National Aeronautics and Space Administration

Final Report Phase II

Super Ball Bot - Structures for Planetary Landing and Exploration

for the

NASA Innovative Advanced Concepts (NIAC) Program

Principle-Investigator: Vytas SunSpiral, MS.

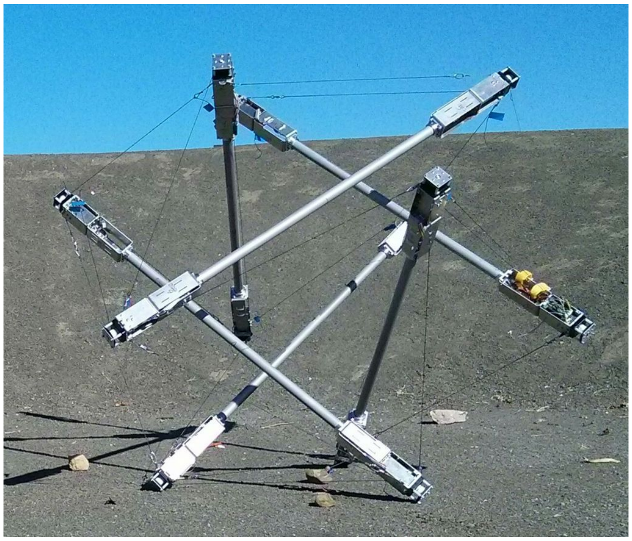
Principle-Investigator: Adrian Agogino, Ph.D.

Co-Investigator: David Atkinson Ph.D.

NASA Ames Research Center

Intelligent Systems Division

September, 2015



Contents

1 Executive Summary

Small, light-weight and low-cost missions will become increasingly important to NASA's exploration goals. Ideally teams of small, collapsible, light weight robots, will be conveniently packed during launch and would reliably separate and unpack at their destination. Such robots will allow rapid, reliable in-situ exploration of hazardous destination such as Titan, where imprecise terrain knowledge and unstable precipitation cycles make single-robot exploration problematic. Unfortunately landing lightweight conventional robots is difficult with current technology. Current robot designs are delicate, requiring a complex combination of devices such as parachutes, retrorockets and impact balloons to minimize impact forces and to place a robot in a proper orientation. Instead we are developing a radically different robot based on a "tensegrity" structure and built purely with tensile and compression elements. Such robots can be both a landing and a mobility platform allowing for dramatically simpler mission profile and reduced costs. These multi-purpose robots can be light-weight, compactly stored and deployed, absorb strong impacts, are redundant against single-point failures, can recover from different landing orientations and can provide surface mobility. These properties allow for unique mission profiles that can be carried out with low cost and high reliability (see Figure 1), and which minimizes the inefficient dependance on "use once and discard" mass associated with traditional landing systems. We believe tensegrity robot technology can play a critical role in future planetary exploration.

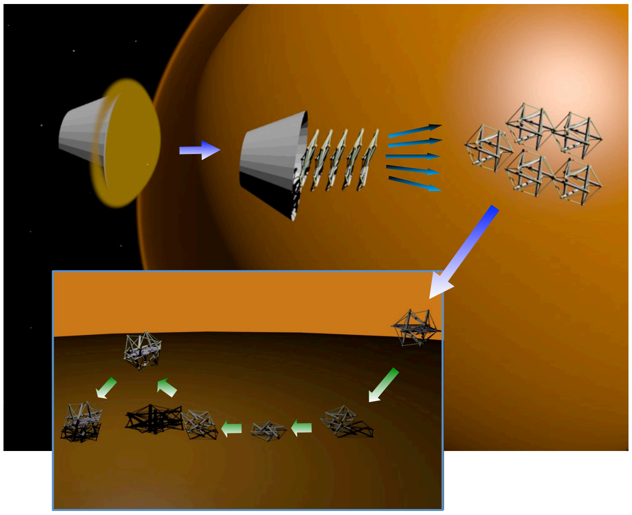


Figure 1: *Tensegrity structures are composed of pure compression and tension elements. They can be lightweight, reliable, deployable, and efficient to manipulate.* **Mission Scenario** - *Tightly packed set of tensegrities, expand, spread out, fall to surface of moon, then safely bounce on impact. The same tensegrity structure which cushioned the landing is then used for mobility to explore moons such as Titan and small asteroids.*

Our Phase II study explored: 1) Designing an untethered hardware prototype of a tensegrity robot ball. 2) Evolutionary and Central Pattern Generator based controls allowing for directed rolling, rolling over hills, rolling over obstacles and rolling up hill. 3) Modeling and verification of landing properties under a wide range of conditions. These studies resulted in three important conclusions:

1. We built a 20 kg tensegrity robot focused on mobility research which was shown to be capable of carrying a 5 kg payload (and expected to be capable of significantly more). This robot successfully rolled on gravel terrain and survived rolling down hills and being driven off a one meter high “cliff” (again, with the expectation that larger falls are possible). The focus of this prototype was on the system engineering for locomotion and future prototypes will be aimed at integrating the full mission capabilities.

2. We demonstrated advanced control algorithms. Our directed evolutionary algorithm dubbed “flop and roll” was able to navigate towards a destination given location feedback. In addition we showed similar capabilities with controls based on Central Pattern Generators and dynamical controls theory. Future work will focus on developing the sensor systems and state estimation algorithms required to migrate these control approaches to the physical prototypes.

3. Using instrumented hardware prototypes, we validated the computational models of landing impact forces we had developed during Phase 1. These models show that tensegrity structures can protect payloads during high speed landing, much like an air bag, and that targeting lading speeds of 15 m/s for the robot appear to be reasonable targets to aim for. Further software analysis of larger scaled structures showed that a tensegrity structure could protect a payload on the scale of the Mars Exploration Rovers. Future work will involve validating that fully engineered and mobile robots can survive the impacts that the structural analysis has shown is possible.

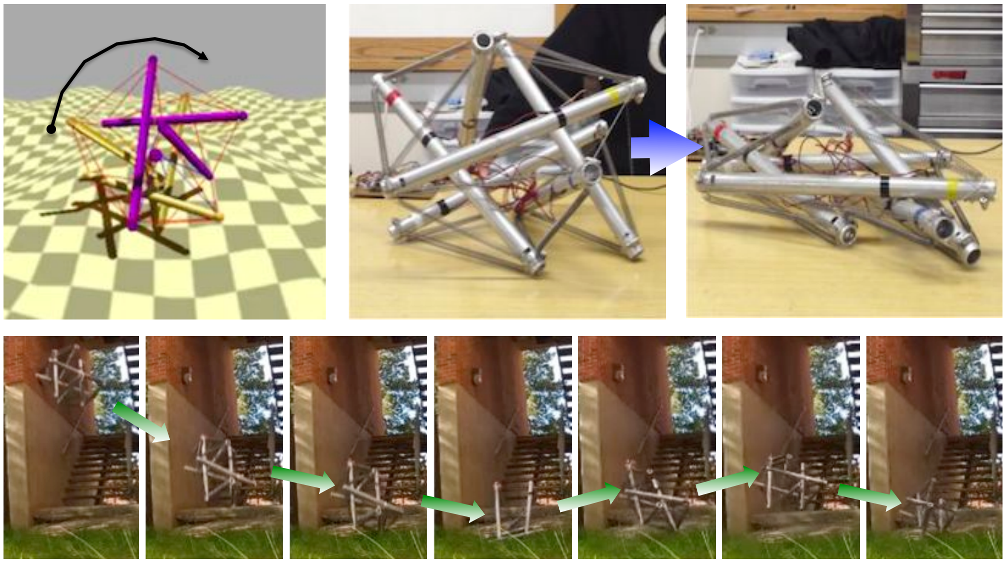


Figure 2: **Tensegrity Evaluations.** *This NIAC study showed: A) That tensegrities can be controlled for mobility, B) That our partially actuated hardware prototype can be efficiently stowed and deployed (with greater stowage efficiency when fully actuated), and C) Analytical and physical drop tests confirm that the probe can serve as an effective landing platform.*

These results are summarized in Figure 2. We are glad to report that our results exceeded all expectations at the start of the project, thoroughly validating the overall tensegrity robot and landing platform concept. This report attempts to summarize experience and insights from both phases of the project, with a particular focus on work that was new in Phase 2. Due to the extremely wide ranging topics covered during this program, it is recommended that those interested in the details also review the many other publications that cover work related to this project. Besides the many details in the NIAC Phase 1 report, this project has resulted in three PhD thesis (and one in progress), five Master's Thesis, two journal articles (and one in revision), and 24 conference papers (See Section 16 for all citations).

2 Organization of this Report

The goal of this final report is to provide an overview of all of our activities related to our NIAC study, with a particular focus on Phase 2 activities. This report is organized as follows:

- Section 3 - Introduction to this NIAC study.
- Section 4 - A summary the basics of tensegrity robotics including their unique structure and possible control methods.
- Section 5 - A summary of our key findings from our prior Phase I study.
- Section 6 - A detailed description of a notional mission to Titan describing both the mission and advantages a tensegrity probe has over traditional alternatives for landing and mobility.
- Section 7 - Overview of our two primary simulation tools, and their cross validation against each other and against an early hardware prototype.
- Section 8 - A description of how dynamical controls and neuroscience inspired central pattern generators may be applied to tensegrity robotics, and new results from Phase 2.
- Section 9 - A description of how to control a tensegrity robot using evolutionary algorithms, including new results from Phase 2. Also includes performance experiments in simulation showing how these algorithms are robust to numerous challenges, and can be accurately steered towards desired goals.
- Section 10 - An exploration of controllers designed to help the tensegrity robot escape from pathological terrain traps, such as getting stuck in holes or crevasses.
- Section 11 - Instrumented experimental validation of our computational models of landing.
- Section 12 - Analysis of the effectiveness of a tensegrity as a landing platform as compared to an airbag system.
- Section 13 - A detailed description of the SUPERball hardware prototype designed to establish foundational engineering principles for locomotion. The development of this prototype represents a significant portion of the effort and results of this project.
- Section 14 - An proof-of-concept experiment demonstrating the use of an actuated payload to induce rolling of the SUPERball robot.
- Section 15 - An outline of future work and the next steps required to mature this concept to Technology Readiness Level (TRL) 3.
- Section 16 - An overview of the academic contributions of this project, including four PhD thesis, five Master's Thesis, three journal articles, and 24 conference papers.

Combined, these results form a strong picture of how a tensegrity robot can be used as an effective landing and mobility platform.

3 Introduction

3.1 Mission Concept

Tensegrity robots can facilitate an intriguing low-cost planetary exploration mission profile (see Figure 1) comprising of the following stages: 1) A set of tensegrity robots can be squeezed into a small launch platform; 2) After initial atmospheric entry and ejection of the heat shield, they can automatically spring away from each other when released at their destination. 3) They “bounce” on impact reducing the need for final descent equipment, such as airbags; and 4) They can reorient themselves from landed position without addition reorientation hardware and efficiently move from scattered initial positions to perform sensor measurements; 5) They can survive significant falls and are resistant to being stuck, simplifying route planning and allowing for more aggressive exploration in the pursuit of science.

Once on the surface, tensegrity robots can perform an array of scientific analysis including soil and atmospheric composition, surface imagery and microscopic analysis. To further reduce complexity, sensors can be suspended on the interior of the tensegrity on cables attached to the nodes, or when appropriate even to the nodes themselves so that the sensors can be moved with movements of the structure itself, eliminating the need for separate sensor arms. In addition, environmental analysis can be performed in-situ at the landing site, at different local locations, or even at distant locations given a tensegrity robot’s potential for efficient locomotion. The biggest advantages of this mission profile are:

1. The structure of the robot itself provides capability for deployment, entry/decent/landing (EDL), and mobility, reducing complexity, risk, and mass compared to using three separate systems.

2. Tensegrity robots are light-weight and can be packed tightly, reducing cost;

3. Multiple robots expand science coverage and reduce risk.

4. Flexibility of robot design allows design reuse, reducing mission project risk.

3.2 Reusability, Redundancy, Reliability, Reduced Mass and Cost

This mission concept has four fundamental desirable properties 1) **Design Reusability** - The core design of a tensegrity robot is simply composed of modular rods which contain actuators, controls, power, and cables. All instances of these components are nearly identical and do not have to be custom designed, thus reducing complexity and project risk. In addition, using the same basic components of actuated tensegrity rods, many different morphologies are possible, with different landing and mobility characteristics that can be tailored to different missions. 2) **Redundancy** - Due to their light-weight and collapsibility, multiple tensegrity robots can be packed together and deployed, then land and carry out their mission independently. In addition, individual tensegrities are robust against actuator failure, as performance gracefully degrades as the number of failures

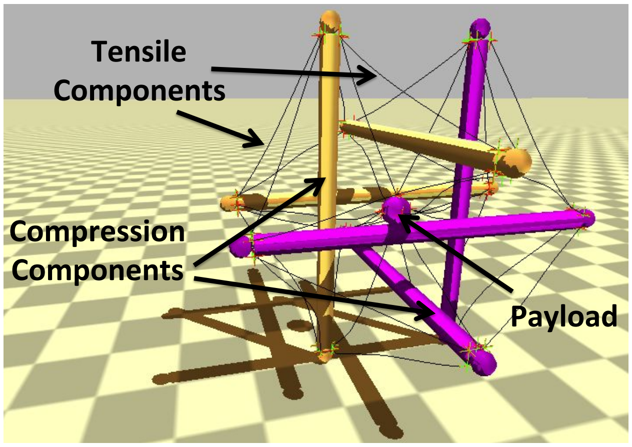


Figure 3: **Tensegrity Structure.** *Tensegrities are composed of pure tension and pure compression elements (e.g. cables and rods) as seen in this picture of a tensegrity robot from our physics based tensegrity simulator. They are light-weight, energy-efficient and robust to failures.*

increases. Note that in this study we mostly focus on the basic engineering, analysis, and testing of a single tensegrity robot, as that is a prerequisite to designing a multi-robot mission. 3) **Reliability** - Structural components can be made robust since they undergo pure linear tension and compression, reducing sheer and bending forces. In addition, the flexible nature of its movements allows the tensegrity robot to recover from awkward landing positions, being stuck, or even falling or rolling from significant heights. Fundamentally, if the robot is reliable enough to land from orbit, it changes the risk-calculus of exploration, and will be reliable even in the face of unexpected slips, collisions, or falls. 4) **Reduced Mass and Cost** - By designing a multi-function landing and exploration robot, we remove the need for the wasted mass of “use once and discard” landing systems like air-bags. Our initial analysis in Phase 1 showed that we can land a higher mass proportion of scientific instruments than a comparable traditional mission design.

3.3 Enabling the Mission

While tensegrities have the potential to dramatically reduce the cost and increase the reliability of robotic missions, significant technology development is still needed to enable such missions. Our study has made large strides towards showing that a tensegrity landing and mobility platform is feasible by showing that:

1. Having analyzed the mission design from trajectory planning, to science instrument selection, deployment, and mass estimation, we showed that tensegrities can be an effective landing and mobility platform for a Titan mission. A tensegrity mission can have a high mass fraction between science payload and overall weight (as measured at atmospheric entry) due to its dual use as a landing system (like an airbag) and as a system for surface mobility. As a result, tensegrity based missions can be cheaper and open up new forms of surface exploration that take advantage of their natural tolerance to impacts.

2. We demonstrated in our physics based simulator that tensegrity probes can be controlled

effectively with evolutionary and dynamical algorithms resulting in robust smooth rolling motion over a variety of terrains.

3. Using multiple analysis and simulation tools we showed that tensegrities are a very robust landing platform, and can protect delicate payloads from a landing impact of 15 m/s (and possibly beyond). This was further confirmed by performing drop tests on multiple physical prototypes.

4. We built multiple prototypes of tensegrity robots, with our most recent one, SUPERball, being built on the scale relevant to the task of carrying a payload of scientific instruments. This work has developed the foundational engineering concepts for this new class of robot, such as appropriate design choices for sensing, actuation, and mechanical design. This prototype has been shown to be capable of locomotion, and has been shown to be robust to impacts and small drops.

These results served our study goals of showing that our tensegrity landing and mobility platform is feasible, relevant, and game changing for NASA.

4 Background on Tensegrity Robotics

4.1 Tensegrity Structures

Tensegrity structures are composed of axially loaded compression elements encompassed within a network of tensional elements, and thus each element experiences either pure linear compression or pure tension. As a result, individual elements can be extremely lightweight as there are no bending or shear forces that must be resisted. An actively controlled tensegrity structure can be packed into small launch volumes and deployed when required. Active motion in tensegrity structures can be performed with minimal energy expenditure since actuators work linearly along the load paths in the tension elements, avoiding the torques caused by long lever arms of traditional robotic designs.

A unique property of tensegrity structures is how they can internally distribute forces. As there are no lever arms, 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 forces applied from any direction. Thus tensegrity structures can be easily reoriented in gravity fields and are ideally suited for operation in dynamic environments where contact forces cannot always be predicted. Likewise, tensegrities can be robust to the failure of individual actuation elements, resulting in a gradual reduction of overall workspace, rather than the loss of entire ranges of motion which are common in serial manipulators.

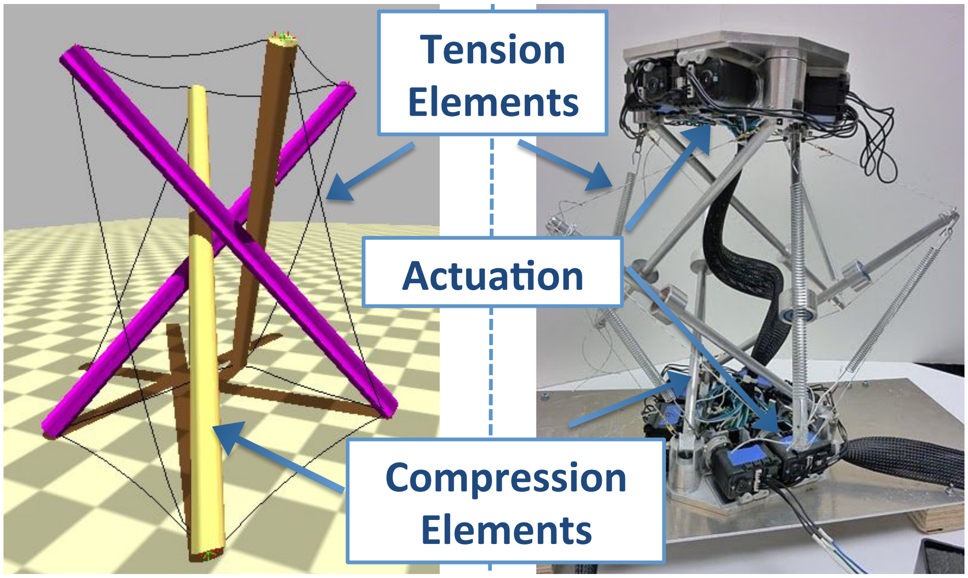


Figure 4: **Tensegrity Structures.** *Left: NASA Ames simulation of tensegrity showing compression and tension elements. Right: An early prototype tensegrity robot at NASA Ames. For the prototype actuation is performed with electric motors and pulleys that change the length of tension elements. Future models may use other simple mechanisms to change cable length such as twisted cable actuation or electrically active polymers (EAP).*

4.2 Tensegrity and Biology

One of the most intriguing aspects of tensegrity structures is that they are being discovered in many aspects of biological systems. This property is being discovered at all scales, from the cytoskeleton of individual cells [39] to mammalian physiology [50]. Emerging biomechanical theories are shifting focus from bone-centric models to fascial-centric models [50, 21]. Fascia is the connective tissue in our bodies (including muscles, ligaments, tendons, etc.), and forms a continuous web of tension, even surrounding and supporting the bones which, unlike our traditional robots, have no rigid connections between them [59]. This new view is challenging the common sense view of our skeletal structure as the primary load bearing elements of our bodies. In the emerging biotensegrity model, bones are still under compression, but they are not passing compressive loads to each other, rather it is the continuous tension network of fascia (muscles, ligaments, tendons) that is the primary load path for forces passing through the body [49]. Recent anatomical research through fresh dissections (i.e. without preserving the cadaver, a process which changes the fascia) [58] is providing evidence of the global network of continuous connective tissue that manages force transfers in the body [59]. Tensegrity structures are a good model for this body-wide tension network [27].

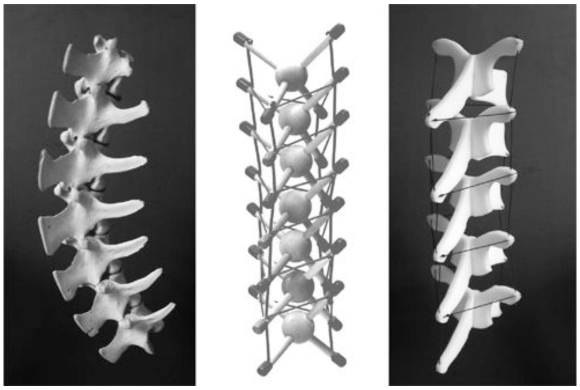


Figure 5: Tensegrity models of the spine showing how vertebrae float without touching, image courtesy of Tom Flemons (copyright 2006) [27]

4.3 Tensegrity Control

Tensegrity structures are a fairly modern concept, having been initially explored in the 1950's by Buckminster Fuller [28] and the artist Kenneth Snelson [82, 81]. For the first few decades, the majority of tensegrity related research was concerned with form-finding techniques [92, 53, 84, 93] and the design and analysis of static structures [5, 46, 80]. Research into control of tensegrity structures was initiated in the mid-1990's, with initial efforts at formalizing the dynamics of tensegrity structures only recently emerging [80]. The very properties that make tensegrities ideal for physical interaction with the environment (compliance, multi-path load distribution, non-linear dynamics, etc) also present significant challenges to traditional control approaches. A recent review [88] shows that there are still many open problems in actively controlling tensegrities. We believe that modern advances in control algorithms based on central pattern generators and distributed learning and genetic algorithms will be key elements in controlling tensegrity robots.

4.3.1 Control Based on Genetic/Evolutionary Algorithms

Instead of defining a control policy directly, evolutionary algorithms can be used to "learn" a control policy. It does this through an iterative cycle, where in simulation a control policy is run and evaluated, and this evaluation is fed back into the genetic algorithm so that it can improve the control policy. Evolutionary algorithms have the following advantages:

1. Complex, nonlinear control policies can be learned.
2. Underlying physics of dynamics does not need to be known.
3. Control policies can be learned from scratch or optimal parameters of existing control policies can be learned.
4. Distributed learning can be used to scale to larger tensegrities and to speed up learning.

Evolutionary algorithms can learn very complex, nonlinear control policies without the need to characterize the physics behind the control policy. Recent advances in distributed evolutionary algorithms applied to control tasks have produced robust learning algorithms that are a perfect match for tensegrity control [26, 33, 20, 83, 89, 87, 2, 86, 35]. Through these methods, each component can individually learn a control policy that decides how to actuate its individual node in such a way that global performance is maximized.

4.3.2 Control Based on Central Pattern Generators

Central pattern generators (CPGs) are neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs. CPGs have been studied from a biological perspective and have been used extensively in robotics [36], especially for walking and other locomotion research (swimming, slithering, flying, etc). They use the term "central" to indicate that rhythms of the CPG are not driven by sensory input, but are self-generated. CPGs are the fundamental building blocks for the locomotor neural circuits both in invertebrate and vertebrate animals, and are also key to other fundamental rhythmic activities such as chewing, breathing, and digesting. In fact, recent research [25] has shown a close relationship between CPG's and motion primitives in the spine which enable both rhythmic and discrete motions. Besides the biological inspiration, CPG's present several interesting properties that are useful to robotic motion including distributed control, robustness to perturbations, the ability to deal with redundancies, fast control loops, and allowing modulation of locomotion by simple control signals.

The complex problem of agile locomotion of a robot can be greatly simplified if the structure and reactive controls of the robot provide a high level of locomotion competence. An inspiring goal is how decerebrated mammals can coordinate complex locomotion behavior without the involvement of their brains [69]. Due to the inherent uncertainty of operating in unstructured natural environments, modern robotic locomotion and manipulation research often focuses on compliant actuation. Tensegrity structures, which model the musculoskeletal system, extend this focus on compliance to the entire structure of the robot, providing desirable qualities such as variable stiffness, robustness to perturbations, and multi-path force distribution. Reactive controls draw inspiration from numerous biological studies showing significant locomotor computation below the brain (for examples and reviews see: cockroaches [71], stick insects [17], and cats [91, 69]), we focus on maximizing the reactive competence of our robots by exploring the combination of compliant tensegrity structures with central pattern generator (CPG) controls [37]. A motivating intuition for pairing tensegrity robots with CPG networks is the similarity in the dynamics of physical forces propagating through a tensegrity structure with the dynamics of control patterns propagating through CPG networks.

Use of tensegrity robots for mobility was initiated in 2004-6 by papers from Masic [54], Aldrich [4], and Paul [67, 68]. Masic's paper included an analytical study of tensegrity based locomotion via periodic waves in a worm-like tensegrity robot; Paul demonstrated mobility both in a physics based simulator and on a hardware prototype. As a result of studies showing the prevalence of tensegrity structures in nature such as cell structure [38] and anatomy [49, 70], and the challenges of controlling tensegrity structures using traditional approaches, the majority of the works in mobile tensegrity robotics have shown biological inspiration in their motivation, using evolutionary algorithms [66, 67, 68, 72, 73, 74, 40], neuroscience inspired CPGs [9, 6, 7], and biomimetic structures such as manta-ray wings [57], or caterpillars [73, 65, 63, 64]. (See also [77, 78, 47, 10, 56] for other works on the locomotion of tensegrity robots) While some work has continued in the analytical understanding of the dynamics of motion for tensegrity mobility [32], the dynamics of contact with the environment are not considered. Since contact dynamics greatly complicate the already difficult task of controller design, most work resulting in simulated or hardware demonstrations of mobility are using non-analytical approaches. This started with Paul's [66, 67, 68] and Rieffel's [72, 73, 74] work which used evolutionary algorithms to discover controllers that resulted in slow crawling and hopping motions. This was followed by Bliss's work using CPGs to control the oscillatory motion of a robotic tensegrity manta ray wing for swimming [9, 6, 7]. Additionally, CPG-like equations have been used by Boxerbaum et al. to control a soft robot moving with peristalsis [11]. Boxerbaum's and Bliss' independent work both confirmed the validity of our approach to using CPGs to control mobile terrestrial tensegrity robots. Other work in our lab, not covered in this report, has also had great success at using CPG's to control the locomotion of a tensegrity inspired "spine" robot over a variety of terrains [85].

5 Summary of Previous Phase I Findings

The purpose of our Phase I study was to validate that the concept of a tensegrity robot landing platform is feasible and useful for real NASA space missions. This validation consisted of three important parts: 1) Define a notional mission to Titan which would provide driving mission requirements for the engineering analysis performed in the later sections. 2) Show that the complexities of non-linear and oscillatory control of a tensegrity robot could be overcome through the use of genetic algorithms and central pattern generators, and 3) Show that in simulation and partially on a hardware platform that a tensegrity robot is in fact a good landing platform capable of surviving a hard landing, protecting its payload, and then act as a mobility platform. We are glad to report that our results exceeded all expectations, thoroughly validating the overall tensegrity robot and landing platform concept.

5.1 Define Notional Titan Mission and Requirements

Section 6 provides full details on our notional mission, which is summarized here. Titans rich atmosphere, the presence of stable bodies of surface liquid, and complex organics make it one of the most complex and Earth-like environments in the solar system. This tensegrity probe mission will build on the science returns from Huygens and will answer many of the new and unresolved questions surrounding Titans ongoing organic processes, geologic history, atmosphere, and surface-atmosphere interactions. Upon arrival at Titan, the tightly packed tensegrity probes enter the atmosphere behind a heat shield until sufficiently slowed to avoid further thermal loads, at which point the heat shield is ejected and the probes separate and expand to fully deployed shock absorbing state. Without requiring parachutes or other landing device, each probe is projected to impact the surface at its terminal velocity on Titan at about 11 m/s, absorbing and distributing impact stresses while protecting its science payload. After the tensegrity probes bounce, roll, and finally come to a rest on the surface of Titan, the actuated tensegrity structure will then begin to function as the primary mobility system for these mobile probes. Once on the surface, a notional science payload containing an atmospheric package, an analytical chemistry package, and an imaging package can begin the probes science mission. Since each probe is equipped with a science payload, mission success is not entirely dependent on the survival of a single probe, and the deployment of only a few probes enables the distributed exploration of a planetary surface.

Our first analysis was to explore the possible packaging and deployment of the probe. We found that the basic structure could be fully flattened to a triangular shape with sides equal to the lengths of the rods. This approach leaves room in the middle of the payload, and can be fully deployed with actuated cables. Next, we investigated possible technologies to use for the three science packages: the atmospheric and meteorology package, the analytical chemistry package, and the imaging package. These were used to drive initial mass estimates and to scale the avionics, power, and structural support required for the mission. This resulted in an initial design that showed an overall system mass of 100kg, of which 70kg is productive scientific payload and associated avionics, while the remaining 30kg is dedicated to structure and actuation for deployment and mobility.

Using these numbers, we were able to show that an individual tensegrity probe has a significantly lower EDL hardware overhead and increased science payload mass percentage when compared to similar planetary surface probe missions, increasing the potential for science return. We evaluate this by comparing the total system mass at the point of atmospheric entry with the mass of the rovers productive payload, which constitutes all the science instruments and associated avionics, power, and controllers. Compared with the Mars Exploration Rovers (MER) and Mars Science Laboratory (MSL) which had a science payload mass fraction of 18% for MER and 22% for MSL, we have shown that a tensegrity probe may be able to operate on Titan as a mobile probe with a science payload mass fraction of 50% due to the dual use of the tensegrity structure for both EDL and surface mobility. While comparing Mars rovers to a Titan probe is not ideal, these are the closest existing missions which include surface mobility. Further analysis may drive this percentage higher as more efficient versions of the tensegrity lander are explored. By increasing the mass ratio of productive science payload vs total system mass, this approach helps drive down future mission costs by allowing smaller lighter missions with the full capabilities of EDL and surface mobility.

5.2 Phase 1 - Developing Control Algorithms using Evolution and Central Pattern Generators

Tensegrity robots have the potential to be fabulous mobility platforms as they can be light-weight, energy efficient, robust against failures, and can traverse across unfavorable terrains. However controlling these robots is difficult since there are many points of control, the controls interact in non-linear ways, and the structure as a whole is oscillatory. Using traditional control algorithms for mobile tensegrity robots is difficult and previous successful attempts have been limited to very slow static gaits, and evolutionary control of very simple structures. A real concern going into the Phase I study was that no reliable control could be found. Fortunately we have had tremendous success with evolutionary algorithms and later with dynamical and Central Pattern Generator based controls (see Figure 6).

We performed several tests of evolutionary control in a detailed physics simulation. Using open loop control, the evolutionary algorithm is able to find a control policy that allows the tensegrity robot to roll quickly in a smooth manner. These results were robust to adding small obstacles to the terrain and was even able to keep rolling after we cut one of the control cables.

The controls used by our evolutionary methods were a simple form of semi-distributed oscillatory control: All the actuations were controlled through sine waves synchronized through their relative phase shifts. These controls proved to be robust and sufficient for our goals of having our tensegrity platform roll smoothly. These controls also gave excellent examples and insight into what fundamental dynamics were required for fast smooth motion. On the other hand, these controls are open-loop and as a result, a major limitation of these evolved controls is that they do not provide a means to direct or steer the motion of the tensegrity probe. Rather, a control law is learned which can be abstracted, but which does not allow reactive changes of direction. Addressing these limitations was a central focus of the Phase 2 controls research described in sections 8 and 9.

During Phase 1 initial steps were taken to explore these closed loop control ideas which were

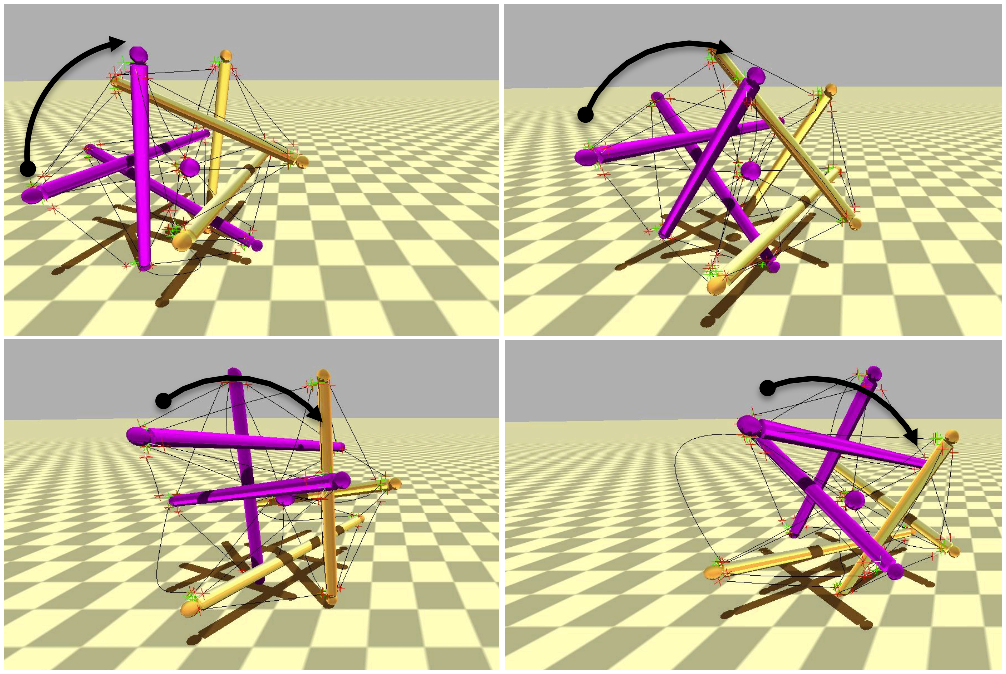


Figure 6: **Tensegrity Dynamics**. *Tensegrity is able to achieve smooth rolling motion. This rolling is accomplished solely by changing the length of the cables. Our learned control policies produce rolling that is also dynamical as the tensegrity does not stop to setup next roll action. This type of rolling can be fast and highly efficient.*

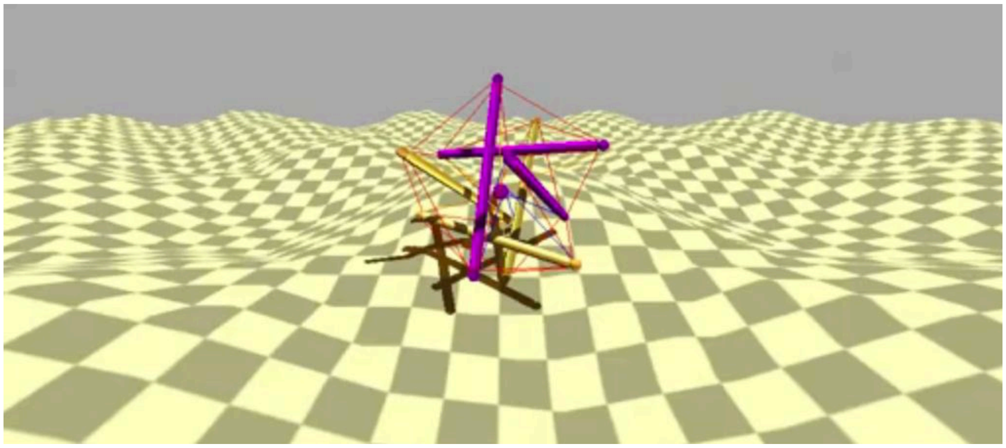


Figure 7: **Dynamical and Central Pattern Generator Based Control**. *Our dynamical controls allows tensegrity to climb over moderate hills.*

taken further during Phase 2. To develop steerable controls in Phase 1, we took the insights gained from the evolved controllers and hand crafted reactive dynamical controllers that used a variety of simulated sensor feedbacks to enable steerable rolling. This approach was very successful and was shown to work on a variety of terrains, including moderate hills (Figure 7). But, this approach relied on significant amounts of sensor feedback which may be difficult to implement, so we wanted to find a hybrid approach that enabled steering while using a minimum amount of sensor feedback. Central Pattern Generators (CPG's) are able to store complex gait cycles in their network dynamics allowing for a significant reduction of sensor feedback required. We explored a variety of ways to use this property and to combine it with aspects of the dynamical controls approach to enable steering. The best result found during Phase 1 uses the hand-coded dynamical controller as a trainer to learn the parameters for a Hybrid CPG which combines Hopf oscillators and Inverse Kinematics.

5.3 Phase 1 - Verification in Simulation and Hardware

In Phase 1 we developed multiple analytical and simulation studies of tensegrity landing events. We cross validated our different computational approaches with each other, and then performed initial hardware verifications which showed broad agreement with the models. We believe that this is the first such analysis showing the characteristics of tensegrity structures during planetary landing. In all these experiments, our Phase I study verified the suitability of a tensegrity robot as a landing and mobility platform in the following three ways:

1. We verified that actuation for compact storage and deployment is possible in a hardware tensegrity probe.

2. We verified that our actuated prototype tensegrity probe remained structurally sound and could continue to operate after landing.

3. We verified that a tensegrity structure could protect a delicate payload, landing at impact speeds of 15 m/s (and possibly higher, pending future analysis).

Our first hardware prototype was built with hollow aluminum rods with the actuators mounted inside the rods for protection. The cables were a mixture of springs and high tension synthetic cables. This prototype was only partially actuated, with six motors, and was able to demonstrate shape change from a compact form to a fully deployed shock absorbing shape. With full actuation it would be able to achieve an even more compact shape when stored. To test the robustness of the design we dropped the probe 10m, as shown in Figure 8. Dropped from this height at earth gravity the probe landed at it's terminal velocity of Titan, and it easily survived the fall and was still functional afterwards, as was shown by its ability to still change shape between its collapsed and deployed states. A useful insight from this test was that the motors should have mechanical brakes on them for the landing impact, as they were slightly back driven during landing.

Since the first prototype did not contain a central payload, we also extensively analyzed how the landing forces would impact a payload in two simulators: our (recently released as open source)

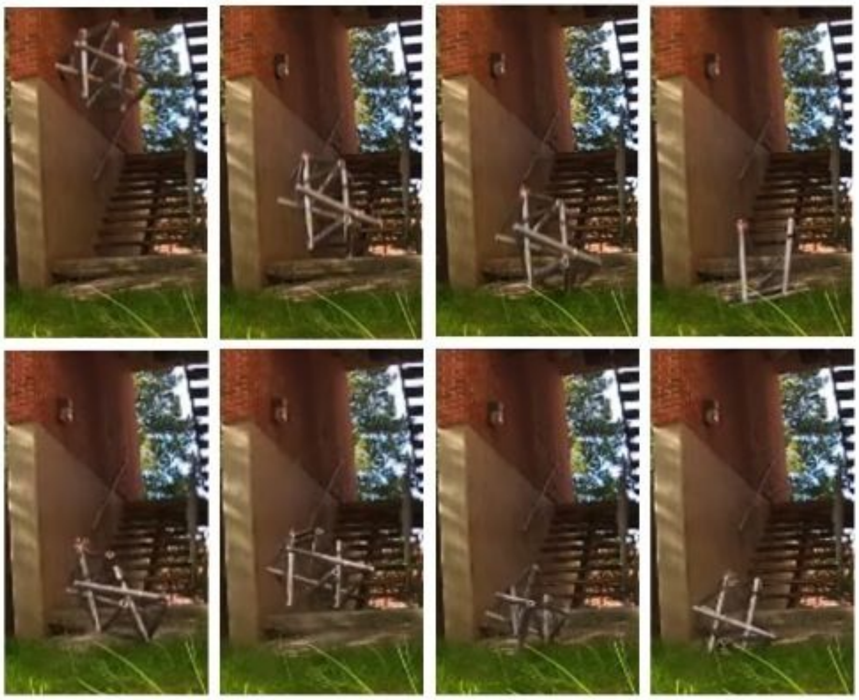


Figure 8: **Prototype drop test** *from 10m – about 14 m/s, greater than its terminal velocity on Titan*

NASA Tensegrity Robotics Toolkt (NTRT) simulator which uses a physics engine and virtual environments, and a hand-built Euler-Lagrange (E-L) simulator based on algorithms developed by Robert Skelton. We used these two simulators because they both have strengths and weaknesses. The NTRT simulator is the most general purpose, allowing us to explore control algorithms and complex environmental interactions, but it is an iterative discrete solver that we were concerned might not be providing accurate answers. The E-L solver, on the other hand, has a much stronger analytical basis and provides very accurate answers, but is of limited use because some of the nodes (i.e. rod ends) must be constrained and locked into place.

We first cross-validated these two simulation tools with each other and found that they predicted similar behaviors. We then performed initial cross-validation to motion control of a prototype tensegrity robot and found a reasonable match, giving us confidence in the accuracy of our modeling tools. Finally, we performed extensive analysis on drop tests and the protection provided to a payload. As expected, we found that by varying the rod lengths, which impacts the stroke distance for the payload to decelerate, we could control the maximum deceleration experienced by the payload while ensuring that it did not collide with the ground or structure. For example, with rods of 1.5 meters in length, our payload experienced a max deceleration of 21.4G when landing at 15 m/s.

These payload simulation results were supported by further drop tests with another prototype probe made of wood, plastic and elastomers. In this test, an egg was suspended in the middle of the tensegrity probe, which had no actuators, and was then dropped from a height of almost 10 meters without the egg breaking. The landing speed of the structure was thus about 14 m/s, and an egg can be expected to break at about 2 m/s, showing a significant amount of payload protection provided by the tensegrity structure, and providing support for the validity of our simulation studies. This probe also had a different morphology, a 30 strut dodecahedron, indicating that future work should explore the advantages of other shapes and configurations of tensegrity probes.

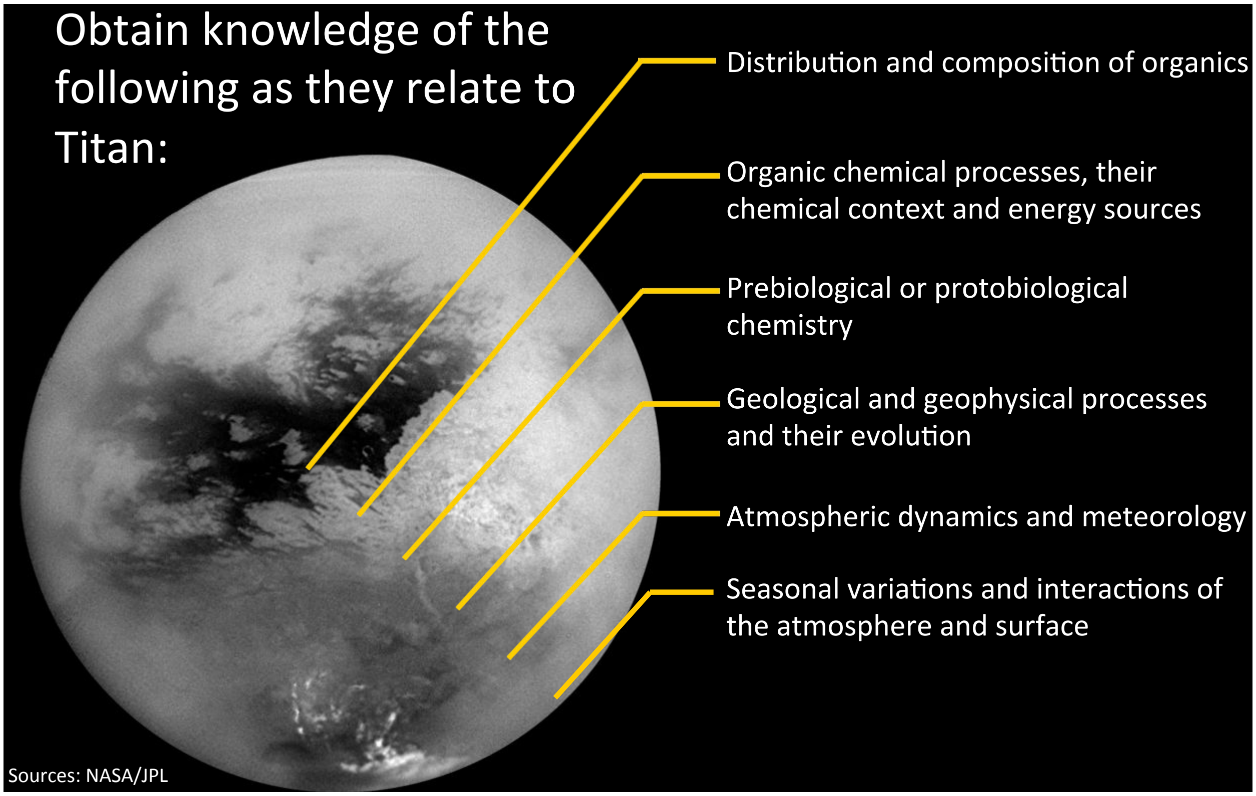
Phase 2 work continued this line of validation through the use of instrumented payload drop tests which enabled us to accurately quantify the match between the models and experimental data. Also, further analysis was performed on scaling the structures up and comparing to larger payloads, such as MER class rovers, for which our approach was found to be a reasonable alternative to airbags.

6 Tensegrity Probes for a Notional Mission to Titan

Based on their robust structural and control properties, we believe that tensegrity robots will have tremendous potential for numerous NASA robotics missions. In this section, we summarize details of what such a mission would look like using a tensegrity robot mission to Titan as an example [1].

Note that this section is not meant to serve as a complete analysis of a Titan mission. Instead it describes how a tensegrity robot would be beneficial for such a mission. Likewise, while the Tensegrity Probes enable multi-robot missions due to their low weight and packing efficiency, we spend most of our technical analysis looking at a single robot mission in order to build the core knowledge required for a full multi-robot mission design effort.

6.1 Motiviation for Titan Mission



Obtain knowledge of the following as they relate to Titan:

- Distribution and composition of organics
- Organic chemical processes, their chemical context and energy sources
- Prebiological or protobiological chemistry
- Geological and geophysical processes and their evolution
- Atmospheric dynamics and meteorology
- Seasonal variations and interactions of the atmosphere and surface

Sources: NASA/JPL

Figure 9: **Scientific Goals for Titan Mission**.

The success of the Huygens probe shed light on Titan's atmosphere and surface conditions which were found to have elements greatly similar to Earth and others vastly different. When it's substantial atmosphere is included, Titan is the largest satellite in the solar system. Titan's thick and very cold atmosphere is composed primarily of nitrogen with traces of methane, argon, and other hydrocarbon molecules. Observation of Titan has shown signs of a significant greenhouse effect, seasons, clouds, evidence of a methane-based hydrological cycle, and an extremely heterogeneous

[1] More details can be found in the Phase I Study

surface that includes methane-ethane lakes, dune fields, evidence of erosion, and possible evidence of cryovolcanism. At the surface, the gravity of Titan is about 1/7th that of Earth, the pressure is about 1.5 bar, and the surface temperature is 94 Kelvins. With the cold temperatures, the surface density of the atmosphere is approximately 4 times that of the Earth.

In addition to methane, Titan's atmosphere also contains nitriles, more complex hydrocarbons, and aerosols formed from the nitrogen and the products of methane photolysis, and it is the precipitation onto the surface of these complex hydrocarbon compounds and aerosols that are the primary source of Titan's surface organics. Once on the surface, these organics can undergo further processing when exposed to brief periods of liquid water resulting from impact events or possible episodes of cryovolcanism. Since the products of complex atmospheric chemistries have precipitated to and been collected on the surface over geologic time, and since it is these products that form the basis of possible biotic and prebiotic chemistries, detailed studies of the chemical composition of the surface and subsurface of Titan are of high interest. In particular, regions of current and previous cryovolcanism and possible impact events are important since it is in these regions that material from the deeper interior of Titan may have found its way to the surface, and where episodes of liquid water in the geologic past are possible.

Along with this interest in possible pre-biotic chemistries in Titan's atmosphere, surface, and subsurface [94] there is also high interest in understanding the wide diversity of surface compositions, textures, and structures, including extensive dune fields and geologies due to fluvial and lacustrine processes, and surface reservoirs of liquid methane and ethane mixtures.

A planetary mission based on tensegrity robotics differs from traditional missions based on its unique unpacking, landing and mobility properties. We would therefore expect a total mission profile for a Titan mission to be similar to the Huygens mission for the approach and entry phase, but much different for the decent, landing and surface phase. The summary of our mission profile is as follows:

1. Enter Titan's atmosphere at 6.4 km/s after a 8-10 years in transit
2. Slow down in Titan's atmosphere using heat shield
3. Unpack and reach a terminal descent velocity of 11 m/s
4. Land at 11 m/s using the tensegrity structure to protect payload
5. Roll to lakeshore or other target destination to perform scientific measurements

6.2 Mobility

Once the tensegrity robot has landing it can move to locations of interest. The need for mobility is essential if several different surface locations need to be examined. In addition, even if only a single point of interest exists, mobility may still be essential since landing mechanisms may not be accurate enough to land in exactly the desired spot, or the point of interest may be in a location where landing is hazardous. One such location is likely to be near a lake shore on Titan. In such a mission, landing too close to the lake would be hazardous since the robot may land in the lake itself. Therefore in such a mission the tensegrity robot could land a safe distance from the lake, and then roll to its shore as shown in Figure 10.

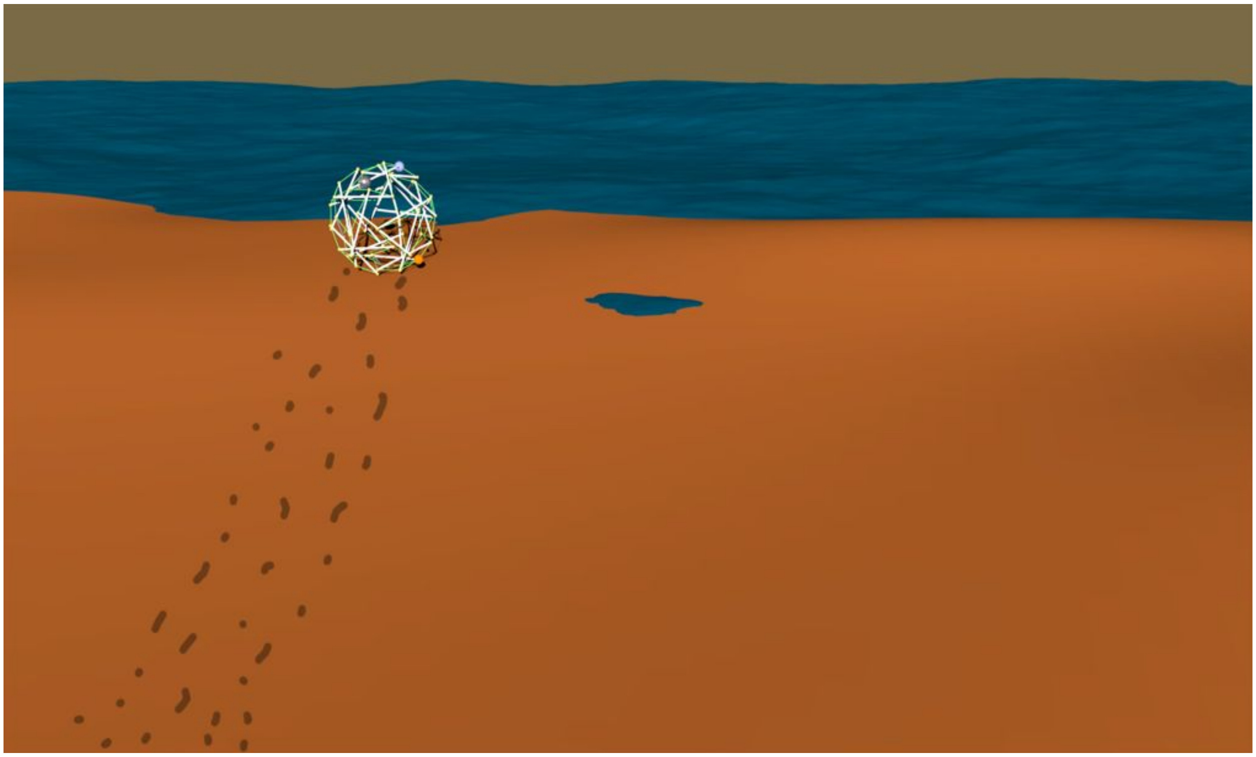


Figure 10: **Tensegrity Robot Moves to Lake Shore**. *The tensegrity is both a landing and mobility platform. After a terminal velocity landing, the tensegrity can move to desired areas of interests such as a lake shore.*

6.3 Scientific Study of Titan's Surface and Instrument Packages

Titan is a volatile rich, chemically active moon with a complex methane cycle. Dunes, craters, fluvial valleys, and lake shores are among the primary target sites for future Titan missions. While a tensegrity platform can potentially be used on all these sites, here we focus on lake shores since they provide a high scientific return due to the interaction between a liquid and solid surface, and the atmosphere. To complete these tasks, we selected a notional set of instrument packages for this mission to support the fundamental science objectives of a future Titan tensegrity Mission. Our instruments are:

- **Analytical Chemistry Package**, containing a Gas Chromatograph and a Mass Spectrometer [24].
- **Atmospheric and Meteorology Package**, containing instruments to measure Temperature, Wind Speed, and Methane Humidity [31].
- **Imagery Package**, containing Navigation Cameras and a Field Microscope [52, 34]

The analytical chemistry package will provide detailed analysis of the organic molecules present on the surface. Sites on Titan's surface where complex organic chemistry can produce and sustain organic molecules may provide an insight to prebiotic chemistries within the solar system. An atmospheric and meteorology instrument package will continuously monitor temperature, pressure, wind speed, and methane humidity as the tensegrity robot travels across Titan's surface. The atmospheric and meteorology package will measure diurnal and possibly seasonal changes in temperature, wind speed and directory, and methane humidity variations at many locations across the surface, allowing for the identification and monitoring of local weather systems. The Imaging package will consist of both a NAVcam and a field microscope to be used in a variety of surface operations. The tensegrity robot NAVcam will be similar to the Mars Exploration Rover (MER) NAVcams with new larger heating elements to handle the harsh Titan environment and will be used to navigate the varied surface [51].

6.4 Analysis of Power Generation from Titan Winds

Given the presence of a thick atmosphere and winds on Titan, we explored the possibility of wind generated power. After analysis of different turbine types, we selected the Twisted Savonius for further study due to its efficient and low profile design. An offshoot of the Savonius turbine, the wind scoops are twisted helically around the main shaft. This removes much of the cyclic loading by allowing constant wind engagement and are self-starting. Efficiency can be increased by closing off the top and bottom of the helix with end caps. Figure 11 shows two possible designs where the Twisted Savonius would be mounted on the rods of the SUPERball. One of the constraints that must be addressed is ensuring that the turbines do not interfere with the cables of the SUPERball, nor contact the ground during locomotion.

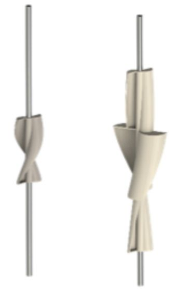


Figure 11: Two potential turbines. The larger one provides significantly more surface area, but comes closer to the cables.

Currently, SUPERball has struts that are about 1.5 meters long. Given this length as a starting requirement, it has been estimated that a Twisted Savonius turbine could have an effective area between $0.2m^2$ and $0.05m^2$. In Figure 11, the design on the right has an area of $0.2m^2$, however it sits very close the outer strings, limiting potential motion. The design on the left has an area of $0.05m^2$, but is much more feasible in packaging. When the tensegrity is stationary for charging, it is believed turbines on up to four struts may be satisfactorily exposed to cross winds so as to generate energy, as shown in Figure 12.

Potential power available from wind power can be determined by,

$$P = \frac{1}{2} * C_p * \rho * A * v^3 \tag{1}$$

where P is the power in Watts, C_p is the power coefficient, ρ is the density of the ambient air in kg/m^3, A is the projected area of the turbine blade, and v is the velocity of the undisturbed

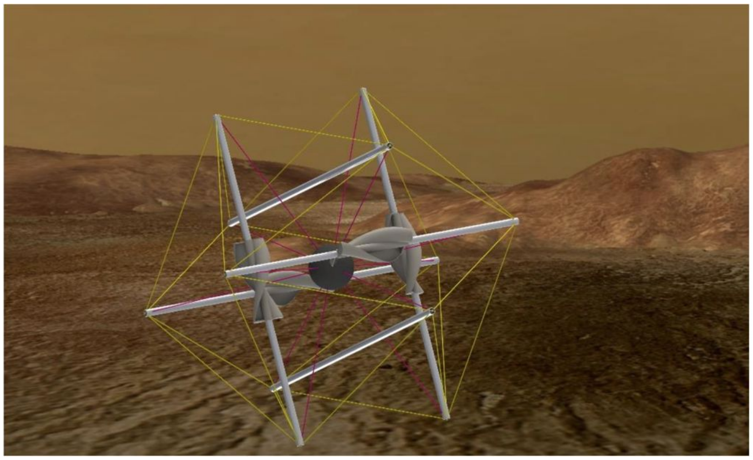


Figure 12: A visualization of four small turbines on a tensegrity

wind. Average parameters near the surface of Titan are given below in Table 1. Maximum power coefficients for Savonius type turbines is 0.3, but in practice power coefficients vary from 0.1 to 0.3.

Table 1: Titan Surface Conditions

Parameter	**Value**
Density	$5kg/m^3$
Wind Speed	$0 - 2m/s$

Using the values above, a maximum expected power output from all four turbines is

$$P = \frac{1}{2}(0.3)\left(5\frac{kg}{m^3}\right)(0.2m^2\,x\,4)\left(2\frac{m}{s}\right)^3 \approx 4.8W \tag{2}$$

As the above calculation used best case efficiency and the “large version of the turbine we also made a more conservative analysis, using the smaller surface area of $0.05m^2$ per strut, and C_p of the turbines as 0.15 and a wind speed of $1m/s$, which gives:

$$P = \frac{1}{2}(0.15)\left(5\frac{kg}{m^3}\right)(0.05m^2\,x\,4)\left(1\frac{m}{s}\right)^3 \approx 0.075W \tag{3}$$

Finally, we examined the effects of these parameters on final output and generated the following sensitivity graph. The x-axis is a normalized axis of each varying parameter, i.e. 0 equates to 0 and 1 equates to the largest value in the parameter's range. The y-axis is the total power output from four turbines in Watts. Each line is varying a single parameter, while all others are held at the nominal value of our more conservative analysis.

We conclude from this study that it is possible to generate a small amount of power from rod-mounted twisted savonius turbines, but that at the size rods we are studying here, the amount of

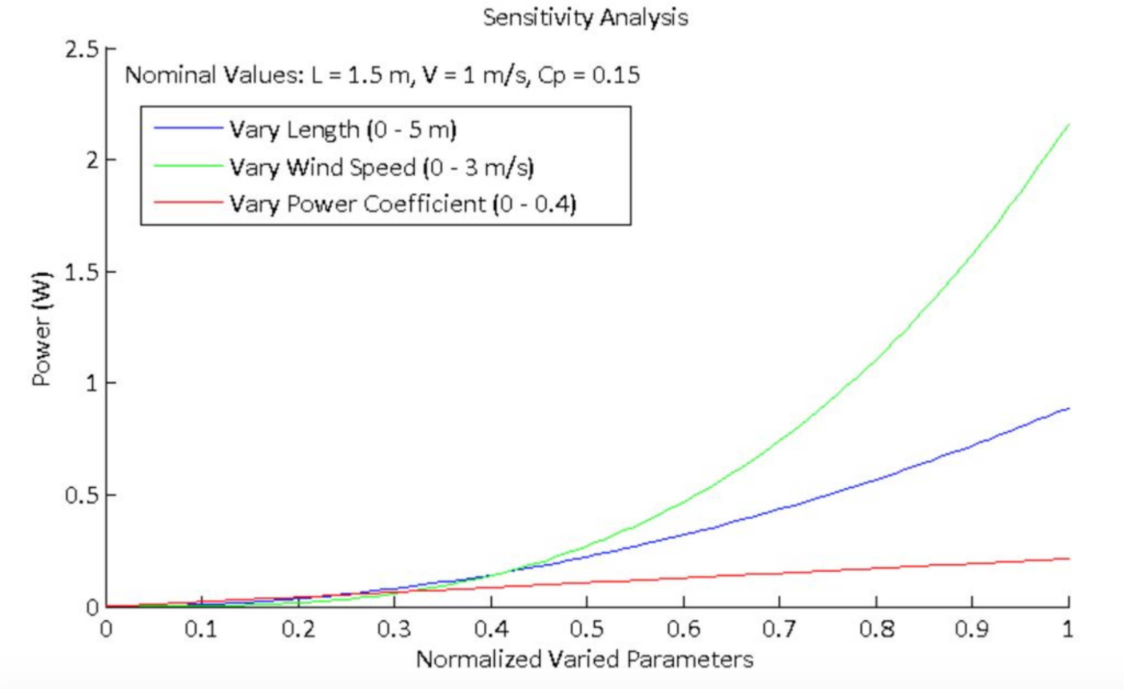


Figure 13: A sensitivity graph examining the influence of strut length, wind speed, and turbine efficiency.

power is unlikely to be sufficient. Granted, this is based on fairly limited knowledge of Titan winds, and if future observations show much higher wind speeds, it would be worth revisiting this analysis. For the rest of the NIAC project we will assume that a future Titan mission would be designed around other power sources, such as RTG's.

7 Hardware Verification of Simulation Tools

7.1 Systems and models

We introduce three systems to evaluate the design and control of tensegrity structures. The first two are simulation environments, while the third is an untethered, lightweight robot prototype. Our simulators and hardware designs are open source and can be obtained from our website at http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/.

7.1.1 Spring-cable assemblies

All tensile members of the structures we study in this work are compliant and we refer to them as *spring-cable assemblies*. Although various implementations are possible, all spring-cable assemblies in this work can be modeled as a zero rest length passive spring in series with a non-elastic cable. The tension on spring-cable assembly i is given by

$$f_i \;=\; k_i \max(\|\mathbf{p}_{i,0} - \mathbf{p}_{i,1}\| - \ell_i, 0), \tag{4}$$

where k_i is the spring stiffness, $\|\mathbf{p}_{i,0} - \mathbf{p}_{i,1}\|$ is the Euclidean distance between the attachments of the spring-cable assembly and ℓ_i is the cable length. Actuated members have a controllable rest length ℓ_i.

7.1.2 Euler-Lagrange simulator

We extended the Euler-Lagrange formulation described in Skelton's reference work on Tensegrity Systems by adding support for ground contacts and gravity [79]. In Skelton's work, the struts of the tensegrity structure are modeled as cylinders with infinitesimal radius. Strut-to-strut contacts are not modeled, which is an acceptable approximation for NASA-scale missions.

We found that this environment provides particularly accurate results for two types of experiments: tests of structural forces, and tests of effective structural stiffness. This simulator is used for payload acceleration prediction during impacts from drop tests, as well as stiffness analyses and form-finding. However, its simple underlying model make it unsuited for the study of complex interactions with the environment.

7.1.3 NASA Tensegrity Robotics Toolkit

Our main simulation environment, the NASA Tensegrity Robotics Toolkit (NTRT), is built on the discrete time Bullet Physics engine (a game physics simulator) [22]. The primary value of NTRT is that it enables research into the control of tensegrity robots while they interact and move through simulated environments. The Euler-Lagrange simulator discussed above cannot support rich and dynamics environmental interaction, and is thus only suited for studying specific limited situations. NTRT allows different terrains and obstacles to be created and for controllers to drive tensegrity robots through those terrains with active sensor feedback, etc.

Since game physics requires real time simulation, Bullet is designed to handle collisions without excessive processing power. However, the Bullet physics library does not currently provide models of ropes, cables, or springs with realistic material properties and stress analysis. Instead of using these default soft body models, we built an additional library to simulate spring-cable assemblies as two point tensional elements that apply directional forces to rigid bodies.

This approach gives the ability to calculate the amount of stretch and tension for each simulated cable, as well as the force exerted to the bodies, using more mathematically rigorous models. Initially we had a limitation to this method that the cables did not exist in the simulation world as physical bodies, and thus their collisions and interaction with rigid bodies was not simulated. For much of our work, this limitation was not relevant, as we were studying locomotion over reasonably easy terrain. Recently we implemented an efficient approach to include soft body contact dynamics for the elastic cables. We believe that, as a result, NTRT is one of the only open source libraries that includes an efficient and realistic model of elastic cable dynamics. This new capability has allowed us to start investigating locomotion over more complex terrain where one can expect the cables to contact and interact with the environment through the locomotion gait. The open source NTRT library can be found at: `http://irg.arc.nasa.gov/tensegrity/ntrt/`.

7.1.4 Prototype

In addition to these two software environments, a physical prototype of the 6-bar tensegrity was constructed. ReCTeR (Reservoir Compliant Tensegrity Robot) is a highly compliant, lightweight (1.1kg), underactuated tensegrity icosahedron robot shown in Fig. 14. The robot's 24 outer tensile elements are passive spring-cable assemblies with low stiffness springs (28.4N/m) under moderate pretension ($\approx$ 10N). Six actuated spring-cable assemblies run through the robot connecting active and passive end caps (see also Section 7.1.5). Their rest length is adjusted by a rotational DC motor (4.5W, 4.4:1 gear ratio), which spins a snag resistant bobbin (5.5mm diameter). The other end of the cable attaches to a stiffer spring (81N/m) which is fixed to a passive end cap. We use low stiffness springs to allow active folding (Fig. 14) without plastic deformation of the 24 passive tensile members or excessive motor power requirements.

The six active spring-cables run through the robot and connect non-parallel struts in an advantageous way. Stiffness analysis revealed that this pattern allows for large shape deformations with low motor power requirements. As a consequence of the low spring stiffnesses, the lowest natural frequencies of oscillatory modes for the structure are on the order of a few Hz.

Sensing and feedback control is achieved by 24 tension sensors using strain gages, six ground reaction force sensors, and three 6-DOF inertial measurement units distributed evenly among the actuated end caps. To allow dynamic motion and rolling, each self-contained strut contains a hardware module with battery power and wireless communication. The battery is mounted in the center of the strut to minimize the moment of inertia around its longitudinal axis. This makes ReCTeR fully untethered.To the best of the authors' knowledge, this is the first example of an untethered 6-strut tensegrity robot with such capabilities. ReCTeR provided excellent initial insight and results on our work, but prototype development eventually switched to the more recent SUPERball robot (described in Section 13 of this report).

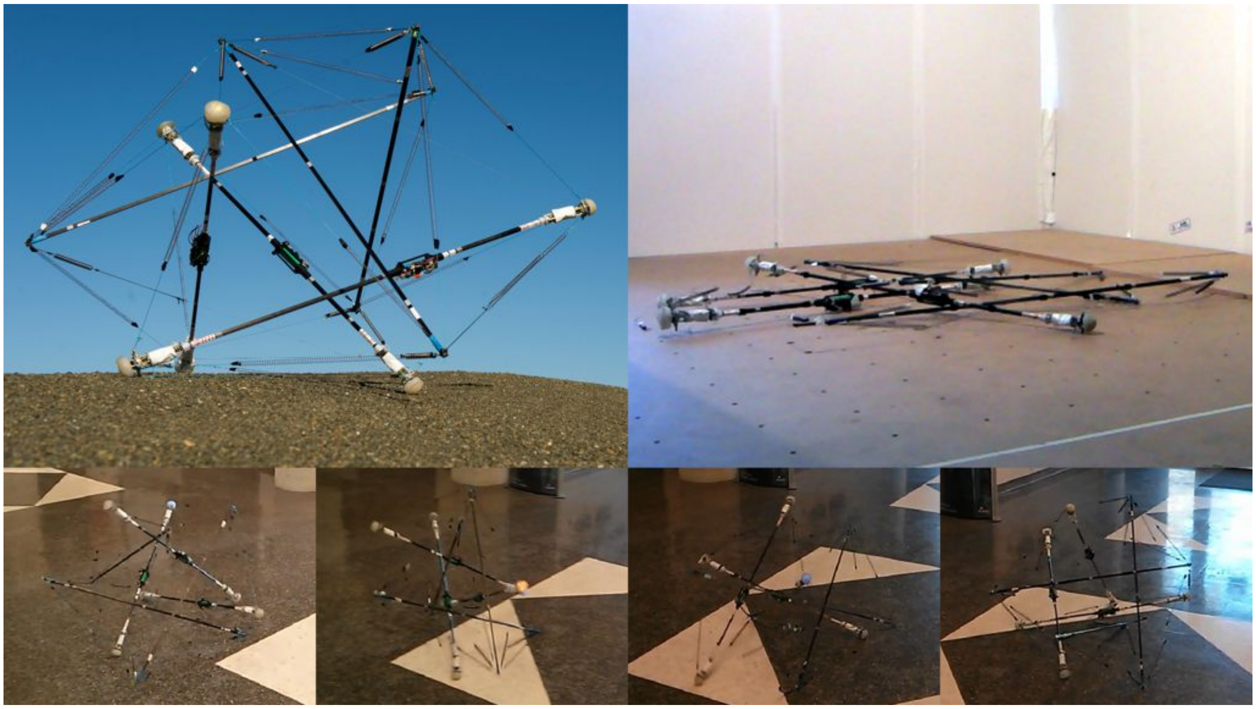


Figure 14: ReCTeR: an untethered, highly compliant, spherical tensegrity robot. Top left: deployed robot. Center right: active folding. Bottom: ReCTeR rolling from right to left.

7.1.5 Robot models

Our control methods are implemented on platforms with varying configurations. Fig. 15 shows the three tensegrity icosahedra analyzed in this paper.

We put a particular emphasis on spherical icosahedron tensegrities. This symmetric configuration provides a large interior volume with a moderate number of members and can be folded easily. It lends itself naturally to rolling locomotion because of its triangular faces. Additionally, tensile member failure will result in reduced locomotion capabilities instead of full failure, due to its redundant number of tensile members [23].

The basic tensegrity icosahedron is shown on the left of Figure 15. This structure has 24 spring-cable assemblies and 6 rigid rods. The spring-cable assemblies are also referred to as *outer shell* elements.

A tensegrity icosahedron with a payload is displayed in the center. This structure has an additional 12 spring-cable assemblies to suspend the payload in the center of the robot. We will also refer to these as *inner* members.

The model on the right displays ReCTeR's configuration, with the robot's actual dimensions. ReCTeR has the basic tensegrity icosahedron configuration with 6 extra actuated spring-cable assemblies running through the robot.

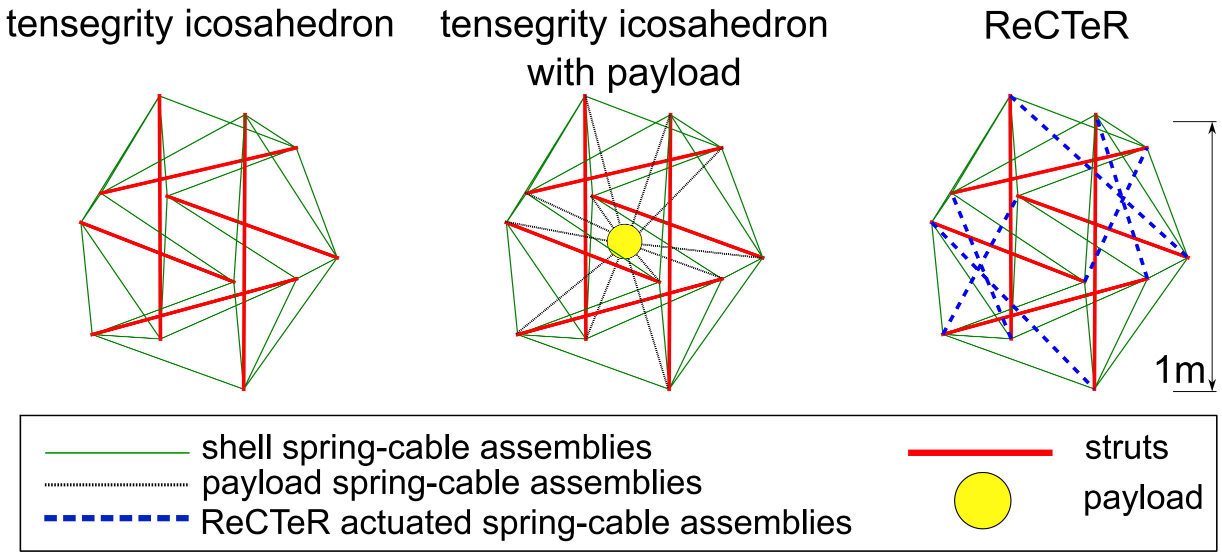

tensegrity icosahedron
tensegrity icosahedron with payload
ReCTeR
1m
shell spring-cable assemblies
payload spring-cable assemblies
ReCTeR actuated spring-cable assemblies
struts
payload

Figure 15: The various tensegrity configurations used in this paper. Left: tensegrity icosahedron with only outer shell members. Center: tensegrity icosahedron with a payload by inner elements. Right: ReCTeR configuration with passive outer shell and actuated spring-cable assemblies inside the robot.

7.2 Validation of simulations

7.2.1 Experimental setup

To track the full state of the robot, an active marker motion capture setup was used. Each passive strut was fitted with 2 markers, while each active strut received 3 markers.

7.2.2 Kinematics

The forward kinematics of the Euler-Lagrange and NTRT simulators were compared against motion capture data from ReCTeR.

The six strut ReCTeR robot was placed on one of its triangular faces and two of the top spring-cable assemblies were actuated, as shown in Fig. 16. We tracked the vertical displacement of an end cap that is not directly actuated by one these two members. The incident strut was suspended in the air by a total of 10 springs.

The lengths of the two actuated spring assemblies were varied from the point of no tension in the given configuration (slack) to 0.32m beyond this length. Each range was sampled at 10 equally spaced lengths, resulting in 100 measurement positions in total. These ranges were manually tuned to maximally deform the robot, without causing it to roll. This experiment was repeated three times, with no meaningful difference in the observed displacements.

The average observed difference between the motion capture data and the Euler-Lagrange simulator was 6.5mm. For NTRT, we obtained an average error of 15mm (0.5% and 1.3% of the robot's diameter respectively).

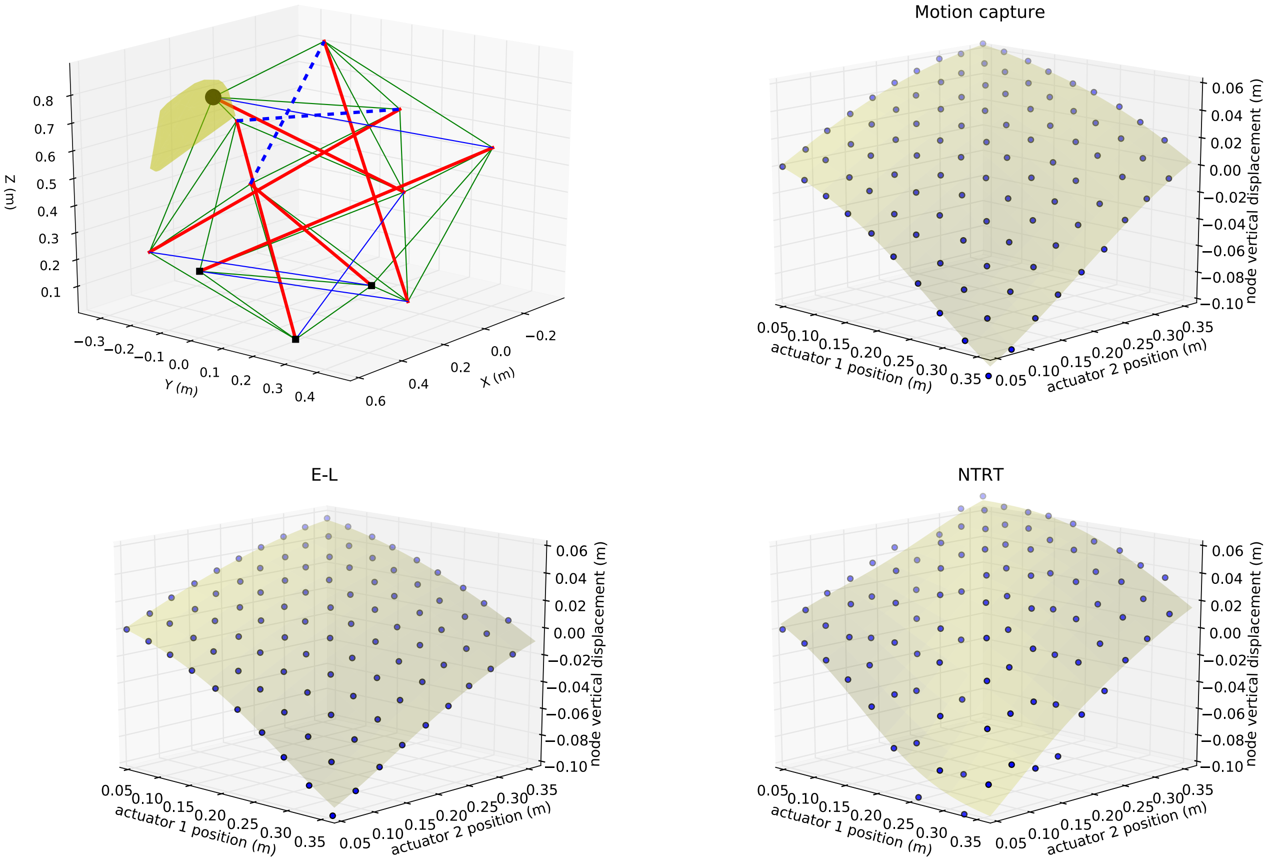


Figure 16: Kinematic comparison of the Euler-Lagrange and NASA Tensegrity Robotics Toolkit simulators and ReCTeR motion capture data. The top left plot shows the experimental setup. The rest length of two actuated spring-cable assemblies (dashed lines) is modified. The full range of motion of the tracked end cap during the experiment is shown in light yellow (convex hull). The end caps indicated by small black squares are on the ground. The 3 other plots show the vertical displacement of the end cap indicated by the large black dot in the top left plot as a function of the lengths of the two actuated cables. The end cap of which we trace the displacement is not directly actuated and is floating. The nodal displacement as a function of the actuator position is non-linear, even for modest displacements. Note that the leftmost point (0.05,0.05,0) is the reference point, as the displacements are relative to this initial state.

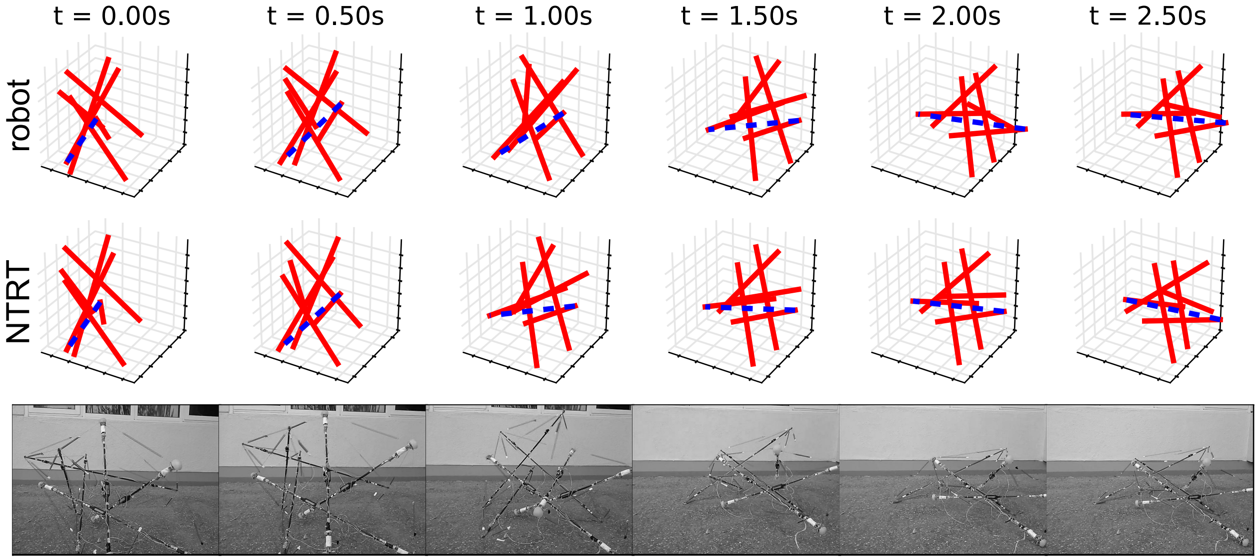


Figure 17: Comparing robot and NTRT dynamics. The tensioned spring-cable assembly indicated by the dashed line is released (0.32m to 0.535m at 0.6m/s), causing the robot to topple. Two other actuated members are also tensioned, while the other three actuated springs are at their initial lengths, resulting in two slack springs. We observed a time averaged error of the end caps' vertical positions of less than 5% of the robot's diameter for all end caps.

7.2.3 Dynamics

Next, we compared the dynamics of the NTRT simulator to the ReCTeR hardware. This experiment was designed for two purposes: to verify first that the simulator can replicate ground interactions, and second, that it can simulate the conversion of potential energy into kinematic energy when a spring is released. The experimental setup is shown in Fig. 17. The robot initially has a non-minimal ground contact (four end caps on the ground instead of just three), and three of the actuated springs are tensioned. Next, one of the tensioned, actuated springs is loosened by its actuator, which causes the robot to roll over.

As the experiment also depends on the initial state of the robot, the observed initial state from the motion capture data was copied to the NTRT simulator. The ReCTeR model in the NTRT simulator was then released from this initial configuration, allowing it to reach the simulated, predicted equilibrium. The recorded motor positions from the physical test were then applied into the simulator, causing a similar rolling-over motion.

Our conclusion from these experiments is that, within reasonable error margins, our various simulators are doing a good job of predicting the behavior of the actual hardware prototypes. While there is remaining un-modeled error, it is small enough that we can use our simulators to explore control architectures with confidence that the resulting approaches are reasonable for use. Clearly, the resulting controllers will need to be ported, tuned, and validated on the hardware prototypes. Furthermore, we can use our simulators for initial estimates for engineering requirements (within reasonable safety margins) for our robots, such as motor specifications, sensors, and power requirements. As we continue to develop prototypes and refine our models, we expect that the remaining differences will slow be reduced as the technology matures towards mission readiness.

8 Dynamical and Central Pattern Generator Controls

This section summarizes the results from the line of controls research which focused on dynamical systems, including some work done in Phase 1, and the most recent work from Phase 2. The controls used by our Phase 1 evolutionary methods were a simple form of semi-distributed oscillatory control: All the actuations were controlled by sine waves synchronized through their relative phase shifts. These controls proved to be robust and sufficient for our goals of having our tensegrity platform roll smoothly. They also gave excellent examples and insight into what fundamental dynamics were required for fast smooth motion. On the other hand, a major limitation of these evolved controls is that they do not provide a means to direct or steer the motion of the tensegrity probe. Rather, a control law is learned which can be abstracted, but which does not allow reactive changes of direction. To develop steerable controls we took the insights gained from the evolved controllers and hand crafted reactive dynamical controllers that used a variety of simulated sensor feedbacks to enable steerable rolling. This approach was very successful and was shown to work on a variety of terrains, including moderate hills (Figure 21). But, this approach relied on significant amounts of sensor feedback which may be difficult to implement, so we wanted to find a hybrid approach that enabled steering while using a minimum amount of sensor feedback. Central Pattern Generators (CPG's) are able to store complex gait cycles in their network dynamics allowing for a significant reduction of sensor feedback required. We explored a variety of ways to use this property and to combine it with aspects of the dynamical controls approach to enable steering.

8.1 Bio-inspired control

The idea behind these control laws is to create a torque by moving the center of mass of the robot with respect to the ground contact surface, in order to cause the robot roll as illustrated in Fig. 18. State feedback will be used to increase the rolling performance of the tensegrity robot with payload simulated with the NTRT (Fig. 15). This motion is achieved by a two layer control architecture: the robot's heading and speed is controlled by the displacement of the central payload using the inner spring-cable assemblies, and the motion is simplified by actuating the outer shell.

Three control approaches are tested for the inner spring-cable assemblies: reactive controls, inverse kinematics (IK) based controls, and CPG-based controls. The outer spring-cable assemblies are controlled with a hand-tuned approach. Actuation of the outer shell reduces ground contacts and does not directly influences heading or speed. This affects the motion in several ways. First, it allows the creation of greater torques with the same payload displacement. Secondly, it smoothens the rolling behavior of the structure, by preventing discontinuities due to excessive ground contact.

For each of the control approaches, inputs were taken as functions of the robot state. The height of each strut is computed using red simulated omnidirectional distance sensors located at the end of each rods. The height assigned to each spring-cable assembly is computed as the average of the two end points' height. We note that omnidirectional distance sensors can be difficult to realize in hardware; it is not practical to rely on the full state of the robot as input. However, multiple solutions to this problem exist. An interesting approach is to embed ground reaction force sensors in a protective soft cap on each rod end. A second possibility, motivated by the separation principle of control theory [13], is to estimate the system states from other sources such as accelerometer

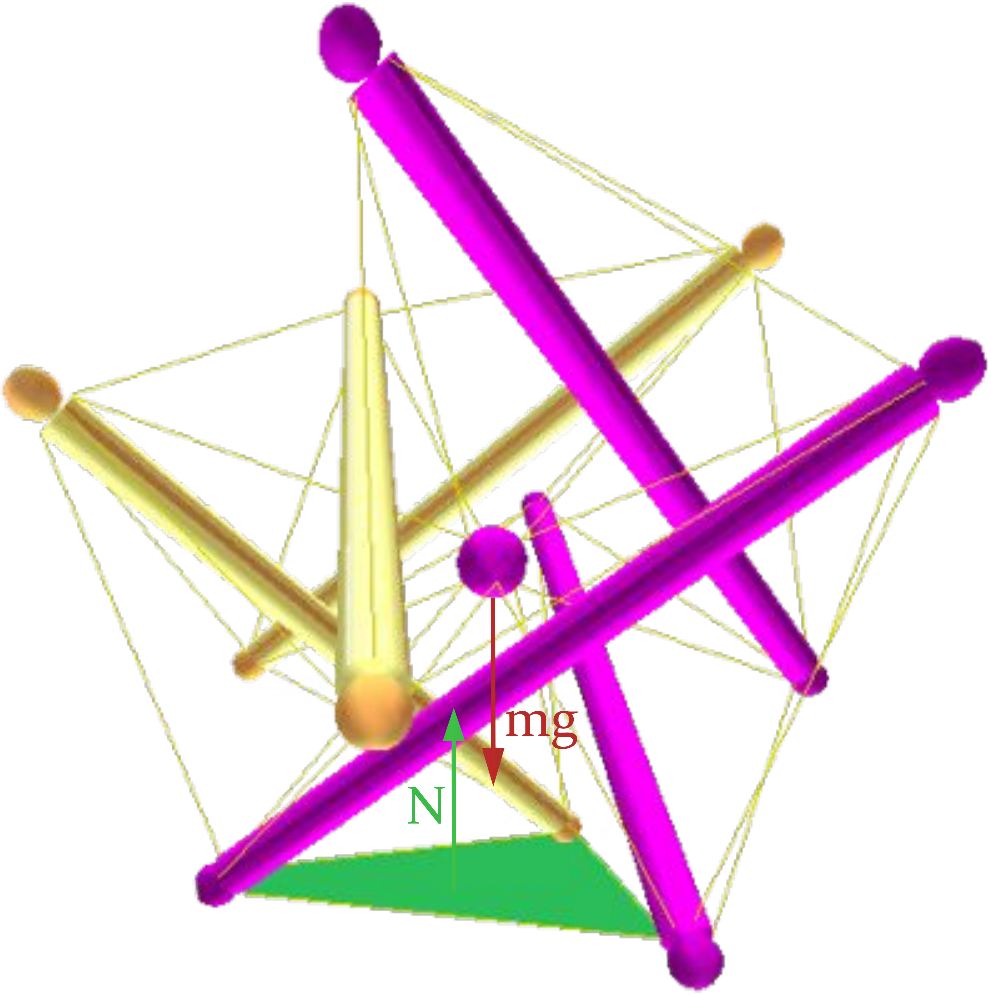


Figure 18: Regular icosahedron tensegrity shape with central payload (see Fig. 15 center). The triangular contact surface with the ground, highlighted in green, creates a reaction force N that, at rest, balances the weight of the structure, represented on the figure by the red arrow mg. A torque is created on the whole structure when a displacement of the center of mass from its rest position occurs.

and gyroscope measurements. This second approach could be augmented with the knowledge that this icosahedron robot often rolls over an edge of a face triangle [48]. Finally, one could use a CPG to emulate the rhythmic activation of the sensors, similar to the approach in Section 8.3.

The control for the outer shell cables was designed to tighten the bottom part of the structure when rolling, changing the lever arms of the gravitational force from the robot's center of mass, such that the robot does not require as much force to induce a roll. Typically in the presence of a slope, the reduction of the ground contact surface is sufficient to cause the robot to roll down the slope. In order to take this into account, we added a measure of speed, which is computed as the dot product between the center of mass position and the robot's overall heading direction vector. With this method, the speed is a scalar number and its sign depends on the heading of the tensegrity (positive if heading in the desired direction and negative otherwise). It can thus be used as feedback to influence the spring actuator command. The rest lengths of the *shell* spring-cable assemblies are computed using the following actuation rule:

$$\begin{cases} \dot{\ell}_i = w_s \left(\ell_0 + \min(h_i^2, h_0^2) - \ell_i\right) & \text{, speed} \geq 0 \\ \dot{\ell}_i = w_s \left(\bar{\ell} - \ell_i\right) & \text{, otherwise} \end{cases} \tag{5}$$

where h_i is the height of spring-cable assembly i as measured from the distance sensors, ℓ_i is its current rest length, ℓ_0, h_0 and $\bar{\ell}$ are constant parameters and $w_s \in \mathbb{R}_+$ accounts for the time scale at which length corrections occur. ℓ_0 and h_0 represent the offset rest length of the spring and the maximum height measurement, respectively. The parameter $\bar{\ell}$ represents the default rest lengths of the springs that, if given as a command to all motors, puts the tensegrity in a stable position on the ground. The input and output parameters of the above control law are updated continuously through feedback control. Impedance control, which was adapted to tensegrities previously [63, 85], is used to modify the spring-cable rest lengths.

8.2 Reactive controls

The first technique for actuation of the inner payload spring-cable assemblies was the use of reactive controllers. We note that the only controllable parameter is the length of a cable. The variables ℓ_i here are the rest length of the inner springs. The global heading direction in a chosen inertial reference frame is defined by the unit vector $\mathbf{v}$ and the orientation of each spring in this same reference frame is represented by the vector $\mathbf{v}_i$. For each *inner* spring-cable assembly we use the dot product $d_i = \mathbf{v} \cdot \mathbf{v}_i$ as feedback to control the position of the payload as follows:

$$\begin{aligned} \dot{\ell}_i &= (\ell_0 + d_i\gamma - \|\mathbf{p}_{i,0} - \mathbf{p}_{i,1}\|)w_r & (6) \\ \ell_i(0) &= \ell_0 & (7) \end{aligned}$$

where the weight w_r determines the reactivity of the system and $\gamma < 0$ is a fixed parameter. Thus, without any external perturbation, the system has a stable equilibrium position at $\ell_0 + d_i\gamma$. The rest length of the spring-cable assemblies of which the orientation aligns with the global heading, is reduced. Vice versa, the springs pointing in the opposite direction are elongated. The global result is a displacement of the payload in the direction of the heading vector as shown in Fig. 19. Note that the heading direction $\mathbf{v}$ can be chosen arbitrarily and can be adjusted dynamically. This method resulted in stable and smoothly rolling gaits allowing the tensegrity to roll up to 1m/s (1

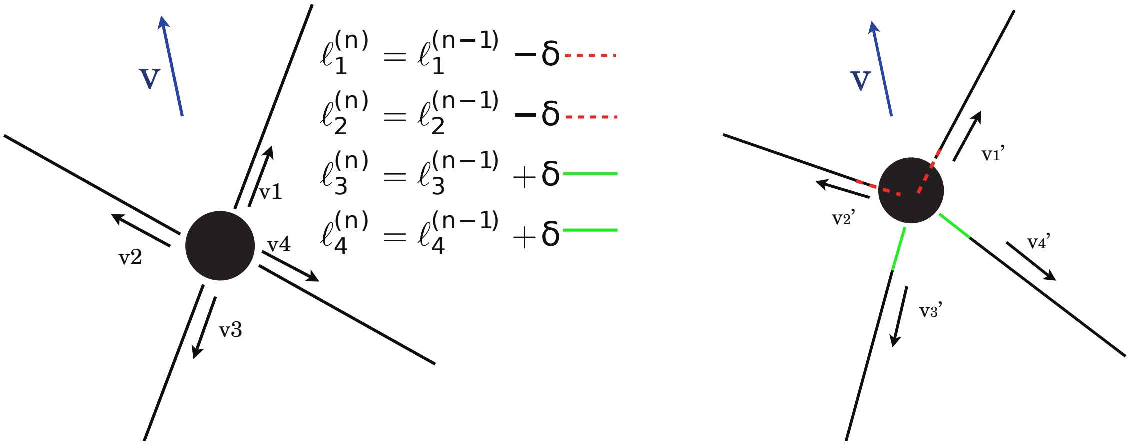

V
v1
v2
v3
v4
$\ell_1^{(n)} = \ell_1^{(n-1)} - \delta$
$\ell_2^{(n)} = \ell_2^{(n-1)} - \delta$
$\ell_3^{(n)} = \ell_3^{(n-1)} + \delta$
$\ell_4^{(n)} = \ell_4^{(n-1)} + \delta$
V
v1'
v2'
v3'
v4'

Figure 19: computation of the new rest lengths according to the spring-cable assemblies' individual orientations $\vec{v}_i$ (time $t^{(n-1)}$). Length modification is indicated by the colored lines, dashed red if reduced and green if elongated. The resulting effect is a displacement of the central payload in the desired direction $\vec{v}$ (time $t^{(n)} = t^{(n-1)} + dt$).

body length per second) over flat terrain. The robot could also handle slopes up to $8°$, bumpy terrain, obstacles and collisions.

The main disadvantage of the reactive method is the type and amount of sensor feedback required to implement this approach in hardware. This issue is addressed by the control methods presented next, which are based on the same physical principle but require less feedback information.

8.3 CPG controls

Central Pattern Generators (CPGs) have been successfully used in past tensegrity systems [8]. Such controls are a feasible alternative to the reactive controllers that enable the generation of regular motion patterns. For this control, full state information is used to generate a smooth motion under the reactive controls, and then the resulting periodic commands were stored as a stable limit cycle of a CPG. Once this process is over, the tensegrity can be driven by the CPG output with much less feedback. We used an adaptive frequency Hopf oscillator [75] during the learning phase where the tensegrity is driven by the reactive controls. The underlying dynamical system reads:

$$\dot{u} = \gamma(\mu - (u^2 + g^2))u - \omega g + \epsilon b(t) \tag{8}$$

$$\dot{g} = \gamma(\mu - (u^2 + g^2))g + \omega u \tag{9}$$

$$\dot{\omega} = -\epsilon b(t)\frac{g}{u^2 + g^2} \tag{10}$$

where u, g and ω are the state variables of the dynamical system, γ is a time constant, μ is the target frequency and ω the target pulsation of the signal. Note the element and time indices are dropped above to simplify the notation: u designates $u_i(t)$, with i the index of the spring-cable assembly. The output u has the ability to synchronize to any periodic input signal $b(t)$ and replaces the feedback signal from the previous section.

$$d_i(t) = u_i(t). \tag{11}$$

Once the signal is learned, the time dependency of the pulsation is removed, i.e. ω is held constant, and a term accounting for coupling with ground contact is added to the dynamical system:

$$\begin{align}
\dot{u} &= \gamma(\mu - (u^2 + g^2))u - \omega g - \eta h(t) \tag{12}\\
\dot{g} &= \gamma(\mu - (u^2 + g^2))g + \omega u \tag{13}\\
\dot{\omega} &= -\epsilon b(t) \frac{g}{u^2 + g^2} \tag{14}
\end{align}$$

where $h(t)$ denotes the height signal fed back by the omnidirectional ranging sensors and $\eta \in \mathbb{R}_+$ is a coefficient. This method has the advantage of requiring minimal feedback and thus only a small amount of computations. However, it is important to note that the dynamical system runs on a much larger time scale than the perturbations disturbing the system. A tensegrity driven only by a CPG would then only, in the best case, have a stable rolling gait on a flat, obstacle free terrain. As a result, it is necessary to include also a second control method that can work on this smaller time scale and give an appropriate response to these external perturbations.

8.4 Hybrid CPG - Inverse Kinematics controls

The final control method tested here is a hybrid technique with inverse kinematics. First, the position of the central payload $\mathbf{p} = (p_x, p_y, p_z)$ is defined as a function of the inner cable lengths $\boldsymbol{\ell} = (\ell_1, ..., \ell_n)$. We can this write a small displacement $\delta\mathbf{p}$ of the payload as:

$$\begin{align}
\delta p_i \approx p_i(\boldsymbol{\ell}^{(0)}) \quad &+ \quad \sum_{j=1}^{n} \frac{\partial p_i(\boldsymbol{\ell}^{(0)})}{\partial \ell_j} \delta \ell_j \nonumber\\
&+ \quad \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 p_i(\boldsymbol{\ell}^{(0)})}{\partial \ell_j \partial \ell_k} \delta \ell_j \delta \ell_k \tag{15}
\end{align}$$

or

$$\begin{align}
\delta p_i \approx p_i(\boldsymbol{\ell}^{(0)}) \quad &+ \quad \left(J(\mathbf{p}^{(0)})\delta\boldsymbol{\ell}\right)_i \nonumber\\
&+ \quad \frac{1}{2} \delta\boldsymbol{\ell}^T H\left(p_i(\boldsymbol{\ell}^{(0)})\right) \delta\boldsymbol{\ell} \tag{16}
\end{align}$$

for $i \in \{x, y, z\}$ and where $H(p_i) = \left[\dfrac{\partial^2 p_i}{\partial \ell_j \partial \ell_k}\right]_{jk}$ is the Hessian matrix associated with p_i.

Considering Eq. 16, we additionally define $\Delta\mathbf{p} = \delta\mathbf{p} - \mathbf{p}(\boldsymbol{\ell}^{(0)})$ and $\mathbf{f}(\delta\boldsymbol{\ell}; \Delta\mathbf{p})$ as

$$f_i(\delta\boldsymbol{\ell}; \Delta\mathbf{p}) = \left(J(\mathbf{p}^{(0)})\delta\boldsymbol{\ell}\right)_i + \frac{1}{2} \delta\boldsymbol{\ell}^T H\left(p_i(\boldsymbol{\ell}^{(0)})\right) \delta\boldsymbol{\ell} - \Delta p_i. \tag{17}$$

	reactive	CPG	Hybrid
average speed [m/s]	1.00	0.50	0.38
complex terrain	Yes	No	No

Table 2: Bio-inspired control strategies summary

Computation of the spring-cable rest length changes $\delta\boldsymbol{\ell}$ for a desired payload displacement $\Delta\mathbf{p}$ corresponds to finding the $\delta\boldsymbol{\ell}$ which cause $\mathbf{f} = \mathbf{0}$. As this last equation is over-determined, non-linear, and might not possess a real solution, we employed a quasi-Newtonian iterative method to approximate the solution. Starting from a candidate solution, e.g. $\delta\boldsymbol{\ell}_0 = \mathbf{0}$, the next iteration is computed as:

$$\delta\boldsymbol{\ell}_{k+1} = \delta\boldsymbol{\ell}_k - J_k^{-1}\mathbf{f}(\delta\boldsymbol{\ell}_k; \Delta\mathbf{p}) \tag{18}$$

until convergence. J^{-1} here denotes the Moore-Penrose pseudo inverse of the Jacobian defined by

$$J = \left[\frac{\partial f_i}{\partial(\delta\ell_j)}\right]_{ij}. \tag{19}$$

Note that this matrix is not the same as the one in the Taylor expansion of $\mathbf{p}(\boldsymbol{\ell})$.

The outputs of the IK algorithm $\xi = \delta\boldsymbol{\ell}_\infty$ (where ∞ indicates convergence) represent the length corrections that have to be made to reposition the payload at the desired location. The outputs ξ can be used together with the adaptive frequency oscillator as presented in Section 8.3. This approach is inspired by two previous works by Ajallooeian et al. [3] and Gay et al. [29]. We update ξ if and only if the payload position lies on the opposite side of the robot's center of mass, as well as continuously adjust ξ with time according to the following evolution rule:

$$\frac{d}{dt}\xi(t) = -\alpha\xi(t) \tag{20}$$

with $\alpha \in \mathbb{R}_+$. In this way, the corrections are made only if the tensegrity can potentially roll in an undesired direction. Note also that in order to use this method, both the position of the payload and the center of mass are required inputs. Combining a corrective term with the output of the oscillator, the resulting dynamical system reads:

$$\dot{u} = \gamma(\mu - (u^2 + g^2))(u - \xi(t)) - \omega g - \eta h(t) \tag{21}$$

$$\dot{g} = \gamma(\mu - (u^2 + g^2))g + \omega(u - \xi(t)), \tag{22}$$

If the value of $\xi(t)$ is constant over time, the dynamical system converges asymptotically to $u(t) = \xi$ [29]. While the pure CPG implementation does not allow any steering control of the robot, this implementation enables the guidance of the robot on a desired trajectory on flat terrain (see Fig. 20). Table 2 provides a summary of the results obtained with the different control strategies over regular flat terrain.

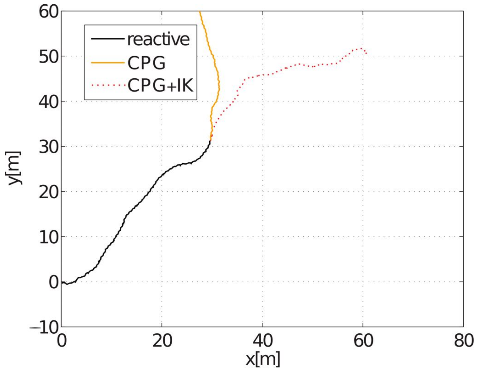


Figure 20: Trajectory of the tensegrity (top view). The black curve represents the trajectory while the robot is driven by the reactive control algorithm and the CPG is in the learning mode (50s). The motion is regular and the heading is maintained throughout the whole period. The yellow and red trajectories represent the path traveled once the CPG controller takes over (40s). When the CPG is coupled to the height signal and receives inputs from the second order inverse kinematics algorithm (red curve), the resulting trajectory is a long and relatively straight line extending well the reactive control.

Note that the results do not take into account the trajectory of the path and, as a consequence, even if the distance traveled using the CPG controller without any trajectory control is larger than with the hybrid control, the "quality" of the path is not as good (see e.g. Fig. 20). Interestingly, we observe that the stable gait pattern obtained in simulation is a sequence of contacts defined as energetically optimal by Koizumi et al. [48] for a tensegrity icosahedron. With the current implementation, only the reactive controller manages to get the robot to roll in an efficient way over complex terrain and obstacles. To the best of our knowledge, this last result is the only implementation of a tensegrity robot controller demonstrating such capabilities. Fig. 21 presents such results from within the NTRT simulator.

Experimentation showed that the hybrid controller's performance is highly sensitive to the choice of some parameters appearing in the CPG equations, such as the ones presented in Eq. 21 and 22. As a result, we looked to incorporate other methods to optimize the feedback data and to compute the corrections, in order to more accurately navigate in complex environments. A good example of such an improvement can be found in Gay et al. [30] where sensory information is preprocessed by a neural network and trained using particle swarm optimization methods before being fed back to the CPG. In the same idea, our next step was to turn to Reservoir Computing which can also be a suitable tool for feedback computation, as detailed in the following section.

8.5 Learning a Matsuoka oscillator with Physical Reservoir Computing

This section presents the final controls result of our work on dynamical system approaches: an implementation of the Physical Reservoir Computing (PRC) principle on the ReCTeR hardware prototype. These results validate the use of the NTRT in an untethered, underactuated feedback control of tensegrity icosahedra with string force sensors [19, Section 5.1.1]. Here, closed-loop feedback control is used in which the motor signals are generated by a Matsuoka oscillator. This

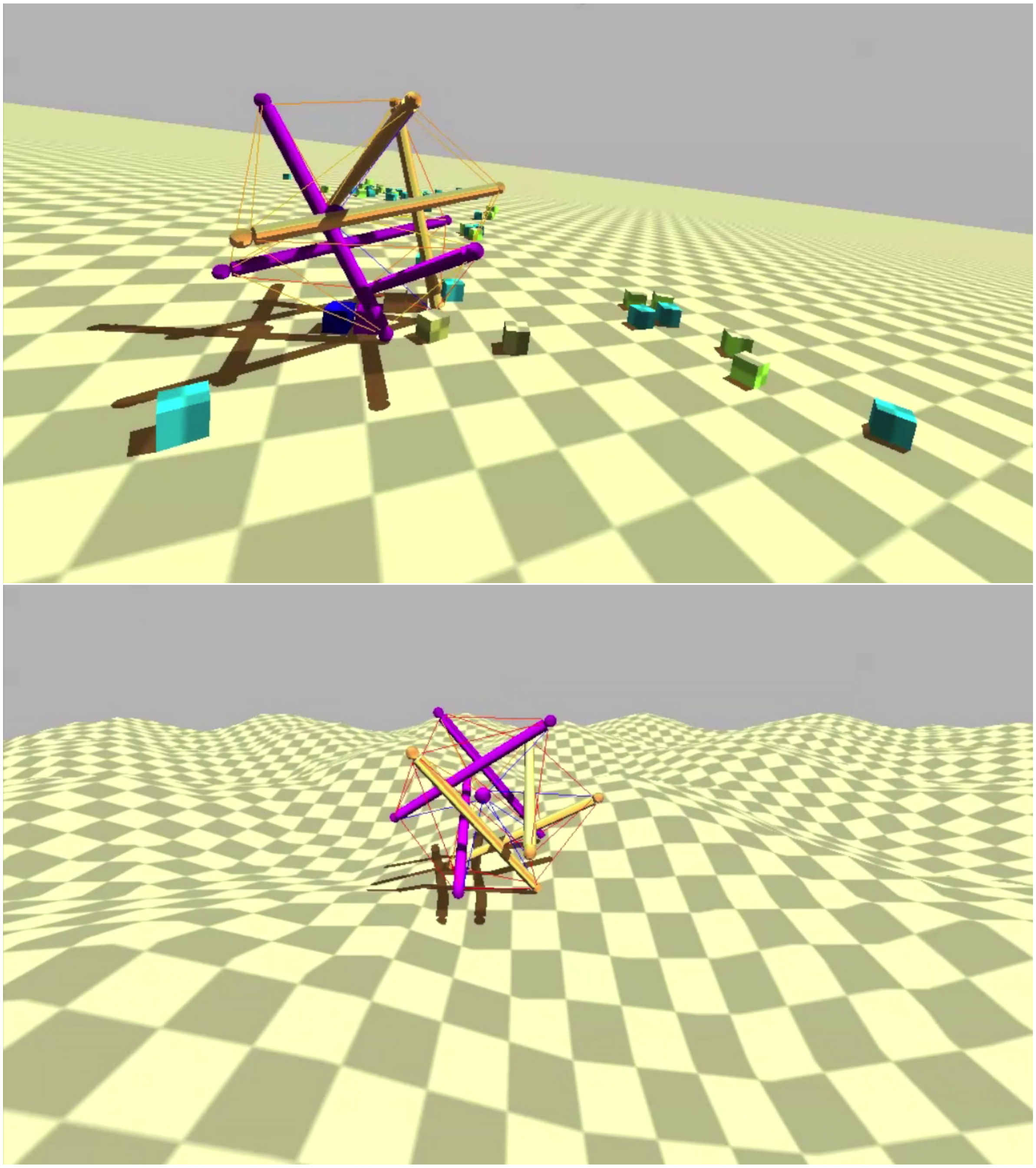


Figure 21: Examples of successful locomotion over complex terrains such as slopes, bumps and obstacles

demonstrates a successful adaptation of a physical platform (ReCTeR) to our simulation results, with similar learning times and robustness.

A static linear feedback controller is designed, which robustly generates a set of desired oscillatory motor signals after a short learning phase. For this experiment, the target spring-cable rest lengths (ℓ_i) are generated using a random Matsuoka oscillator [55] (see [19, Section 3.1] for oscillator parameters and motivation). These signals represent the desired actuator signals. We manually scaled the target signals such that the resulting behavior corresponds to a motion pattern with large shape deformations, while keeping the physical structure from moving too fast (as this impeded motion tracking). The resulting gait was a slow crawling motion, which allowed motion-capture tracking of the full experiment.

The algorithm proceeds by first applying the target rest lengths to the actuators in an open loop setup, which causes the robot to start moving. Next, online learning is applied to approximate the desired signals based on the sensor readings. These approximations are the feedback signals. The ratio of open loop versus feedback signals is gradually decreased until the signals are generated by the feedback loop alone. At this point, the robot will robustly maintain the oscillatory patterns. The precise equations and parameters used in the experiment are provided in [19] and the supplementary material.

In our prior simulation work, we used the term Physical Reservoir Computing to describe how the nonlinear computations, which are inherently performed by a physical system, can be easily exploited to simplify the control of tensegrities [19]. PRC extends the Reservoir Computing (RC) concept, which is at its origin a simple technique to train recurrent neural networks [90, 44]. The common idea is that a system with complex dynamics is perturbed externally but left untouched otherwise. Instead, a simple readout mechanism is trained to perform the desired computational task. A number of related demonstrations have recently appeared in the soft robotics fields, e.g. RC applied to a soft, simulated octopus arm [61, 60].

The controller feedback signals are obtained from ReCTeR's 24 force transducers. As these sensors are mounted perpendicularly to the robot's struts, the output values depend on the angle of attack and the tension of the attached spring-cable assembly. Thus, the sensors provide a readout of the robot's state, similar to the state observations in RC. The robot's behavior was evaluated using the motion capture setup described in Section 7.2.1.

Fig. 22 shows the result of an experiment in which we first outsource the motor signal generation to the feedback loop by online learning of the feedback weights. After training, we disturb the system (lifting the robot up and constraining it). The robot stops moving in this case and switches back to its original mode of oscillation when it is released, demonstrating the robustness of the learned feedback controller, corresponding to our simulation results.

This experiment is a first demonstration of a simple, robust feedback control strategy implemented in both hardware and simulation for this class of untethered tensegrity robots. Additionally, this result shows the usefulness of tension sensors for tensegrity robot control. These PRC experiments are part of a continuous effort to develop low-level controllers for compliant robots that maximally exploit the robots' proper dynamics, and which allow to mitigate stringent sensor requirements. We discussed many variations and extensions on the hardware experiment presented here in our prior simulation work [19].

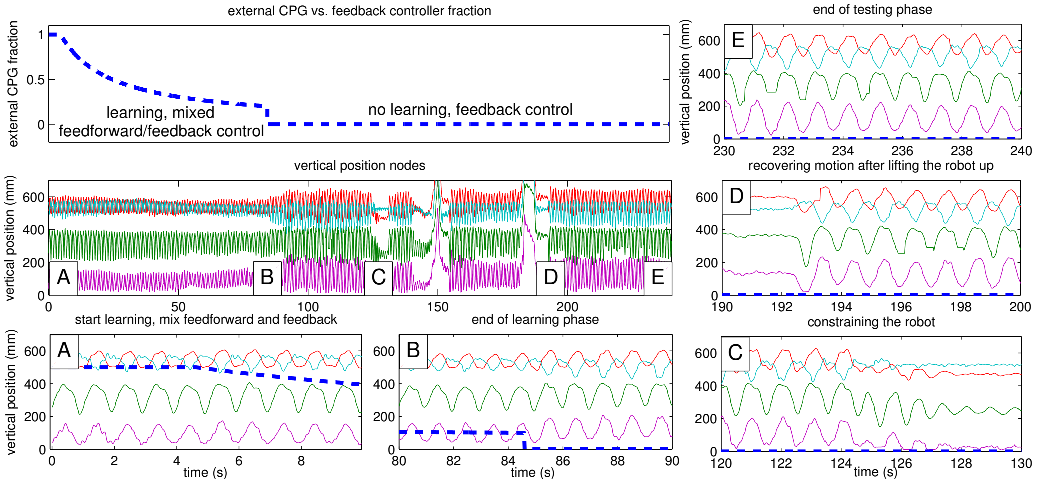


Figure 22: Fast online learning of a static feedback controller for a Matsuoka oscillator on the ReCTeR robot based on uncalibrated strain gauge sensors. The top left plot shows the fraction of feedforward vs. feedback control. During learning, both the feedback and feedforward controllers (training signals) are active. The influence of the open loop feedforward controller decreases and when its fraction is below 0.2, learning stops and only the trained feedback controller is active. The left plot on the second row shows the vertical coordinates (in mm) of the four end caps with the largest vertical displacements as a function of time. The five surrounding plots are details of this plot, showing the different training and testing phases. A) Fully open loop control. B) Switch from partially open loop and feedback control to full feedback control, learning stops. C) We perturb the robot by pushing it down, preventing all movements. D) The feedback controller recovers after the robot was lifted up from the ground. E) The behavior of the robot after 250s (170s closed loop).

9 Evolutionary Controls

In this section we explore the use of Evolutionary algorithms to develop controllers for surface mobility of the tensegrity probe once on Titan. We initiated this line of research in Phase 1, and developed a significant new algorithm during Phase 2, which is reported here. Evolutionary algorithms, genetic algorithms and reinforcement learning are all part of the broad field of machine learning that can be applied to sophisticated control problems. What makes these algorithms unique is that the control solution is not created in a top-down manner, but is instead "evolved" or "learned." These methods of controls are ideally suited for control problems that exhibit complex and non-linear characteristics making it difficult to apply traditional control algorithms. In this report, we describe how we successfully applied "evolutionary" algorithms to the task of controlling a tensegrity robot. The evolutionary aspect comes from our use of a population of control policies that is then slowly evolved to create populations of higher and higher performance. Note that this work does not exclude the possibility of using reinforcement learning for this problem, which would involve the tensegrity robot exploring the environment and learning to improve its control policy through negative and positive rewards.

9.1 Target Tensegrity Platform

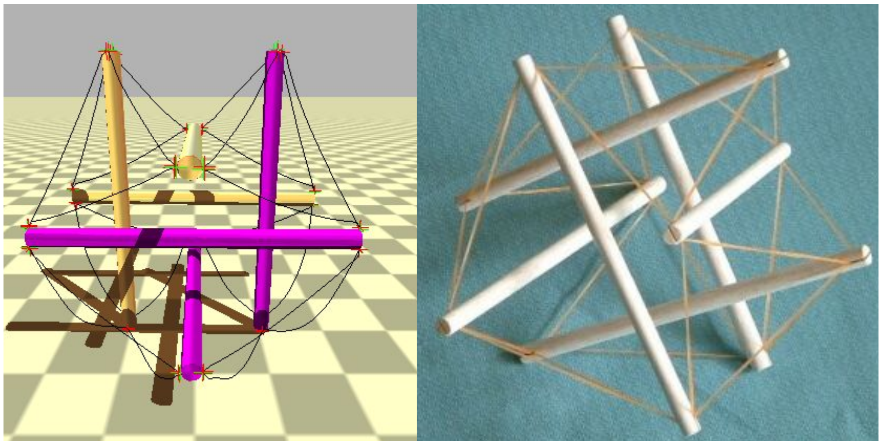


Figure 23: **Structure for Tensegrity Robot.** *This six-rod design is one of the simplest designs that can behave as a "ball." It is capable of rolling, changing shapes, and can be robust against failures.*

Our experiments for evolutionary controls of a tensegrity robot involves our basic "6-bar" tensegrity used throughout most of this report. This platform has 6-rod and 24-cable tensegrity as shown in Figure 23.

9.2 Tensegrity Actuation

The tensegrity is controlled by changing the lengths of the cables. Many hardware implementations do this by using a motor to wind the cable onto a spool that is either interior of the tensegrity structure or inside a rod. Other concepts involve using dynamic cable twisting or elastomers to change the length of the cable. In this section we do not consider the hardware implementation, though we do limit our abstract model of actuation to reasonable performance characteristics for velocity, acceleration, and string elasticity.

9.3 Flop And Roll

Rolling as a means of locomotion has many advantages. First, it uses gravity to its own advantage as the main driving force. In addition, this locomotion does not have balancing issues. Considering tensegrity structures with a near spherical shape, it is the most natural way to move.

On the other hand, the locomotion algorithm to roll for a tensegrity robot is not trivial. The structure has to deform itself by changing the lengths of the actuated cables to create any motion. Finding the right configuration to create motion in a direction is already a hard problem, it consists of 24 interconnected elements that all affect each other, resulting in a nonlinear system. Compared to deformation, figuring out continuous and smooth rolling adds inertia as a component to the problem, which creates another level of complexity.

The first step that we take is to divide the problem of rolling into consecutive flops. Considering that the structure is stable on one of its surfaces, we define 'a flop' as deforming the structure and falling to one side only to end up lying on another surface. Doing one flop towards the target will move the robot towards the flop direction and change its orientation. Following the same routine over and over will end up moving the tensegrity robot in the desired direction. Learning to do a single flop in a desired direction is a simpler problem than learning smooth rolling.

On the other hand, learning a single flop and repeating it does not necessarily provide a smooth rolling locomotion. As an analogy, the difference is similar to the difference between repeating the routine 'one step forward and stop' and smooth walking. During smooth walking, steps taken are optimized for consecutive steps and they differ from taking one single step forward. To avoid such a difference, we chose our fitness function to evaluate overall rolling. During the evolution of policies, the policy that we evaluate makes a single flop, but these single flops will evolve according to their success when they are executed consecutively over 60 seconds. With the same analogy of walking, we evolve the robot to learn how to make a step, but the policy to perform each step is evolved so that it will maximize the walking behavior when executed over and over. The details of the learning algorithm and the state and the fitness function will be explained in section 9.5

The first advantage of the approach that we take is making the control policy simpler (single flop), while learning a more complex behavior (smooth rolling). Additionally, the algorithm can handle external and unexpected forces during rolling motion. This robustness is mainly provided by the fact that the algorithm is composed of smaller pieces to make each flop as opposed to previous research that provides the whole rolling sequence on a given surface [41]. Let's imagine a tensegrity robot that encounters a large external force while rolling using "flop and roll" towards a target point. The external force applied will break the sequence of flops for the robot, and the robot will end up in a random orientation. Since the robot has a spherical and symmetrical nature, it will land on one of its faces. The robot then executes the algorithm in the new orientation, and will deform itself to undertake the first flop that will be followed by a rolling behavior towards the desired target point.

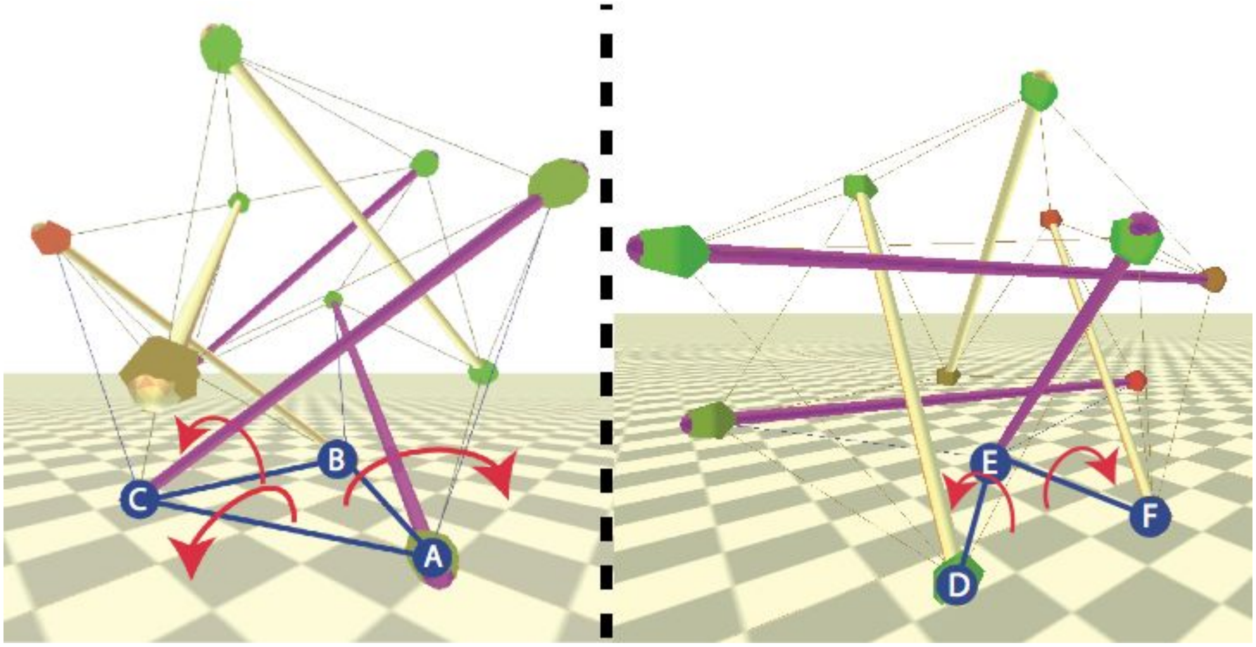


Figure 24: Possible bases and flops for an icosahedron tensegrity robot. There are two types of possible base triangles when the robot is balanced: an equilateral triangle (left) or an isoceles triangle (right). For the first configuration (left), there are three possible flops (over AB, BC, or CA). For the second configuration, there are two possible flops (over DE and EF). We do not consider the flop over DF since it requires a lot more deformation.

Finally, we test the robustness of the algorithm with an environment where executing a flop is harder due to different terrain properties or external forces. As learned, the robot will try to execute the single flop until it lands on a different surface. The algorithm considers itself at the same state and works on completing the flop until it succeeds. This property allows the algorithm to work on uneven terrains, hills, small obstacles, and unexpected external forces as shown in the results in section 9.6.

9.4 Distributed Controls via Pooling

We first discuss how to control the robot using distributed controls while still taking advantage of the symmetrical nature of the structure. The ideal control algorithm for the robot provides rolling motion that is steerable in a desired direction and robust to external forces. This is the main significance of the rolling algorithm that we present in our paper. In terms of input, the algorithm takes sensor information from the robot. Traditionally, robots are equipped with many different sensors, such as cameras and infrared sensors. We will use the minimal information, such as pressure or contact sensors, at the end of the rods and the desired direction.

The robot has 24 cables that are controlled by 24 independent controllers. Each controller is responsible for selecting the desired length for the cable that they control. One important point here is that the algorithm does not directly control the lengths of the cables at a given time. Since the algorithm is based on simple flops for the robot, the lengths provided are the desired lengths for the cables, so that when reached to the configuration, the structure will flop and start rolling. This particular point makes the algorithm time-independent.

The actual lengths of the cables at a particular time depends on the speed of the motors and previous configuration, but the algorithm still works with slower motors since reaching the desired configuration will make the tensegrity flop. This fact is also supported with experimental results in section 9.6. Another advantage of time-independency is that the controllers do not need to have precise synchronization. The robot can tolerate latency in coordination and will still perform the flop.

At each moment, the robot has a state (contact points and goal direction), the controllers (24 total) and actions to choose (preferred lengths). The goal of the algorithm is to provide rolling behavior that is composed of single flops as explained in section 9.3. The set of policies are defined as

$$\pi_x : (\phi, \upsilon) \rightarrow (l_x) | x \in \{1, 2, .., 24\}$$

where ϕ is the contact points and υ is the desired direction, and l_x is the desired length for the cable x. Now we will reduce the complexity of the policies to learn by what we call 'pooling', which takes advantage of the symmetrical nature-repetitive pattern of the tensegrity structures.

Let's first analyze the stable configurations for the robot in its default configuration (equal lengths for all the cables). When the structure will be stable on a surface, it will have 3 points of contact forming the base triangle. Here, we use this advantage for discretization of the state space. Instead of having the orientation of the structure, we use the base triangle that the structure is lying on. For the desired direction, we use the sides of the base triangle, which gives us 3 possible values. We define the new state variables as:

$$s : (\overline{XYZ}, d),$$

where X, Y, and Z represent the edges of the base triangle and d takes values of 0, 1 or 2 (XY, YZ, or ZX), representing the side of the triangle that encapsulates the desired direction to roll.

There are twenty possible surfaces for the icosahedron robot. Since tensegrity robots have repetitive patterns, there are only two types of base triangles: an equilateral triangle where all 3 nodes are connected, or an isosceles triangle where only two of the sides are connected. Figure 24 shows the only two types of possible base configurations. Out of twenty triangle surfaces, eight of them are equilateral (Figure 24 - left) and twelve of them are iscoceles triangles (Figure 24 - right). At any one of these stable situations, the goal of the controllers is to make the robot flop on one of the sides of the triangle. The possibilities are sides AB, BC, and CA on the left of Figure 24 and DE and EF on the right of Figure 24. We do not consider flopping over DF, because not only is DF not connected, it is impossible to perform that flop with a small deformation, since the projection of the center of mass of the structure is much closer to point E, as opposed to the edge DF.

Let's assume that all the controllers have the policy to flop in a given state of $\overline{ABC}, AB$. It can be seen that the structure is symmetrical for all 3 sides (Rotating 120 degrees around a gravitational axis will give the exact same structure). Moreover, if the structure is lying on any other equilateral triangles, we will see the same pattern of connections. This resemblance lets us reuse the knowledge of the policies for state $\overline{ABC}, AB$ for every equilateral base, and $\overline{DEF}, DE$ for every isosceles triangle. The idea is similar transfer learning. We do not directly copy the knowledge, but the policies that perform these flops are reused for flops in all possible orientations.

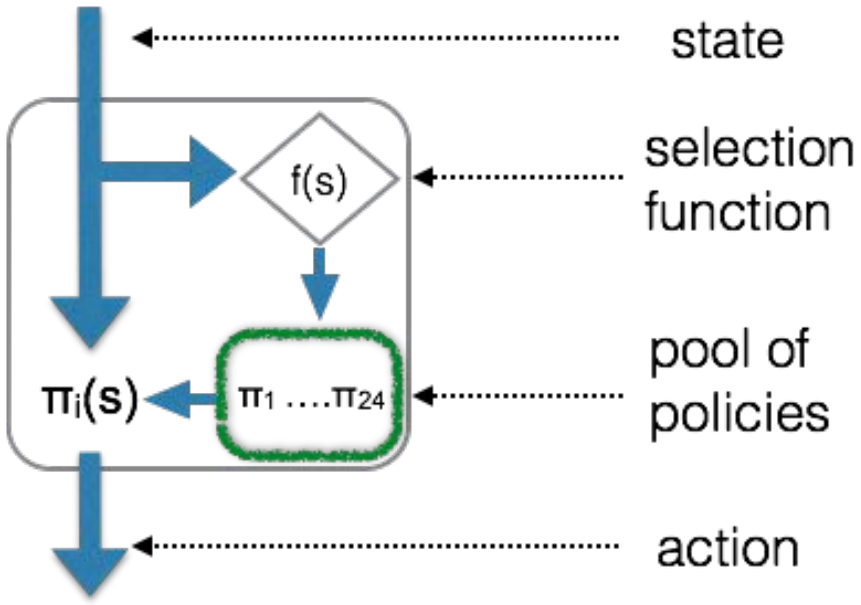


Figure 25: The decision flow for each agent in Flop and Roll. The pool of policies is same for all the agents, and the decision of policies is done according to the state.

To reuse the knowledge gathered, we use a virtual pool of policies, assuming that all the learned policies ($\pi_1, .., \pi_{24}$) for those particular states ($\overline{ABC}, AB$ and $\overline{DEF}, DE$) are available to all of the controllers. In a new state s' , we only need a function F that returns the policy that each controller select from the pool so that the desired flop will happen.

$$F : (s, i) \rightarrow (j_\pi),$$

where s is the state, i is the unique ID of the controller, and j_π is the ID of the policy that the the controller i should pick from the pool. Figure 25 illustrates the agents on the left side. These agents are used for each of the cables to perform rolling, and the pool of all the agents is updated at the end of rolling. Please note that the given method is not specific to our robot, F can easily be designed or discovered using the repetitive nature of tensegrity structures. Using pooling, we now reduced the complexity of the problem to learn 24 policies with only two different states. 24 policies combined with the pooling function F will provide the capability to flop in every possible orientation.

An alternative explanation to this pooling mechanism uses ‘roles.’ Depending on the new state, each controller selects one of the 24 roles. For example, the controllers of the base cables in current condition will select the roles of the cables AB, BC, and CD. The policies in the pool actually represent what to do for that specific role to make a flop. This selection function F, is hand-coded according to the structure, but the policies for the roles are learned using coeovolutionary algorithms, as we explain in section 9.5.

Sharing a pool of policies gives the impression of excessive communication. Yet, there is no active communication during rolling. The policies to use are decided before each episode, and they are not updated during each trial of rolling. Since we use evolutionary algorithms (section 9.5), and delayed fitness assignment, the policies are only updated before each episode. The only time that controllers have to communicate is before starting an experiment, to make sure that their pool is synchronized. Once it is synchronized, the robot can start the experiment and roll without the need of communication for policies. A small amount of communication is used to make sure all the agents figure out and learn the current state. The state is basically a binary value of a pressure sensor for each node (twelve bits total), and this communication is not sensitive to timing as discussed earlier. It is also possible to derive the state without communication (i.e., using tension sensors, proximity sensors, etc.), but we leave that for future research.

9.5 Learning to Roll via Flops

Pooling (section 9.4) reduced the problem to learn to flop with 24 agents in two possible states. On the other hand, the higher-level function to learn is to roll. We will learn the pool of policies, where each policy controls one cable of the robot for a specific orientation. We use coevolutionary algorithms since they are more suited to the distributed nature of the problem.

Evolutionary algorithms consist of the cycle of forming new members, assigning fitness, and selecting the most fit members. This cycle is called a "generation" and it is repeated until the desired behavior is obtained. In coevolution particularly, each one of the 24 policies has its own population. Each of these populations evolve separately, but the policies have to form a team for fitness assignment, because the task needs cooperation of policies to maximize the rolling behavior. At each experiment, one policy from each population is chosen randomly to form a team. This set of policies forms the pool for that experiment.

The problem is episodic, the agents have 60 seconds to test the policies in the pool. At the start of the episode all the agents copy the pool to minimize communication during rolling. At each state s, each agent calls the function $F(s, i)$ to select the policy π_j from the pool. Then agents use the policy until the state changes to a different state, meaning that the structure performed one flop and ended up on another surface.

At the end of each episode, all of the policies forming that team are evaluated according to the performance of the whole robot after 60 seconds. The global fitness function is the distance covered towards the desired goal point. Once again, fitness is not related to the flops, it is only related to the overall rolling. Although it is possible to evaluate all these policies with the same fitness function (distance traveled), a better fitness assignment can evaluate each policy according to their contribution to the overall performance. The method of providing different fitness assignments for each agent is called fitness shaping. In this work, we use historical average fitness-shaping that is previously used with tensegrity robots with success [41]. In the historical average, each policy in the pool shapes the global fitness according to its previous experience by taking the average of all the teams that it has been previously tested with.

For each generation, we test the individuals using their performances with different teams. To form teams we use random sampling. 50 random teams are formed and the members are assigned fitness according to these experiments. After 50 experiments, each population eliminates half of its members to keep the most fit ones. These selected members form new members by mutation.

9.6 Results

In this section, we present the results of the experiments that we conduct using the NASA Tensegrity Robotics Toolkit (NTRT). As discussed in section 7.2, NTRT has been validated with a physical tensegrity robot and has been shown to produce less than 1% error for a passive tensegrity and for semi-static controls of the active tensegrity robot used. In our experiments, each strut is 1.5m in length, 3 kg in weight, and the same as in the specifications of the SUPERball design. We used 24 active cables controlled by 24 controllers, where the cables have a rate of change in length of 0.3 m/s, and the elasticity coefficient for the cables is 3kN/m.

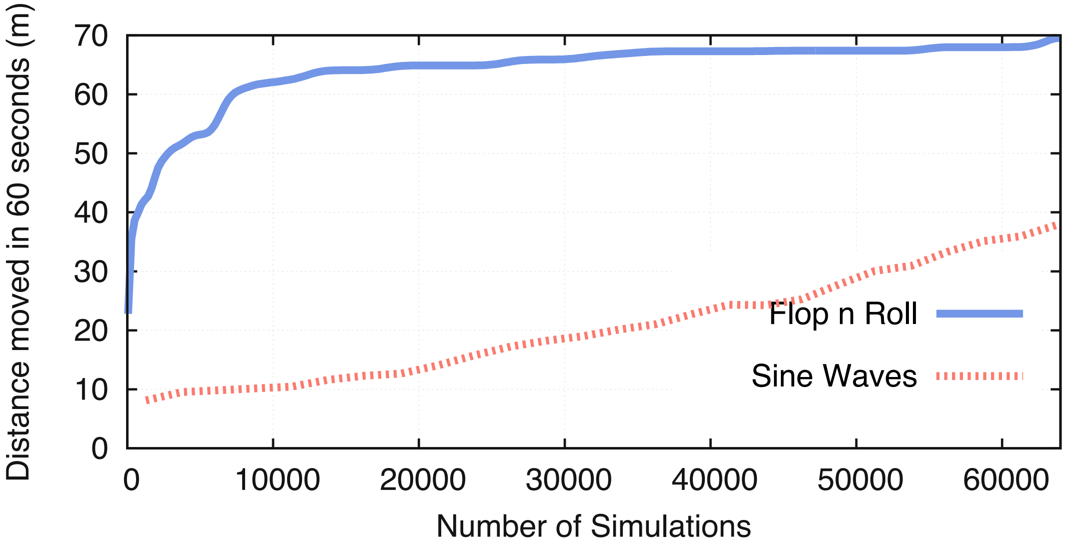


Figure 26: The performance of the best policies over time during the learning process. Flop and roll is compared to the non-directional sine wave approach. Flop and roll learns directional rolling behavior in a short amount of time.

The first experiment is to learn to roll using the algorithm ‘flop and roll,’ as described in previous sections. We compare it to the sine wave rolling algorithm that was previously presented in [41] and discussed in section 5.2. First, flop and roll is a method for learning to roll towards a goal; on the other hand, the sine wave approach that we compare it to is a less-capable algorithm that provides uncontrollable rolling motion in one direction. To be fair on the sine wave approach, we compare the distance traveled in any direction by the sine wave approach vs. the distance traveled towards the goal using flop and roll.

Figure 26 shows that the flop and roll algorithm actually takes less time to learn and performs better. The flop and roll algorithm takes advantage of the reuse of knowledge and has a learning curve with a jumpstart. Flop and roll learns a basic rolling behavior in a few generations, and it reaches rolling with a speed of 60 meters per minute towards a desired direction.

In this figure, the result of the previous sine wave approach is slightly different than the original result that was published in 2013. Since the publication of the sine wave algorithm, the realism of the simulation environment is increased using improvements that are made on NTRT. Moreover, in this work, the specifications of the tensegrity model is closer to the prototype that is currently under development [15]. In this work, we are testing the two algorithms in a more realistic environment with a robot that is harder to control.

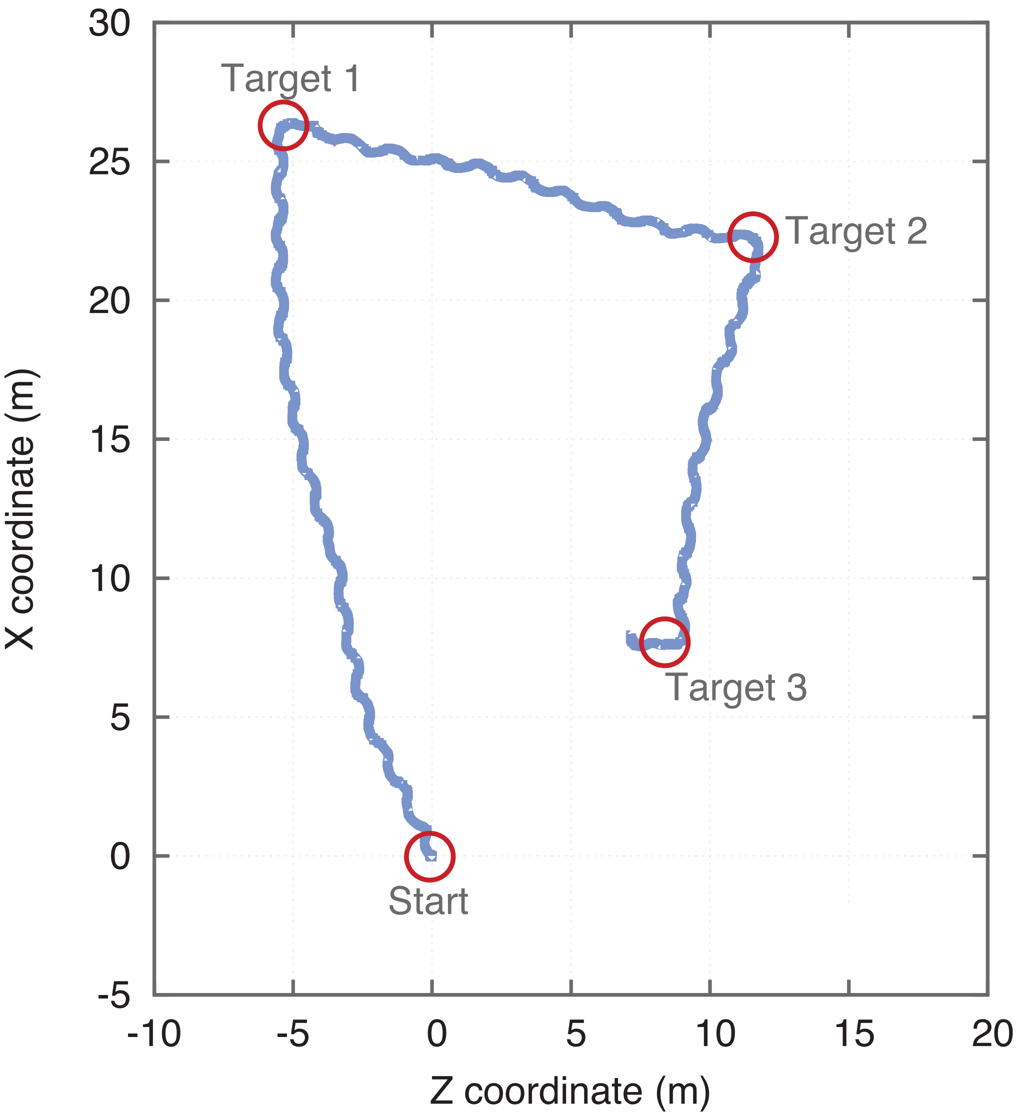


Figure 27: The 2D path followed by the robot with the learned policy tested with 3 consecutive targets. Flop and roll provides successful rolling and changing directions.

Second, we analyze the learned rolling behavior. We take the best policy and test it with 3 consecutive targets. Figure 27 shows the targets and the path of the center of the robot. The robot successfully navigates to all of the targets. The zig-zag behavior that is observed is due to the fact that the robot does not have a perfect spherical shape, it falls on two sides in order to roll.

As part of the same experiment, we analyze the lengths and tensions of the cables during this navigation. These results are a keypoint for the applicability of the algorithm, because we have to make sure that resulting behavior does not require an unrealistic amount of tensions or deformations on the cables. Figure 28 shows that the stationary robot (first 2 seconds) has 50N of average tension and 100 N maximum tension. During the rolling motion (after 2 seconds), the maximum tension

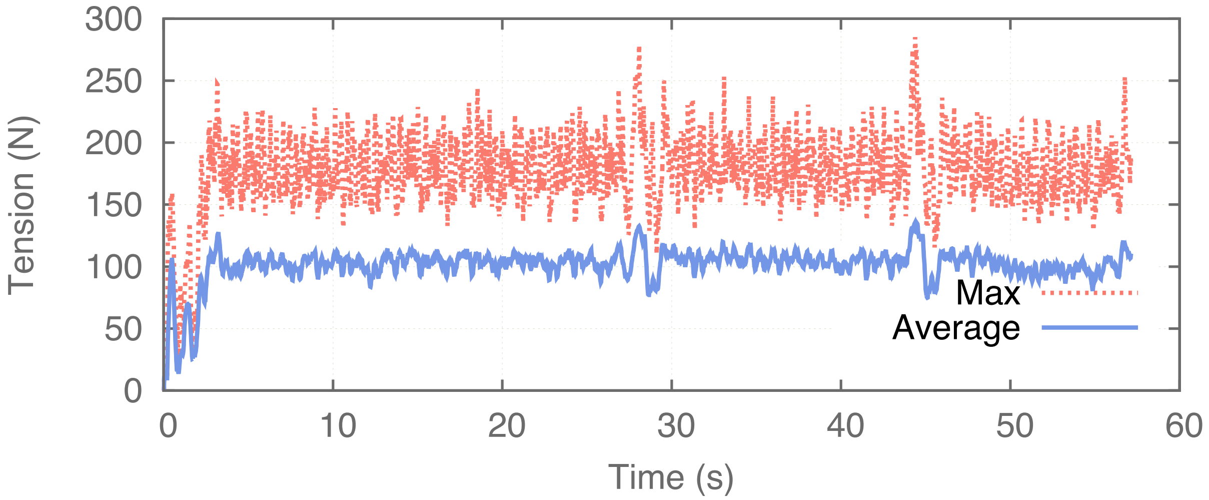


Figure 28: The average and maximum tensions of the cables during the experiment with 3 targets. The tensions stay within reasonable limits and the peaks are reached during change of direction while switching to different targets.

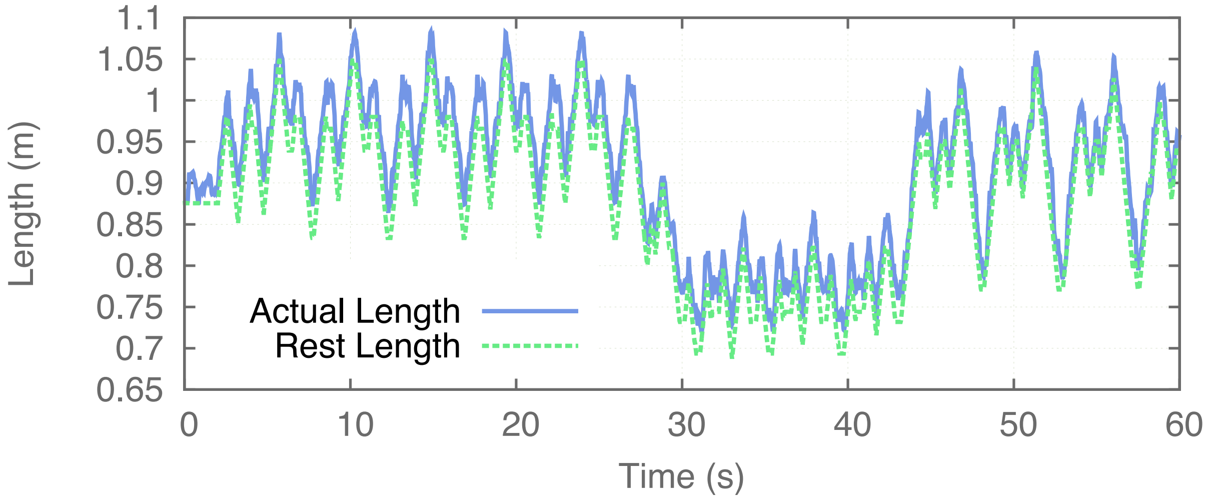


Figure 29: The rest length and real length of one of the cables during the experiment with 3 targets. The patterns emerge between the 0-30's, the 30-45's, and the 45-60's due to the rolling to different directions for 3 different targets. Rolling to different directions give different roles to this particular cable.

is around 180N, with the average around 100 N. The graphs shows two peaks around 30 and 45 seconds. These peaks are due to right-turning when the robot reaches target 1 and 2. Even in those cases, the tensions stay below 300N. These numbers gives an idea about the applicability of the algorithm to the physical robot.

If we look at the lengths of the cables, Figure 29 shows the pattern used for one of the cables. Since rolling is a repeating behavior, the same pattern repeats itself until the robot changes direction. During this first phase, the rest length of that cable oscillates between 0.8m and 1.05m. The actual length of the cables (since they are elastic) is slightly higher then the rest length, which provides the tension analyzed in the previous graph. When the robot starts moving towards target 2 (around the 30's), the direction of rolling is different and the same cable has a different role in the rolling locomotion. It is shorter and oscillates around the 0.7 - 0.85m range. The third phase is completely different and it can be seen that the pattern is different once again.

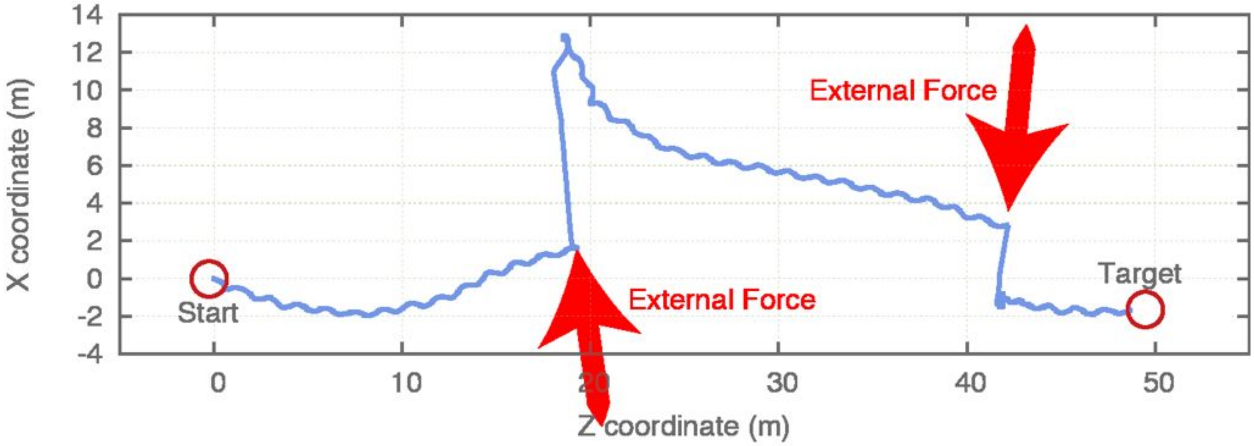


Figure 30: The 2D path followed by the robot with two intense unexpected external forces. The robot is pushed with an impulse that causes the robot to roll outside its path. After stabilization, the robot starts rolling to its target once again.

To test how the learned behavior handles an unexpected external force, Figure 30 shows the path of the robot when it is pushed sideways (twice) during the rolling motion. The robot pushed by the impulse rolls sideways but stabilizes and starts rolling towards its target again.

Next, we test the ability of the algorithm to overcome a terrain with different types of hills. After learning using virtual terrains with randomly-generated hills, we take the best policy and analyze its performance. The robot can continuously climb an inclined uniform terrain with a 20$\%$ grade.

Our next tests are about the applicability and robustness of the learned behavior. We first test the algorithm with slower motor speeds. We train the robot using the default configuration of simple motor models with a constant speed of $0.2m/s$ independent of the load. We take the best policy and test it with motors as slow as $0.02m/s$. Figure 31 shows that the robot can still move in the desired direction. The learned behavior does not necessarily require fast motors. The slower motors mean slower rolling, but we obtained a distributed rolling algorithm independent of motor speeds.

The last set of experiments test the algorithm against hardware failures. We compare standard conditions with broken controls and broken cable. In the first scenario, we disable the controller,

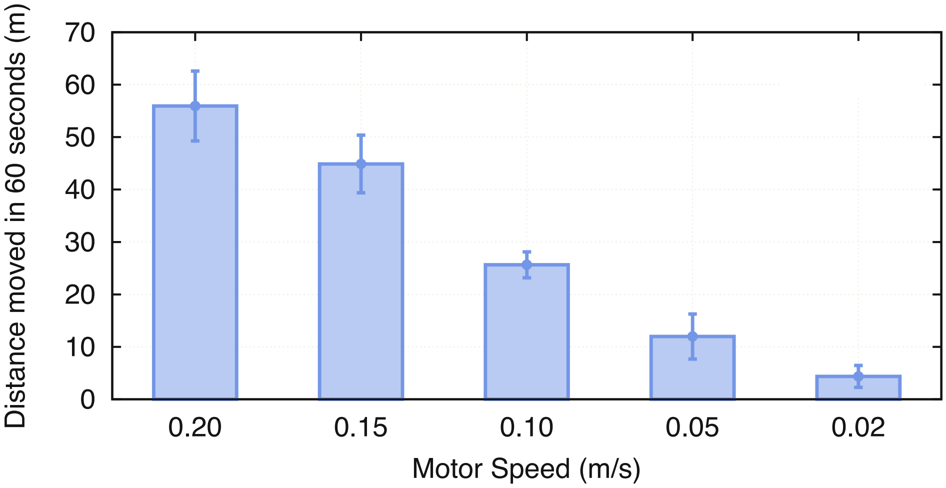


Figure 31: Flop and roll tested with slower motor speeds. Even if it is trained with a robot that has motors with 0.2 m/s speed, it can still roll with slower motors.

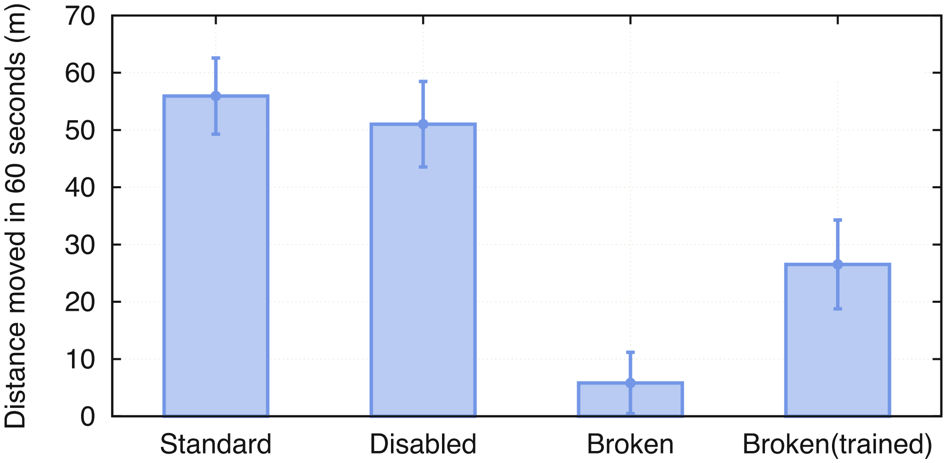


Figure 32: Flop and roll tested with non-functioning controllers and broken cables. With one controller not functioning, the robot still rolls close to its original performance. When one of the cables breaks completely (does not exist anymore) and symmetry is broken, the robot can perform rolling when it is trained for breakage of one random cable.

meaning that one randomly chosen cable becomes stuck with the initial length. In the second scenario, we break one randomly chosen cable assuming that the cable does not exist anymore. Figure 32 shows that if one of the controllers stops working, the structure can still roll successfully. Imagine a legged robot where the controller of one of the joints stops working. The algorithm for running will be highly affected by this failure. In our case, the distance covered by the robot in 60 seconds only decreases from 55m to 50m. For the case of the broken cable, one random part of the structure will be affected by the cut cable causing the robot to be deformed and to lose symmetry. In this case, the learned algorithm is highly affected. On the other hand, the fourth column shows that if we train the algorithm for such cases, we can obtain a rolling behavior that provides reasonable locomotion, even with a broken cable. Please note that during training the broken cable is selected randomly so that the algorithm can be robust to failure for any of the 24 cables.

10 Escaping Terrain Traps

10.1 Introduction

In this section, we explored a technique which enables SUPERball to escape from craters or ditches. We assume that missions planners will do their best to avoid getting the robot stuck in terrain traps, but the possibility is always present. Tensegrity rovers have already been shown to survive extreme impacts of landing, climb slopes, and traverse rough terrain [43], so demonstrating that they can also escape pathological terrain traps is an important step in showing that they can be seriously considered for exploration missions.

10.2 Implementation

Our proposed solution to this is to employ a machine learning module that discovers controllers to enable the rover to escape ditches and craters on-the-fly. Our solution consists of two tiers of sampling simulations utilizing the The NASA Tensegrity Robotics Toolkit (NTRT).

The tensegrity structure used in this paper is a full actuated version of SUPERball, consisting of a 6-rod icosahedron and 24 actuated cables. For the cable model, a linear spring model consisting of two points implementing Hooke's law with a linear damping term is used as defined in (23).

$$F = -kX - bV \tag{23}$$

A simple change in length motor model is also implemented within each cable, where the cable is deformed by changing the rest length of the spring model.

10.2.1 Controller Definition

It has been shown by Kim et al in [45] that controlled locomotion may be achieved by deforming only the equilateral, triangular faces on a tensegrity structure. By consulting the results in [45], we not only ensured locomotion in the robot, but we were also able to reduce our search space by segregating the cables according to the equilateral faces. Each cluster is a set of three cables conjoined to form one of the 8 equilateral triangular faces. To actuate our tensegrity structure, a control algorithm was created to generate synchronized motion between these clusters.

An approach, based on this method, has been used to produce a rolling motion in 6-rod tensegrities as illustrated by Atil Iscen et al in [42]. To further reduce the solution space for our multi-level Monte Carlo simulation, we simplified Iscen's controller such that all actuation in the structure was determined by 8 sine functions, one for each cluster of cables. Each cable actuates towards a shared goal length within its cluster, but this goal length varies by cluster. The exact goal length at any time step in the simulation is dictated by a sine wave. Specifically, our goal cable lengths are regulated by four sine wave parameters: amplitude, angular frequency, phase change, and DC offset. The equation defined by these four parameters is shown in equation (24).

$$y = Asin(\omega t + \varphi) + D \tag{24}$$

Where A is amplitude, ω is angular frequency, φ is phase change, and D is the DC offset.

The amplitude is the maximum deviance from the initial pretension in length to which a cable can actuate. The angular frequency is how often a cable cycles between its maximum length and its minimum length. Lower angular frequencies are due to greater acceleration by the motor. The phase change is the length of a cable at the start of the simulation (i.e. at which point during its period the sine wave begins). The DC offset is the default length of a cable. These four sine wave parameters together dictate the resulting goal length of a cable. Each controller, therefore, requires 32 input parameters to function (the 4 sine wave parameters for 8 different clusters).

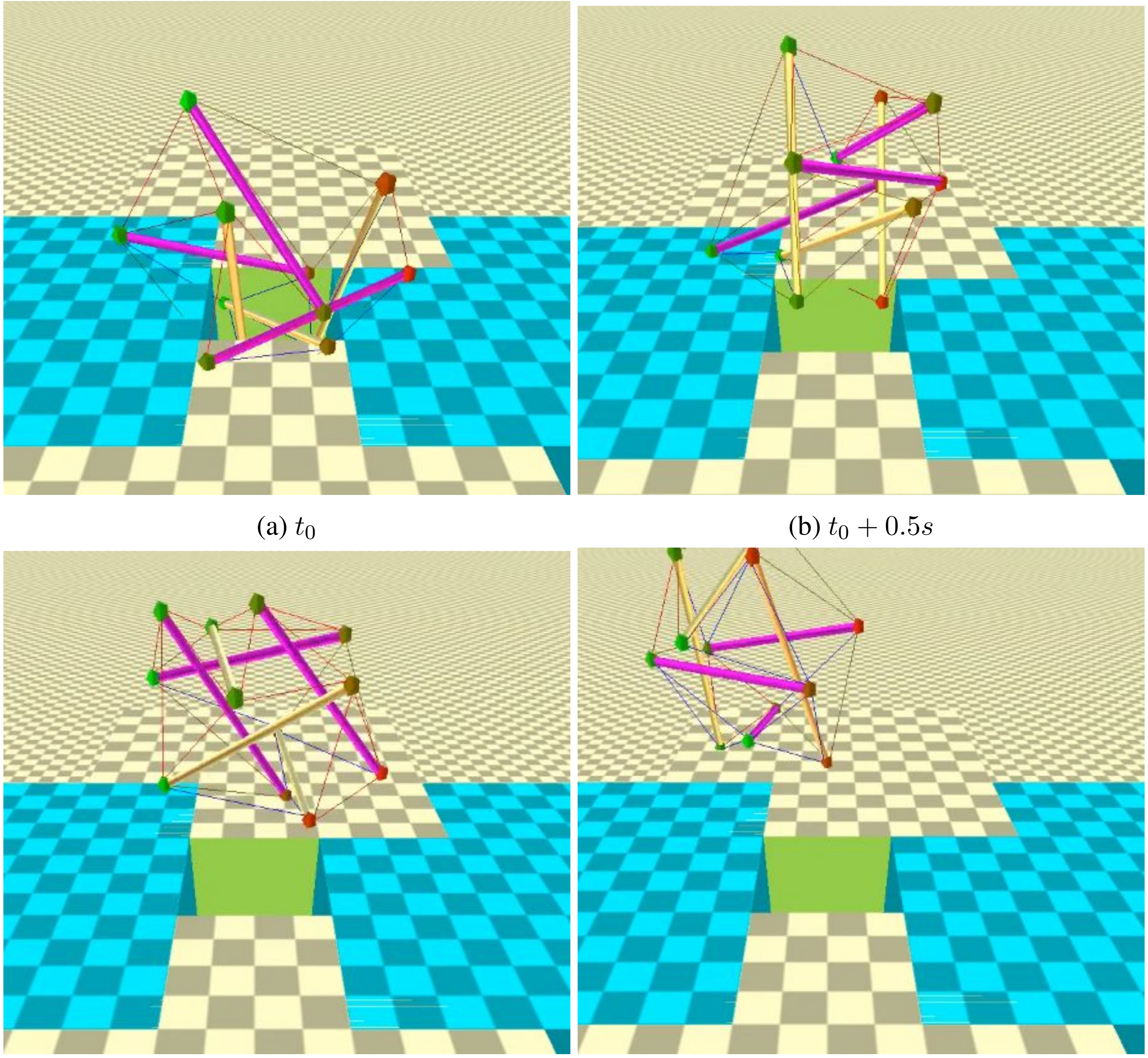
The control policy of the tensgrity structure is dictated by these 32 sine wave parameters. Each simulation run in our experiment has a unique control policy. The 32 values in a given control policy are Monte Carlo values within the range of $[0, 1]$. These values are then scaled according to their corresponding sine wave parameter. The result is that the control policy, consisting of the 32 sine wave parameters, commands all of the motion generated by our tensegrity structure.

10.2.2 Control Parameter Constraints

Each sine wave parameter was constrained to real-world limitations. Since actuators outside of simulation can only ever stretch as far as their material properties allow, the amplitudes in simulation follow this physical constraint. DC offsets less than the minimum length of an actuator are nonsensical, so this sine parameter was limited to the rest length of each cable. Likewise, angular frequency was limited to mimic the acceleration limitations of mechanical actuators. For this experiment, we limited our amplitudes to be less than fifty percent of the initial length and actuations to occur at frequencies between 0.3 and 20 Hz. The phase change of any sine wave was limited to be within $[-\pi, \pi]$. The amalgamation of these constraints formed a more accurate model for simulating the tensegrity robot, providing us with more useful results. To test for actuator robustness and redundancy, we simply limited which cables were able to change length during the simulation. We can artificially select cables to "fail", that is, to never change from their original length. The cables are still present in the structure, but due to their imposed inability to move, become passive components to the robot as a whole. For our experiments, we compared structures with 100% of their 24 actuators still functioning to structures with 75% (18) of the cables functioning, and to structures with only 50% (12) of the actuators changing with respect to their sine wave control policies. This comparison illustrates whether or not the reduction of capable actuators in the tensegrity structure dooms the entire robot.

10.2.3 Simulation Execution

To simulate a tensegrity structure becoming trapped in a crevice, four walls were created to enclose the robot. As seen in Figure 33, the robot was placed in the middle of these four walls such that the only way it could escape was by climbing out over an edge of the crevice. Once the robot had completely removed itself, it simply falls off the wall. This not only prevented the robot from accidentally re-entering the crater, but prevented the robot from returning within 25 meters of its origin (the center of the crater). As a result, recorded displacements of over 25 meters indicate a successful escape.



(a) t_0

(b) $t_0 + 0.5s$

(c) $t_0 + 1.0s$

(d) $t_0 + 1.5s$

Figure 33: A tensegrity structure escaping a crater. The crater has roughly a 25m radius at its closest edge and is elevated above the ground to keep the escaped tensegrity robots out.

To test our controllers in the simulator, we established a two-tier system for sampling control policies. We define a sample as a single simulation run and we define a tier as the set of samples that are run according to the same high level policy. We also define a generation to be an individual tier within a given experiment. In the first tier of sampling, we set control policies according to a uniform distribution of Monte Carlo values. Each of these Monte Carlo values correspond to one of the 32 parameters which dictate the actuations of our robot with respect to time. By determining which control policies caused the tensegrity robot to escape from its crater, we can segregate samples as either successes (escaped) or failures (not escaped). The successful samples from the first tier are used to centralize the sampling ranges for the second tier. In this second tier, the control policies of the simulated robots are made to imitate already-successful control policies. The only difference in sampling policy for this new tier is that each Monte Carlo value in the second tier is sampled uniformly within an acute range which is centered around "trustworthy" Monte Carlo values for that particular parameter.

Each of our samples run for 60,000 steps. As a result, a total of 100 seconds are simulated each sample. To find enough control policies that successfully generate an escape path, at least one thousand samples were executed in each generation. Once a sample has finished executing, the control policy of that sample as well as the final displacement of the robot are recorded.

To test actuator robustness, we took 100 samples of the simulated robot at each of the three actuator settings (100%, 75%, and 50% actuators enabled). Each of these samples were run for 100 simulated seconds as well.

10.2.4 Reward System

In this experiment, the control policy of an individual sample was judged to be either a success or a failure. This judgement depends on that sample's displacement (the distance traveled by the tensegrity rover from its initial position). The magnitude of the displacement of a simulated tensegrity robot determined the categorization of that sample's control policy. Specifically, if a displacement was larger than the distance between the crater's origin and the edge of the crater, that displacement indicates the robot successfully escaped the crater. As a result, the control policies of samples that resulted in a displacement of 25 meters or more were deemed successes. All other control policies were considered failures. The sine wave values that constituted a successful control policy were saved for the next generation while the others were cast out. This procedure, in summary, sorts data as either capable of generating viable escape paths or as data that is incapable of generating viable escape paths.

10.2.5 High Level Policy for Generations

Each sample represents a simulation based on a set of 32 control values and its resulting performance. Each generation is then taking samples from the previous generation to help determine which samples to take. In Generation 0, control policies of each sample are uniformly random and are determined by the Monte Carlo method. The Monte Carlo values ranged from 0 to 1 inclusive. These values were eventually scaled to match the physical requirements as dictated in sub-section 10.2.2. These assigned sine wave values follow an even distribution such that the probability of any given legal value (i.e. $[0, 1]$) being assigned is equal to the probability of any other legal value. For our experiment, we ran 1000 samples in Generation 0. When every sample in this base generation has finished executing, the recorded results were parsed. Once the successful samples have been filtered out, their control policies are stored for use in the next generation.

In Generation 1, control policies of each sample consider the already-proven successful policies of Generation 0. Control policy values in Generation 1 are centered around the respective input policy values from Generation 0. To optimize the results from Generation 0, the control policy values for Generation 1 are then tweaked by as much as 0.5%. Each of the successful samples from Generation 0 were re-run in Generation 1 ten times, each with a slight variation. The purpose of this action is to discover what improvements can be made on already successful control policies. The exact variation on the Generation 1 control policy values is dictated by the Monte Carlo method. Once again, the selection distribution of this Monte Carlo sampling is even, so each value within $\pm 0.05\%$ of the input Generation 0 sine wave values is equally likely to be chosen. These modified control policies are then simulated en masse. For our two-tier experiment, we ran 1100 samples, ten times the number of successful control policies discovered in Generation 0. As in the base generation, the recorded results are analyzed and the successful samples are filtered out.

To test actuator redundancy, we ran 3 generations of 100 samples each. The first generation was our control group (all 24 actuators functioned according to their sine wave control policies). The second generation experienced a forced actuator failure in 25% (6) of the 24 actuators. These “failed” actuators became passive components as the other 18 actuators behaved normally. The third and final generation only had half (12) of its actuators functioning according to sine wave control policies.

10.3 Escape Results

10.3.1 Results of Two-Tier Monte Carlo Simulations

We ran experiments on the controller of our tensegrity robot to optimize escape paths out of an arbitrary crater. To measure the success of each control policy, we record the displacement of the corresponding simulated robot. This displacement is used as a score for comparison against the displacements of other samples.

In addition to the displacement of the robot, the 32 Monte Carlo values, the control policy of that sample, that generated said displacement were also recorded. To escape the simulated obstacle, a displacement of at least 25 meters was required. Because of the height at which the robot rested on top of the fissure, once a robot escaped (and consequently reached 25 meters from its origin), it fell out of the crater. Consequently, all robots that were able to escape remained outside of the crater for the remainder of that trial. This means that there were no samples in our simulation where a tensegrity robot escaped a crater only to fall back into that crater.

The scatter plots, Figures 34 and 35, feature data points for each control policy. Since each sample has exactly one control policy, there are 1000 data points in the scatter plot for Generation 0 and 1100 data points in the scatter plot for Generation 1. Each data point represents a 32-dimensional value, the 32 sine wave parameter control policy.

Generation 0 yielded diverse, but uneven results (see Figure 34). Displacements ranged from 0.01

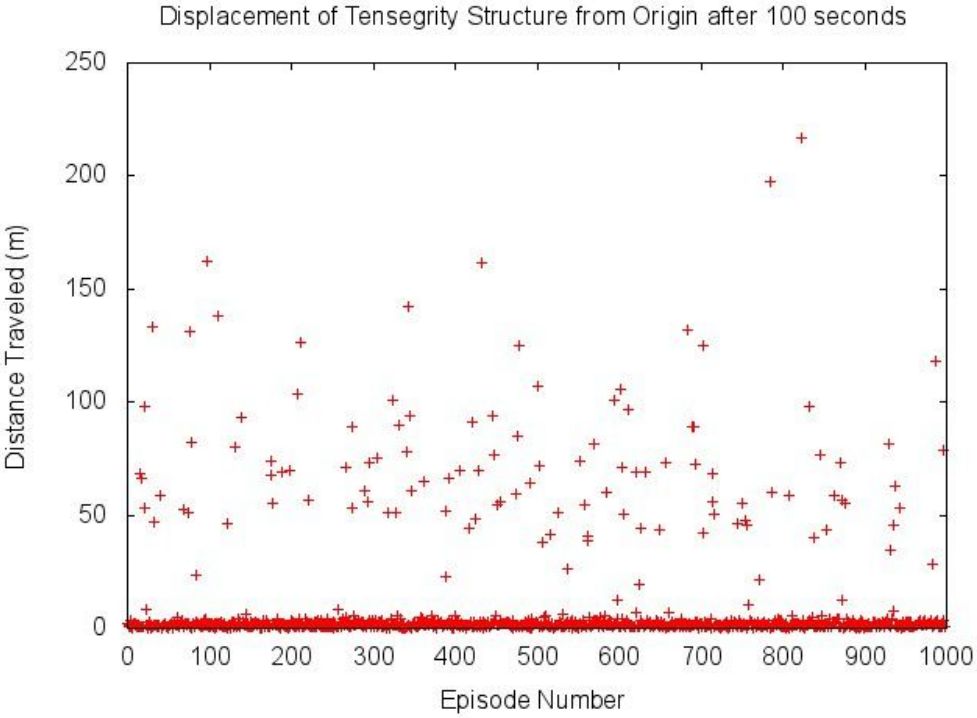


Figure 34: The displacements reached in 1000 independent samples using Monte Carlo generated control policies.

Table 3: Displacement (m) by Generation

Generation	Minimum	Average	Maximum	% Escaped
0	0.01	9.91	216.55	11.1
1	0.05	20.53	223.59	24.0

meters to 216.55 meters. The mean displacement traveled by the robot was 9.91 meters and 11.1% of samples resulted in a successful escape. The vast majority (88.9%) of Generation 0 samples featured a failed control policy (i.e. the robot did not escape the crater).

Generation 1 yielded a larger range of results than Generation 0, most notably including a greater number of successful samples. Displacements ranged from 0.05 meters to 223.59 meters. The mean displacement traveled by the robot was 20.53 meters and 24.0% of samples resulted in a successful escape. The majority (76.0%) of Generation 1 samples featured a failed control policy (i.e. the robot did not escape the crater).

The combined results from the two high level policies have shown significant improvement from Generation 0 to Generation 1. The minimum, average, and maximum displacements in Generation 1 were all improvements over the previous generation. A randomly selected sample from Generation 1 was 2.16 times more likely to generate a successful escape path than an arbitrary sample from Generation 0. The mean displacement of a Generation 1 robot was less than 5 meters from its goal displacement; a 207% increase from the comparable mean Generation 0 robot.

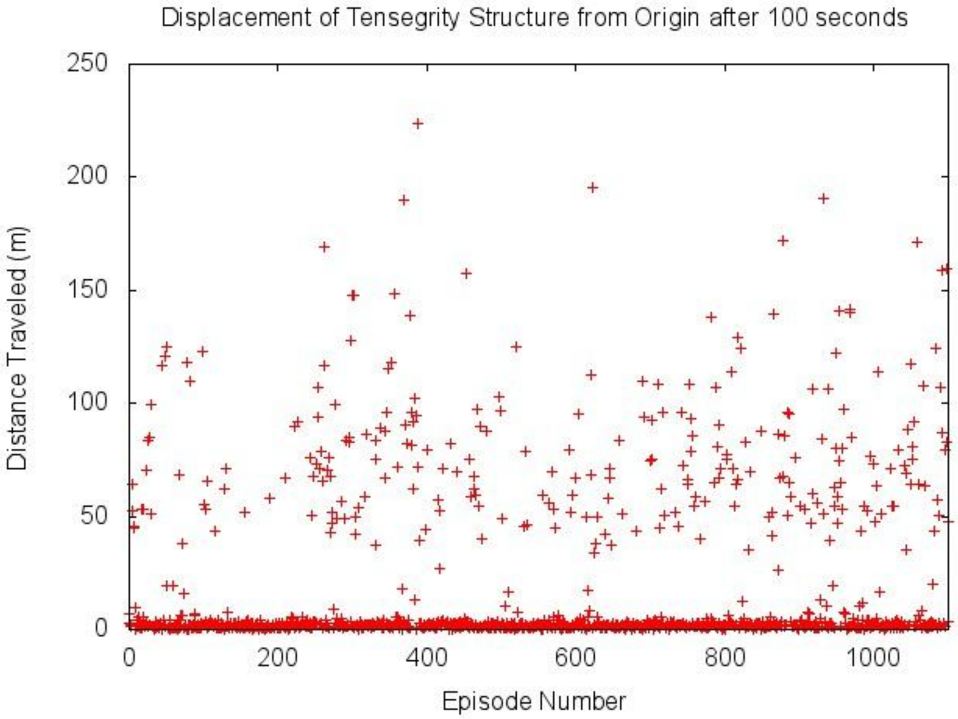


Figure 35: The displacements reached in 1100 independent samples using control policies dictated by the successful samples from Generation 0. The values in the set of sine wave parameters from Generation 1 samples were each centered around one of the successful corresponding sine wave parameters in Generation 0 control policies. These Generation 1 values were then modified to be within 0.5% of their respective Generation 0 values with the goal of optimizing Generation 0 control policies.

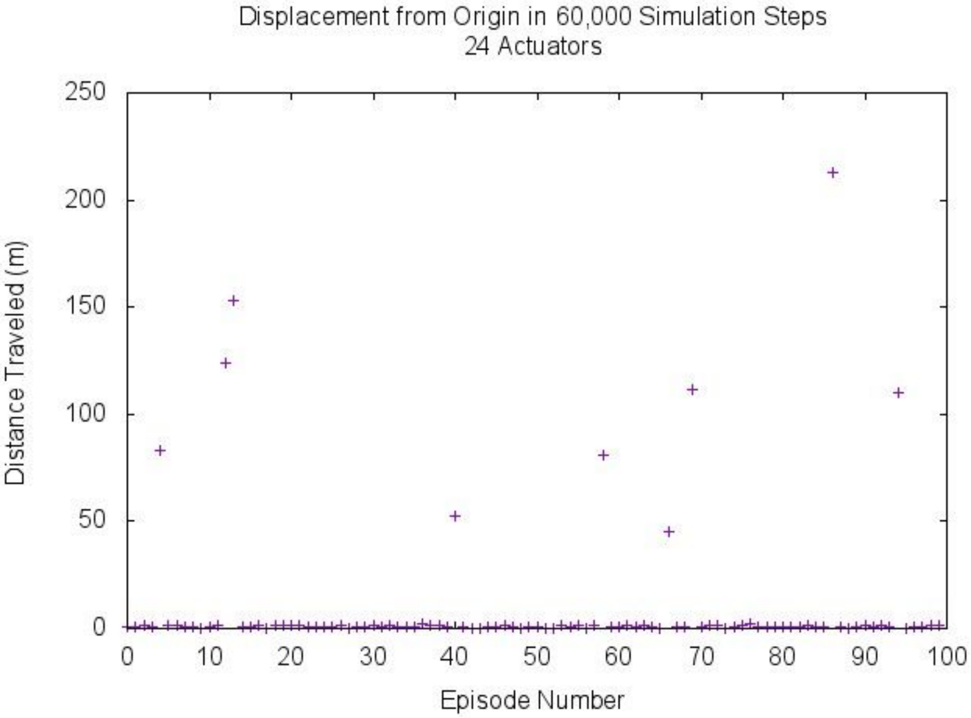


Figure 36: 100 Samples of Tensegrity Structures with 24 Actuators

10.3.2 Results of Actuator Redundancy Experiments

By reducing the number of functioning actuators on our simulated tensegrity, we can see how the system gracefully decays even as actuators fail. In all cases, we were able to find control strategies which enabled an escape, even with up to half the actuators frozen. Granted, our search algorithm found fewer successful controllers under those conditions, yet we have shown that solutions exist. Looking at the percentage of successful control strategies helps see how actuator failures do have an impact on system motion, but that the degradation in capability is gradual.

With all 24 actuators on the tensegrity functioning correctly, we see the greatest chances of finding successful escape strategies during our monty carlo search (9% success rate) (see Figure 36).

With 18 actuators on the tensegrity functioning correctly, a successful escape is more challenging, but not impossible (see Figure 37). In this experiment, 6% of the tensegrity structures were able to escape.

With just half of the original actuators functioning correctly, a successful escape is even more difficult (see Figure 38), but still possible. In our tests, 4% of the randomly sampled controllers were still able to escape the ditch in the allotted time.

10.4 Conclusions for Escape Analysis

The key result is that it is possible for a tensegrity robot designed like SUPERball to escape pathological terrain traps. This can even be accomplished if some of the actuators have failed and gone

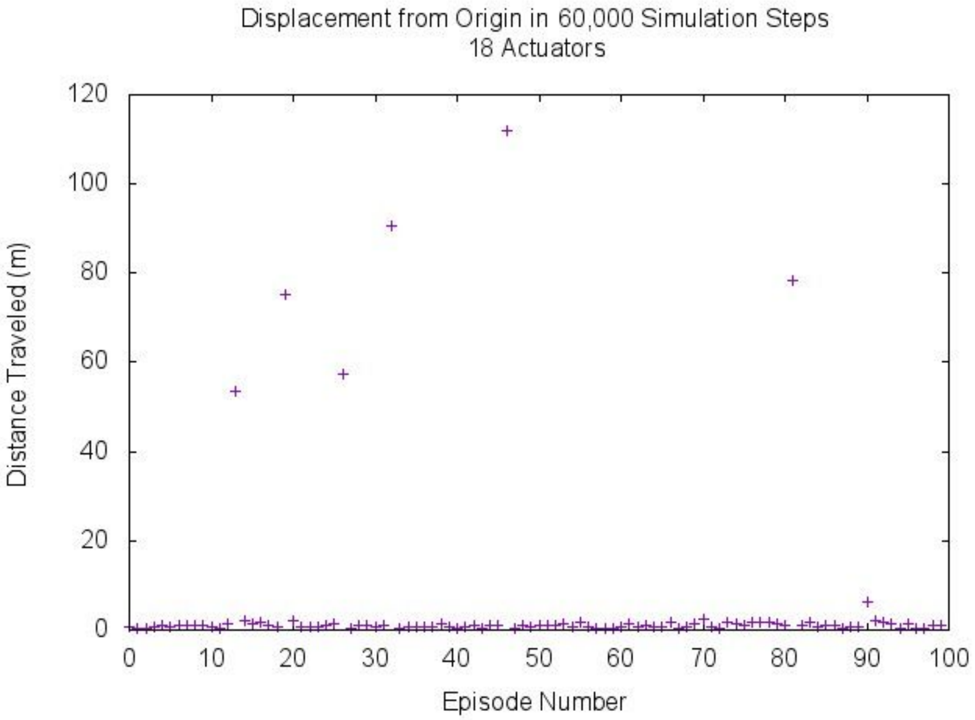


Figure 37: 100 Samples of Tensegrity Structures with 18 Actuators

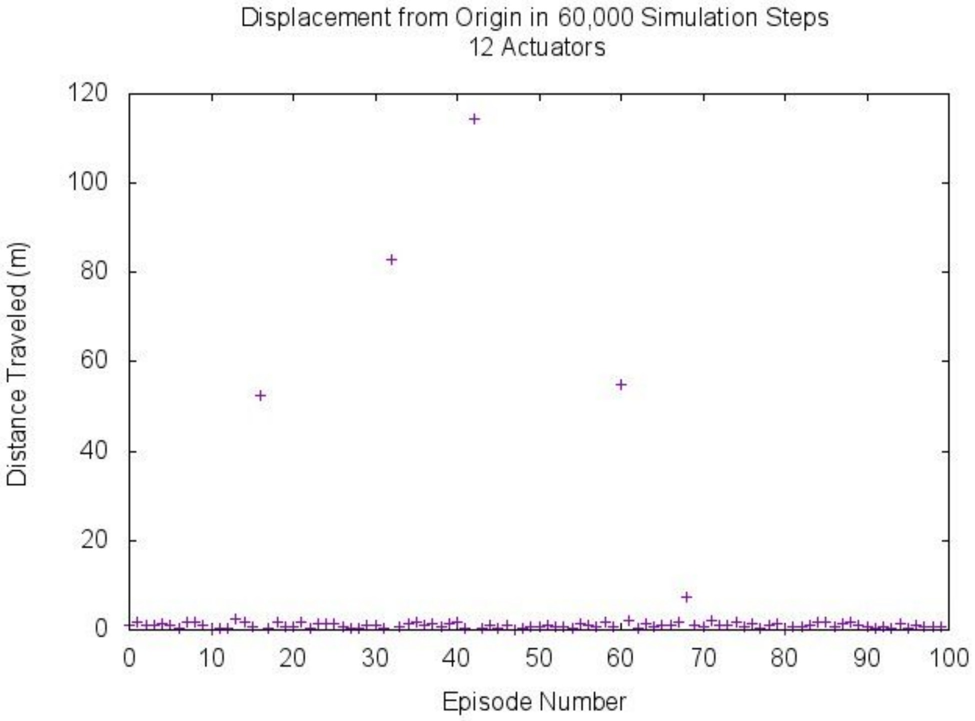


Figure 38: 100 Samples of Tensegrity Structures with 12 Actuators

passive.

Our solution intelligently mapped cables and used very simple control policies (sine waves). By clustering the cables in our tensegrity structures into triplets, we reduced our problem space by two thirds. Sine wave control functions meant that we could completely define the actuator goal length with respect to time of each cluster in only four parameters. By reducing our problem space, we were able to run Monte Carlo simulations to quickly obtain useful data on escape controllers.

The results from the two generations of simulation have illustrated the compounding improvements that a two-level Monte Carlo simulation can yield. In Generation 0, uniformly random control policies produced by the Monte Carlo method established a baseline success rate for tensegrity structures escaping our crater. Generation 1 built off of the information obtained from Generation 0 by sampling around the corresponding values of Generation 0's previously-tested control policies that were shown to succeed. The result is a proven system that optimizes control parameters for tensegrity robots that must escape craters.

Tensegrity structures, due to their abundance of cables, are able to actuate in a variety of manners. This redundancy, coupled with the flexibility of the suspended system, allows tensegrity robots to successfully actuate even when limited by "failed" actuators. If our tensegrity rovers lose actuation capability in 6 or even 12 of their 24 actuators, it can still successfully escape the crater with which it was presented. This is particularly relevant to the original mission of the rover because loss of mechanical capability is not uncommon in extraterrestrial robots. This ability to function, even under adverse conditions, can allow for missions to continue when more traditional rovers would fail. Also, robots with greater redundancy can be trusted to explore more dangerous terrains and environments than their traditional counterparts.

11 Validation of Landing Analysis

11.1 Introduction

During Phase 1, we developed initial computational models for analyzing the forces experienced by a tensegrity robot as it landed on another planet. Also during Phase 1 we built some hardware prototypes with which we conducted drop tests, which empirically validated the concept. What was missing was a series of instrumented drop tests that could be used to calibrate and validate our computational models.

This section discusses research performed by a University of Idaho Senior Design Team during Phase 2. This research included the construction of a six-strut icosahedron structure and incorporation of an internal payload system that housed an accelerometer. Testing was performed using both linear and non-linear springs with a goal of collecting acceleration and deflection data of the payload. The results include the use of internal, linear springs and suggestions for structure and model improvements to yield more accurate results.

In order to fully validate the computational model, an experiment was designed in which the three key variables (i.e. the stresses in any of the members, the acceleration of the payload, and the deflection of the payload) could be accurately and repeatedly measured.

This provided the following three preliminary goals (in order of importance):

1. Collect acceleration data of the payload at varying drop heights, with varying springs, and while carrying varying payload masses
2. Determine a method to measure payload deflection with an accuracy of $\pm 0.5in$
3. Calculate the stress in critical cables by using acceleration and deflection data

In planetary exploration missions, the structures can possibly land in any number of orientations. To account for this, we setup the model and perform drop tests with the structure in the worst case landing configuration. If the structure can survive the worst case landing scenario, it should be able to survive an impact in any scenario. The data was consequently collected in experiments where the SUPERball model was positioned in the worst possible orientation upon landing.

The final goal of our landing analysis is to build a structure capable of surviving potentially any landing orientation at a height corresponding to terminal velocity on Titan. Terminal velocity on Titan for this structure is approximately $14m/s$. This corresponds to the velocity achieved from a 10 meter drop here on earth. The goal for the final structure is to drop from a height of 10 meters with a payload mass of $5kg$. We were able to accomplish this feat and the structure illustrated the robustness of the tensegrity principles. Even when the overall tensegrity lander is tested to failure and multiple attachment points failed (as expected), the overall structure holds together.

11.2 Project Overview

To perform this analysis we built many prototypes and collected experimental data to validate the computational model. The first step was to run the provided computational model and determine a set of starting conditions for designing a prototype. The first prototype used shock cord as a means to attenuate the payload. The spring constant of the shock cord had to be determined experimentally due to its non-linear behavior. Throughout all the experiments, all of the acceleration data was collected using the Gulf Coast Data Concepts X250-2 accelerometer, which was suspended in the middle of the tensegrity structure inside the mock payload.

Comparison between the model and the data obtained by dropping the first structure revealed a large disagreement in the magnitude of acceleration. Unable to properly calibrate the non-linear behavior of the shock cord, the computational model was converted back to using linear spring inputs and the shock cord was replaced with linear springs. This second structure was used in a series of tests involving different springs and measurement methods.

Two different methods to determine deflection of the payload and of each spring were used. The data was compiled and compared to the model. The results of these experiments showed fairly good match between the computational model and the experimental data. From these results we are confident that the scale of expected forces output by the model is accurate. Although some small differences still existed between the model and the experimental data, which may be caused by tangential oscillations of the springs after landing. To test this, we developed a third and final prototype that incorporated internal springs in the rods in order to remove their deflection dynamics. This structure was dropped to failure.

11.3 Experimental Setup

11.3.1 Approximate Stiffness of Inner and Outer Springs

To conduct the payload and drop test experiments, several iterations of SUPERball prototypes were constructed. The main structure, a six-strut icosahedron, required 24 springs to assemble the outer tension network. To attach a central payload, the inner tension network required 12 additional springs. Since the inner structure springs and outer structure springs did not have to be mechanically equivalent, several combinations of stiff and pliable springs were considered when constructing these prototypes. Ultimately, a configuration of springs was selected where the outer tension network consisted of relatively stiff springs and the inner network consisted of relatively pliable springs.

11.3.2 Acceleration Measurement

The computational model provided used input design parameters to output a maximum value of acceleration which the payload experiences during an impact event. The acceleration of the payload is a very important component of this design, as the main purpose of the structure is to protect the payload and a very high acceleration will damage the payload. The easiest possible way to measure the acceleration the payload experiences is to directly attach some form of data collection device to the payload (in other words, an accelerometer). In order to avoid complications with wiring and powering a micro-controller, we used an accelerometer that would collect and store data. This allowed for collection of data for multiple tests much faster, as there would be no need to configure any wires, power supplies or micro-controllers. Throughout all the experiments, all

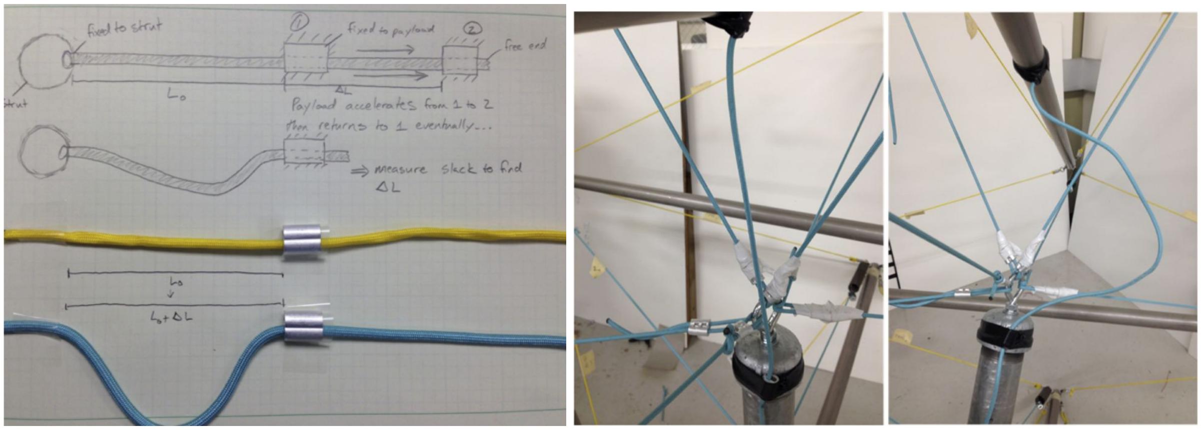


Figure 40: Left: Deflection Measurement System, Right: Deflection Measurement System attached to payload and upper bar of structure.

of the acceleration data was collected using the Gulf Coast Data Concepts X250-2 accelerometer, which was suspended inside the mock payload with all required batteries, etc.

11.3.3 Mechanical Fuse Concept - Spring Deflection Method 1

For any linear spring, if the spring constant is known and the amount the spring deflects is also known, then the force that the spring experiences can be determined. The dynamics of ideal linear springs are expressed by Hooke's Law which states:

$$F = kx \tag{25}$$

where F is the applied force, k is the spring constant, and x is the deflection.

The deflection of a particular spring within the structure can be experimentally determined by utilizing the concept of a mechanical fuse. A mechanical fuse is a member that is loaded up, and carefully designed to break under a very specific load. This concept was adapted in order to measure the deflection of each spring. The mechanical fuses in these prototypes were threads that were cut to a specific length, longer than the length of an unstretched spring. Upon impact, the spring deflects and causes one of two outcomes. Either it deflects farther than the thread allows, breaking the thread, or less than the thread allows, in which case the thread is unaffected. By starting with a small piece of thread and increasing its length by the desired margin of accuracy, we could obtain an upper and lower bound on the deflection of a spring. Similarly, one can attach multiple strings of varying length for the same outcome.

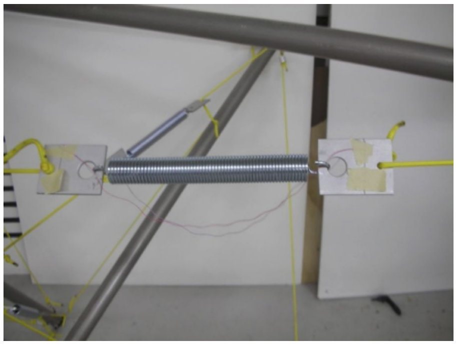


Figure 39: Linear spring with mechanical fuses

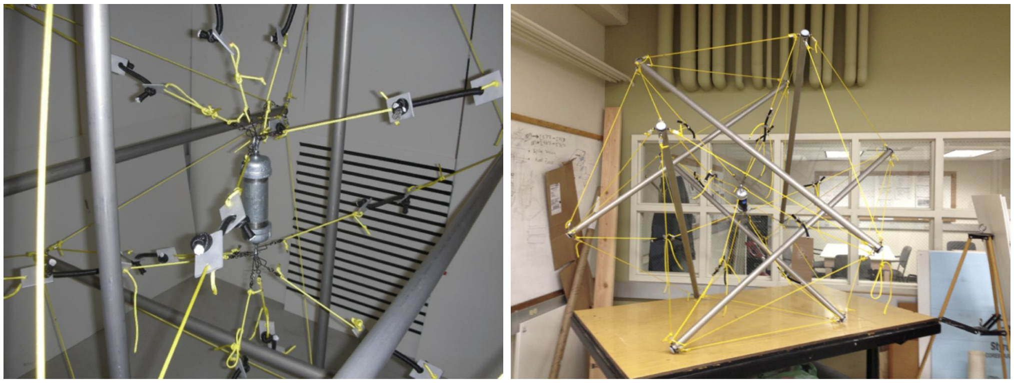


Figure 41: First Iteration structure made with light weight shock cord.

11.3.4 Friction Concept - Spring Deflection Method 2

Although the first method described was effective, the process was very time intensive, often taking several trials to obtain one data point. This made it clear that a faster method would have much higher value. The method we eventually developed utilized a short piece of paracord, an un-crimped crimp and the friction between the two. The design was fairly simple; the paracord was secured to a horizontal strut directly above the payload, such that there was no slack in the cord. This is the constant cord length, and it was measured and marked carefully. An un-crimped crimp was then secured to the payload such that the cord could slide through. Upon impact, the strut located above the payload deflects very little, as the outer structure is relatively rigid. However, the payload accelerates downward causing the secured crimp to also slide downward, increasing the cord length. When the payload recoils in the upward direction, the friction between the crimp and cord keeps the crimp from sliding back up the cord. This new length can then be recorded, and the total deflection found by subtracting the constant length from the deflected length.

11.4 Experimental Results

11.4.1 First Iteration Structure - Shock Cord

One of our primary concerns for the structure was to build a prototype that closely resembled the structure being used in the model. The model provided used linear springs, however they were modeled as massless and only allowed deflection in one direction. Although linear springs were already in the model, most are very heavy and could alter the dynamics of the physical structure. The addition of linear springs would add lumped masses to the physical structure, and the model could not account for such masses. In order to alleviate this issue, options for other types of springs were explored. Shock cord is commonly used in rock climbing applications, and closely resembles bungee cord. This cord is very lightweight, more closely resembling the models mass-less springs. The downside to shock cord is that it is a non-linear spring. Specifically, instead of having a constant spring rate, the spring rate increases as the load applied increases. Unfortunately, there is no data available from the manufacturers on the properties of the material, so a fairly significant amount of testing is required in order to determine the relationship. It was assumed that it would be fairly simple to implement into the model (as far as changing code), once the spring characteristics

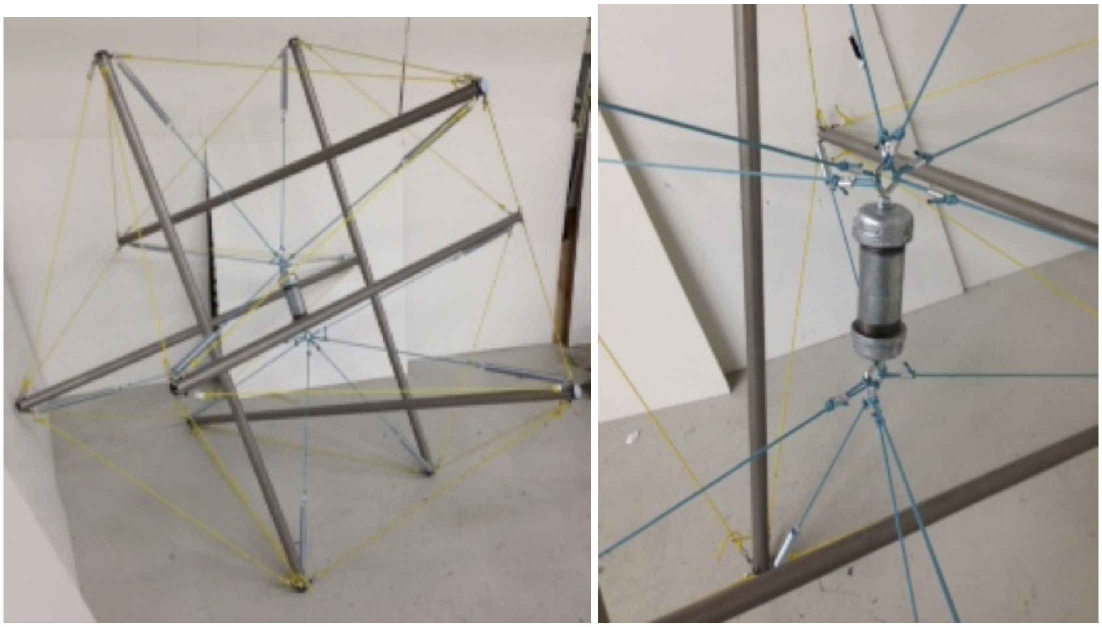


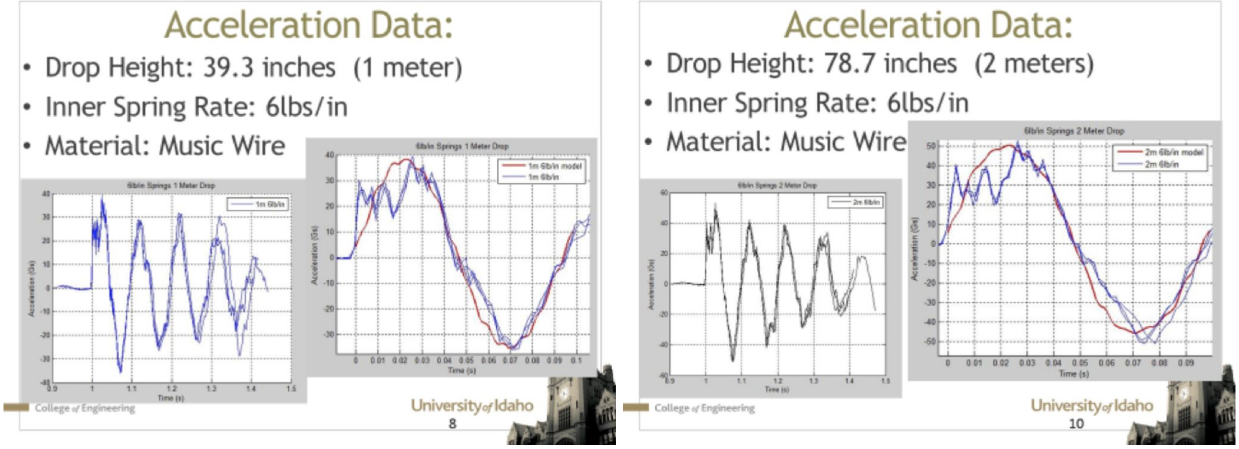
Figure 42: Second Iteration with Linear Extension Springs

were determined. As mentioned above, the students were unfortunately not able to build reliable spring models for the shock cord and this approach was quickly abandonded.

11.4.2 Second Iteration Structure - Linear Springs

Because non-linear springs were not successfully implemented into the model, linear springs were used instead of shock cord. This led into the second design iteration, where excess components associated with the shock cord were eliminated, and the over-all symmetry of the structure was improved. The data acquired using these springs was far closer to the model and intuitively more reasonable than with non-linear springs. Although the structure was much more symmetric and tight, the weight of the external springs could affect our model/experimental correlation. The springs in the computational model were massless and one-dimensional. Thus we were concerned that using real linear springs could lead to increased disagreement between experimental values and model outputs. If linear springs were attached to the physical structure and dropped, they would oscillate in more than just one direction, which could add discrepancy.

Using collected data and comparing it to the model, we showed that the computational model roughly matched the experimental data in terms of expected magnitude of accelerations (see accelerations graphs below). Yet, there were still extra oscillations in the experimental data that did not appear in the computational model. It was assumed that these extra oscillations were caused by the mass of the springs deflecting towards the ground during impact. This was confirmed upon filming several drop tests with a high speed camera where it was clear that the springs were oscillating not only in the axial direction, but also the transverse direction due to the inertia associated with the mass of each spring. Because the computational model was unable to capture this complexity, the we began to consider ways to alleviate this issue in the third prototype.



Acceleration Data:

- Drop Height: 39.3 inches (1 meter)
- Inner Spring Rate: 6lbs/in
- Material: Music Wire

College of Engineering

University of Idaho

8

Acceleration Data:

- Drop Height: 78.7 inches (2 meters)
- Inner Spring Rate: 6lbs/in
- Material: Music Wire

College of Engineering

University of Idaho

10

Figure 43: Second Iteration Acceleration Data when dropped from various heights. This showed a good match to the model, except for the extra oscillations caused by the tangential deflection of the springs themselves.

11.4.3 Third Iteration Structure - Internal Linear Springs

One solution to the oscillation of the linear springs was building another structure where the springs for the payload are located inside of the struts. With springs located inside the struts, they effectively become one dimensional because they can only oscillate in the axial direction. The added mass of the springs could then be simply added to the struts. It was expected that this prototype would remove the remaining delta from the computational model. This approach also ends up matching the design of the SUPERball prototype, which also holds its springs inside the rods.

Building this prototype took a much deeper redesign. For instance, the internal springs could be either extension or compression, which would be designed completely differently. A method of attachment from the payload to each of the internal springs would need to be developed without adding additional complexities not accounted for in the model.

The selection of springs for the newest structure was based on the goals of being able to support a larger payload mass as well as allowing for a range of drop heights to be implemented. The constraints for spring selection included dimensional bounds set by the length and diameter of the struts. Other considerations were the force required to separate the spring coils, yield strength, and the spring constant. The springs selected to meet these parameters were tension springs made of music wire. The force required to separate the spring coils was $10lb$ with a spring constant of $23.5lb/in$. Dimensionally, the springs were $1in$ in diameter and $9in$ long.

The original inner payload system allowed for the inner system to stem directly from the structure nodes. Since the new design implemented internal springs, there was no possible way for the inner payload cables to stem from the same location as the nodes without significantly reducing the cross sectional area of the strut. Structural integrity was heavily considered in the new strut design since the new structure harbored a larger payload mass and would be tested at greater drop heights. It was ultimately decided to set eyelets into the struts $59in$ apart. This distance was selected since it was the node separation used in the original structure, and due to its close proximity to the new structures nodes. The eyelets were assembled so that the eye was inside the strut. Essentially, the eyelets served as a guide for the inner payload cable being routed from the springs to the payload. On the opposite side of the eyelets, a slot $2.5in$ long and $0.5in$ wide was cut into the strut. This slot allowed for excessive movement of the inner payload cable. This design feature was implemented in order to avoid having any portion of the inner payload cable coming into contact with the strut during payload attenuation.

Although the structure was built with internal springs, lack of time and spring manufacturer error prevented the collection of useful data. Because cables had to be routed into the rods, the ideal solution would have been to use pulleys, but there was not enough time to manufacture and implement pulleys. Instead of using pulleys, the use of eye bolts was implemented, as discussed above, which created too much friction. Further, the springs ordered were said to have a pre-tension of approximately $10lb$ and during testing were discovered to be closer to approximately $40lb$. Thus, this final prototype was expected to fail, and was not used for calibrated drop tests, as done with the second iteration structure. Instead, the team performed a test to failure drop test by throwing it off a $10m$ cliff (Figure 44). This size fall in earth gravity results in a $15m/s$ landing event, approximately the landing speed expected for a Titan mission. As expected, a number of the connections between the cables and rods failed yet a significant result is that the structure maintained its integrity and shape due to the cables suspending the payload.

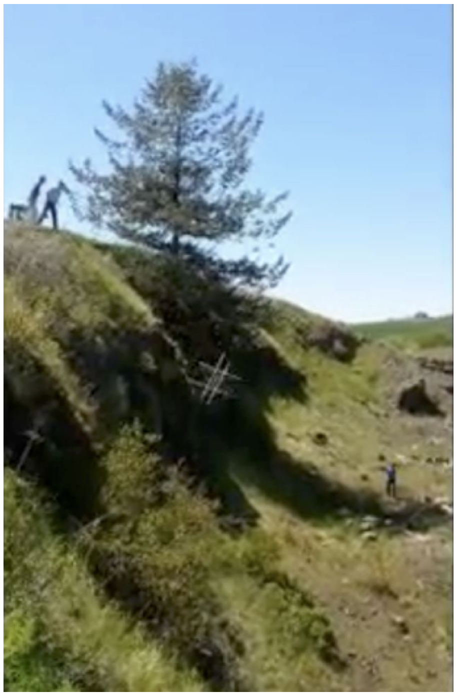


Figure 44: The final test of the Internal Drop Test Structure with internal compression springs.

We had noted in the past that the payload suspension cables helped stabilize the structure acting much like the 3D equivalent of bicycle spokes and this was our first confirmation that the redundancy they provide also supports the structure even in the face of cable failures. We feel that this result supports an expectation of graceful failure even under extreme landing scenarios, and we look forward to analyzing it further in the future.

11.5 Conclusions of Drop Test Analysis

The instrumented drop tests with the second iteration structure confirmed that our computational models were reasonable. While a perfect match was not established, we were able to show that the models and tests agreed on the general magnitude of the expected landing forces and decelerations experienced by the payload. The remaining divergence between the model and tests can be directly attributed to the mass of the linear extension springs deflecting sideways, which were not in the computational model. While the final test structure, which was intended to confirm this source of divergence, was not able to be used for calibrated testing, it demonstrated one of the other fundamental advantages of tensegrity landers – that even under failure of multiple cables, the structure held together and would have been able to continue with the exploration mission.

12 Scaling Analysis: Comparison of Tensegrity Lander versus MER scale Airbag

One of the interesting things about our tensegrity based approach to landing is that it appears to scale very well compared to other approaches, such as an airbag. The MER rovers used airbags, but since the mass of airbags grows with the square of the radius (area of a sphere), that approach was too heavy for the much larger MSL rover, necessitating the use of the far more complex sky-crane technology. Our tensegrity lander's mass scales linearly with the radius of the structure, as we simply use longer rods and cables, and do not need to cover the entire surface of the sphere. Thus, we have always suspected that at larger scales a tensegrity lander would outperform airbags and other options, purely as a landing system. This section takes a deeper dive into exploring this concept, and indeed shows that by using long enough rods, we can protect large payloads. This section only looks at this from a landing systems perspective, so it should not be forgotten that the tensegrity lander also includes the ability to provide surface locomotion too. Future work will look into integrating telescoping rods into the SUPERball design, which will enable very large tensegrity landers to be deployed from small launch volumes.

12.1 MER Mission and Airbag Systems

In 2004 NASA landed Spirit and Opportunity on Mars for the Mars Exploration Rovers mission. The EDL sequence for the mission included an initial velocity decrease via the aeroshell during entry, further velocity reduction with a parachute followed by retro-propulsion with solid rockets during descent, with landing being accomplished via airbags [12]. Both Spirit and Opportunity were 177 kg rovers that were each part of a 540 kg landed mass including the airbag system [62]. Spirit landed at a vertical velocity of 8 m/s and a horizontal velocity of 11.5 m/s, while Opportunity landed at a milder 5.5 m/s vertical and 9 m/s horizontal. This makes the normalized impact velocity for Spirit 14 m/s and 10.5 m/s for Opportunity [12]. The rovers each had a Warm Electronics Box (WEB) which was 0.86 meters long, 0.55 meters wide and 0.37 meters high. The total mass of the instruments and support electronics was 22 kilograms [62]. When packaged inside the airbag system the landed payload had a height of 1.14 meters, while the inflated airbags had a diameter of 4.56 meters [1]. Before the mission, the MER lander was tested to determine the performance of the airbag system at a range of impact velocities from 8 to 18 m/s. An upper and lower bound was determined for the acceleration as a function of the impact velocity through linear regression. The upper bound was modelled by $y = 2.151x\text{-}4.625$ and the lower bound by $y = 2.151x\text{-}9.825$ [1]. These results formed a baseline that was used for comparison with the tensegrity structure's ability to attenuate impact.

12.2 Simulation

For the simulations, two orientations were used during the runs. The equilateral orientation shown in Figure 45 is close to a best case configuration for landing as it provides the largest stroke distance for the payload to decelerate since there are no struts the payload may hit before the ground.

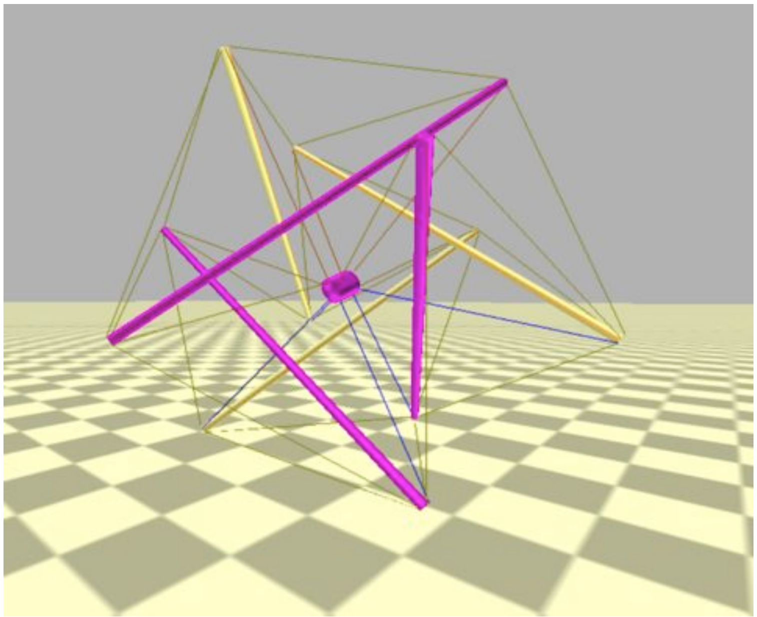


Figure 45: Equilateral triangle impact orientation.

The two strut orientation show in Figure 46 is the worst case orientation for impact due to the minimal stroke distance for the payload to decelerate before it contacts the strut below it.

Most simulations were run in the best case, equilateral configuration, though some simulations were also run in the two strut configuration for comparison at the end of this section. The constant parameters for the struts in the simulation are listed in Table 4. Unless otherwise noted the payload parameters were identical to the parameters listed here in Table 4. Due to the dimensions of the struts listed in Table 4 the mass of the rod scaled linearly as a function of length (m = 0.74*L). A 6 kilogram offset was added to the strut mass to account for actuators and support electronics in the rods. The payload was based off of the MER WEB described earlier based on the assumption that a tensegrity lander would only land with instruments since the system is its own rover.

Parameter	**Value**
Strut outer diameter	1.25 inches
Strut inner diameter	1.01 inches
Strut density	$2700 kg/m^3$ (Aluminum)
Payload mass	22 kg
Payload height	0.86 m

Table 4: Constant parameters for acceleration versus impact velocity simulations.

For most simulations the impact velocity was varied between 10 and 18 m/s to match the MER experimental data previously mentioned. Strut lengths for the equilateral orientation landing were varied between 3 and 5 meters by half meter increments. Struts that were 3.8 meters long were also simulated as this length represents the longest six strut tensegrity that could be packed behind the 4.5 meter aeroshell used for MSL assuming the structure was packed as a triangle.

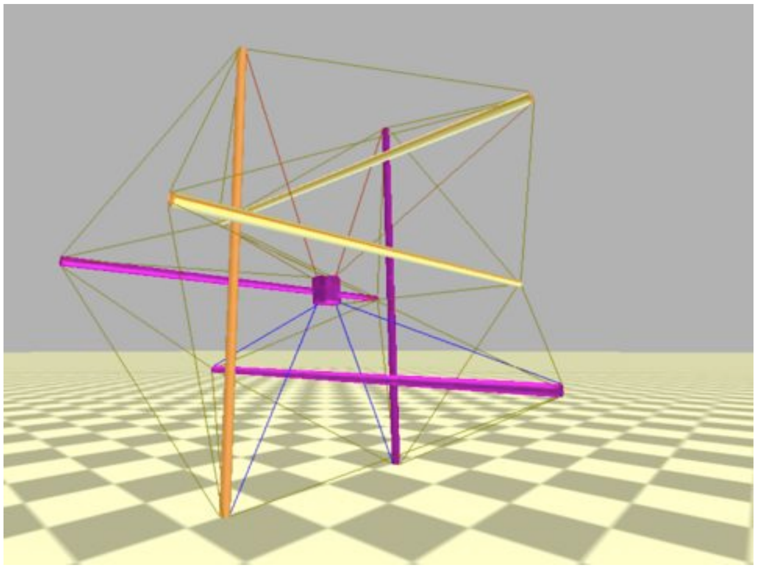


Figure 46: Two strut impact orientation.

12.3 Equilateral Landing

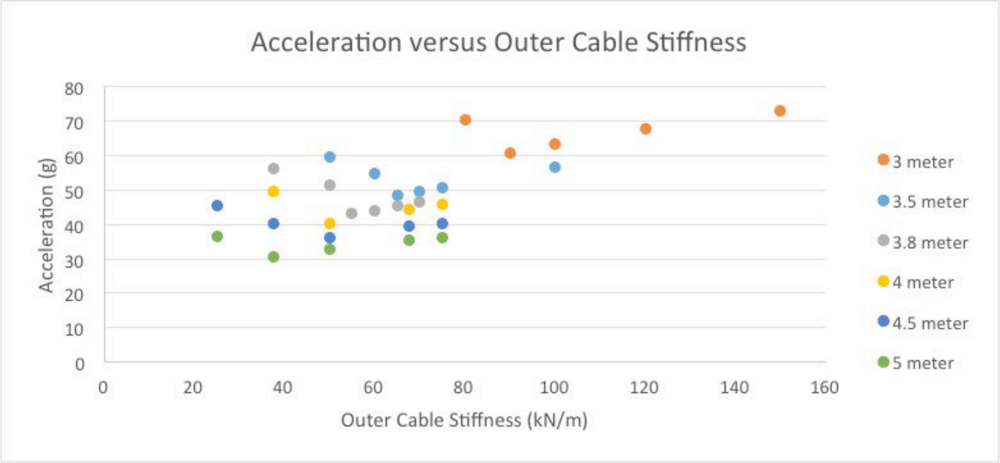


Figure 47: Change in payload acceleration for different strut lengths as a function of outer cable stiffness after the inner cable stiffness has been tuned to minimize acceleration.

The performance of tensegrity structures as impact attenuators may be optimized by tuning the stiffness of the outer and inner cables. The plot shown in Figure 47 illustrates the procedure used to find the optimal parameters for minimizing payload acceleration. For each strut length a range of outer cable stiffnesses was chosen. Smaller structures required stiffer cables to reduce the distance the payload deflected which intuitively lead to larger acceleration values. For each outer cable stiffness, the inner cable stiffness was tuned to minimize acceleration with the constraint that no contact with the ground or other struts was permitted. This procedure lead to the plot shown in Figure 48, which enabled a straightforward selection of the optimal inner and outer cable stiffness for each strut length.

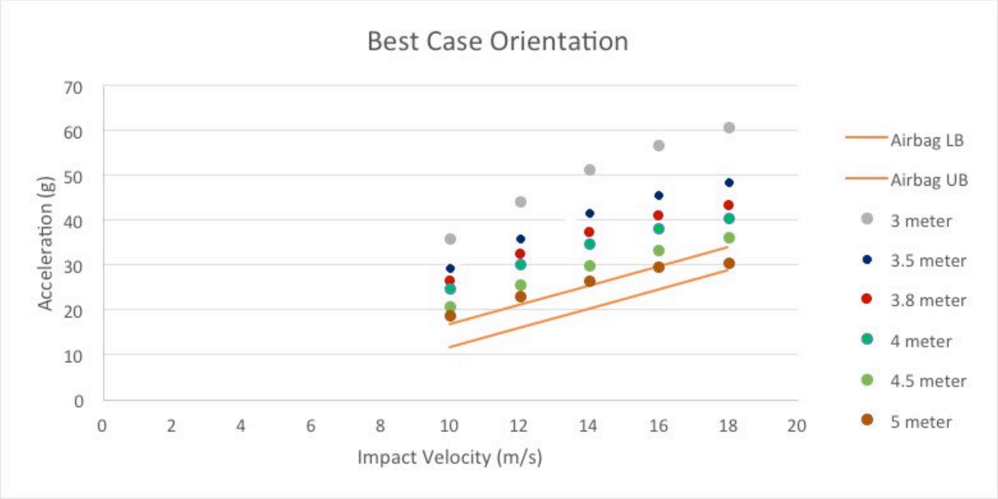


Figure 48: Payload Acceleration as a function of impact velocity for various strut lengths as compared to the upper and lower bounds found when testing the MER airbags for landing in equilateral orientation.

It should be noted that these results were obtained using the largest impact velocity of 18 m/s and those cable stiffnesses were used for all the subsequent simulations of that strut length over the range of velocities. Of course, the accelerations plotted in Figure 48 could have been further reduced at lower impact velocities by optimizing the cable stiffnesses at each velocity. However, it was believed to be more useful for this analysis to see how a structure tuned for a large impact velocity would behave at lower impact velocities. The results shown in Figure 48 show nearly linear behavior for the acceleration as a function of velocity. The linear increase in acceleration appears to flatten out towards the larger velocities, but that may simply be because it was optimized for that high impact velocity. Interestingly, after increasing the strut length from 3 to 3.5 meters the benefit of increasing the strut length appears to be more incremental, though still significant. Overall a 5 meter strut length appears to get close to achieving the attenuation behavior of the airbag system. It should be remembered that tuning the structure for lower impact velocities than 18 m/s would enhance its performance, although it would also decrease the range of acceptable impact velocities.

The plots show data on the peak forces experienced by the entire structure during impact. Figure 49 shows the peak axial stress in the rods, while Figures 50 and 51 show the peak force in the outer and inner cables respectively. Overall the peak loads appear to decrease linearly with decreasing velocity. For the outer cables the forces range from 5.4 to 13.6 kN, while the forces in the inner cables range from 1.4 to 4 kN. As the structure scales up the peak stresses in the rods increase even as the peak forces in the cables decrease. The stresses range from a low of around 70 MPa to a high of nearly 150 MPa. These are well below the yield strength of the 6061-T6 Aluminum alloy. The stress increase for larger structures may be due to rod mass increases, while the force decrease in the cables for larger structures is likely due to the lower cable stiffness required at the larger structure sizes.

Overall the stress in the rods appeared to decrease roughly linearly as the velocity decreased for each unique rod length. As the structure size was scaled up from 3 to 5 meter struts the peak

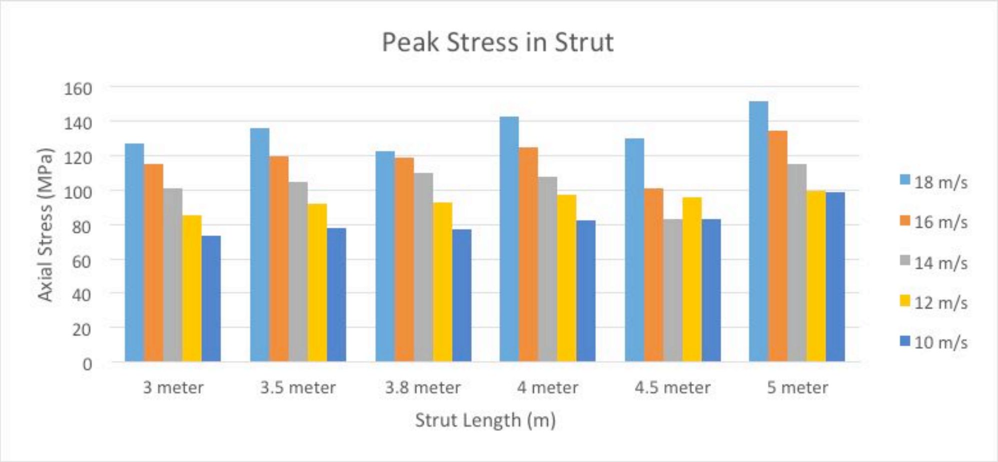


Figure 49: Peak axial stress occurring in strut during impact simulation.

stress in the struts did tend to increase, but only by around 25 MPa. The peak simulated stress of 150 MPa is still well below the yield strength of the 6061-T6 Aluminum Alloy that is a potential candidate for the strut material.

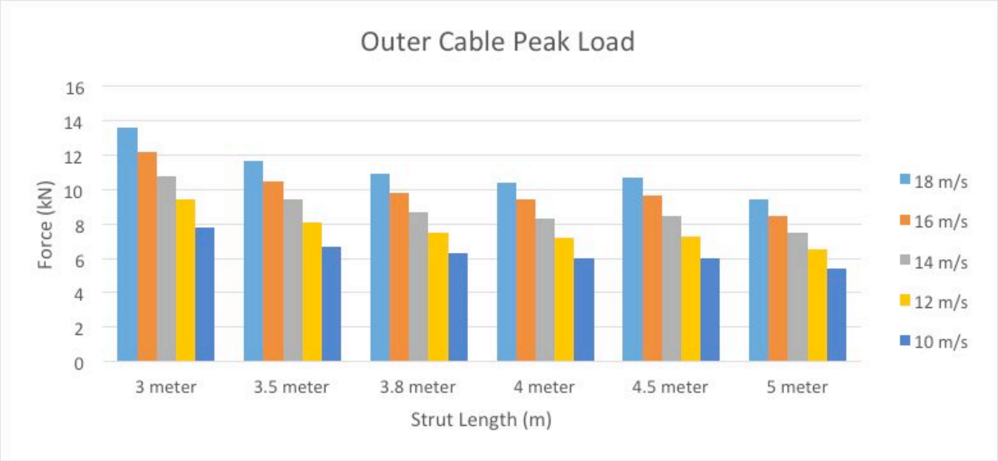


Figure 50: Peak force occurring in any outer cable during impact simulation.

For both the inner and outer cables the peak forces decreased with both velocity and strut length. The decreased force due to velocity is intuitive as is the decreased force with increased strut length since the increased strut length allows for more compliance in the cables because more space exists for the payload to deflect. For the outer cables the peak load centered around 10 kN, while the peak load in the inner cables was generally closer to 2 kN.

The tensegrity system was further compared with the airbag system by means of results that were obtained through testing and simulation of the airbag system [1]. The first two results shown in Table 5 are for simulations of the airbag system at a 12 m/s impact. The third value is experimental data and the fourth value is a result achieved by the tensegrity structure in a simulation run with the parameters listed in Table 6. This result showed that a tensegrity structure small enough to fit behind an aeroshell could achieve very similar performance to an actual airbag system given

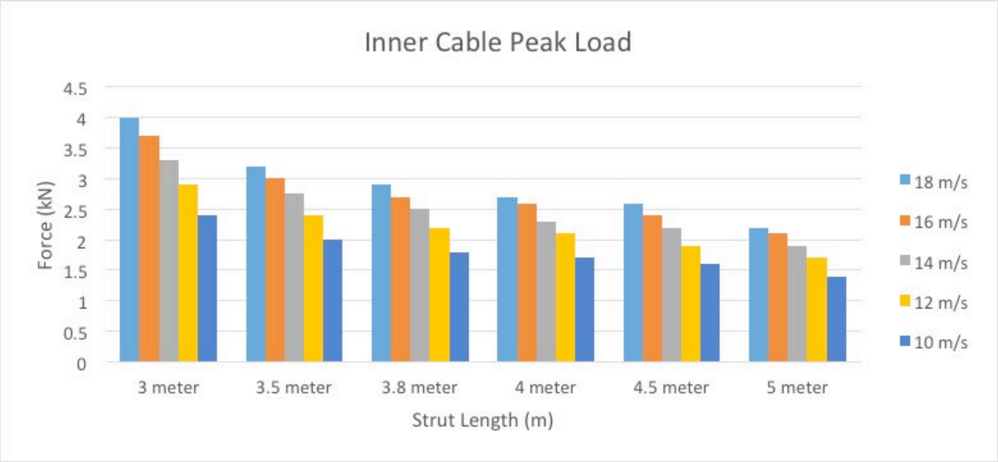


Figure 51: Peak force occurring in any inner cable during impact simulation.

correct tuning of the outer and inner cable stiffness.

Bag-M (g)	ADAMS (g)	Superflat Test (g)	Tensegrity EL (g)
16.4	14.7	16.0	16.4

Table 5: Impact at 12 m/s simulations and tests of airbag and simulation of 3.8 meter strut tensegrity.

Parameter	Value
Outer cable stiffness	220 kN/m
Inner cable stiffness	10.6 kN/m
Strut length	3.8 m
Payload mass	177 kg
Payload height	1.14 m
Orientation	Equilateral

Table 6: Parameters for simulation for 12 m/s comparison with MER system.

12.4 Two Point Landing

For the two strut landing struts were simulated as being either 4.5 or 8 meters long. The longer struts were required to get similar performance to the airbag system due to the shorter stroke distance for deceleration mentioned previously. The inner and outer cable stiffness was optimized for the peak impact velocity of 18 m/s then held constant for the simulations at lower impact velocities as was done in the equilateral configuration above. Two payloads were simulated. The parameters for the WEB payload are listed in Table 4. The parameters for the rover payload are listed in Table 6. Figure 52 shows how the peak acceleration after impact changes nearly linearly

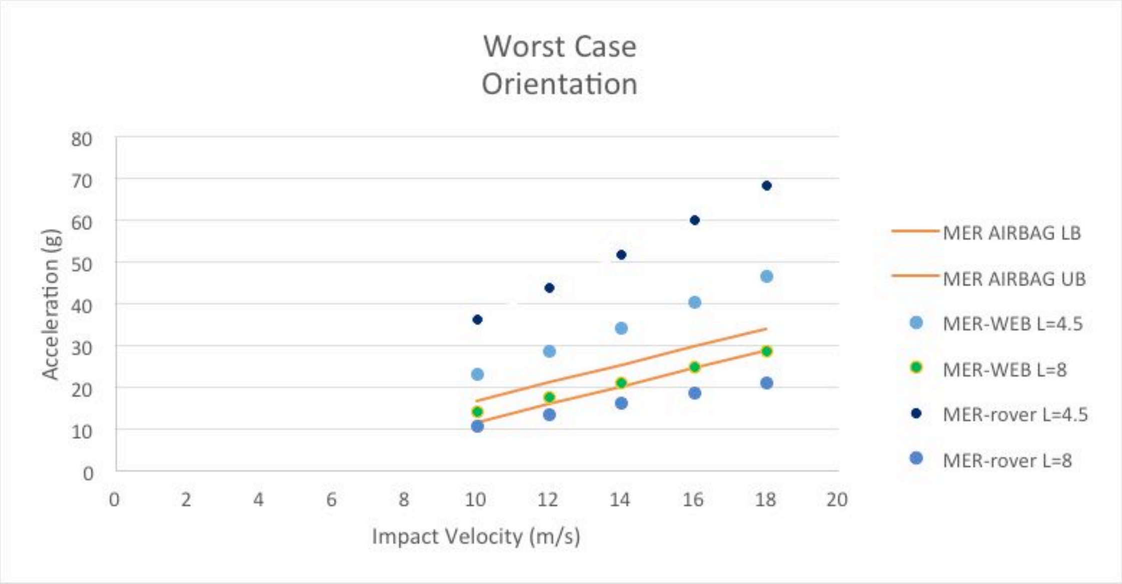


Figure 52: Payload Acceleration as a function of impact velocity for various strut lengths as compared to the upper and lower bounds found when testing the MER airbags for two strut impact orientation.

as a function of impact velocity. While the 4.5 meter struts do not perform as well as the airbag system, the 8 meter struts perform as well or better.

12.5 Conclusion for Comparison to MER Airbag

The results of these simulations have shown that when sized appropriately tensegrity systems can match and even outperform the impact attenuation capabilities of traditional airbag systems when used in place of airbags during the landing phase of EDL. This is because unlike airbags, tensegrities appear to scale well to larger sizes. This is because a larger tensegrity provides a longer stroke length for deceleration. Forces in the cables also go down since the larger structure can be more compliant. In other words the cables in a larger structure do not have to be as stiff as they do for smaller structures. Furthermore, the low increase in system mass due to increased strut length combined with the more compliant cables creates a minimal increase in peak stress in the struts even at larger impact velocities. This permits structure size to increase significantly before mechanical loads become a greater issue than the benefit received for impact attenuation due to the larger size. The low system mass, particularly when compared with airbags, brings several advantages with it. First, more of the mass budget can go to instrumentation instead of being spent on the lander or rover. The lower mass may also open up more areas to exploration. This is because a lower mass exploration vehicle can be more easily slowed and directed during EDL. This in itself is a great aspect of the tensegrity system. The current limiting parameter on size appears to be the size of aeroshells which the structure can be packed behind. For this reason, a number of simulations were done with a structure with 3.8 meter struts, since that length of struts would allow it to be packed behind an aeroshell such as the 4.5 meter diameter aeroshell used for MSL. At an impact velocity of 12 m/s these simulations showed that the tensegrity could match the capability of the airbag system despite the tensegrity's smaller overall diameter. Still, development of telescoping or otherwise deployable struts would be a useful direction for research. Given the ability to both deploy the cable elements and also the struts, one can imagine very large tensegrity structures packing down to very small, dense, packages for launch and transit. While the simulations in this paper will require more extensive experimentation to validate, tensegrity systems appear to have significant potential for impact attenuation of large payloads for space exploration.

13 SUPERball Prototype

Building and testing the SUPERball prototype has been the primary defining feature of our Phase 2 work. During Phase 1 we performed a number of landing impact analysis and drop tests with simple prototypes which convinced us that our structure was capable of landing at 15 m/s, which exceeds the actual landing speeds of the MER rovers in their airbags. During Phase 2 we performed more tests to verify our landing analysis, and extended that work to examine how our tensegrity landers would scale to larger payloads, as is discussed in Section 12.

Yet there is a big difference between looking at how an abstracted structure or simplified payload will survive a landing, and how a fully actuated mechanism, with actuators, sensors, and other design details, would survive the landing events. To that end, our motivation with the SUPERball prototype was to understand how to engineer a tensegrity robot which could roll and explore on the surface of another planet, while carrying an instrument payload. By focusing on the mechanism design to support locomotion, we were ignoring high-speed landing scenarios for this physical prototype. Once we have shown good design for locomotion, we can circle back and harden that design during a future prototype which will be designed to both land at high speeds, and to provide surface mobility.

13.1 SUPERball Design Requirements

Using NTRT, design requirements were developed from the learned locomotion controls. Prior work showed that a 6-strut robot with $1.5\,\mathrm{kg}$ rods, which could withstand average cable tensions of $75\,\mathrm{N}$, peak cable tensions of $200\,\mathrm{N}$, and $800\,\mathrm{N}$ compressive loads on the struts, would be able to execute learned control trajectories [16]. The actuators were also required to adjust cable lengths at $0.2\,\mathrm{m\,s^{-1}}$. As shown below, the current SUPERball version is heavier than intended, at $3.5\,\mathrm{kg}$ per strut including batteries. However, the capabilities of the robot exceed the other design requirements to compensate for this. Though SUPERball has successfully performed some motion primitives, discussed below, work is ongoing to learn more complicated locomotion trajectories with the increased mass with respect to our prior simulations.

13.2 Comparison with other Spherical Tensegrity Robots

The size, weight, and capabilities of SUPERball are unique in comparison with other tensegrity robots, including those made within our Dynamic Tensegrity Robotics Lab (DTRL). Table 7 compares this version of SUPERball to the previous iteration, and to other similar robots. A primary differentiator is SUPERball's autonomy and onboard power, making it entirely untethered, as opposed to such robots as Hirai's pneumatic tensegrities [48].

Notably, SUPERball only has 12 actuators for 24 spring-cable assemblies, due to the extra space required for the $100\,\mathrm{W}$ motors used in its end caps (see Section 13.3). This is more than ReCTeR, with 6 actuators, and less than the UC Berkeley Rapid Prototyping Kit robot, with a full complement of 24 actuators (one per cable). However, SUPERball has significantly more capabilities than the prototyping kit: SUPERball's motors are many times faster than the kit's, and can retract fully. This advantage of range can be illustrated by the minimum number of actuated cables for a single

Table 7: SUPERball and Related Robots Design Overview.

	l_{strut}	Δl_{act}	$k_{passive}$	tethered?	control	f_{act}	#act.	mass	sensors	actuators	ref.
Pneumatic	0.57 m	-	-	Y	open loop	800 N	24	3.3 kg	none	McKibben	[48]
ReCTeR	1 m	0.3 m	$28.4\,\mathrm{N\,m^{-1}}$	N	closed loop	12 N	6	1.1 kg	F, L, IMU	DC	[18]
Rapid Proto Kit	0.69 m	5 mm	$1193\,\mathrm{N\,m^{-1}}$	N	open loop	<45 N	24	2.7 kg	none	linear DC	[45]
SUPERball 2014	1.5 m	0.2 m	$613\,\mathrm{N\,m^{-1}}$	N	closed loop	140 N	12	9 kg	F, L, τ, IMU	BLDC	[14]
SUPERball 2015	1.7 m	0.42 m	$998\,\mathrm{N\,m^{-1}}$	N	closed loop	250 N	12	21 kg	F, L, τ, IMU	BLDC	

The variable l_{strut} indicates the length of a strut, Δl_{act} is the nominal spring-cable retraction length in tension, $k_{passive}$ is the linear stiffness coefficient of a passive spring-cable (or active spring-cable if fully actuated), tethered indicates if the robot is powered externally or by internal systems, control indicates whether sensor feedback is used, f_{act} is the nominal actuated spring-cable tension and #act. is the number of actuators. In the sensors column, F represents a linear force sensor (for cables), L is cable length sensor (in the form of motor encoders), τ represents a torque sensor for motors, and IMU represents an accelerometer/gyroscope inertial motion sensing unit. Actuators are specified as DC motors or brushless DC (BLDC) motors. The SUPERball 2014 values are revised original design requirements based on NTRT simulations, and changed to the 2015 values after additional detail design.

punctuated roll: the prototyping kit requires 3 actuated cables to induce a roll, but SUPERball only requires one.

Figure 53 shows the size of SUPERball as well as an initial hookup pattern of the actuated cables (with motor) versus passive cables (unactuated). A symmetric pattern of "actuated triangles" was chosen for SUPERball, where four out of eight equilateral faces on the robot are fully actuated, and other faces are fully passive [14]. This initial approach mimics some of our earlier simulated controls research, such as the Flop and Roll algorithm, which focused on actuating the edges of the triangular face in contact with the ground. This connection pattern conservatively guarantees basic motion primitives - when placed on an actuated face, and is the approach used for the work reported in this section. This approach can leave the robot stuck if it accidentally ends up on a non-actuated face. Thus, current studies, not yet ready to be published, are using a different connection pattern which solves that limitation.

13.3 System Design

SUPERball is composed of 12 fully independent, autonomous units, termed *end caps*, which slide into each end of hollow aluminum tubes to create the 6-strut robot. Each of these end caps has a single 100 W Maxon brushless DC motor for actuation (Maxon 386674), four custom PCBs which serve various purposes, batteries, wireless communication, and two internal springs as part of the spring-cable assemblies. Prior work has discussed the mechanical design of these end caps [18, 14, 76]. Figure 54 shows one end cap in detail. Novel mechanical and electronic elements of SUPERball's end caps are discussed below.

13.3.1 Spring-Cable System and Sensors

In SUPERball, the tensile elements are called *spring-cable assemblies* and consist of a combination of steel wire cable, Vectran cable, a compression spring, sensors and optionally an actuator. Figure 55 shows the conceptual model of such a system. Each of SUPERball's motors is attached to a 1.4 mm Vectran cable (Cortland 7012 Vectran HT, 2.2 kN breaking strength). The opposite

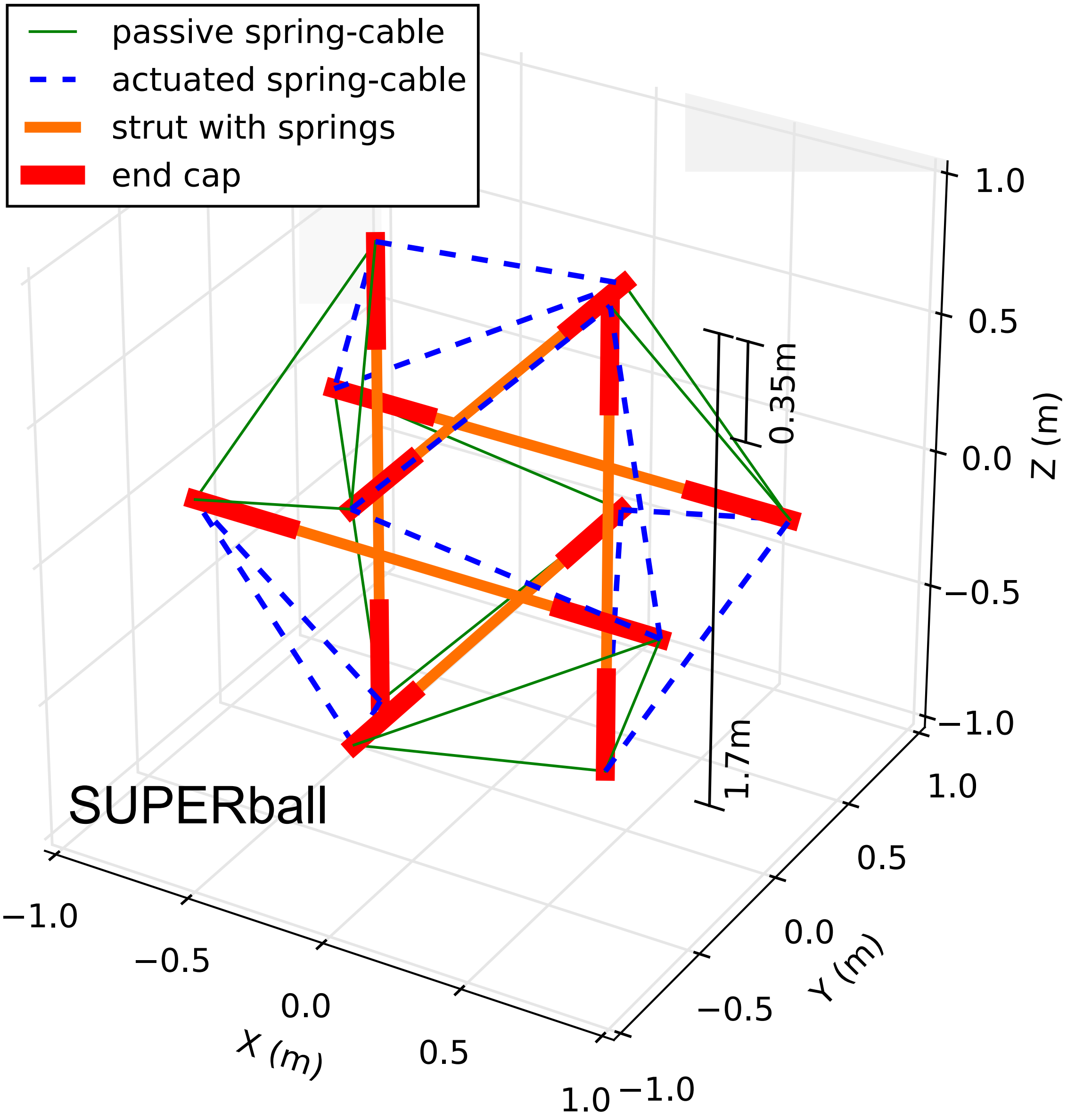


Figure 53: SUPERball size and active cable hookup pattern, which currently actuates half of the equilateral triangular faces.

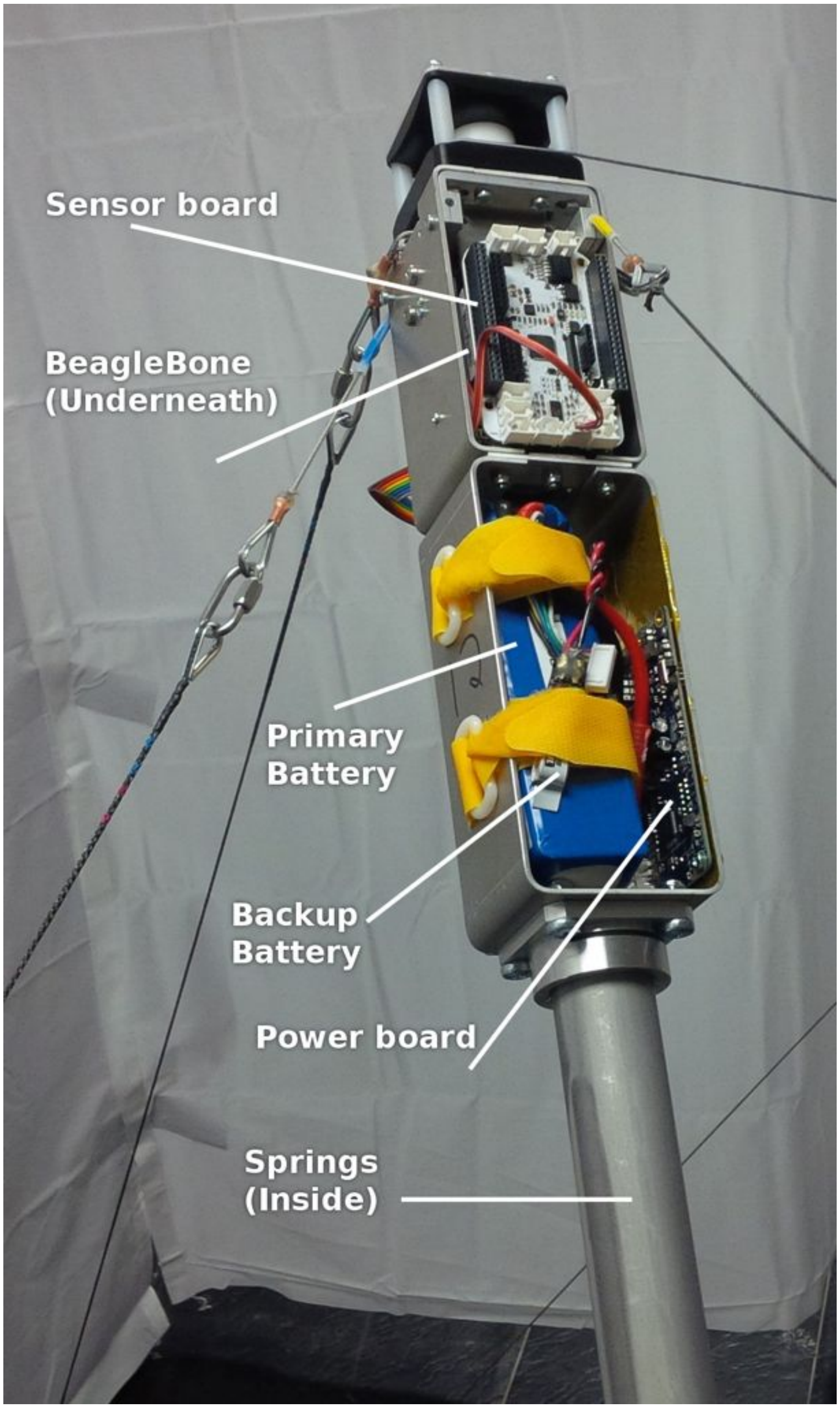


Figure 54: One end cap of SUPERball. The springs of the spring-cable assemblies are inside the hollow aluminum tube. Steel wire cable transfers motion from the springs to the outer cables, and is routed through the end cap assembly using various PTFE tubes and pulley-bearing elements.

end of that cable is looped onto the free end of a steel cable, close to the opposite rod, which then transfers force through the cable into a spring inside that other rod. This internal enclosure for the springs is motivated by observations from past work where environmental snagging occurred on a similar robot's external springs [18]. Figure 56 shows the internals of the entrance point of this cable, which winds over a bearing then into a PTFE tube and then travels through the back of the end cap to the spring area. Unactuated (passive) cables are attached directly to one end cap and to a compression spring and sensor inside another end cap.

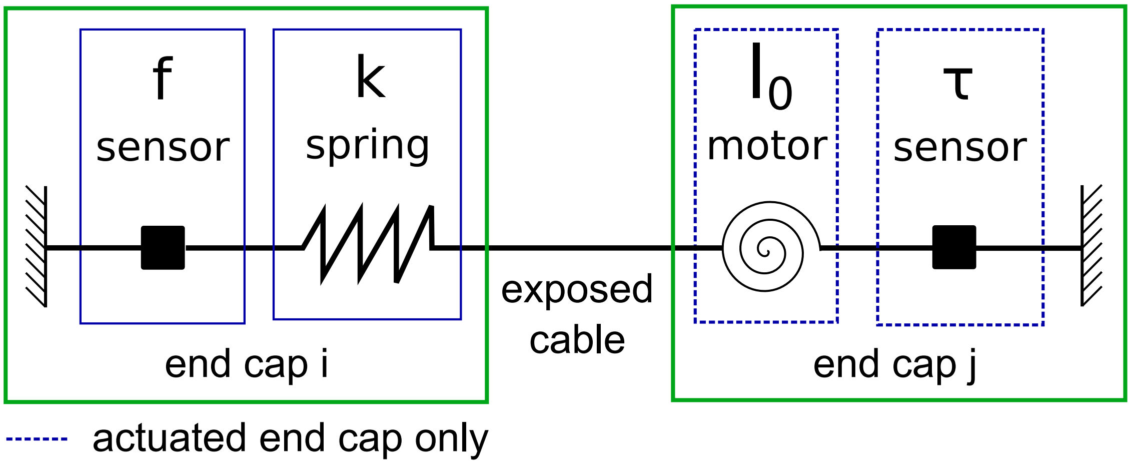


Figure 55: Conceptual model of a SUPERball spring-cable assembly. Each spring-cable assembly contains a (compression) spring with linear stiffness k. The current spring force or cable tension f is measured by an in-line compressive force sensor. Only the cables are exposed to the environment as all sensors and actuators are embedded in the end caps. The dashed elements - motor & torque sensor - are available on actuated spring-cable assemblies. Such assemblies are effectively series elastic actuators (SEA) with a significant amount of passive compliance. The remote cable tension f on end cap i is only available to the motor controller on end cap j through a wireless link. Motor-side torque sensing τ allows for local tension control on end cap j, without introducing stringent requirements for the wireless link, which would reduce the flexibility of SUPERball's distributed design.

Each spring-cable assembly has minimum of two sensors. For direct motor torque measurements, the motor mount itself is designed as a torque sensor: a strain gauge attached to one of its legs measures torsional displacement. These sensors, previously discussed in [14, 76], were calibrated externally before assembly. Each of the springs, located at the bottom of the endcap for both active and passive cables, are paired with a linear force sensor in the form of a small aluminum mechanism with a strain gauge attached. Also, each sensor board is equipped with a 9-axis inertial measurement unit (IMU).

13.3.2 Sensing, Control, and Power Electronics

SUPERball employs four circuit boards per end cap. A custom sensor collection board, motor control board, and power distribution board are connected over a Controller Area Network (CAN) bus. A single Beaglebone Black is also housed on each endcap to directly implement high level code and facilitate WiFi communications. Each of these four PCBs has at least one microcontroller, with the three custom boards using Microchip's DSPIC33E series. SUPERball also has two types of wireless communication: WiFi on the Beaglebone, for data collection and high-level control, and a chip from Nordic Semiconductor on the power board that control the emergency kill switch.

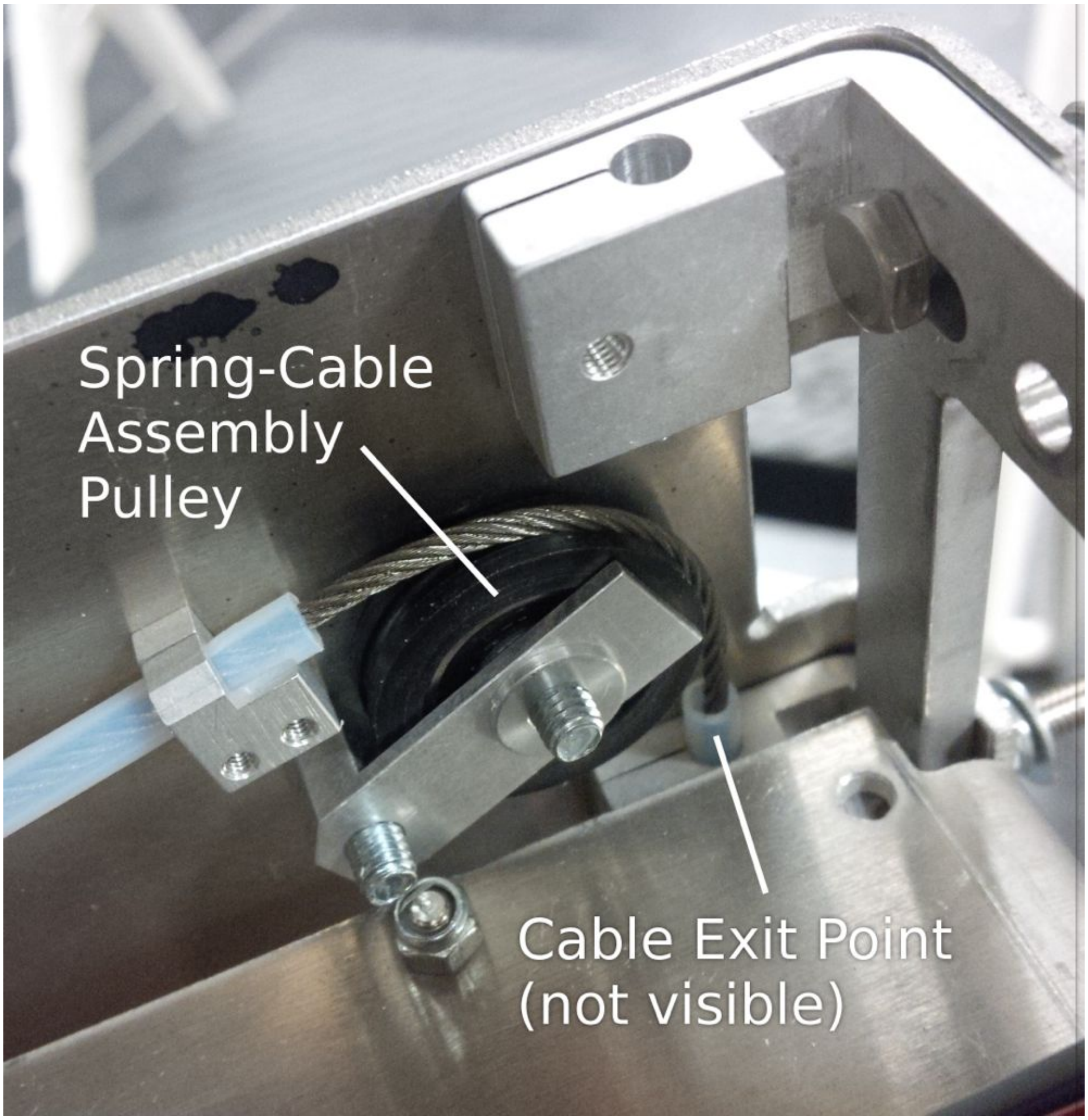


Figure 56: Internal view of the pulley and exit point of the steel cable, which connects to springs inside the strut tube on one end and to the external vectran cables on the other.

Each sensor collection board takes in sensor data from strain gauges and processes the information for embedded controls on each end cap. For example, a 24-bit ADC reads sensor values from the motor mount torque sensor (discussed in detail in [14]). After processing and conversion into suitable inputs for the motor control, these data are sent over CAN to the motor control board. The sensor board thus calculates the primary closed-loop feedback controller currently on SUPERball. Future work will integrate higher level control on the BeagleBone's more powerful ARM microprocessor.

The power distribution board regulates the two independent lithium-polymer batteries on each end cap. Since calibration for SUPERball's cable lengths is expected to be a challenge, a backup battery is used to avoid recalibrating cable positions when changing out the primary cell. This board automatically switches between the larger $3\,\mathrm{A\,h}$ cell and the backup $160\,\mathrm{mA\,h}$ cell when the larger cell is disconnected.

13.3.3 Motor and Actuation Electronics

The motor control board implements the lowest level of control. Cable-length position control and speed estimation are achieved through the integration of hardware quadrature encoders on a dsPIC33E. Thus, the motor control board uses its own lowest-level controller with just the motor encoder positions as feedback, and then acts as part of a mid-level controller in coordination with the sensor board over CAN. Commutation is performed through space vector modulation, and the motor board's position controller uses a system ID'd model and a tuned proportional control. To assure temporal accuracy and synchronization during data acquisition and motor control, dedicated hardware peripherals of the dsPIC33E are leveraged to reduce the amount of operations which are otherwise costly and frequently non-reentrant computationally in an embedded environment. Figure 57 shows a motor control PCB on an end cap.

13.4 Control Schemes

Prior work has discussed the locomotion goals of SUPERball [14, 18]. Since SUPERball has the capability to sense tensions on all cables, and includes inertial motion sensors for optional sensor fusion capabilities, full state estimation is possible. However, high-level trajectory-tracking control requires robust low-level feedback controllers for the brushless DC motor. In order to show that SUPERball is capable of complex movement, a linear-quadratic-Gaussian (LQG) position controller on the motor board was tested while sensor data was collected from multiple motor mount torque sensors.

For the results discussed below in all tests, a time series of inputs (a trajectory) was commanded using feedback from the motor encoder. This emulates future control schemes that will be based on state trajectory following, and shows the capabilities of SUPERball to perform such control.

13.5 Results

13.5.1 Position Feedback Control and Sensor Validation for Trajectory Tracking

For all the sensor readings below, tensions were calculated from torque values using an a-priori calibration curve.

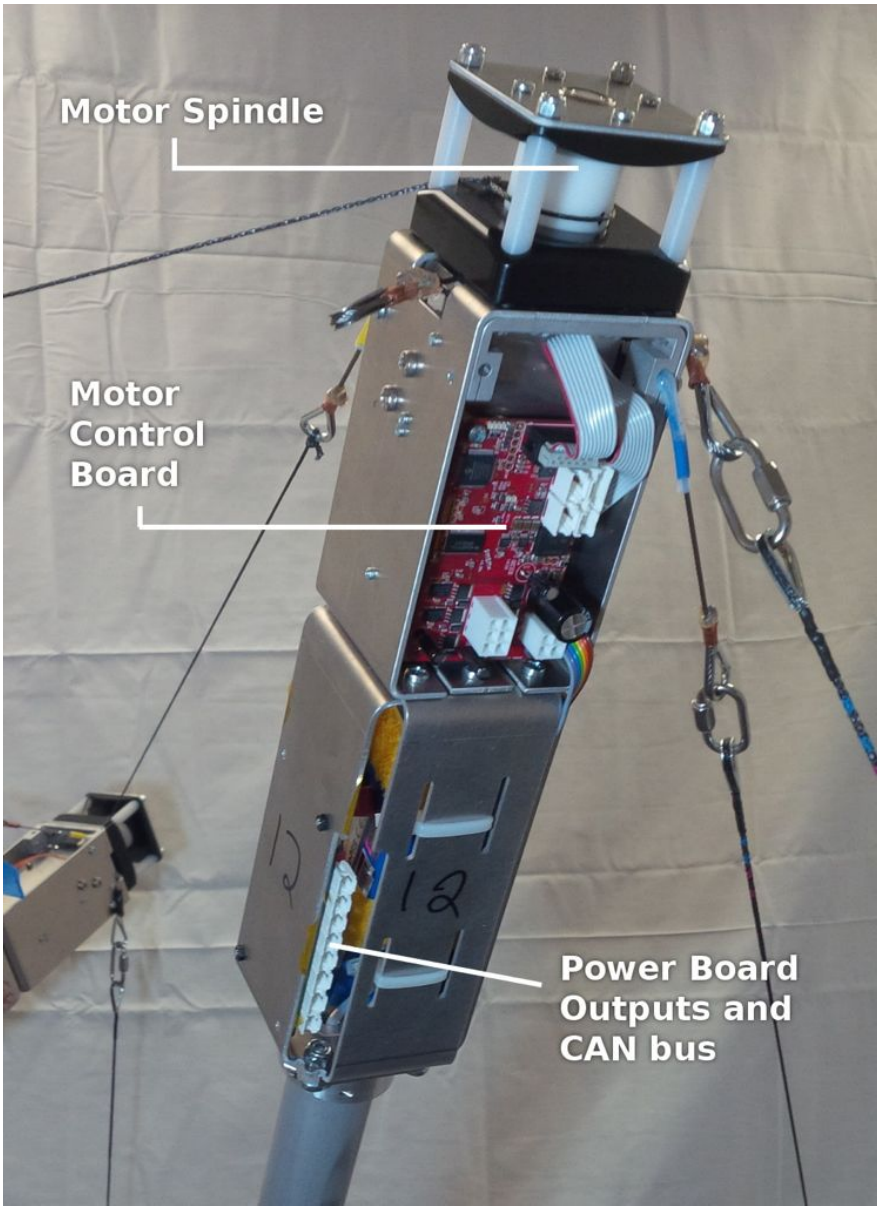


Figure 57: Reverse view of SUPERball's end cap. This view shows the motor control PCB, the back side of the power management board, and the motor spindle.

13.5.2 Pseudo-static Kinematic Sensor Testing

Three tests were performed to validate the distribution of tension throughout the system, and to show that all sensors can work in conjunction simultaneously. Figure 58 shows tension readings from a different motor-mount torque sensor on the opposite side of SUPERball (*Cable 2*) from a cable which is being retracted (*Cable 1*.) *Cable 2* was not actively actuated during each test. For each plot in Figure 58, the actuated cable was retracted with various step inputs marked in the figure. Each data point in this figure (yellow) was collected by averaging data from the sensorboard for a total of 5 seconds at 1 kHz, after waiting 2 seconds after the step input actuation to avoid dynamic effects. These tests were done with different levels of pretension on the sensed cable: this pretension was adjusted by changing the length of the sensed cable. Though the lower-pretension tests show smaller changes in readings, the higher pretensions show monotonically-increasing readings which demonstrate the ability to sense forces throughout the tension network in pseudo-equilibrium states, as well as SUPERball's passive force redistribution properties.

13.5.3 Dynamic Sensor Testing

One additional test was performed to demonstrate the force sensors' ability to capture data under dynamic motion. Figure 59 shows a plot of sensor data from one end cap whose motor is commanded in a square-wave position trajectory. The position trajectory had a period of $13\,\mathrm{s}$, and oscillated between $10\,\mathrm{rad}$ and $15\,\mathrm{rad}$ of the output shaft measured before the gearbox, by the encoder. The trajectory of sensor torque values reasonably tracks the position square wave: the commanded position trajectory starts at 10 seconds and ends at 62 seconds, as does the sensed tension square wave. The overshoot on the torque sensor measurements is due to the system inertia and spring dynamics.

These results show that SUPERball should be able to combine both closed-loop feedback with state estimation for trajectory following of multiple cables. Such tracking is crucial to future work, where system-level control will be employed for following a highly dynamic whole-robot rolling trajectory.

13.5.4 Basic Locomotion

Using this step-input controller, SUPERball can perform simple motion primitives. Though more intricate and complex locomotion results are to come in future work, these motion primitives of punctuated rolling ("flopping") are performed under (currently limited) feedback control, a novel feat for this class of tensegrity robot.

Figure 60 shows still frames from an experiment where the motor position controller retracts a cable. The retraction distance required for movement was hand tuned, and will be validated against simulations in future work. In practice, one single face-change movement required retraction of roughly half of the starting length of one of SUPERball's cables.

In this case, the active cable being controlled is on the bottom triangle, and punctuated rolling is induced when the center-of-mass of the robot moves outside the shrinking triangle base. Similar to

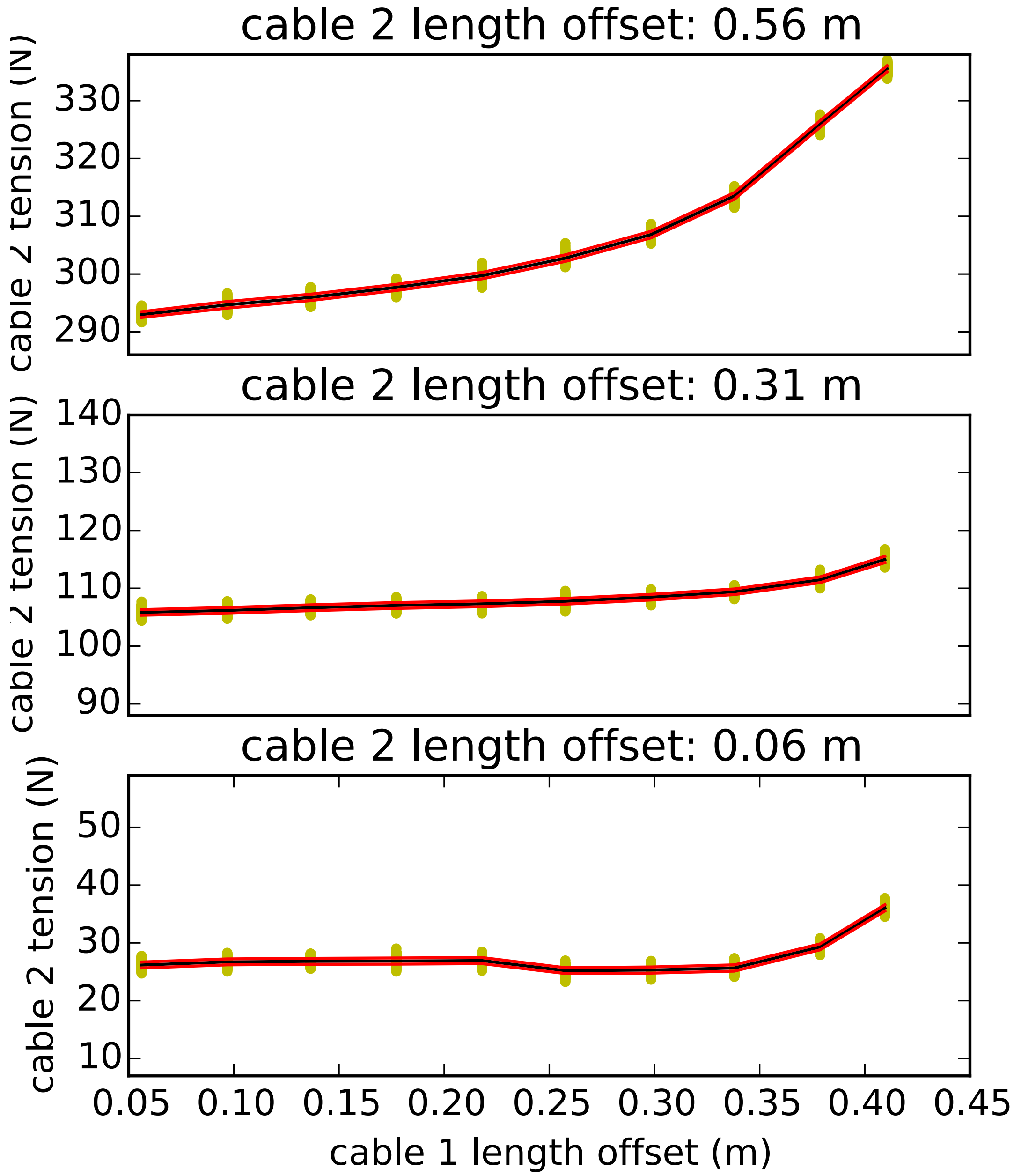


Figure 58: Global force redistribution test. Yellow marks are the means of roughly 5,000 tension sensor measurements of *cable 2* opposite that which is actuated (*cable 1*.) The black line shows the linear interpolation between points, with the red boundary as standard deviation. The pretension in the sensed cable is adjusted in each test, showing measurement sequences at increasing pretensions.

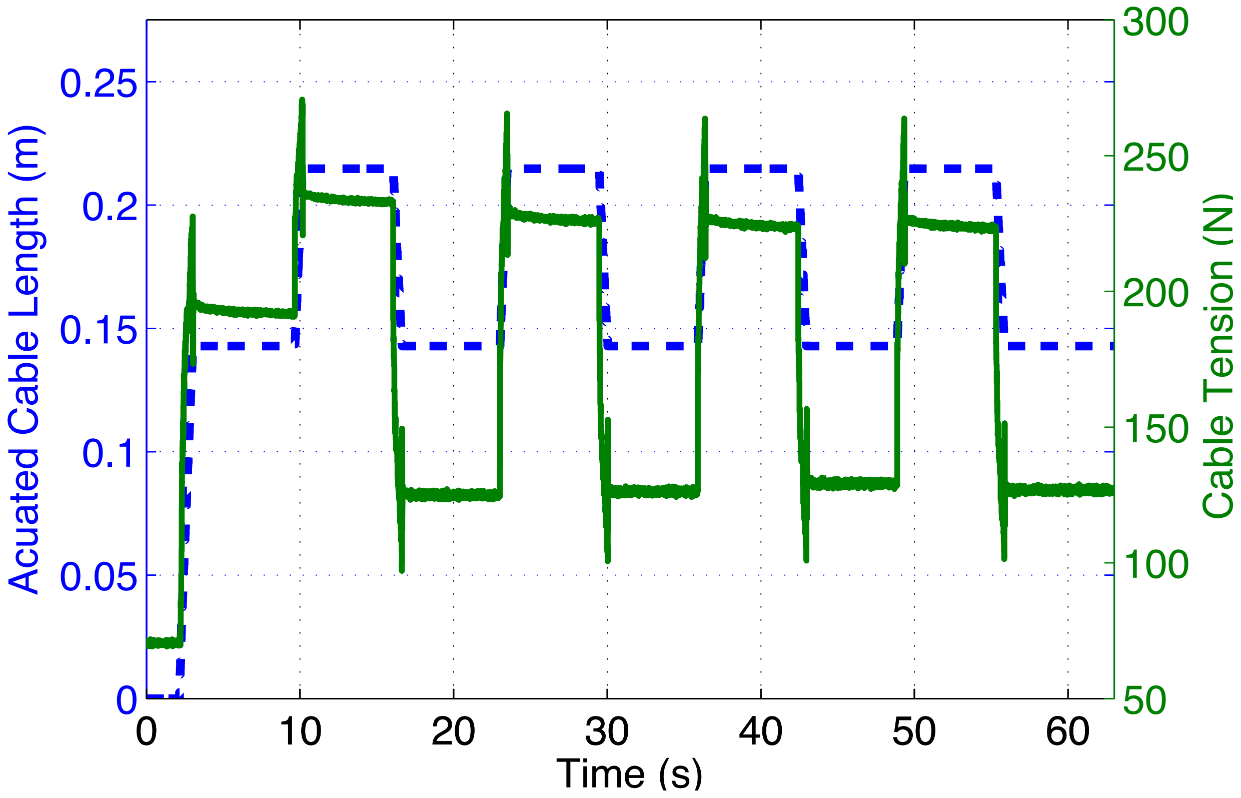


Figure 59: Motor mount torque sensor data and motor position data recorded during a square wave input position trajectory for a single motor. This plot shows measured tension from the sensor and cable length from motor encoder measurements as a function of time for this dynamic movement.

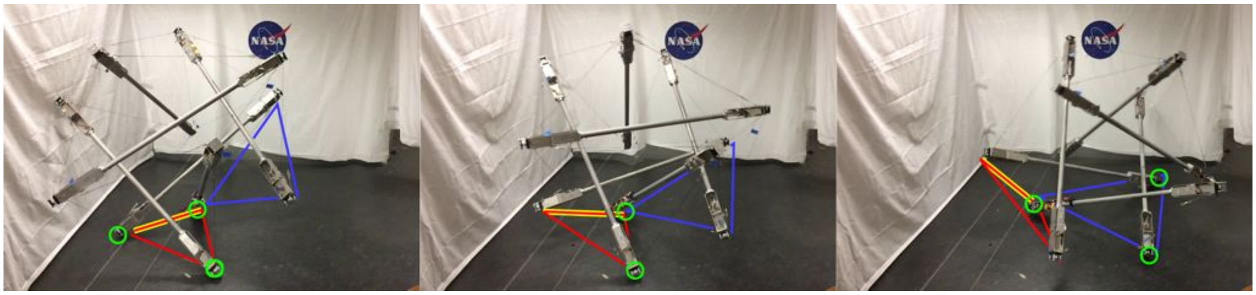


Figure 60: SUPERball performing a single *punctuated roll* or face-change movement, from one equilateral triangular face to another. The robot begins with all endcaps of the red triangle touching the ground, indicated by the green circles. Note that one of these endcaps is also part of the blue triangle. Then, SUPERball retracts the yellow-highlighted cable on the red triangle, inducing movement. Frame 2 shows SUPERball halfway through the movement, in the midst of tipping over, with only two points of contact on the ground. Finally, frame 3 shows SUPERball at the end of this *puncutated roll*, with all 3 points of the blue triangle in ground contact.

the UC Berkeley Rapid Prototyping tensegrity kit [45], this method represents one motion primitive for spherical tensegrity systems.

Shibata and Kim both describe the different possible face-change movements of icosahedral tensegrity structures [48, 45]; in Figure 60, SUPERball demonsrates motion from one equilateral triangle face onto another equilateral face. Though not discussed here, SUPERball is capable of performing punctuated rolls in the other two modes: from an equilateral to isosceles and vice-versa. Additionally, future work will include locomotion in which the dynamics and inertia of rods are used to propel the robot forward, as in [40, 41].

13.5.5 Locomotion Results

The robot is now fully assembled and can now roll in indoor and outdoor environments. The indoor test consisted of rolling across corridors with smooth floor tiles.

The outdoor test was performed in the RoverScape, which is designed to simulate terrain conditions on planetary destinations such as mars. The rolling test were successful as the rover was able to roll across both indoor and outdoor environments. See Figure 61.

In addition, several drop tests of about one meter were performed by driving the SUPERball off of a loading dock. For video of the rolling and dropping tests, see: `https://www.youtube.com/watch?v=ypeAhdNa7-U&feature=youtu.be`

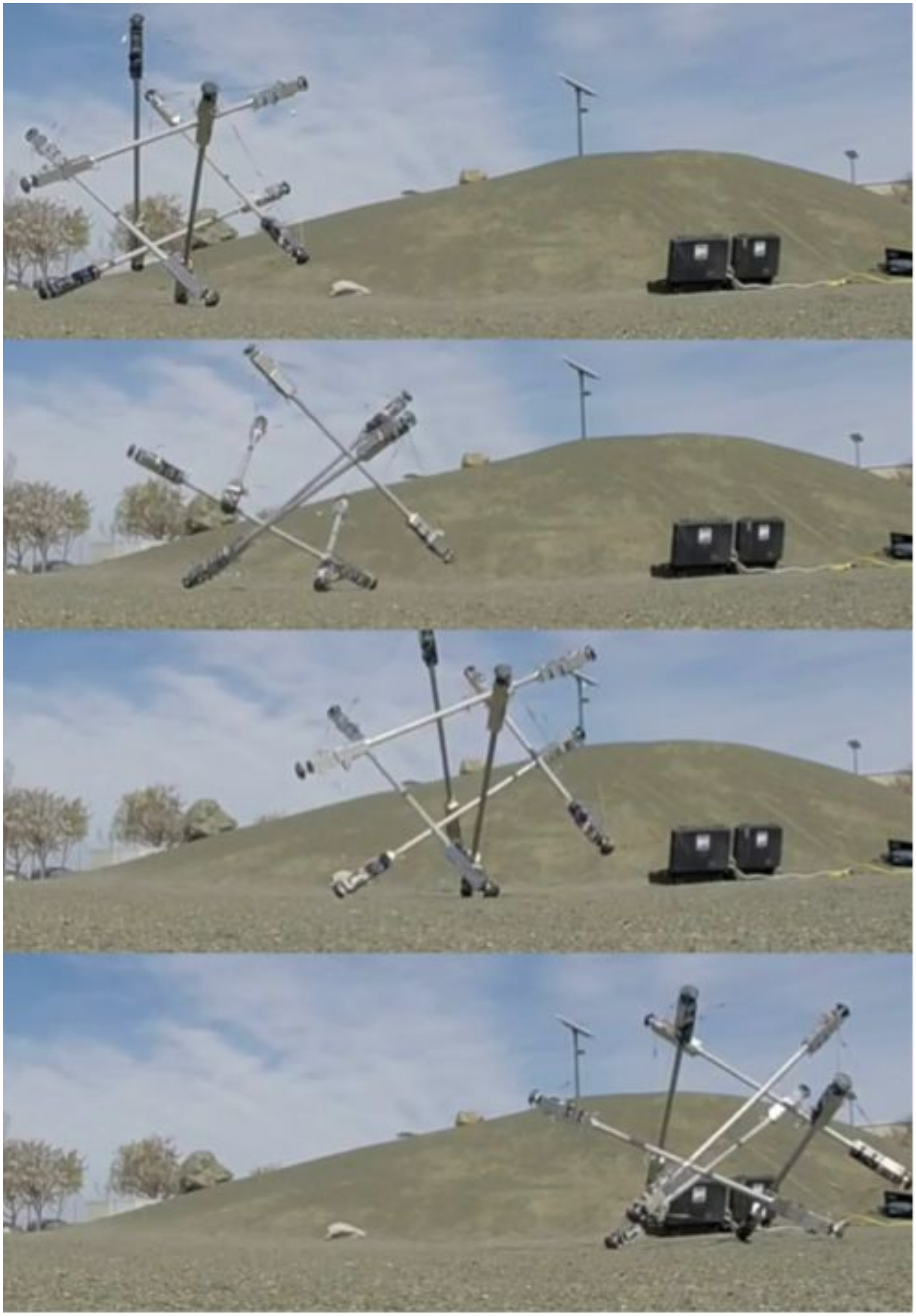


Figure 61: SUPERball rolling in the Ames Roverscape test facility

14 Payload Based Actuation

The current prototype of SUPERBall is actuated through 12 of the 24 cables forming its exterior structure. Greater mobility and agility will be enabled in a future prototype that fully actuates all 24 cables. However, it is also possible to augment current design through actuation of the cables which suspend the payload. This section describes a proof of concept experiment of such an actuated payload which was successfully integrated with the existing prototype and which demonstrated basic rolling capability. This experiment was performed by a group of senior engineering students from the University of Idaho. In this design a payload is suspended at the center of SUPERBall with 12 cables. Within the payload are twelve brushed DC motors connected to spools with vectran cabling extending from the payload to the structure. The 12 motors work in unison to adjust the payloads position inside the tensegrity structure. These movements are used to change the center of gravity of the tensegrity structure, causing the structure to tip. Localization of the payload is done by measuring the lengths of the 12 vectran cables. This is accomplished by using a geared potentiometer system, which provides absolute positioning of the payload. To control this motion a low-level control system has been developed using seven controllers working in a master-slave architecture.

In this design, the payload is able to move around in the entirety of the tensegrity structure (See Figure 62). This task requires both conceptual design and experimentation in order to meet certain specifications. Necessary components included a structure to house the payload, actuators and a control system to provide locomotion, and a power system. The final objective is to have a payload system capable of causing multiple controlled tips of the outer tensegrity structure by moving the payload and changing the center of gravity. This method of locomotion had been previously unexplored.

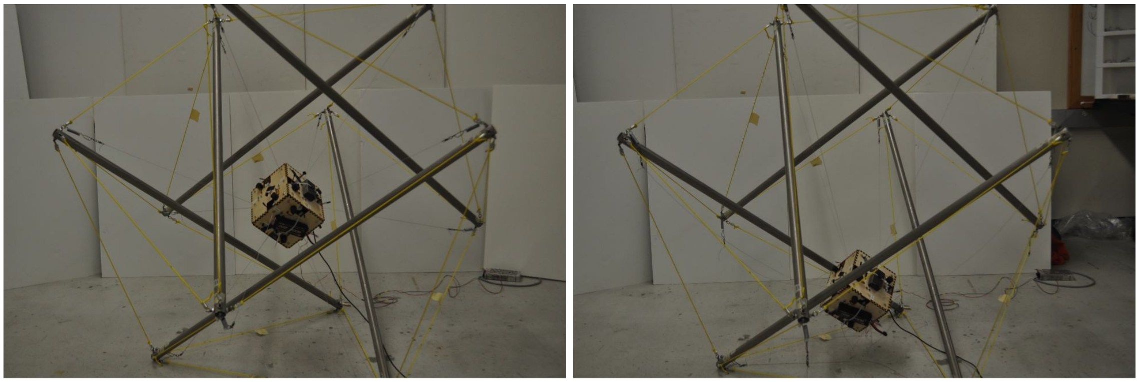


Figure 62: Actuated Payload mounted on the Test Structure, moving to two different locations. The image on the right also highlights how the payload could be brought into contact with the environment for direct interaction by science instruments.

The overall goal of this experiment was to develop a prototype sufficient to demonstrate that the idea was feasible, included moving the controls and power on board, having a run time of at least 30 minutes from an onboard battery, incorporating an accelerometer, having a payload mass of 5 kilograms, and achieve locomotion through 1-10 tips of the structure. It was also desirable to have pre-programmed controls to move the payload and to replace the brushed DC motors with brushless DC motors.

14.1 Payload Structure

The payload structure consist of six square walls laser cut out of plywood (see Figure 63). Each wall has two cutouts on opposite edges for the motors, and one hole on each of the remaining edges for strings, as well as holes to mount the motors. When the box is assembled, the edges of the walls mesh together. Each motor is mounted with a spool attached facing the cutouts on an adjacent wall and positioned over a hole for the string to pass through. Two motors are attached to each wall, and the microcontroller and power bus were also fastened to a wall. Screws are attached to a 3D printed block in each of the eight corners. The square walls of the payload make it relatively simple to design and manufacture, and the positioning of the strings on the faces matches the positions of the ends of the struts on the outer tensegrity structure.

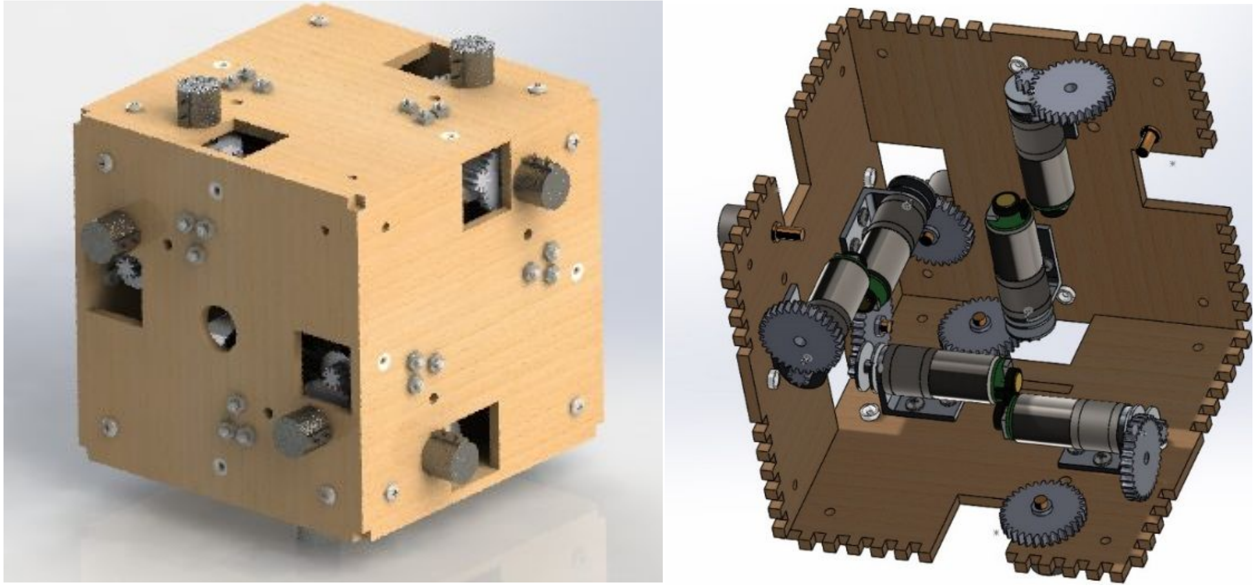


Figure 63: Payload Structure

14.2 Sensing

While motor encoders enable tracking of cable length, they do not provide absolute cable length without careful calibration, which can be time consuming. We desired a means of directly measuring the absolute cable length which would require less frequent recalibration than the motor encoder appraoch. Thus, a geared potentiometer was adopted as the means of position sensing. One potentiometer is used for each motor, and the potentiometers are mounted on the walls adjacent to their corresponding motor. For this method, a gear is attached to a 10-turn potentiometer, and a pinion attached to the spool on the motor that held the cable. When the cable is wound in, the potentiometer turns and feeds the position back to the controller. The gear ratio allows the 10-turn pots to cover the full cable length within its ten turns. Figure 64 shows a CAD model of this new setup.

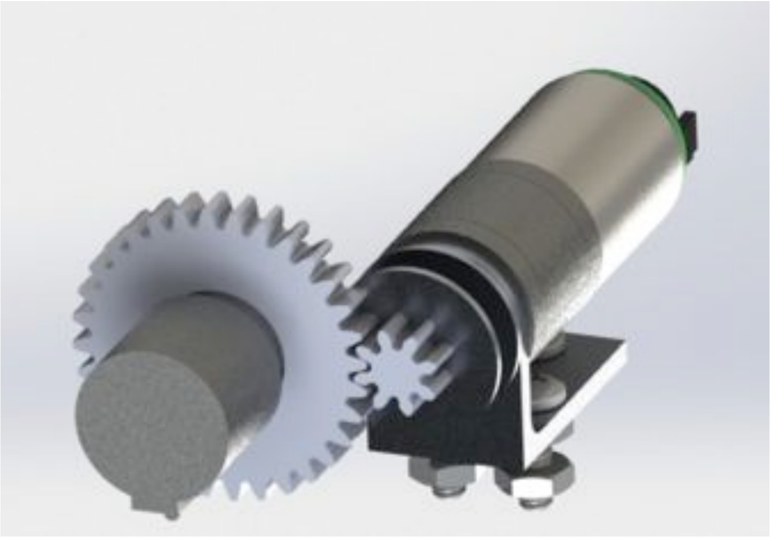


Figure 64: Geared potentiometer measurement system.

14.3 Accelerometer

In order to determine the orientation of the payload, and to open the door to motion sensing in the future, an accelerometer is included in the third iteration. The accelerometer is mounted to one of the struts of the tensegrity structure and tethered to the payload, since the payload itself has too much slack in the cables to produce reliable results. By comparing experimentally determined values to the current value read from the accelerometer, the orientation of the tensegrity structure can be determined.

14.4 Testing

14.4.1 Tipping Testing

The goal of this design project was to develop a prototype payload and explore cable routing and payload movement during a tipping sequence of the tensegrity structure. This tensegrity structure has two unique tipping faces. One is the isosceles face and the other is the equilateral face. These two faces are repeated on the structure. The string lengths for tipping over one of the isosceles faces can be transformed and used for any other isosceles faces. The same situation exists for the equilateral. Because of this only two faces were tested on the tensegrity structure.

To induce tipping on either face additional weight is added to the payload. The payload was originally 3 kg. An additional 3 kg were added to ensure tipping even after errors. This is done by securing steel blocks to the payload. This addition represents our mock science payload and battery weight.

14.4.2 Isosceles Tipping Sequence

For the isosceles face the tipping points were straight forward. Using the static tipping model the string lengths are calculated to induce tipping over the long edges of the triangle. These theoretical points are tested and modified for the actual structure. The main reason for differences comes from the addition of springs, cable routing, and geometric inconsistencies between the actual structure and the model.

Starting from home the payload travels directly to the tipping point. Each controller runs a simple on/off control. Figure 65 models how the payload moves through the structure.

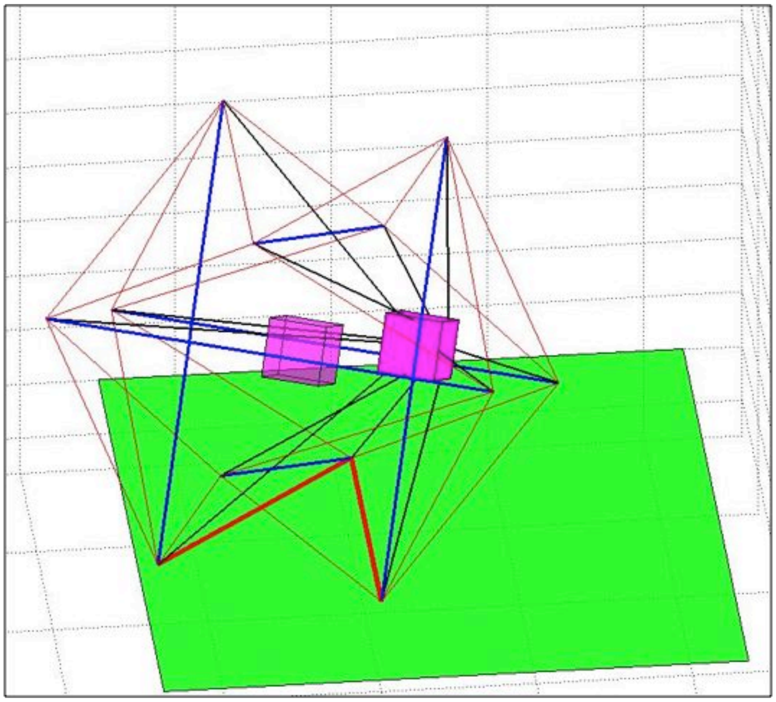


Figure 65: Payload motion to induce tip-over from an isosceles face.

The transparent box displays home position and the opaque box displays the tipping position. The stings controlling the motion move linearly with time, but the payloads trajectory is more complex due to the different speeds of each of the motors. The motion from home position to tip-over takes approximately 25 seconds.

Experimentally this tipping sequence works very well. 90% of the time this initiates a successful tip. The structure also lands successfully. Initial tests revealed some issues with robustness. These included poorly secured gears and cables. After fixing this issue the payload was robust to the landing forces after each tip.

14.4.3 Equilateral Tipping Face

The equilateral tipping face is much less straight forward. The tipping point calculated from our model is harder to get to. First this tipping point is adjusted to reflect real world inaccuracies with the model including: cable routing, springs, and errors. Then it was determined that a new travel sequence was needed in order to reach the tipping point safely.

This new movement sequence has an intermediate point. This intermediate point is directly above home position. By moving the payload up before it moves over allows for all 3 motors to take the load. If the payload were to move over first it would cause too much strain on the main load bearing sting. After the payload has moved to the intermediate point the payload then swings down and over to the equilateral tipping point. Figure 66 demonstrates the tipping sequence for the equilateral face.

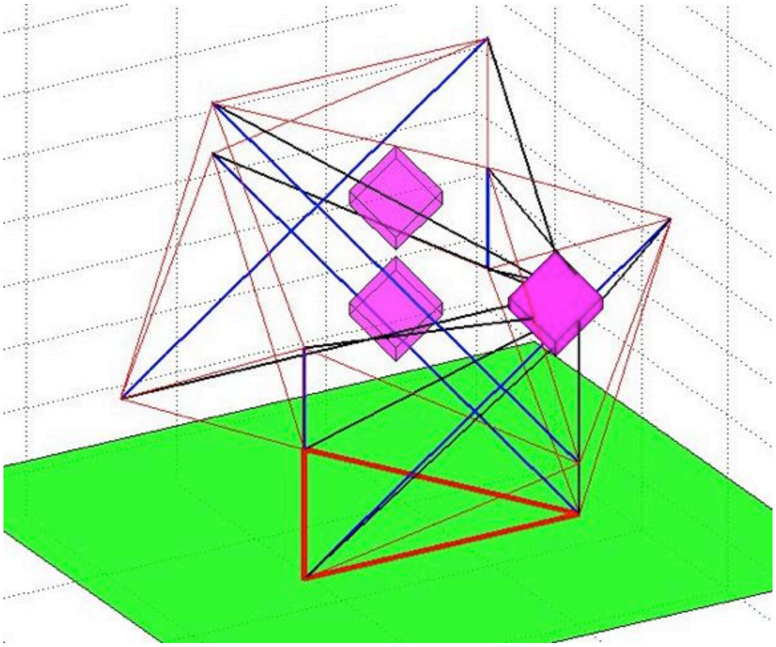


Figure 66: Payload motion to induce tip-over from an equilateral face.

14.5 Conclusions for Payload Based Actuation

The actuated payload was brought to NASA Ames in the spring of 2015 and integrated with SUPERball, where it was able to demonstrate the ability to roll the robot over one tipping point (see Figure 67). This experiment successfully showed the possibility of rolling the SUPERball through actuation of the payload suspensions cables. This opens up a few new possibilities. First, one could combine the current 12 actuated cables of SUPERball with the payload actuation to achieve sufficient mobility without requiring additional actuators to be mounted in the rods. Or, a future fully actuated SUPERball prototype could be further enhanced through the use of actuated payload cables, as we have shown in simulation. Finally, such an "over-actuated" design would provide even greater redundancy in case of partial failure of any actuators.

While the current prototype was successful at demonstrating the concept, there are many engineering improvements which could be performed. Most valuable would be to implement tension sensing in future payload designs. Currently the payload can only move to preprogrammed points, and has no way of knowing whether each cord is slack or taught. Tension sensing would allow feedback to control motor speeds in order to avoid over-tightening cables as the payload moves, and to act as a safety measure in case there are problems with controls or calibration. It would also allow better control over cable slack when the payload reaches a tip over point. Cables could be tightened to a certain tautness rather than a certain length, which would also help eliminate problems with variable spring stretch affecting calibration. Finally, in light of the eventual goal of integrating the payload with SUPERBall, it would allow the payload to respond to the changing shape of an outer frame.

While the spooling system used was generally adequate for the proof of concept, an improved spooling system would increase the reliability of the system in future iterations. Currently the cable winds somewhat unevenly at times causes problems with calibration. In addition, the cable can wedge itself if pulled too hard, causing issues when attempting to unspool. A spooling method that incorporated a level wind system to distribute the cable evenly would eliminate or greatly reduce these issues. Another option that would likely be slightly less effective, but much simpler, would be to design the spool with threads or grooves to help the cable wind evenly.

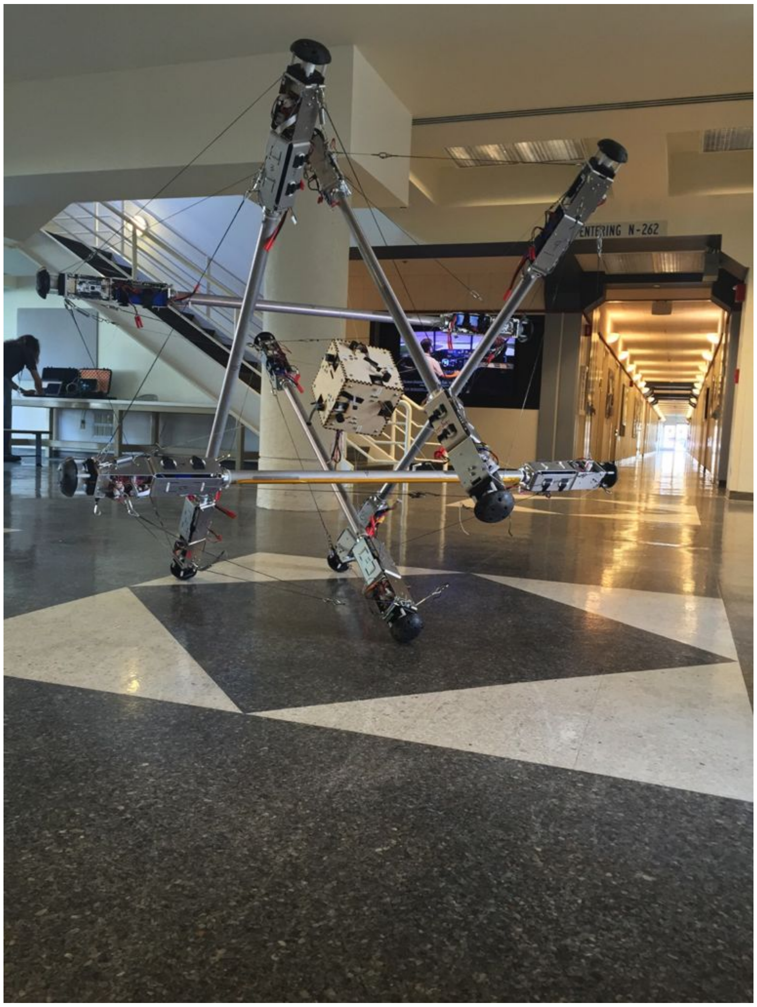


Figure 67: Actuated Payload Integrated with SUPERball prototype.

15 Conclusions and Future Work

When this project was first proposed, the idea of using a tensegrity probe for planetary landing and mobility bordered on science fiction, and it has been our objective over the last years to determine if the concept was sound or if there were fundamental flaws. We are extremely pleased that after many forms of analysis, simulation, and prototype construction and testing we find that the concept is indeed sound and worthy of further research and engineering effort. Clearly there is much more to be done on both the foundational engineering and on the mission design fronts, but at this point we can safely make the following claims based on the work done:

- Tensegrities can be an effective landing and mobility platform for a Titan mission. Compared to existing flown missions, a tensegrity probe can have a high mass fraction between science payload and overall weight (as measured at atmospheric entry) due to its dual use as a landing system (like an airbag) and as a system for surface mobility. As a result, tensegrity based missions can be cheaper and open up new forms of surface exploration that take advantage of their natural tolerance to impacts. Since the robot is designed to land safely from orbit, mission controllers may be willing to take greater risks in order to access high priority science targets, such as exposed rocks on the edges of cliffs or traverse complex jagged terrain on icy moons, on the assumption that the robot could survive unplanned falls.
- Tensegrity robots can be applicable to a wide range of scales and solar system destinations. Fundamentally, their core benefit is to enable landing at much higher speeds than traditional surface robots, and thus reducing the complexity of entry systems, decreasing the need for fuel and extra deceleration phases. Using multiple analysis and simulation tools we showed that tensegrities can protect delicate payloads from a landing impact of 15 m/s (and possibly beyond) and at different size and mass scales. This was further confirmed by performing drop tests on multiple physical prototypes. We do not yet know what the upper limits of those landing speeds are, as that will require more foundational engineering and testing with future prototypes that combine both mobility features and landing resilience.
- We demonstrated in our physics based simulator that tensegrity probes can be controlled effectively with evolutionary and dynamical algorithms resulting in robust smooth rolling motion over a variety of terrains.
- We cross validated the NTRT simulator with other analytic tools and multiple hardware experiments, giving us confidence in results from our simulator. NTRT was also released as an open source project, and has attracted contributions and use by an international community of tensegrity robotics researchers.
- We built the SUPERball robot prototype which is focused on discovering foundational engineering principles for locomotion of tensegrity robots. This effort has provided rich insight into the needs of untethered locomotion, such as mechatronic design guidelines, sensor and actuator design principles, and will be an experimental platform for the migration of control theory developed in simulation.

With these many successes behind us, we are also aware of the many challenges ahead of us as we mature this new technology to TRL 3. The most immediate steps are to use the SUPERball prototype we have built for further research into the control of tensegrity robots. Our first steps will be to continue exploring sensor technologies that can provide critical information needed for state estimation and localization. Once we can reliable measure the full state of the robot (such as distances between the rods, velocities, cable tensions, orientation, etc), then we can start migrating the advanced control theories we developed in simulation onto the hardware prototype.

We will also need to build a next generation prototype to increase the performance and reliability of our robot. One design principle we were not aware of during the construction of this first prototype is the centrality of cable routing. While we built a compact and efficient end cap housing actuation and all supporting avionics, the cable routing became too complex and introduces significant friction. Thus, we would like to redesign the endcap with a priority given to the cable routing, and force the avionics to work around those pathways. During this redesign, we would like to address a number of other open issues, such as integrating two actuators into each end-cap for a fully controllable robot, and adding brakes such that the robot can hold poses while efficiently powering down the actuators.

While we have studied the structural response of tensegrity robots during extreme landing events, and we have validated those models by dropping passive test structures, and we have now made inroads to the engineering for mobility, we have yet to integrate these two separate aspects into a single prototype. Our next generation prototype will also be designed with high speed landing in mind and will include clutches that can be engaged to protect the actuators from landing shocks, and will improve on details of the construction of that endcap housing which we have identified as overly fragile.

Thus, this next prototype will enable us to start exploring the full end-to-end mission concept, from deployment to landing and exploration, in a single integrated hardware prototype. From there, future design iterations can start focusing on payload capacity, energy efficiency, and other optimizations that will progress the technology towards inclusion on a future planetary exploration mission.

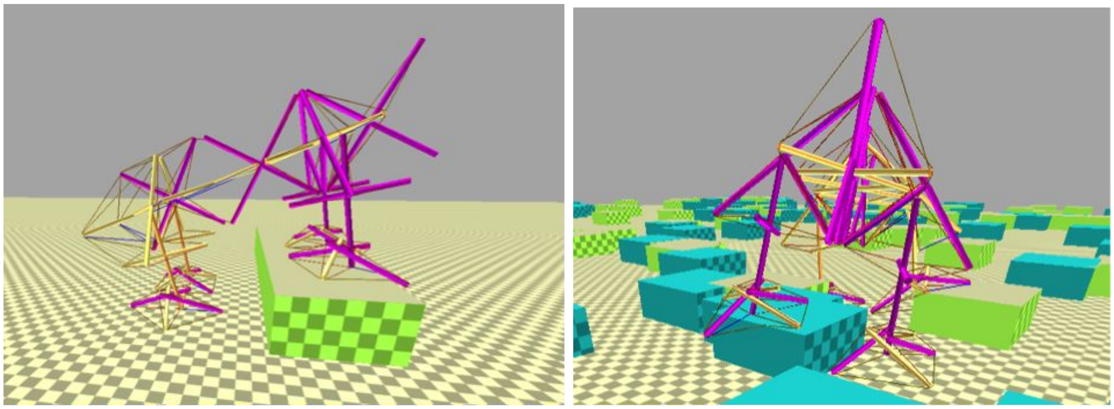


Figure 68: The Structurally Compliant Robot has natural passive dynamics that allow it to adapt to the terrain by automatically distributing the ground reaction loads across its feet, leading to improved stability as a property of the structure. This enables agile and reliable access to rough terrain by giving the robot the ability to push off any surface for climbing.

Along the way, we hope to continually inspire and lead a growing research community focused on tensegrity robotics. While much as been accomplished during this initial NIAC study, which has established the value and feasibility of tensegrity robots, these are only the first steps. Due to their unique structural design, tensegrity robots do not directly inherit many of the design and engineering fundamentals from traditional robots. Thus, we are in the midst of exploring and establishing the fundamental engineering principles for this new class of robot, which will enable the growing community of researchers to design and build tensegrity robots with increasing complexity and sophistication. SUPERball is only a single example of what is possible with this new class of robot. Many other applications are possible, including lightweight and robust legged robots that have the terrain access capability of a mountain goat – opening up whole new possibilities for accessing high priority science targets on the planets and moons of our solar system. (See Figure 68).

16 Academic Involvement

This NIAC project involved significant collaboration with students from many different universities. As a result, a significant accomplishment of the project has been the many papers and theses that were written in conjunction with research supported by this program. While this report attempts to cover many of the activities of the last few years, interested readers are encouraged to review our full publication history for further insights and results.

16.1 Associated Ph.D. Dissertations

1. Jonathan Bruce, “Design, Building, And Testing Of SUPERball: A Tensegrity Robot To Enable Space Exploration”, PhD Thesis, UC Santa Cruz, In Progress.

2. Brian Tietz Mirletz, “Adaptive Central Pattern Generators For Control Of Tensegrity Spines With Many Degrees Of Freedom”, PhD Thesis, Case Western Reserve University, Jan 2016.

3. Ken Caluwaerts, “Design and Computational Aspects of Compliant Tensegrity Robots”, PhD Thesis, Ghent University, Ghent, Belgium, 2014.

4. Atil Iscen, “Multiagent Learning for Locomotion and Coordination in Tensegrity Robotics”, PhD Thesis, Oregon State University, 2014.

16.2 Associated Master’s Theses

1. Alexander Lawrence Xydes, “Simulating DuCTT and optimizing control for DuCTT with machine learning,” Master of Science in Computer Science, University of California, San Diego, 2015.

2. Andrew P. Sabelhaus, “Mechanism and Sensor Design for SUPERball, a Cable-Driven Tensegrity Robot,” Master of Science in Engineering - Mechanical Engineering University of California, Berkeley, 2014.

3. George Korbel, “Central Pattern Generator Gait Evolution Of A Robotic Modular Tetrahedral Tensegrity”, Masters Thesis, University of Idaho, Moscow Idaho, 2013.

4. Jérémie Despraz, “Superballbot - Structures For Planetary Landing And Exploration”, Masters Thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2013.

5. Steve Burt, “Kinematics Algorithms For Tensegrity Structures”, Masters Thesis, UC Santa Cruz, Santa Cruz, CA. 2013.

16.3 Journal Papers

1. Brian T. Mirletz, Perry Bhandal, Ryan D. Adams, Adrian K. Agogino, Roger D. Quinn, Vytas SunSpiral, "Goal directed CPG based control for high DOF tensegrity spines traversing irregular terrain", Journal of Soft Robotics, In Revision.

2. Atil Iscen, Ken Caluwaerts, Jonathan Bruce, Adrian Agogino, Vytas SunSpiral, Kagan Tumer, "Learning Tensegrity Locomotion using Open-Loop Control Signals and Coevolutionary Algorithms," Artificial Life Journal, Vol. 21, No 2. pp 119-140, May 2015.

3. Ken Caluwaerts, Jrmie Despraz, Atil Isen, Andrew P. Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytas SunSpiral, "Design and Control of Compliant Tensegrity Robots through Simulation and Hardware Validation," Journal of the Royal Society Interface, Vol. 11, No 98. September 6, 2014.

16.4 Conference Papers

1. Ken Caluwaerts, Jeffrey M. Friesen, Jonathan Bruce, and Vytas SunSpiral, "State Estimation for Tensegrity Robots", In Review.

2. Paul Glick, Michael Fanton, Jonathan Bruce, Ken Caluwaerts, Jeffrey Michael Friesen, Vytas SunSpiral, "Increased Efficiency and Impact Tolerance for Cable Driven Robots via Multistage Braking System, Applied to NASAs SUPERball", In Review.

3. Steven Lessard, Jonathan Bruce, Erik Jung, Mircea Teodorescu, Vytas SunSpiral, "A lightweight, multi-axis compliant tensegrity joint", In Review.

4. Jeffrey Michael Friesen, Michael Fanton, Paul Glick, Pavlo Manovi, Alexander Xydes, Thomas Bewley, Vytas SunSpiral, "The Second Generation Prototype of A Duct Climbing Tensegrity Robot, DuCTTv2", In Review.

5. Brian T. Mirletz, In-Won Park, Roger D. Quinn, Vytas SunSpiral, "Towards bridging the reality gap between tensegrity simulation and robotic hardware," In Proceedings of of The 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Sept 2015, Hamburg, Germany.

6. Andrew P. Sabelhaus, Hao Ji, Patrick Hylton, Yakshu Madaan, ChanWoo Yang, Alice M. Agogino, Jeffrey Friesen, Vytas SunSpiral, "Mechanism Design and Simulation of the ULTRA SPINE, A Tensegrity Robot." In Proceedings of The ASME 2015 International Design Engineering Technical Conferences And Computers and Information in Engineering Conference (IDETC/CIE 2015). August 2015, Boston, Massachusetts.

7. Steven Lessard, Jonathan Bruce, Adrian Agogino, Vytas SunSpiral, Mircae Teodorescu, "Robust Monte Carlo Control Policies to Maneuver Tensegrity Robots out of Obstacles," In Proceedings of Autonomous Robots and Multirobot Systems (ARMS). May 2015, Istanbul, Turkey.

8. Andrew P. Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Pavlo Manovi, Roya Fallah Firoozi, Sarah Dobi, Alice M. Agogino, Vytas SunSpiral, "System Design and Locomotion of SUPERball, an Untethered Tensegrity Robot." In Proceedings of 2015 International Conference on Robotics and Automation (ICRA2015), May 2015, Seattle, Washington, USA.

9. Brian Mirletz, Roger D. Quinn and Vytas 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 And Alternatives, May 2015, Seattle, Washington, USA.

10. Jonathan Bruce, Ken Caluwaerts, Mircea Teodorescu, Vytas SunSpiral, "Design and Simulation of Compliant Tensegrity Robots for Planetary Exploration." In Proceedings of 2015 International Conference on Robotics and Automation (ICRA2015) Workshop on Soft Robotics: Actuation, Integration, and Applications Blending Research Perspectives., May 2015, Seattle, Washington, USA.

11. Lee-Huang Chen, Peadar Keegan, Michelle Yuen, Alice M. Agogino, Rebecca K. Kramer, Adrian K. Agogino,and Vytas SunSpiral, "Soft Robots Using Compliant Tensegrity Structures and Soft Sensors", In Proceedings of 2015 International Conference on Robotics and Automation (ICRA2015) Workshop on Soft Robotics: Actuation, Integration, and Applications Blending Research Perspectives., May 2015, Seattle, Washington, USA.

12. Kyunam Kim, Adrian K. Agogino, Deaho Moon, Laqshya Taneja, Aliakbar Toghyan, Borna Dehghani, Vytas SunSpiral, Alice 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), Dec 2014, Bali, Indonesia. Finalist Best Student Paper.

13. In-Won Park, Vytas SunSpiral, "Impedance Controlled Twisted String Actuators for Tensegrity Robots," In Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), KINTEX, Korea, October 2014.

14. Atil Iscen, Adrian Agogino, Vytas SunSpiral, Kagan Tumer, "Flop and Roll: Learning Robust Goal-Directed Locomotion for a Tensegrity Robot," In Proceedings of The 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, USA, September 2014.

15. Andrew P. Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Yangxin Chen, Dizhou Lu, Yuejia Liu, Adrian K. Agogino, Vytas SunSpiral, Alice 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, July 2014.

16. Brian T. Mirletz, In-Won Park, Thomas E. Flemons, Adrian K. Agogino, Roger D. Quinn, and Vytas 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), Barcelona, Spain, July 2014.

17. Jonathan Bruce, Andrew Sabelhaus, Yangxin Chen, Dizhou Lu, Kyle Morse, Sophie Milam, Ken Caluwaerts, Alice Agogino, Vytas 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, June 2014.

18. Jeffrey Michael Friesen, Alexandra Pogue, Thomas Bewley, Mauricio de Oliveira, Robert E. Skelton, Vytas SunSpiral, “A Compliant Tensegrity Robot for Exploring Duct Systems”, In Proceedings of International Conference on Robotics and Automation (ICRA), Hong Kong, June 2014

19. Jonathan Bruce, Ken Caluwaerts, Atil Iscen, Vytas SunSpiral, “Design and Evolution of a Modular Tensegrity Robot Platform”, In Proceedings of International Conference on Robotics and Automation (ICRA), Hong Kong, June 2014.

20. Brian Tietz, Ross Carnahan, Richard Bachmann, Roger Quinn, and Vytas 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 (AIM 2013), Wollongong, Australia, July 2013.

21. Vytas SunSpiral, George Gorospe, Jonathan Bruce, Atil Iscen, George Korbel, Sophie Milam, Adrian Agogino, David Atkinson, “Tensegrity Based Probes for Planetary Exploration: Entry, Descent and Landing (EDL) and Surface Mobility Analysis,” in Proceedings of 10th International Planetary Probe Workshop, San Jose, California, June 2013.

22. Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Robust Distributed Control of Rolling Tensegrity Robot” In proceedings of Autonomous Robots and Multi-robot Systems (ARMS) Workshop, Saint Paul, Minnesota, May, 2013.

23. Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Controlling Tensegrity Robots through Evolution” In proceedings of Genetic and Evolutionary Computation Conference, (GECCO 2013), Amsterdam, The Netherlands, July 6-10, 2013.

24. Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Learning to Control Complex Tensegrity Robots.” In proceedings of Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Saint Paul, Minnesota, May, 2013.

17 Acknowledgements

As part of our Phase 1 and 2 studies, we directed and collaborated with numerous students in our labs and at Universities around the Nation, and were honored to work with two International Visiting Researchers who were inspired to donate their time contributing fundamental research to this exciting project. The goals of our study were very broad and ambitious, covering many disciplines, and we greatly appreciate the work done by these teams in helping bring this study together. This projects success has also been strongly supported by Terry Fong and all the members of the Intelligent Robotics Group at NASA Ames Research Center. In addition this project has been supported by Guillaume Brat and the Robust Software Engineering Lab at NASA Ames Research Center. Thank You!

References

[1] D. Adams. Mars exploration rover airbag landing loads testing and analysis. In *AIAA Structures, Structural Dynamics & Materials Conference*, 2004.

[2] A. Agogino and K. Tumer. Distributed evaluation functions for fault tolerant multi rover systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Seattle, WA, July 2006.

[3] Mostafa Ajallooeian, Sébastien Gay, Alexandre Tuleu, Alexander Sproewitz, and Auke Ispeert. Modular control of limit cycle locomotion over unperceived rough terrain. In *IROS*, pages 3390–3397, 2013.

[4] J. B. Aldrich and R. E. Skelton. Backlash-free motion control of robotic manipulators driven by tensegrity motor networks. *IEEE Conference on Decision And Control*, pages 2300–2306, 2006.

[5] N. Bel Hadj Ali and et al. Design optimization and dynamic analysis of a tensegrity-based footbridge. *Engineering Structures*, 32:3650–3659, 2010.

[6] T. Bliss. Central pattern generator control of a tensegrity swimmer. *Ph.D. thesis, Dept. Mech. Aerosp. Eng., Univ. Virginia,*, December 2011.

[7] T. Bliss, J. Werly, T. Iwasaki, and H. Bart-Smith. Experimental validation of robust resonance entrainment for CPG-controlled tensegrity structures. *IEEE Transactions On Control Systems Technology*, 2012.

[8] T. Bliss, J. Werly, T. Iwasaki, and H. Bart-Smith. Experimental validation of robust resonance entrainment for CPG-controlled tensegrity structures. *IEEE Transactions On Control Systems Technology*, 21:666–678, 2012.

[9] T. K. Bliss, T. Iwasaki, and H. Bart-Smith. CPG Control of a Tensegrity Morphing Structure for Biomimetic Applications. *Advances in Science and Technology*, 58:137–142, 2008.

[10] V. Böhm, A. Jentzsch, T. Kaufhold, F. Schneider, and K. Zimmermann. An approach to compliant locomotion systems based on tensegrity structures. *56th International Scientific Colloquium, Ilmenau University of Technology*, pages 1–6, August 2011.

[11] A. S. Boxerbaum, A. D. Horchler, K. M. Shaw, H. J. Chiel, and R. D. Quinn. A controller for continuous wave peristaltic locomotion. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 197–202, September 2011.

[12] R.D. Braun and R.M. Manning. Mars exploration entry, descent and landing challenges. In *Journal of Spacecraft and Rockets*. 2007.

[13] Claude Brezinski. *Computational Aspects of Linear Control (Numerical Methods and Algorithms)*. Springer, 2002.

[14] Jonathan Bruce, Ken Caluwaerts, Atil Iscen, Andrew P. Sabelhaus, and Vytas SunSpiral. Design and evolution of a modular tensegrity robot platform. In *ICRA*, 2014. To Appear.

[15] Jonathan Bruce, Ken Caluwaerts, Atil Iscen, and Vytas SunSpiral. Design and evolution of a modular tensegrity robot platform. In *IEEE International Conference on Robotics and Automation*, 2014.

[16] Jonathan Bruce, Andrew P. Sabelhaus, Yangxin Chen, Dizhou Lu, Kyle Morse, Sophie Milam, Ken Caluwaerts, Alice M. Agogino, and Vytas SunSpiral. SUPERball: Exploring tensegrities for planetary probes. In *12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2014.

[17] A. Büschges, T. Akay, J. P. Gabriel, and J. Schmidt. Organizing network action for locomotion: insights from studying insect walking. *Brain research reviews*, 57(1):162–71, January 2008.

[18] Ken Caluwaerts, Jérémie Despraz, Atıl Işçen, Andrew P. Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytas SunSpiral. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface*, 11(98), 2014.

[19] Ken Caluwaerts, Michiel D'Haene, David Verstraeten, and Benjamin Schrauwen. Locomotion without a brain: physical reservoir computing in tensegrity structures. *Artificial Life*, 19(1):35–66, 2013.

[20] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia, July 2003.

[21] Fascia Research Congress. http://www.fasciacongress.org/2012/about-the-conference/.

[22] Erwin Coumans. Bullet physics engine, 2005.

[23] Calladine C.R. Buckminster Fuller's Tensegrity structures and Clerk Maxwell's rules for the construction of stiff frames. *International Journal of Solids and Structures*, 14(2):161–172, 1978.

[24] MR Darrach, R Kidd, JA MacAskill, A Chutjian, J Simcic, E Neidholdt, M Sinha, and B Bae. An overview of mass spectroscopy based instrument development at the jet propulsion laboratory. mr. *LPI Contributions*, 1683:1114, 2012.

[25] Sarah Degallier and Auke Ijspeert. Modeling discrete and rhythmic movements through motor primitives: a review. *Biol. Cybern.*, 103(4):319–338, October 2010.

[26] T. G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence*, 13:227–303, 2000.

[27] Tom Flemons. The bones of tensegrity.
http://www.intensiondesigns.com/bones_of_tensegrity.html, 2012.

[28] Buckminster Fuller. Tensegrity. *Portfolio and Art News Annual*, 4:112–127, 1961.

[29] Sébastien Gay, Sarah Dégallier, Ugo Pattacini, Auke Ijspeert, and José Santos Victor. Integration of vision and central pattern generator based locomotion for path planning of a non-holonomic crawling humanoid robot. In *IROS*, pages 183–189, 2010.

[30] Sébastien Gay, José Santos-Victor, and Auke Ijspeert. Learning robot gait stability using neural networks as sensory feedback function for central pattern generators. In *IROS*, pages 192–201, 2013.

[31] J Gomez-Elvira and REMS Team. Environmental monitoring station for mars science laboratory. *LPI Contributions*, 1447:9052, 2008.

[32] A. Graells Rovira and J. M. Mirats-Tur. Control and simulation of a tensegrity-based mobile robot. *Robotics and Autonomous Systems*, 57(5):526–535, May 2009.

[33] C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, 2002.

[34] KE Herkenhoff, SW Squyres, JF Bell III, JN Maki, HM Arneson, P Bertelsen, DI Brown, SA Collins, A Dingizian, ST Elliott, et al. Athena microscopic imager investigation. *Journal of Geophysical Research*, 108(E12):8065, 2003.

[35] C. HolmesParker and A. Agogino. Agent-based resource allocation in dynamically formed cubesat constellations. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Taipei, Taiwan, May 2011.

[36] A Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21:642–653, May 2008.

[37] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks : the official journal of the International Neural Network Society*, 21(4):642–53, May 2008.

[38] D. E. Ingber. The architecture of life. *Scientific American*, 278(1):48–57, 1998.

[39] Donald E. Ingber. Cellular tensegrity: defining new rules of biologic design that govern cytoskeleton. *Journal of Cell Science*, 104:613–627, 1993.

[40] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer. Learning to control complex tensegrity robots. In *To appear in: Twelfth International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[41] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. Controlling tensegrity robots through evolution. In *GECCO*, 2013.

[42] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. Controlling tensegrity robots through evolution. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 1293–1300. ACM, 2013.

[43] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. Robust distributed control of rolling tensegrity robot. In *The Autonomous Robots and Multirobot Systems (ARMS) Workshop at AAMAS*, volume 2013, 2013.

[44] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[45] Kyunam Kim, Adrian K. Agogino, Deaho Moon, Laqshya Taneja, Aliakbar Toghyan, Borna Dehghani, Vytas SunSpiral, and Alice M. Agogino. Rapid prototyping design and control of tensegrity soft robot for locomotion. In *To Appear in ROBIO*, 2014.

[46] H. Klimke and S. Stephan. The making of a tensegrity tower. In *IASS Symposium*, Montpellier, 2004.

[47] Y. Koizumi, M. Shibata, and S. Hirai. Rolling Tensegrity Driven by Pneumatic Soft Actuators. *Robotics*, pages 1988–1993, 2012.

[48] Y. Koizumi, M. Shibata, and S. Hirai. Rolling tensegrity driven by pneumatic soft actuators. In *ICRA*, pages 1988–1993, 2012.

[49] S. M. Levin. The tensegrity-truss as a model for spine mechanics: biotensegrity. *Journal of Mechanics in Medicine and Biology*, 2:375–388, 2002.

[50] Stephen Levin. The tensegrity-truss as a model for spine mechanics. *Journal of Mechanics in Medicine and Biology*, 2:375–388, 2002.

[51] J. N. Maki, D. Thiessen, A.Pourangi, P. Kobzeff, T. Elliott L. Scherr and, A. Dingizian, and Beverly St. Ange. The mars science laboratory (msl) navigation cameras (NAVCAMS). In *42nd Lunar and Planetary Science Conference*, pages 268–278, 2011.

[52] JN Maki, D Thiessen, A Pourangi, P Kobzeff, L Scherr, T Elliott, A Dingizian, and B St Ange. The mars science laboratory (msl) navigation cameras (navcams). In *Lunar and Planetary Institute Science Conference Abstracts*, volume 42, page 2738, 2011.

[53] M. Masic and et al. Algebraic tensegrity form-finding. *International Journal of Solids and Structures*, 42:4833–4858, 2005.

[54] M. Masic and R. E. Skelton. Open-loop control of class-2 tensegrity towers. *Proceedings of the 11th Smart Structures and Materials Conference*, 5383:298–308, 2004.

[55] K Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56(5-6):345–353, 1987.

[56] J. M. Mirats-Tur. On the movement of tensegrity structures. In *International Journal of Space Structures*, volume 25, 2010.

[57] K. W. Moored, III, S. A. Taylor, and H. Bart-Smith. Optimization of a Tensegrity Wing for Biomimetic Applications. *Proceedings of SPIE*, 6173:617313, March 2011.

[58] Thomas W. Myers. Dissection of the anatomy trains: http://www.anatomytrains.com/explore/dissection.

[59] Thomas W. Myers. *Anatomy Trains: Myofascial Meridians for Manual and Movement Therapists, 2nd Edition*. Churchill Livinstone, Edinburgh, 2009.

[60] Kohei Nakajima, Helmut Hauser, Rongjie Kang, Emanuele Guglielmino, Darwin G Caldwell, and Rolf Pfeifer. Computing with a muscular-hydrostat system. In *ICRA*, pages 1504–1511. IEEE, 2013.

[61] Kohei Nakajima, Helmut Hauser, Rongjie Kang, Emanuele Guglielmino, Darwin G Caldwell, and Rolf Pfeifer. A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience*, 7(91), 2013.

[62] E. N. Nilsen. *Exploring mars: an overview*. 2012.

[63] O. Orki. *A Model Of Caterpillar Locomotion Based On Assur Tensegrity Structures*. PhD thesis, Tel Aviv University, 2012.

[64] O. Orki, A. Ayali, O. Shai, and U. Ben-Hanan. Modeling of caterpillar crawl using novel tensegrity structures. *Bioinspiration & Biomimetics*, 7(4):046006, 2012.

[65] O. Orki, O. Shai, A. Ayali, and U. Ben-Hanan. A Model of Caterpillar Locomotion Based on Assur Tensegrity Structures. *Proceedings of the ASME 2011 IDETC/CIE*, August 2011.

[66] C. Paul, H. Lipson, and F. J. V. Cuevas. Evolutionary form-finding of tensegrity structures. *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO)*, pages 3–10, 2005.

[67] C. Paul, J. W. Roberts, H. Lipson, and F. J. V. Cuevas. Gait production in a tensegrity based robot. In *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, January 2005.

[68] C. Paul, F. J. Valero-Cuevas, and H. Lipson. Design and control of tensegrity robots for locomotion. *IEEE Transactions on Robotics*, 22(5), October 2006.

[69] K. Pearson. Role of sensory feedback in the control of stance duration in walking cats. *Brain research reviews*, 57(1):222–7, January 2008.

[70] Schleip R, T. W. Findley, L. Chaitow, and P. Huijing, editors. *Fascia: The Tensional Network of the Human Body: The science and clinical applications in manual and movement therapy, 1e*. Churchill Livingstone, 1 edition, April 2012.

[71] A. L. Ridgel and R. E. Ritzmann. Effects of neck and circumoesophageal connective lesions on posture and locomotion in the cockroach. *Journal of comparative physiology. A, Neuroethology, sensory, neural, and behavioral physiology*, 191(6):559–73, June 2005.

[72] J. Rieffel, R. Stuk, F. Valero Cuevas, and H. Lipson. Locomotion of a Tensegrity Robot via Dynamically Coupled Modules. *Proceedings of the International Conference on Morphological Computation*, 2007.

[73] J. Rieffel, B. Trimmer, and H. Lipson. Mechanism as Mind : What Tensegrities and Caterpillars Can Teach Us about Soft Robotics The Manduca Sexta Caterpillar : Morphological Communication in Tensegrity Robots. *Artificial Life*, pages 506–512, 2008.

[74] J. A. Rieffel, F. J. Valero-Cuevas, and H. Lipson. Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of the Royal Society, Interface / the Royal Society*, 7(45):613–21, April 2010.

[75] Ludovic Righetti, Jonas Buchli, and Auke Jan Ijspeert. From dynamic hebbian learning for oscillators to adaptive central pattern generators. In *AMAM*, page 45, 2005.

[76] Andrew P Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Yangxin Chen, Dizhou Lu, Yuejia Liu, Adrian K Agogino, Vytas SunSpiral, and Alice M Agogino. Hardware design and testing of SUPERball, a modular tensegrity robot. In *The 6th World Conference of the International Association for Structural Control and Monitoring (6WCSCM)*, 2014.

[77] M. Shibata and S. Hirai. Rolling Locomotion of Deformable Tensegrity Structure. *Mobile Robotics: Solutions and Challenges*, pages 479–486, 2009.

[78] M. Shibata, F. Saijyo, and S. Hirai. Crawling by body deformation of tensegrity structure robots. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 4375 –4380, may 2009.

[79] R. E. Skelton and M. C. De Oliveira. *Tensegrity Systems*. Springer, 2009 edition, June 2009.

[80] Robert E. Skelton and Mauricio C. Oliveria. *Tensegrity Systems*. Springer, New York, 2009.

[81] Kenneth Snelson. http://www.kennethsnelson.net/.

[82] Kenneth Snelson. Continuous tension, discontinuous compression structures. united states patent 3169611, Feburary 1965.

[83] P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 2005.

[84] Tibert and et al. Review of form-finding methods for tensegrity structures. *International Journal of Space Structures*, 18:209–223, 2003.

[85] Brian R Tietz, Ross W Carnahan, Richard J Bachmann, Roger D Quinn, and Vytas Sunspiral. Tetraspine : Robust Terrain Handling on a Tensegrity Robot Using Central Pattern Generators. In *2013 IEEE/ASME Advanced Intelligent Mechatronics*. IEEE, July 2013.

[86] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 330–337, Honolulu,HI, May 2007. **Best Paper Award**.

[87] K. Tumer and D. Wolpert, editors. *Collectives and the Design of Complex Systems*. Springer, New York, 2004.

[88] J. M. Mirats Tur and S. H. Juan. Tensegrity frameworks: Dynamic analysis review and open problems. *Mechanism and Machine Theory*, 44:1–18, 2009.

[89] Katja Verbeeck, Maarten Peeters, Ann Nowe, and Karl Tuyls. Reinforcement learning in stochastic single and multi-stage games. In *Adaptive Agents and Multi-Agent Systems II, Lecture Notes in Artificial Intelligence*, pages 275–294. Springer Verlag, Berlin, 2005.

[90] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20:391–403, 2007.

[91] P. J. Whelan. Control of locomotion in the decerebrate cat. *Progress in Neurobiology*, 49(5):481 – 515, 1996.

[92] J. Y. Zhang and M. Ohsaki. Adaptive force density method for form-finding problem of tensegrity structures. *International Journal of Solids and Structures*, 43:5658–5673, 2006.

[93] L. Zhang and et al. Form-finding of nonregular tensegrity systems. *Journal of Structural Engineering*, 132:1435–1440, 2006.

[94] W. Zimmerman, J. Lunine, and R. Lorenz. A post-huygens titan surface science mission design. In *Aerospace Conference, 2005 IEEE*, pages 268–278, 2005.