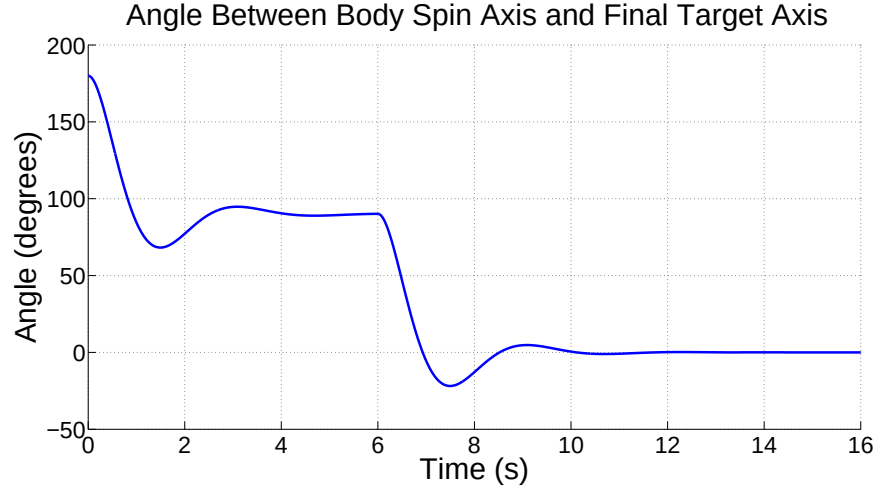


Figure 5.13: Change of angle between the body spin axis $\mathbf{e}_3$ and the final target axis $\mathbf{t}_3 = -\mathbf{E}_3$ over time for the singular case. The angle first converges to 90° , at around $t = 6$, as $\mathbf{e}_3$ stabilizes about $\mathbf{t}'_3 = \mathbf{E}_2$. Then it converges to zero at around $t = 12$, meaning that $\mathbf{e}_3$ and $\mathbf{t}_3$ are aligned at the equilibrium [6].

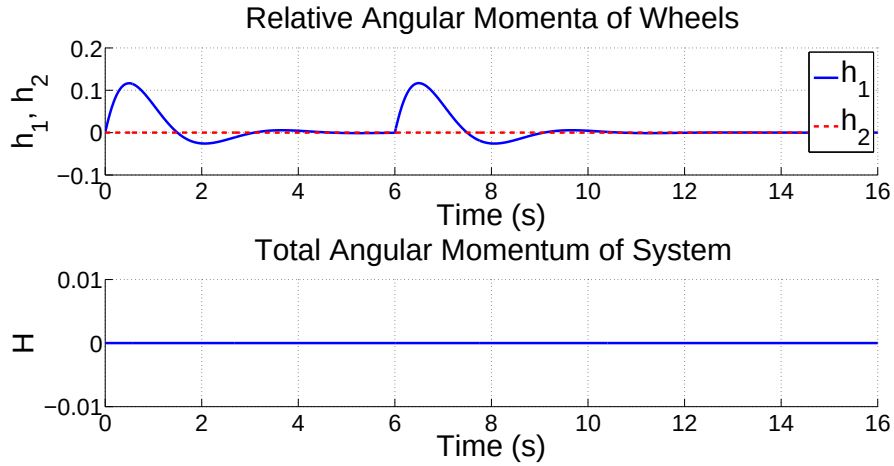


Figure 5.14: Change of relative angular momenta of the wheels with respect to the primary body over time for the singular case. h_1 first converges to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. h_2 remains zero throughout. The total angular momentum of the system, H , is conserved and is equal to zero [6].

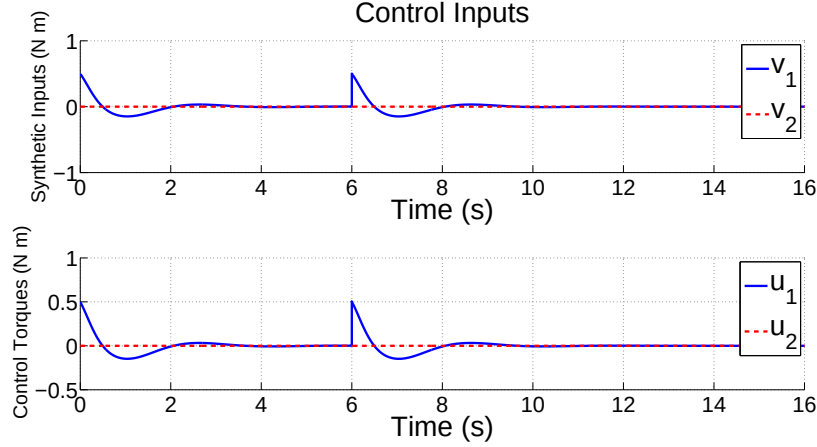


Figure 5.15: Change of control inputs over time for the singular case. Both v_1 and u_1 first converge to zero at around $t = 6$ and then again to zero at around $t = 12$. The first convergence corresponds to the stabilization of the body spin axis $\mathbf{e}_3$ about $\mathbf{t}'_3 = \mathbf{E}_2$, and the next one about $\mathbf{t}_3 = -\mathbf{E}_3$. v_2 and u_2 remain zero throughout [6].

5.5 Conclusions and Future Research

In this chapter, we presented several methods that can be used for thrust vectoring of a thruster-based tensegrity robot. It was pointed out that such thrust vectoring mechanisms are needed in order to prevent the robot from diverging away from its desired path when hopping. With the methods proposed in this chapter, the robot will be able to save its propellant and travel to the target location in an energy-efficient manner.

Among the proposed methods, we presented a detailed analysis on a reaction wheel based system and developed a control law achieving the spin-axis stabilization of a thruster system, modeled as a rigid body, about an arbitrary direction using two reaction wheels. Specifically, our control law successfully achieved two goals: 1) it oriented the spin axis of the body to an arbitrary direction, and 2) it made the body rotate about the spin axis only, if the angular momentum of the system is not zero. The control law was developed using a feedback linearization technique and it was shown that the control is globally and asymptotically stable about the equilibrium (5.45) by using Lyapunov's direct method in conjunction with LaSalle's invariance principle. In addition, the new set of differential kinematic equations that are based on the (z, w) -parameterization is derived. The new formulation explicitly considers the arbitrary target direction about which the spin axis of the body is stabilized, and describes the attitude of the body with respect to the target orientation.

We also demonstrated the stabilizing behavior of the controller by simulating it on a thruster system of hopping tensegrity robots. It was shown that our control law can quickly orient the thruster nozzle to a desired target direction using lightweight wheels and small control torques that can potentially be provided by lightweight actuators. We further simulated the strategy to overcome the singularity that arises from the stereographic projection (5.10).

The work in this chapter assumes that the perfect knowledge of the states is provided when developing and simulating the controller. In practice, however, this information may not be available and there may exist signal noises, model uncertainties, and external disturbances that challenge the controller. It is recommended that the future research will focus on the development of a robust controller that performs well even under these unfavorable conditions.

Chapter 6

Conclusions and Future Research

This research provided an extensive study on spherical tensegrity robots with an emphasis on their rolling and hopping locomotion. The work showed that tensegrity robots have prominent future robotic applications such as space exploration when provided with mobility. Tensegrity robots can choose their mode of locomotion depending on task requirements: 1) if the robot needs to precisely position itself, it can do so by rolling, or 2) if the robot needs to quickly travel long distances, hopping would be more useful.

Chapter 2 presented detailed hardware designs of four spherical tensegrity robots that are cable-driven, untethered and fully-actuated (except the current prototype of T12-R). The design space of tensegrity robots was not well-known at the beginning of this research. Therefore, the design of six-rod tensegrity based robots went through multiple design iterations and various materials were tested over the iterations. Modularity was a key concept in our hardware design such that the design of one spherical tensegrity robot can be easily modified and adapted to other forms of spherical tensegrity robots. This was demonstrated by showing that the design of T12-R was largely inherited from that of TT-3. While TT-1 and TT-2 include a controller unit as their center payloads, TT-3 and T12-R have a distributed controller scheme, clearing their center spaces for larger scientific payloads.

Rolling locomotion of spherical tensegrity robots was studied in details in Chapter 3. A spherical tensegrity robot rolls by deforming its outer shape and thus shifting its center of mass. To enable a step (a piecewise continuous motion between ground impacts), the ground projection of center of mass (GCoM) should leave the supporting polygon of the robot, and two methods were proposed to develop actuation policies that result in the deformation satisfying this condition. The first method uses a greedy search algorithm in the NASA Tensegrity Robotics Toolkit (NTRT) simulation with a heuristic function defined as the distance between GCoM and an edge of a supporting polygon serving as a rotation axis of the step to be performed. This method quickly found actuation policies for TT-1 and TT-2 but did not guarantee the reliability of steps in hardware experiments. The second method relies on a dynamic relaxation technique and a multi-generation Monte Carlo based algorithm. It was shown that the dynamic relaxation can quickly and effectively solve the form-finding problem of tensegrity structures and can find equilibrium configurations of our hardware

robots for given sets of actuation. On top of the dynamic relaxation, the multi-generation Monte Carlo was run to sample a number of equilibrium configurations of tensegrity robots and to find the best shapes that would result in reliable steps of the robots. The actuation policies obtained from both methods were tested on our hardware robots and their successful rolling was demonstrated. The detailed procedure of extending the latter method to other class 1 spherical tensegrity robots was also described by taking T12-R as an example.

In Chapter 4, tensegrity robots capable of hopping using a cold-gas thruster system were introduced and their specific mission goals were stated. Unlike other conventional robots, hopping is a viable mean of locomotion for spherical tensegrity robots as they can survive from significant impact shocks due to their natural compliance. A thermodynamic analysis was performed to estimate the deliverable thrusts on the Moon under the hardware constraint limiting the total weight of the thruster-based robot to be less than 10 kg. Examination of several hopping profiles showed that longer hops are more fuel-efficient but are subject to higher impulse at landing, which may lead to damaging the robots. Hence there is a trade-off between energy-efficiency and safety of the robots. Localization based on height measurements of surroundings and path planning based on the A-star algorithm on the lunar surface were also discussed.

In Chapter 5, it was observed from NTRT simulation that thrust vectoring is required for energy-efficient hopping of thruster-based tensegrity robots. For thrust vectoring, a system with two reaction wheels is proposed, in addition to the other three possible mechanisms: gimbal-enclosed, gimbaled-nozzle, and cable-actuation based systems. A controller for spin-axis stabilization of the thruster system about an arbitrary direction is designed based on the (z, w) -parameterization of rotation. It was also proved that the controller is globally and asymptotically stable by using the Lyapunov's direct method in conjunction with LaSalle's invariance principle. The analysis also included a discussion of singular cases.

Although this study covered a wide range of topics on the locomotion of spherical tensegrity robots, there still exists a rich set of research problems that need to be addressed in the future. This work would like to conclude with some suggestions on the future research directions.

6.1 Towards Dynamic Rolling of Spherical Tensegrity Robots

Perhaps the leading branch of robotic locomotion systems would be legged robots, especially bipeds. In this section, potential research directions on *dynamic* rolling of spherical tensegrity robots are suggested by adapting results from biped research. Although tensegrity robots and bipeds are seemingly different, the theory behind their locomotion may be shared because both systems aim to achieve a periodic gait.

To understand what the meaning of “dynamic” rolling is, consider the following definition on different types of gaits for bipeds, which is brought from [132]:

A *statically stable gait* is periodic locomotion in which the biped's center of mass (CoM) does not leave the support polygon, that is, the convex hull formed by all of the contact points with the ground. A *quasi-statically stable gait* is one where the center of pressure (CoP) of the biped's stance foot remains strictly within the interior of the support polygon, and hence does not lie on the boundary. Loosely speaking, a *dynamically stable gait* is then a periodic gait where the biped's CoP is on the boundary of the support polygon for at least part of the cycle and yet the biped does not overturn.

According to this definition of gaits, rolling of spherical tensegrity robots that this work presented is classified as static locomotion because the methods developed in Chapter 3 are concerned with the robots' centers of mass. However, there is a critical difference between bipeds and tensegrity robots in this definition; while bipeds try to keep their CoM inside of their feet for stable gaits, tensegrity robots try to do the opposite to unbalance themselves for movement. This difference arises because bipeds are naturally unbalanced system and they tend to fall over if not controlled properly, while tensegrity robots are naturally well-balanced system and they tend to remain at a stable pose when not actuated. Therefore, the control objectives for these systems are opposite to each other; control for bipeds aims to move the systems while not losing their balance and control for tensegrity robots aims to move the systems by losing their balance.

The reason why the definition of a quasi-statically stable gait relies on the CoP is because the CoP is closely related to the zero-moment point (ZMP), the concept of which is widely used in biped control [133]–[137]. By definition, the ZMP is a point on the ground where the tipping moment due to the gravity and inertia forces is zero, see Figure 6.1 [138]–[140]. The tipping moment is the moment that is perpendicular to a sagittal plane and tangential to a supporting surface. If the ZMP is located inside of a supporting polygon, that is, a foot sole or a convex hull formed by all ground contacting feet when multiple feet are in contact with the ground, then the tipping moment generated by the gravity and inertia forces is perfectly balanced out by the ground reaction force, and the point is equivalent to the CoP. If the ZMP does not satisfy the condition, then the tipping moment is nonzero and the system will start to tip over. The CoP remains on the boundary of the supporting polygon. The ZMP in this case is often referred to as an imaginary ZMP (IZMP) [137], a fictitious ZMP (FZMP) [140], or a foot-rotation indicator (FRI) point [141], but all of them are mathematically identical [137]. Hence the ZMP serves as the criterion of dynamic equilibrium for biped and other legged robot locomotion.

The main control objective for bipeds is then to maintain the ZMP within its supporting polygon during locomotion. As mentioned before, the control objective for tensegrity robots is opposite to this and is to shift the ZMP outside of the supporting polygon. In other words, the FZMP is desirable for tensegrity robots unlike other legged locomotion systems. The distance between the FZMP and the nearest point on the boundary of the supporting polygon can be interpreted as a measure of how reliable the rolling motion of tensegrity robots would be because the greater distance implies a larger tipping moment. This interpretation

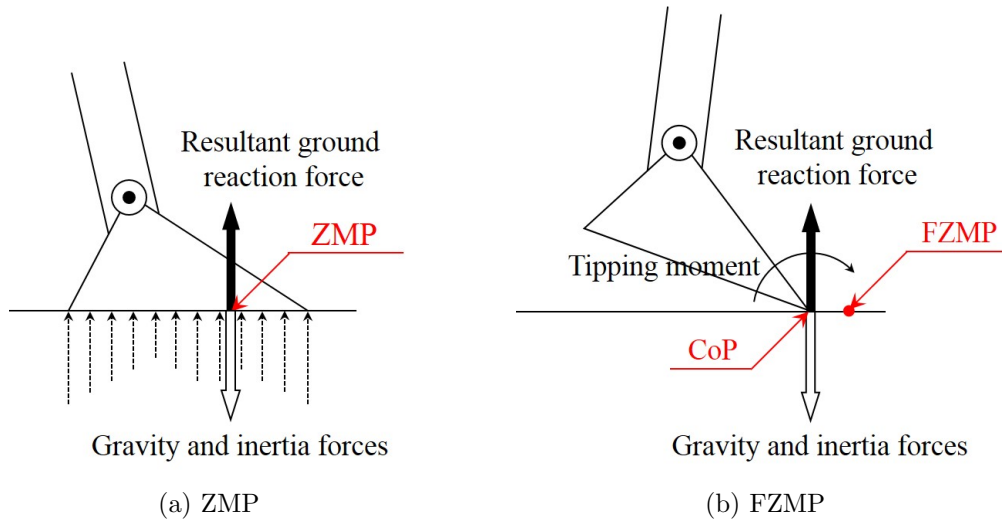


Figure 6.1: ZMP is a point where the tipping moment due to the gravity and inertia forces is zero. (a) ZMP is equivalent to CoP when it is located inside of a supporting polygon. The tipping moment in this case is zero as the moment generated by the gravity and inertia forces is balanced out by the ground reaction force. (b) If ZMP is located outside of a supporting polygon, then it is often referred to as FZMP, IZMP, or FRI point. The tipping moment in this case is not equal to zero as the ground reaction force cannot balance out the moment generated by the gravity and inertia forces.

is analogous to the discussion of the reliability of steps in Chapter 3. In biped research, the ZMP is usually measured by measuring the CoP using force sensors on soles because the ZMP stays within the supporting polygon, if not fallen over. However, this approach does not work for tensegrity robots as the ZMP is supposed to be outside of the supporting polygon. Recall that the CoP should remain on the boundary of the supporting polygon in this case. For tensegrity robots, this simply means that the ground reaction forces exerted on the leading rods that are pivoted during the rotation have positive normal components, while those exerted on the hind rods that are leaving the ground have zero normal components. This is so because tensegrity robots have only finite contact points with the ground and the ground reaction forces can only be applied to a subset of these points that remain in contact with the ground during the robots' motion. This is in contrast with a biped whose sole has an area or line contact with the ground (except when the biped is *tiptoed*, which is a very unusual situation, or the biped has point feet) and the ground reaction force is continuously distributed over the contact area or line. Many locomotion controllers were developed for bipeds based on the ZMP in previous research [142], and they may be modified or extended to meet the control objectives of tensegrity robots for their quasi-static locomotion. However, the gaits generated by pure ZMP-based controllers often do not look natural.

The definition of a dynamically stable gait of a biped requires that its ZMP leaves the supporting polygon for at least part of the gait cycle. This implies that the biped becomes

briefly unbalanced and may fall over if the situation lasts. Obviously, this is not desirable and the controller should bring the robot back to the balanced state. A popular approach for the development of such controllers is based on the design of *limit cycles* [132]. A limit cycle is an isolated periodic orbit of state trajectories of a nonlinear system [130]. If all of the state trajectories in the neighborhood of a limit cycle converge to it, then the limit cycle is said to be stable and the maximum neighborhood that has this property is called the limit cycle's basin of attraction. If all of the trajectories starting from points arbitrarily close to a limit cycle diverge from it, then the limit cycle is said to be unstable. For bipedal locomotion, the existence of a limit cycle corresponds to the existence of a dynamic walking gait. If the limit cycle is stable, then the gait is also stable, which means that even if the biped is disturbed and is instantaneously unbalanced within the limit cycle's basin of attraction, it can recover its stable gait. Similarly, dynamic locomotion of spherical tensegrity robots would be achieved if stable limit cycles are designed for them. For tensegrity robots, a limit cycle would correspond to a periodic state evolution during which their structures are unbalanced and the stability would mean that even if the robots are briefly well-balanced, they will return to the state trajectories where they become unbalanced again.

The design of controllers that give rise to limit cycles, however, is quite challenging, if not impossible. To reduce the complexity of the controller design problem, the method of Poincaré sections and Poincaré return maps is often exploited in the literature [143]–[145]. The method can also incorporate impulse effects that arise from legs of bipeds or rods of tensegrity robots hitting the ground [132]. Controllers for a dynamically stable gait of bipeds were developed based on this method in [132], [145], [146], and similar approaches maybe taken for the design of controllers achieving dynamic rolling of spherical tensegrity robots.

Another way of mitigating the difficulty in the controller design is to use simple models. For example, although bipeds are often mechanically complex and have many degrees of freedom, controllers that are developed based on variations of a simple inverted pendulum model are shown to be effective [147]–[149]. Many spherical tensegrity robots have some sort of symmetry in them, and the robot models may be simplified by exploiting the symmetry. This was already discussed in Chapter 3, where it was shown that the motion of T12-R can be described on a two-dimensional plane instead of a three-dimensional space as long as the robot maintains its symmetry about the sagittal plane. The simple dynamics model of T12-R will facilitate its controller design for dynamic rolling. In fact, this is another reason why T12-R is designed based on a rhombicuboctahedron twelve-rod tensegrity structure. The model of T12-R may further be simplified if additional assumptions are made. For instance, if the two pairs of parallel rods of the T12-R model on a sagittal plane are kept parallel and if the distances between parallel rods remain fixed, the robot can be approximated as a planar two-rod tensegrity system as illustrated in Figure 6.2.

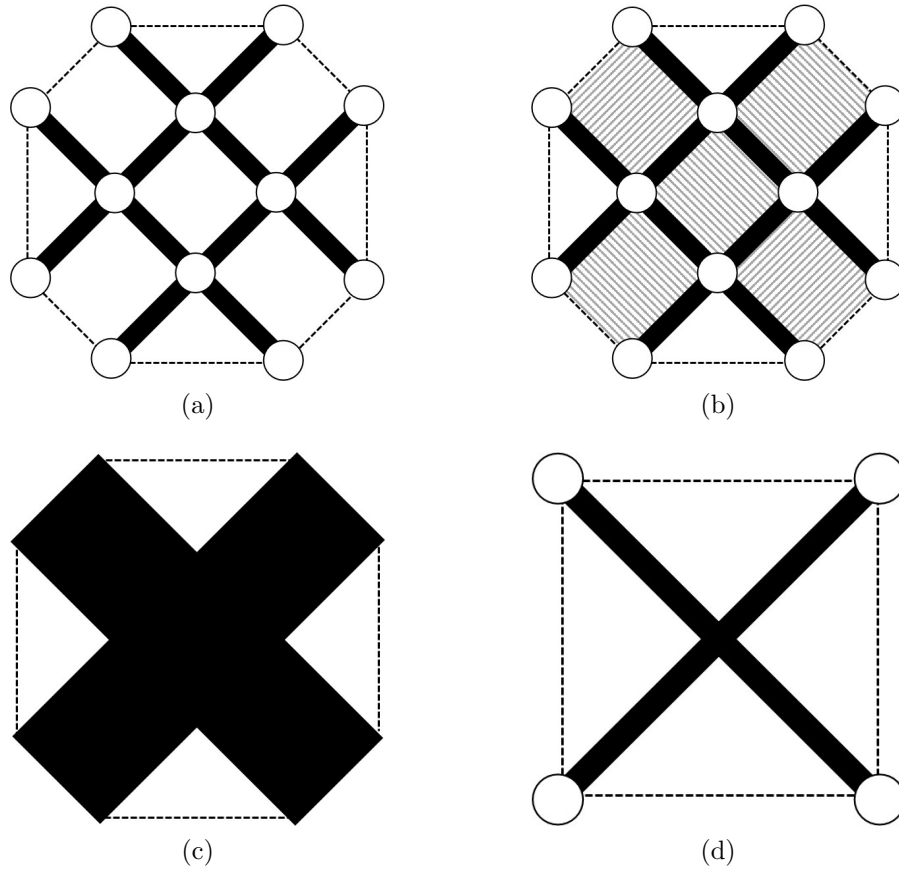


Figure 6.2: The procedure of further simplifying the two-dimensional model of T12-R. (a) The planar model of T12-R when the robot maintains its symmetry about the sagittal plane and the members are projected onto the plane. This model is the same as what is presented in Figure 2.9c. White circles are nodes, bold black lines are rods, and dashed lines are cables. (b) Two assumptions are made to pairs of parallel rods to further simplify the model. First, both pairs of parallel rods remain parallel during deformation. Next, the distances between both pairs of parallel rods remain constant. (c) Under these assumptions, each pair of parallel rods behave like a single rigid rod. (d) The system can be approximated as a planar two-rod tensegrity system. Controllers for dynamic rolling of T12-R may be developed based on the simplified model and tested on a full dynamic model to evaluate their performances.

6.2 Thrust Vectoring within Tensegrity Robots

In Chapter 5, a reaction wheel based thrust vectoring system was studied. However, the system was considered to be isolated and it was not placed inside of a tensegrity robot. It is anticipated that the dynamics of the reaction wheel system will become more complicated than what is presented in Section 5.3 when the system is connected to the outer tensegrity structure. This is because the reaction wheel system will interact with the tensegrity via the inner cables connecting the two, and the inner cable forces will affect the rotational motion of the reaction wheel system. To understand the interaction between the two systems happening through the inner cables, it will be necessary to complement the dynamics of the reaction wheel system with that of the outer tensegrity structure, which will likely result in the complicated dynamics of the coupled system.

There are two separate cases to consider for the thrust vectoring of the coupled system: 1) the robot is on the ground, and 2) the robot is in mid-air. In the first case, the rotation of the thruster system induced by the rotation of the reaction wheels may be hindered by the inner cable forces because the rotation of the outer tensegrity structure may be restricted by the ground reaction forces. As a result, the reaction wheel controller may face a saturation type of nonlinearity during its action. One possible way to minimize the saturation effect is to reposition the robot by punctuated rolling into a pose that minimizes the angle difference between the current thrust direction and the target thrust direction. That way, the amount of rotation of the thruster system induced by the reaction wheel control action would be decreased, and the saturation effect on the controller would be reduced. An initial work in this direction has been done in [118], [150].

For the latter case, it may be more practical to treat the inner cable forces as disturbance forces and improve the reaction wheel system controller presented in Chapter 5 in a way that it is robust to such disturbances, instead of updating the controller based on the explicit modeling of the inner cable forces. An initial study on the behavior of the thruster system inside of a simple and symmetric two-dimensional tensegrity system in mid-air shows that the thruster system exhibits sinusoidal type of angular motion relative to the outer tensegrity because the line of action of the thrust may not pass through the center of mass of the overall system due to the unbalanced inner cable forces, and thus the thrust generates a torque on the system [151]. While a further study is needed to verify this behavior of the thruster system in three-dimensional tensegrity robots, it is still reasonable to assume that the thruster system will show similar sinusoidal type of angular motion in this case because the inner cables would act like springs and their net force would be oscillating back and forth about some equilibrium, just like the spring in a simple spring-mass system, but in a more complicated way. The problem of sinusoidal disturbance rejection has been actively studied in the controls community [152]–[155], and the techniques developed in the literature may be extended and applied to our problem.

Bibliography

- [1] K. Kim, A. K. Agogino, D. Moon, L. Taneja, A. Toghyan, B. Dehghani, V. SunSpiral, and A. M. Agogino, “Rapid prototyping design and control of tensegrity soft robot for locomotion,” in *Proceedings of 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO2014)*, Best Student Paper Finalist, Bali, Indonesia, Dec. 2014.
- [2] K. Kim, A. K. Agogino, and A. M. Agogino, “Emergent form-finding for center of mass control of ball-shaped tensegrity robots,” in *Autonomous Robots and Multirobot Systems (ARMS) Workshop, 2015 International Conference on Autonomous Agents and Multiagent Systems*, ACM, 2015.
- [3] K. Kim, A. K. Agogino, A. Toghyan, D. Moon, L. Taneja, and A. M. Agogino, “Robust learning of tensegrity robot control for locomotion through form-finding,” in *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 5824–5831.
- [4] L.-H. Chen, K. Kim, E. Tang, K. Li, R. House, E. Jung, A. K. Agogino, V. SunSpiral, and A. M. Agogino, “Soft spherical tensegrity robot design using rod-centered actuation and control,” in *Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 40th Mechanisms and Robotics Conference*, American Society of Mechanical Engineers, 2016.
- [5] 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,” in *Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2016, pp. 4369–4376.
- [6] K. Kim and A. M. Agogino, “Spin-axis stabilization of a rigid body about an arbitrary direction using two reaction wheels,” in *Proceedings of the 55th IEEE Conference on Decision and Control*, IEEE, 2016.
- [7] R. B. Fuller, “Tensile-integrity structures,” pat. US 3063521A, Nov. 1962.
- [8] R. E. Skelton, R. Adhikari, J.-P. Pinaud, W. Chan, and J. W. Helton, “An introduction to the mechanics of tensegrity structures,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, IEEE, vol. 5, 2001, pp. 4254–4259.

- [9] S. H. Juan and J. M. M. Tur, "Tensegrity frameworks: Static analysis review," *Mechanism and Machine Theory*, vol. 43, no. 7, pp. 859–881, 2008.
- [10] K. Snelson, "The art of tensegrity," *International Journal of Space Structures*, vol. 27, no. 2&3, pp. 71–80, 2012.
- [11] ———, *Kenneth snelson; art and ideas*, http://kennethsnelson.net/KennethSnelson_Art_And_Ideas.pdf, [Online; accessed November 22, 2016], 2013.
- [12] A. Pugh, *An introduction to tensegrity*. Univ of California Press, 1976.
- [13] R. W. Burkhardt Jr, *A practical guide to tensegrity design*, http://www.angelfire.com/ma4/bob_wb/tenseg.pdf, [Online; accessed November 23, 2016], 2008.
- [14] R. Motro, *Tensegrity: Structural systems for the future*. Elsevier, 2003.
- [15] M. Arsenault and C. M. Gosselin, "Kinematic, static and dynamic analysis of a planar 2-DOF tensegrity mechanism," *Mechanism and Machine Theory*, vol. 41, no. 9, pp. 1072–1089, 2006.
- [16] B. N. R. Abadi, S. M. Shekarforoush, M. Mahzoon, and M. Farid, "Kinematic, stiffness, and dynamic analyses of a compliant tensegrity mechanism," *Journal of Mechanisms and Robotics*, vol. 6, no. 4, p. 041 001, 2014.
- [17] C. D. Crane, J. Duffy, and J. C. Correa, "Static analysis of tensegrity structures," *Journal of Mechanical Design*, vol. 127, no. 2, pp. 257–268, 2005.
- [18] J. Y. Zhang and M. Ohsaki, "Adaptive force density method for form-finding problem of tensegrity structures," *International Journal of Solids and Structures*, vol. 43, no. 18, pp. 5658–5673, 2006.
- [19] Y. Li, X.-Q. Feng, Y.-P. Cao, and H. Gao, "A monte carlo form-finding method for large scale regular and irregular tensegrity structures," *International Journal of Solids and Structures*, vol. 47, no. 14, pp. 1888–1898, 2010.
- [20] L. Zhang, B. Maurin, and R. Motro, "Form-finding of nonregular tensegrity systems," *Journal of Structural Engineering*, vol. 132, no. 9, pp. 1435–1440, 2006.
- [21] G. Fagerström, "Dynamic relaxation of tensegrity structures," in *Proceedings of the 14th International Conference on Computer Aided Architectural Design Research in Asia/Yunlin (Taiwan)*, vol. 22, 2009, pp. 553–562.
- [22] C. Sultan, "Modeling, design, and control of tensegrity structures with applications," PhD thesis, Purdue University, 1999.
- [23] E. Fest, K. Shea, and I. F. Smith, "Active tensegrity structure," *Journal of Structural Engineering*, vol. 130, no. 10, pp. 1454–1465, 2004.
- [24] R. E. Skelton and C. Sultan, "Controllable tensegrity: A new class of smart structures," in *Smart Structures and Materials' 97*, International Society for Optics and Photonics, 1997, pp. 166–177.

- [25] E. Fest, K. Shea, B. Domer, and I. F. Smith, "Adjustable tensegrity structures," *Journal of structural engineering*, vol. 129, no. 4, pp. 515–526, 2003.
- [26] J. B. Aldrich, R. E. Skelton, and K. Kreutz-Delgado, "Control synthesis for a class of light and agile robotic tensegrity structures," in *Proceedings of the 2003 American Control Conference*, IEEE, vol. 6, 2003, pp. 5245–5251.
- [27] N. B. H. Ali and I. Smith, "Dynamic behavior and vibration control of a tensegrity structure," *International Journal of Solids and Structures*, vol. 47, no. 9, pp. 1285–1296, 2010.
- [28] G. Tibert, "Deployable tensegrity structures for space applications," PhD thesis, Royal Institute of Technology, 2002.
- [29] S. Djouadi, R. Motro, J. C. Pons, and B. Crosnier, "Active control of tensegrity systems," *Journal of Aerospace Engineering*, vol. 11, no. 2, pp. 37–44, 1998.
- [30] C. Sultan and R. Skelton, "Deployment of tensegrity structures," *International Journal of Solids and Structures*, vol. 40, no. 18, pp. 4637–4657, 2003.
- [31] A. G. Tibert and S. Pellegrino, "Deployable tensegrity reflectors for small satellites," *Journal of Spacecraft and Rockets*, vol. 39, no. 5, pp. 701–709, 2002.
- [32] C. Sultan, M. Corless, and R. E. Skelton, "Peak-to-peak control of an adaptive tensegrity space telescope," in *1999 Symposium on Smart Structures and Materials*, International Society for Optics and Photonics, 1999, pp. 190–201.
- [33] A. G. Tibert and S. Pellegrino, "Deployable tensegrity masts," in *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, 2003.
- [34] M. Cefalo and J. M. Mirats-Tur, "A comprehensive dynamic model for class-1 tensegrity systems based on quaternions," *International Journal of Solids and Structures*, vol. 48, no. 5, pp. 785–802, 2011.
- [35] J. Cheong, R. E. Skelton, and Y. Cho, "A numerical algorithm for tensegrity dynamics with non-minimal coordinates," *Mechanics Research Communications*, vol. 58, pp. 46–52, 2014.
- [36] R. E. Skelton, "Dynamics and control of tensegrity systems," in *IUTAM Symposium on Vibration Control of Nonlinear Mechanisms and Structures*, Springer, 2005, pp. 309–318.
- [37] —, "Dynamics of tensegrity systems: Compact forms," in *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 2276–2281.
- [38] R. E. Skelton, J. P. Pinaud, and D. Mingori, "Dynamics of the shell class of tensegrity structures," *Journal of the Franklin Institute*, vol. 338, no. 2, pp. 255–320, 2001.
- [39] C. Sultan, M. Corless, and R. E. Skelton, "Linear dynamics of tensegrity structures," *Engineering Structures*, vol. 24, no. 6, pp. 671–685, 2002.

- [40] N. Kanchanasaratool and D. Williamson, "Modelling and control of class nsp tensegrity structures," *International Journal of Control*, vol. 75, no. 2, pp. 123–139, 2002.
- [41] A. S. Wroldsen, M. C. De Oliveira, and R. E. Skelton, "Modelling and control of non-minimal non-linear realisations of tensegrity systems," *International Journal of Control*, vol. 82, no. 3, pp. 389–407, 2009.
- [42] K. Nagase and R. E. Skelton, "Network and vector forms of tensegrity system dynamics," *Mechanics Research Communications*, vol. 59, pp. 14–25, 2014.
- [43] S. Faroughi, H. H. Khodaparast, and M. I. Friswell, "Non-linear dynamic analysis of tensegrity structures using a co-rotational method," *International Journal of Non-Linear Mechanics*, vol. 69, pp. 55–65, 2015.
- [44] M. Masic and R. E. Skelton, "Open-loop control of class-2 tensegrity towers," in *Smart Structures and Materials*, International Society for Optics and Photonics, 2004, pp. 298–308.
- [45] J. Michielsen, R. H. B. Fey, and H. Nijmeijer, "Steady-state dynamics of a 3d tensegrity structure: Simulations and experiments," *International Journal of Solids and Structures*, vol. 49, no. 7, pp. 973–988, 2012.
- [46] J. M. M. Tur and S. H. Juan, "Tensegrity frameworks: Dynamic analysis review and open problems," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 1–18, 2009.
- [47] C. Sultan, "Tensegrity: 60 years of art, science, and engineering," *Advances in applied mechanics*, vol. 43, pp. 69–145, 2009.
- [48] C. Paul, J. W. Roberts, H. Lipson, and F. J. V. Cuevas, "Gait production in a tensegrity based robot," in *Proceedings of the 12th International Conference on Advanced Robotics*, IEEE, 2005, pp. 216–222.
- [49] C. Paul, F. J. Valero-Cuevas, and H. Lipson, "Design and control of tensegrity robots for locomotion," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 944–957, 2006.
- [50] A. P. Sabelhaus, J. Bruce, K. Caluwaerts, P. Manovi, R. F. Firoozi, S. Dobi, A. M. Agogino, and V. SunSpiral, "System design and locomotion of SUPERball, an untethered tensegrity robot," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation*, IEEE, 2015, pp. 2867–2873.
- [51] J. Bruce, K. Caluwaerts, A. Iscen, A. P. Sabelhaus, and V. SunSpiral, "Design and evolution of a modular tensegrity robot platform," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, 2014.
- [52] M. Shibata and S. Hirai, "Rolling locomotion of deformable tensegrity structure," *Mobile Robotics: Solutions and Challenges (CLAWAR09)*, pp. 479–486, 2009.

- [53] A. P. Sabelhaus, J. Bruce, K. Caluwaerts, Y. Chen, D. Lu, Y. Liu, A. K. Agogino, V. SunSpiral, and A. M. Agogino, "Hardware design and testing of SUPERball, a modular tensegrity robot," in *Proceedings of The 6th World Conference of the International Association for Structural Control and Monitoring (6WCSCM)*, Barcelona, Spain, Jul. 2014.
- [54] J. Bruce, A. P. Sabelhaus, Y. Chen, D. Lu, K. Morse, S. Milam, K. Caluwaerts, A. M. Agogino, and V. SunSpiral, "SUPERball: Exploring tensegrities for planetary probes," in *Proceedings of 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014)*, Montreal, Canada, Jun. 2014.
- [55] M. Shibata, F. Saijyo, and S. Hirai, "Crawling by body deformation of tensegrity structure robots," in *Proceedings of 2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 4375–4380.
- [56] Y. Koizumi, M. Shibata, and S. Hirai, "Rolling tensegrity driven by pneumatic soft actuators," in *Proceedings of 2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 1988–1993.
- [57] K. Caluwaerts, "Design and computational aspects of compliant tensegrity robots," PhD thesis, Ghent University, 2014.
- [58] 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," *Journal of The Royal Society Interface*, vol. 11, no. 98, 2014.
- [59] A. K. Agogino, V. SunSpiral, and D. Atkinson, "Super Ball Bot – structures for planetary landing and exploration," *NASA Innovative Advanced Concepts (NIAC) Program, Phase 1, Final Report*, Jul. 2013.
- [60] V. SunSpiral, A. K. Agogino, and D. Atkinson, "Super Ball Bot – structures for planetary landing and exploration," *NASA Innovative Advanced Concepts (NIAC) Program, Phase 2, Final Report*, Sep. 2015.
- [61] S. Hirai, Y. Koizumi, M. Shibata, M. Wang, and L. Bin, "Active shaping of a tensegrity robot via pre-pressure," in *Proceedings of 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, 2013, pp. 19–25.
- [62] M. Khazanov, J. Jocque, and J. Rieffel, "Evolution of locomotion on a physical tensegrity robot," in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, 2014, pp. 232–238.
- [63] W. Du, S. Ma, B. Li, M. Wang, and S. Hirai, "Dynamic simulation for 6-strut tensegrity robots," in *Proceedings of 2014 IEEE International Conference on Information and Automation*, IEEE, 2014, pp. 870–875.
- [64] S. Hirai and R. Imuta, "Dynamic simulation of six-strut tensegrity robot rolling," in *Proceedings of 2012 IEEE International Conference on Robotics and Biomimetics*, IEEE, 2012, pp. 198–204.

- [65] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer, “Controlling tensegrity robots through evolution,” in *Proceedings of the 15th annual conference on genetic and evolutionary computation*, ACM, 2013, pp. 1293–1300.
- [66] —, “Robust distributed control of rolling tensegrity robot,” in *The Autonomous Robots and Multirobot Systems (ARMS) Workshop at AAMAS*, vol. 2013, 2013.
- [67] —, “Flop and roll: Learning robust goal-directed locomotion for a tensegrity robot,” in *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 2236–2243.
- [68] B. T. Mirletz, R. D. Quinn, and V. SunSpiral, “CPGs for adaptive control of spine-like tensegrity structures,” in *Proceedings of 2015 International Conference on Robotics and Automation (ICRA2015) Workshop on Central Pattern Generators for Locomotion Control: Pros, Cons & Alternatives*, 2015.
- [69] B. T. Mirletz, I.-W. Park, T. E. Flemons, A. K. Agogino, R. D. Quinn, and V. SunSpiral, “Design and control of modular spine-like tensegrity structures,” in *Proceedings of The 6th World Conference of the International Association for Structural Control and Monitoring (6WCSCM)*, 2014.
- [70] B. R. Tietz, R. W. Carnahan, R. J. Bachmann, R. D. Quinn, and V. SunSpiral, “Tetraspine: Robust terrain handling on a tensegrity robot using central pattern generators,” in *Proceedings of 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, 2013, pp. 261–267.
- [71] B. T. Mirletz, P. Bhandal, R. D. Adams, A. K. Agogino, R. D. Quinn, and V. SunSpiral, “Goal-directed CPG-based control for tensegrity spines with many degrees of freedom traversing irregular terrain,” *Soft Robotics*, vol. 2, no. 4, pp. 165–176, 2015.
- [72] A. P. Sabelhaus, H. Ji, P. Hylton, Y. Madaan, C. Yang, A. M. Agogino, J. Friesen, and V. SunSpiral, “Mechanism design and simulation of the ULTRA Spine: A tensegrity robot,” in *Proceedings of ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2015, V05AT08A059–V05AT08A059.
- [73] D. Hustig-Schultz, V. SunSpiral, and M. Teodorescu, “Morphological design for controlled tensegrity quadruped locomotion,” in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2016)*, IEEE, 2016.
- [74] J. Friesen, A. Pogue, T. Bewley, M. de Oliveira, R. Skelton, and V. SunSpiral, “DuCTT: A tensegrity robot for exploring duct systems,” in *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, IEEE, 2014, pp. 4222–4228.
- [75] J. M. Friesen, P. Glick, M. Fanton, P. Manovi, A. Xydes, T. Bewley, and V. SunSpiral, “The second generation prototype of a duct climbing tensegrity robot, DuCTTv2,” in *Proceedings of 2016 IEEE International Conference on Robotics and Automation*, IEEE, 2016, pp. 2123–2128.

- [76] T. Flemons, *Intension designs biotensegrity models*, <http://www.intensiondesigns.com/>, [Online; accessed November 24, 2016].
- [77] G. Scarr, "A consideration of the elbow as a tensegrity structure," *International Journal of Osteopathic Medicine*, vol. 15, no. 2, pp. 53–65, 2012.
- [78] S. Lessard, D. Castro, W. Asper, S. Chopra, L. B. Baltaxe-Admony, V. SunSpiral, M. Teodorescu, and A. K. Agogino, "A bio-inspired tensegrity manipulator with multi-DOF, structurally compliant joints," in *Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2016.
- [79] L. B. Baltaxe-Admony, A. Robbins, E. Jung, S. Lessard, M. Teodorescu, V. SunSpiral, and A. K. Agogino, "Simulating the human shoulder through active tensegrity structures," in *Proceedings of ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, 2016.
- [80] K. Caluwaerts, J. M. Friesen, J. Bruce, and V. SunSpiral, "State estimation for tensegrity robots," in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA2016)*, IEEE, 2016.
- [81] L.-H. Chen, P. Keegan, M. Yuen, A. M. Agogino, R. K. Kramer, A. K. Agogino, and V. SunSpiral, "Soft robots using compliant tensegrity structures and soft sensors," in *Soft Robotics Workshop, IEEE International Conference on Robotics and Automation (ICRA 2015)*, abstract, IEEE, 2015.
- [82] L.-H. Chen, "Soft spherical tensegrity robot design using rod-centered actuation and control," PhD thesis, University of California, Berkeley, 2016.
- [83] M. C. Yuen and R. K. Kramer, "Fabricating microchannels in elastomer substrates for stretchable electronics," in *Proceedings of ASME 2016 11th International Manufacturing Science and Engineering Conference*, American Society of Mechanical Engineers, 2016, V002T01A014–V002T01A014.
- [84] C. Sultan and R. E. Skelton, "Force and torque smart tensegrity sensor," in *5th Annual International Symposium on Smart Structures and Materials*, International Society for Optics and Photonics, 1998, pp. 357–368.
- [85] C. Sultan and R. Skelton, "A force and torque tensegrity sensor," *Sensors and Actuators A: Physical*, vol. 112, no. 2, pp. 220–231, 2004.
- [86] A. G. Rovira and J. M. M. Tur, "Control and simulation of a tensegrity-based mobile robot," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 526–535, 2009.
- [87] J. M. M. Tur, "On the movement of tensegrity structures," *International Journal of Space Structures*, vol. 25, no. 1, pp. 1–13, 2010.
- [88] *NASA Tensegrity Robotics Toolkit*, <http://irg.arc.nasa.gov/tensegrity/NTRT/>, [Online; accessed December 3, 2016].
- [89] B. Jessen, "Orthogonal icosahedra," *Nordisk Matematisk Tidskrift*, pp. 90–96, 1967.

- [90] V. Boehm, A. Jentzsch, T. Kaufhold, F. Schneider, F. Becker, and K. Zimmermann, "An approach to locomotion systems based on 3d tensegrity structures with a minimal number of struts," in *Proceedings of the 7th German Conference on Robotics (ROBOTIK 2012)*, VDE, 2012, pp. 1–6.
- [91] V. Böhm, T. Kaufhold, F. Schale, and K. Zimmermann, "Spherical mobile robot based on a tensegrity structure with curved compressed members," in *Proceedings of 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2016, pp. 1509–1514.
- [92] J. L. Ware, "Spherical tensegrity soft robots for NASA missions: IMU integration for orientation detection and system drop response," Master's thesis, University of California, Berkeley, May 2016.
- [93] V. SunSpiral, G. Gorospe, J. Bruce, A. Iscen, G. Korbelt, S. Milam, A. Agogino, and D. Atkinson, "Tensegrity based probes for planetary exploration: Entry, descent and landing (EDL) and surface mobility analysis," *International Journal of Planetary Probes*, 2013.
- [94] J. M. Friesen, A. Pogue, T. Bewley, M. de Oliveira, R. E. Skelton, and V. SunSpiral, "A compliant tensegrity robot for exploring duct systems," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, 2014.
- [95] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer, "Learning to control complex tensegrity robots," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1193–1194.
- [96] M. Khazanov, B. Humphreys, W. Keat, and J. Rieffel, "Exploiting dynamical complexity in a physical tensegrity robot to achieve locomotion," in *Advances in Artificial Life, ECAL*, vol. 12, 2013, pp. 965–972.
- [97] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: A modern approach*. Prentice hall Upper Saddle River, 2003, vol. 2.
- [98] S. Lessard and A. K. Agogino, "Multi-level monte carlo control algorithms to maneuver tensegrity robots out of obstacles," in *Proceedings of AAMAS 2015*, AAMAS, 2015.
- [99] H.-J. Schek, "The force density method for form finding and computation of general networks," *Computer methods in applied mechanics and engineering*, vol. 3, no. 1, pp. 115–134, 1974.
- [100] M. Barnes, "Form-finding and analysis of prestressed nets and membranes," *Computers & Structures*, vol. 30, no. 3, pp. 685–695, 1988.
- [101] M. R. Barnes, "Form finding and analysis of tension structures by dynamic relaxation," *International journal of space structures*, vol. 14, no. 2, pp. 89–104, 1999.

- [102] A. Tibert and S. Pellegrino, "Review of form-finding methods for tensegrity structures," *International Journal of Space Structures*, vol. 26, no. 3, pp. 241–255, 2011.
- [103] R. E. Skelton and M. C. de Oliveira, *Tensegrity systems*. Springer, 2009.
- [104] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, R. Gellert, J. B. Gilbert, M. Golombek, J. Gómez-Elvira, D. M. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. C. Malin, I. Mitrofanov, J. J. Simmonds, D. Vaniman, R. V. Welch, and R. C. Wiens, "Mars science laboratory mission and science investigation," *Space Science Reviews*, vol. 170, no. 1, pp. 5–56, 2012.
- [105] R. A. Lindemann and C. J. Voorhees, "Mars exploration rover mobility assembly design, test and performance," in *Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 450–455.
- [106] E. Ackerman, "Boston dynamics sand flea robot demonstrates astonishing jumping skills," *IEEE Spectrum Robotics Blog*, vol. 2, no. 1, 2012.
- [107] B. Fuller, "Tensegrity," *Portfolio Artnews Annual*, vol. 4, pp. 112–127, 1961.
- [108] S. M. Levin, "The tensegrity-truss as a model for spine mechanics: Biotensegrity," *Journal of mechanics in medicine and biology*, vol. 2, no. 03&04, pp. 375–388, 2002.
- [109] K. D. Snelson, "Continuous tension, discontinuous compression structures," pat. US 3,169,611, Feb. 1965.
- [110] A. Tibert and S. Pellegrino, "Review of form-finding methods for tensegrity structures," *International Journal of Space Structures*, vol. 18, no. 4, pp. 209–223, 2003.
- [111] J. M. Mirats Tur and S. H. Juan, "Tensegrity frameworks: Dynamic analysis review and open problems," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 1–18, 2009.
- [112] A. Deng, G. Paulson, J. Saltz, and A. Savinov, "Analysis of impact absorption of a tensegrity structure," *Internal Report, Berkeley Emergent Space Tensegrities lab, UC Berkeley*, May 2013.
- [113] J. Mueller, "Thruster options for microspacecraft: A review and evaluation of existing hardware and emerging technologies," *AIAA paper*, vol. 3058, p. 1997, 1997.
- [114] S. L. Nothnagel, "Development of a cold gas propulsion system for the talaris hopper," PhD thesis, Massachusetts Institute of Technology, 2011.
- [115] J. D. Anderson Jr, *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.
- [116] J. J. Sellers, W. J. Astore, R. B. Giffen, and W. J. Larson, *Understanding space: An introduction to astronautics*. Primis, 2000.
- [117] J. Despois, "Localization, path planning and controls for spherical soft tensegrity robots in the context of nasa space missions," Master's thesis, University of California, Berkeley, May 2016.

- [118] E. L. Zhu, “TBD,” in press, Master’s thesis, University of California, Berkeley.
- [119] P. E. Crouch, “Spacecraft attitude control and stabilization: Applications of geometric control theory to rigid body models,” *Automatic Control, IEEE Transactions on*, vol. 29, no. 4, pp. 321–331, 1984.
- [120] H. Krishnan, H. McClamroch, and M. Reyhan, “On the attitude stabilization of a rigid spacecraft using two control torques,” in *Proceedings of 1992 American Control Conference*, IEEE, 1992, pp. 1990–1995.
- [121] C. I. Byrnes and A. Isidori, “On the attitude stabilization of rigid spacecraft,” *Automatica*, vol. 27, no. 1, pp. 87–95, 1991.
- [122] P. Morin and C. Samson, “Time-varying exponential stabilization of a rigid spacecraft with two control torques,” *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 528–534, 1997.
- [123] T. Urakubo, K. Tsuchiya, and K. Tsujita, “Attitude control of a spacecraft with two reaction wheels,” *Journal of Vibration and Control*, vol. 10, no. 9, pp. 1291–1311, 2004.
- [124] Y. Katsuyama, K. Sekiguchi, and M. Sampei, “Spacecraft attitude control by 2 wheels with initial angular momentum,” in *Proceedings of 2013 SICE Annual Conference*, IEEE, 2013, pp. 1890–1895.
- [125] Y. Katsuyama, T. Ibuki, K. Sekiguchi, and M. Sampei, “Attitude controllability analysis of an underactuated satellite with two reaction wheels and its control,” in *Proceedings of 2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 3421–3426.
- [126] S. Kim and Y. Kim, “Spin-axis stabilization of a rigid spacecraft using two reaction wheels,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 5, pp. 1046–1049, 2001.
- [127] P. Tsiotras and J. M. Longuski, “Spin-axis stabilization of symmetric spacecraft with two control torques,” *Systems & Control Letters*, vol. 23, no. 6, pp. 395–402, 1994.
- [128] ———, “A new parameterization of the attitude kinematics,” *Journal of the Astronautical Sciences*, vol. 43, no. 3, pp. 243–262, 1995.
- [129] J. L. Junkins and J. D. Turner, *Optimal spacecraft rotational maneuvers*. Elsevier, 1986.
- [130] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.
- [131] J. LaSalle, “Stability theory for ordinary differential equations,” *Journal of Differential Equations*, vol. 4, no. 1, pp. 57–65, 1968.
- [132] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007, vol. 28.

- [133] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of 2003 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 2003, pp. 1620–1626.
- [134] Y. Fujimoto and A. Kawamura, "Proposal of biped walking control based on robust hybrid position/force control," in *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, IEEE, 1996, pp. 2724–2730.
- [135] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1998, pp. 1321–1326.
- [136] P. Sardain and G. Bessonnet, "Zero moment point-measurements from a human walker wearing robot feet as shoes," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 5, pp. 638–648, 2004.
- [137] M. Vukobratović, B. Borovac, and D. Šurdilović, "Zero-moment point – proper interpretation and new applications," in *Proceedings of The Second IEEE-RAS International Conference on Humanoid Robots, CD-ROM*, 2001.
- [138] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure – zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.
- [139] M. Vukobratovic, A. Frank, and D. Juricic, "On the stability of biped locomotion," *IEEE Transactions on Biomedical Engineering*, no. 1, pp. 25–36, 1970.
- [140] M. Vukobratović and B. Borovac, "Zero-moment point – thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [141] A. Goswami, "Postural stability of biped robots and the foot-rotation indicator (FRI) point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.
- [142] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to humanoid robotics*. Springer, 2014, vol. 101.
- [143] D. E. Koditschek and M. Buehler, "Analysis of a simplified hopping robot," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 587–605, 1991.
- [144] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles and their stability in a passive bipedal gait," in *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, IEEE, vol. 1, 1996, pp. 246–251.
- [145] Y. Hurmuzlu, "Dynamics of bipedal gait: Part II – stability analysis of a planar five-link biped," *Journal of Applied Mechanics*, vol. 60, no. 2, pp. 337–343, 1993.
- [146] J. W. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Transactions on automatic control*, vol. 46, no. 1, pp. 51–64, 2001.

- [147] S. Kajita, O. Matsumoto, and M. Saigo, “Real-time 3D walking pattern generation for a biped robot with telescopic legs,” in *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, IEEE, vol. 3, 2001, pp. 2299–2306.
- [148] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, IEEE, 1991, pp. 1405–1411.
- [149] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” in *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, vol. 1, 2001, pp. 239–246.
- [150] A. B. Cera, “Control applications for mobility of generalized tensegrities,” in press, Master’s thesis, University of California, Berkeley.
- [151] Y. Yan, “Simulation of two-dimensional cable actuated thruster in tensegrity structure,” Master’s thesis, University of California, Berkeley, May 2016.
- [152] Z. Ding, “Asymptotic rejection of unknown sinusoidal disturbances in nonlinear systems,” *Automatica*, vol. 43, no. 1, pp. 174–177, 2007.
- [153] ———, “Adaptive estimation and rejection of unknown sinusoidal disturbances in a class of non-minimum-phase nonlinear systems,” *Control Theory and Applications*, vol. 153, no. 4, pp. 379–386, 2006.
- [154] S.-M. Zhang, G.-Y. Tang, and H.-Y. Sun, “Approximate optimal rejection to sinusoidal disturbance for nonlinear systems,” in *Proceedings of 2008 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2008, pp. 2816–2821.
- [155] A. Jajarmi and M. Hajipour, “Optimal disturbance rejection with zero steady-state error for nonlinear systems with sinusoidal disturbances,” in *Proceedings of the 24th Iranian Conference on Electrical Engineering*, IEEE, 2016, pp. 1154–1159.