Design and Control of Compliant Tensegrity Robots Through Simulation and Hardware Validation

KEN CALUWAERTS[1,2], JÉRÉMIE DESPRAZ[1,3], ATIL IŞÇEN[1,4], ANDREW P. SABELHAUS[1,5], JONATHAN BRUCE[1,6], BENJAMIN SCHRAUWEN[2], AND VYTAS SUNSPIRAL[1,7]

[1]Dynamic Tensegrity Robotics Lab, NASA Ames Research Center, USA
[2]Reservoir Lab, Electronics and Information Systems department, Ghent University, Belgium
ken.caluwaerts@ugent.be, benjamin.schrauwen@ugent.be
[3]Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
jeremie.despraz@epfl.ch
[4]Oregon State University, USA
iscena@onid.oregonstate.edu
[5]Berkeley Institute of Design, University of California Berkeley, USA
apsabelhaus@berkeley.edu
[6]USRA/University of California Santa Cruz, USA
jbruce@soe.ucsc.edu
[7]SGT Inc., NASA Ames Intelligent Robotics Group, USA
vytas.sunspiral@nasa.gov

Abstract

To better understand the role of tensegrity structures in biological systems and their application to robotics, the Dynamic Tensegrity Robotics Lab at NASA Ames Research Center has developed and validated two software environments for the analysis, simulation, and design of tensegrity robots. These tools, along with new control methodologies and the modular hardware components developed to validate them, are presented as a system for the design of actuated tensegrity structures. As evidenced from their appearance in many biological systems, tensegrity ("tensile-integrity") structures have unique physical properties that make them ideal for interaction with uncertain environments. Yet, these characteristics make design and control of bio-inspired tensegrity robots extremely challenging. This work presents the progress our tools have made in tackling the design and control challenges of spherical tensegrity structures. We focus on this shape since it lends itself to rolling locomotion.

The results of our analyses include multiple novel control approaches for mobility and terrain interaction of spherical tensegrity structures that have been tested in simulation. A hardware prototype of a spherical six-bar tensegrity, the Reservoir Compliant Tensegrity Robot (ReCTeR), is used to empirically validate the accuracy of simulation.

Keywords: tensegrity, bio-inspired locomotion, central pattern generators, reservoir computing, compliant robotics, soft robotics, planetary exploration

This is a pre-copyedited, author-produced PDF of an article accepted for publication in Journal of the Royal Society Interface. The definitive publisher-authenticated version is available online at: `http://dx.doi.org/10.1098/rsif.2014.0520` or `http://rsif.royalsocietypublishing.org/content/11/98/20140520`.

Please cite this work as:
Caluwaerts K, Despraz J, Iscen A, Sabelhaus AP, Bruce J, Schrauwen B, SunSpiral V.
Design and control of compliant tensegrity robots through simulation and hardware validation.
J. R. Soc. Interface September 6, 2014; doi:10.1098/rsif.2014.0520 1742-5662

I. Introduction

Prior work has investigated the unique structural properties of tensegrity systems, their role in biology, and control strategies for different tensegrity morphologies. One of the centers for this research is NASA Ames Research Center, where there is interest in these systems for planetary exploration missions.

I.1 Tensegrity Structures

Tensegrity structures are composed of compression elements encompassed within a network of tensional elements; consequently, each element experiences either pure compression or pure tension. This allows individual elements to be extremely lightweight, as designs do not need to resist bending or shear forces. Active motion in tensegrity structures can be performed with minimal energy expenditure since actuators work linearly along load paths in tension elements, avoiding torques caused by long lever arms of traditional robotic designs.

A unique property of tensegrity structures is how they internally distribute forces. Since there are no lever arms, forces do not magnify around joints or other common points of failure. Rather, externally-applied forces distribute through the structure via multiple load paths, creating system-level mechanical robustness and tolerance to forces applied from any direction or failure of individual actuation elements [1]. Thus, tensegrity structures are ideally suited for operation in dynamic environments where contact forces cannot always be predicted.

I.2 Tensegrity and Biology

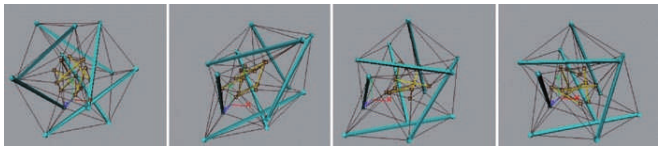

Figure 1: Computer simulations of a nucleated tensegrity cell model exhibits mechanical coupling between the cell, the cytoskeleton, and the nucleus. Adapted by permission from Macmillan Publishers Ltd: Nature Reviews Molecular Cell Biology [2], 2009

Tensegrity structures are being discovered in many aspects of biological systems, which motivates this work's bio-inspired modeling and controls approaches. The tensegrity concept appears at various scales, from the cytoskeleton of individual cells (see Figure 1) [2, 3, 4] to mammalian physiology [5]. Emerging biomechanical theories are shifting focus from bone-centric models to fascia-centric models. Fascia is the connective tissue in our bodies (including muscles, ligaments, tendons, etc.) that forms a continuous web of tension, even surrounding and supporting bones, which, unlike traditional mechanical systems, have no rigid connections between them [6]. This new view is challenging the "common sense" view of skeletal structures as the primary load-bearing elements of human and mammalian bodies (see Figure 2). In the emerging "bio-tensegrity" 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 that are the primary load-bearers [5, 6].

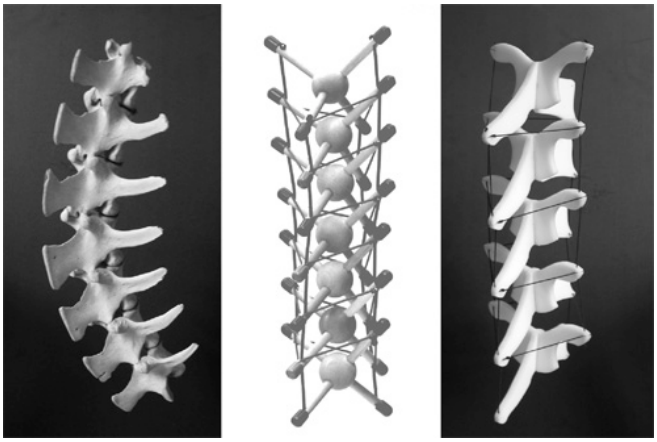

Figure 2: Tensegrity models of the spine show how vertebrae float without touching. Image courtesy of Tom Flemons (copyright 2006) [7]

I.3 Tensegrity Robotics for Space Exploration

NASA is supporting research into tensegrity robotics to create planetary rovers with many of the same qualities that benefit biological systems. The high strength-to-weight ratio of tensegrity structures is attractive due to the impact of mass on mission launch costs. Likewise, large tensegrity structures are deployable from compact configurations, enabling them to fit into space-constrained spacecraft. While these qualities have inspired studies of deployable antennae and other large space structures [8], the unique force distribution of tensegrity robots has only recently been investigated for planetary exploration [9]. Initial work in the NASA Innovative Advanced Concepts (NIAC) project [10] shows that controllable compliance and force distribution properties allow for reliable and robust environmental interactions during landing and planetary surface exploration.

A key goal of this NASA work is to develop a tensegrity probe with an actively controlled tensile network, enabling compact stowage for launch followed by deployment for landing. Compliant tensegrity probes can safely absorb significant impact forces, enabling high-speed Entry, Descent, and Landing (EDL) scenarios where the probe acts like an airbag [9]. However, unlike an airbag that must be discarded after a single use, the tensegrity also provides rolling mobility (Figure 3). This enables compact and lightweight planetary exploration

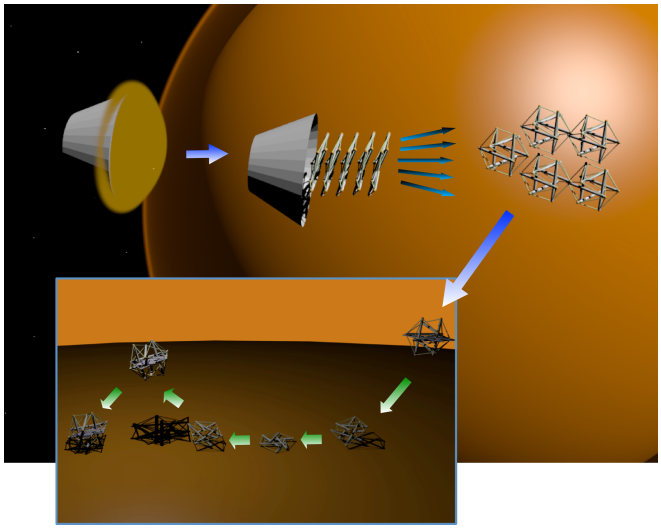

Figure 3: **Mission Scenario** - *A tightly packed set of tensegrities expands, spreads out, falls to surface of moon, and then safely bounces on impact. The same tensegrity structure cushioning landing is then used for exploration.*

missions with the capabilities of traditional wheeled rovers, but with a mass and cost similar to a stationary probe. Dual use of structure allows a tensegrity mission to have a high mass fraction between science payload and overall weight (as measured at atmospheric entry). This reduces mission cost and enables new forms of surface exploration utilizing the tensegrity's natural tolerance to impacts [9].

I.4 Tensegrity Control

Tensegrity structures are a fairly modern concept having been initially explored in the 1960s by Buckminster Fuller [11] and the artist Kenneth Snelson [12]. Initial tensegrity research was mostly concerned with form-finding techniques [13] and the design and analysis of static structures [14, 15]. Research into control of tensegrity structures began in the mid-1990s, with initial efforts at formalizing the dynamics of tensegrity structures only recently emerging [15]. The very properties that make tensegrities ideal for physical interaction with the environment (compliance, multi-path load distribution, nonlinear dynamics, etc.) also present significant challenges to traditional control approaches. A recent review shows that there are still problems actively controlling tensegrities [16]. Work has continued in the analytical understanding of the equations of motion and dynamics of tensegrity structures [17]. However, environmental interactions cause additional modeling difficulties, typically limiting the effectiveness of such approaches. Consequently, successful demonstrations of tensegrity mobility have primarily used non-analytical methods [1, 18, 19, 20]. Our approach to these problems is therefore to develop control algorithms based on Central Pattern Generators (CPGs), distributed learning, and genetic algorithms instead of more traditional control approaches.

I.4.1 Central Pattern Generators

CPGs are neural circuits found in both vertebrate and invertebrate animals that 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 [21]. The term *central* indicates that rhythms of the CPG are not driven by sensory input, but are self-generated. CPGs are the fundamental building blocks for locomotor neural circuits in many animals, and are also key to other fundamental rhythmic activities such as chewing, breathing, and digesting. Recent research shows a close relationship between CPGs and motion primitives in the spine that enable both rhythmic and discrete motions [22]. Alongside their biological inspiration for use in robotic motion, CPGs present several interesting and useful properties including distributed control, robustness to perturbations, inherent tolerance to redundancies, fast control loops, and the ability to modulate locomotion by simple control signals.

CPGs are therefore well suited for controlling tensegrities [23] and other biomimetic structures [24]. Additionally, there is intuition for pairing tensegrity robots with CPG networks: the dynamics of physical forces propagating through a tensegrity structure are similar to the dynamics of control patterns propagating through CPG networks.

I.4.2 Evolutionary Algorithms

Instead of defining a control policy directly, evolutionary algorithms can be used to learn a control policy. This is accomplished 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 dynamics need not be known.
3. Control policies can be learned or parameters of existing control policies can be optimized.
4. Distributed learning can be used to scale to larger tensegrities and to accelerate learning.

Moreover, coevolutionary algorithms provide distributed learning for multiagent problems [25]. Each component can individually learn a control policy that decides how to actuate its individual end cap in such a way that global performance is maximized [26]. Challenges controlling tensegrity robots

using traditional approaches have led some researchers to consider biologically-inspired evolutionary or coevolutionary algorithms [1, 27, 28, 29].

I.5 Outline

This paper is organized as follows: section II introduces our simulators and hardware platform; section III presents our simulator validation results; section IV describes our various control strategies; and section V discusses our future work and our results and control strategies in context of other work. And last, section VI presents our conclusions.

II. 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: http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/.

II.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:

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

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.

II.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 [15]. In Skelton's work, the tensegrity structure struts 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 makes it unsuited for the study of complex environmental interactions.

II.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) [30].

Since game physics require 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. A current limitation of this method is that the cables do not exist in the simulation world as physical bodies. Thus, their collisions and interaction with rigid bodies are not simulated. For locomotion tasks and terrain types we have explored so far, this is not a problem, but will be addressed in the future to simulate more extreme terrain interactions.

II.4 Prototype

In addition to these two software environments, a physical prototype of the six-bar tensegrity was constructed. ReCTeR is a highly compliant, lightweight (1.1kg), underactuated tensegrity icosahedron robot, as shown in Figure 4. The robot's 24

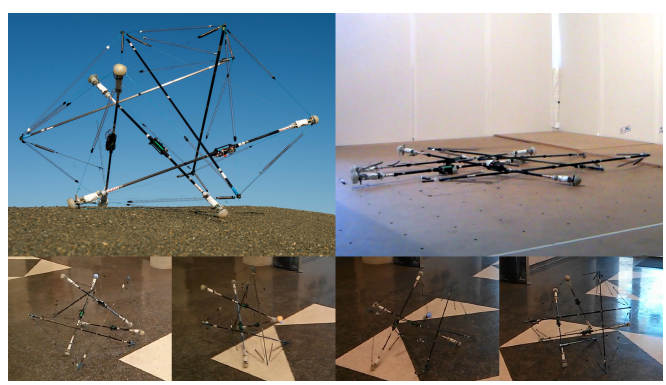

Figure 4: ReCTeR: an untethered, highly compliant, spherical tensegrity robot. Top left: deployed robot. Center right: active folding. Bottom: ReCTeR rolling from right to left.

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 (also see section

II.5). Their rest length is adjusted by a rotational DC motor (4.5W, 4.4:1 gear ratio) that spins a snag-resistant bobbin (5.5mm diameter). The other end of the cable attaches to a stiffer spring (81N/m) affixed to a passive end cap. We use low stiffness springs to allow active folding (Figure 4) 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 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 are achieved by 24 tension sensors using strain gages, six ground reaction force sensors, and three six-DOF inertial measurement units distributed evenly among the actuated end caps. To allow dynamic motion and rolling, each self-contained strut holds 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.

II.5 Robot Models

Our control methods are implemented on platforms with varying configurations. Figure 5 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 [31].

The basic tensegrity icosahedron is shown on the left. This structure has 24 spring-cable assemblies and six 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 also refer to these as *inner* members.

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

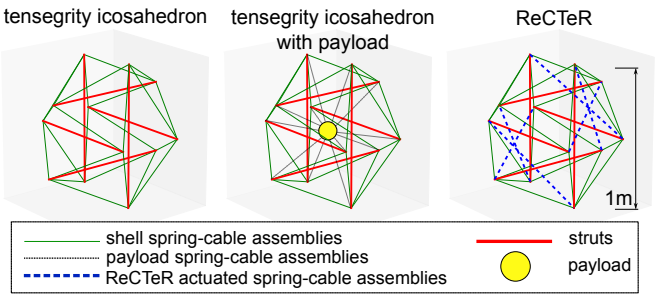

Figure 5: The various tensegrity configurations used in this paper. Left: tensegrity icosahedron with only outer shell members. Center: tensegrity icosahedron with a payload by inner elements. Right: ReCTeR configuration with passive outer shell and actuated spring-cable assemblies.

III. Validation of Simulations

III.1 Experimental Setup

To track the full state of the robot, an active marker motion capture setup was used. Passive struts were fitted with two markers, active struts received three markers.

III.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 top spring-cable assemblies were actuated, as shown in Figure 6. We tracked vertical displacement of an end cap not directly actuated by one these two members. The incident strut was suspended in air by a total of ten springs.

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 ten equally spaced lengths, resulting in 100 measurement positions. 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 observed displacements.

The average observed difference between 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 ReCTeR's diameter).

III.3 Dynamics

Next, we compared dynamics of the NTRT simulator to ReCTeR hardware. This experiment was designed for two purposes: to first verify that the simulator can replicate ground

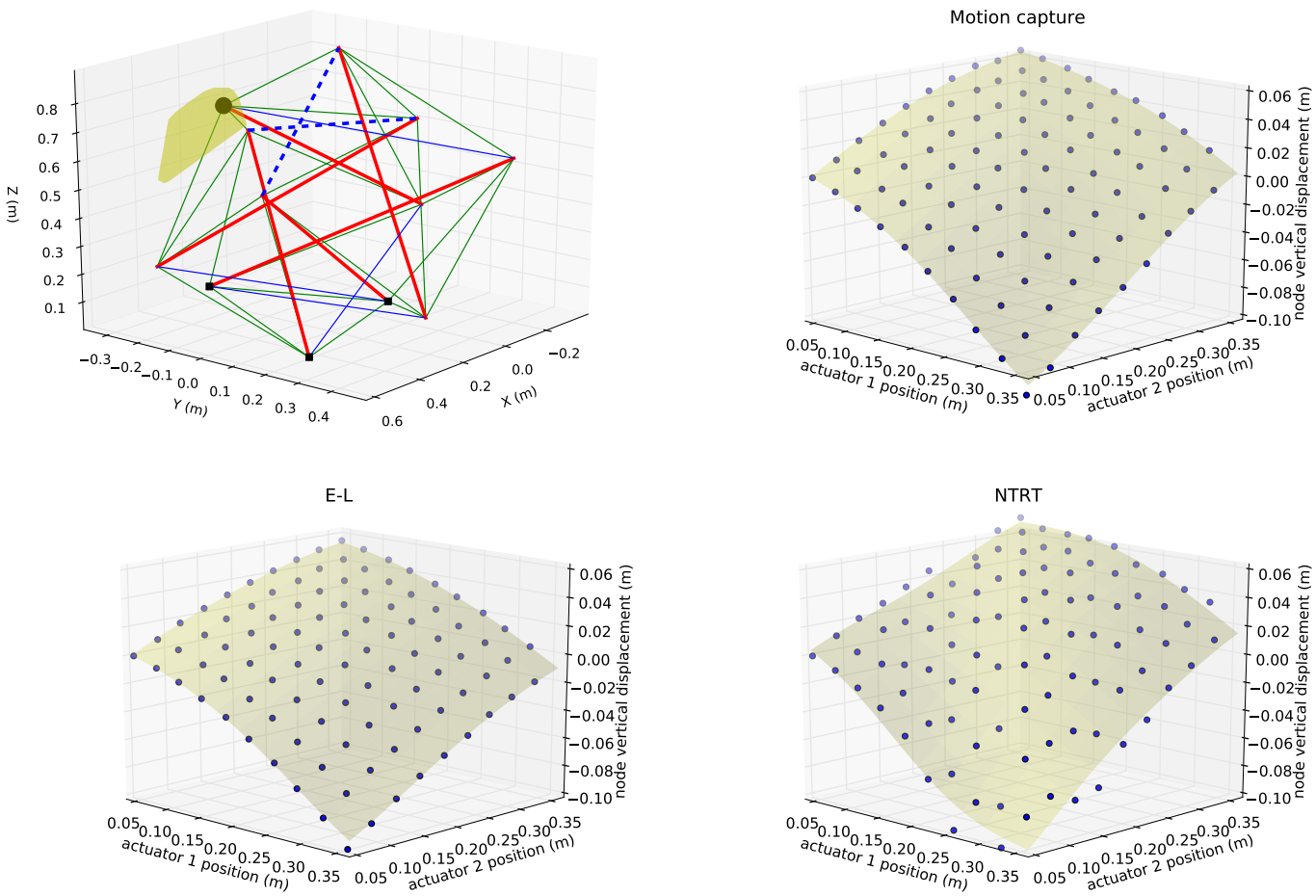

Figure 6: Kinematic comparison of Euler-Lagrange and NTRT simulators and ReCTeR motion capture data. The top left plot shows the experimental setup. The rest length of two actuated spring-cable assemblies (dashed lines) is modified. The full range of tracked end-cap motion during the experiment is shown in light yellow (convex hull). The end caps indicated by small black squares are on the ground. The three other plots show 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 where we trace the displacement is not directly actuated and is floating. The nodal displacement as a function of the actuator position is nonlinear, even for modest displacements. Note that the left-most point (0.05,0.05,0) is the reference point; displacements are relative to this initial state.

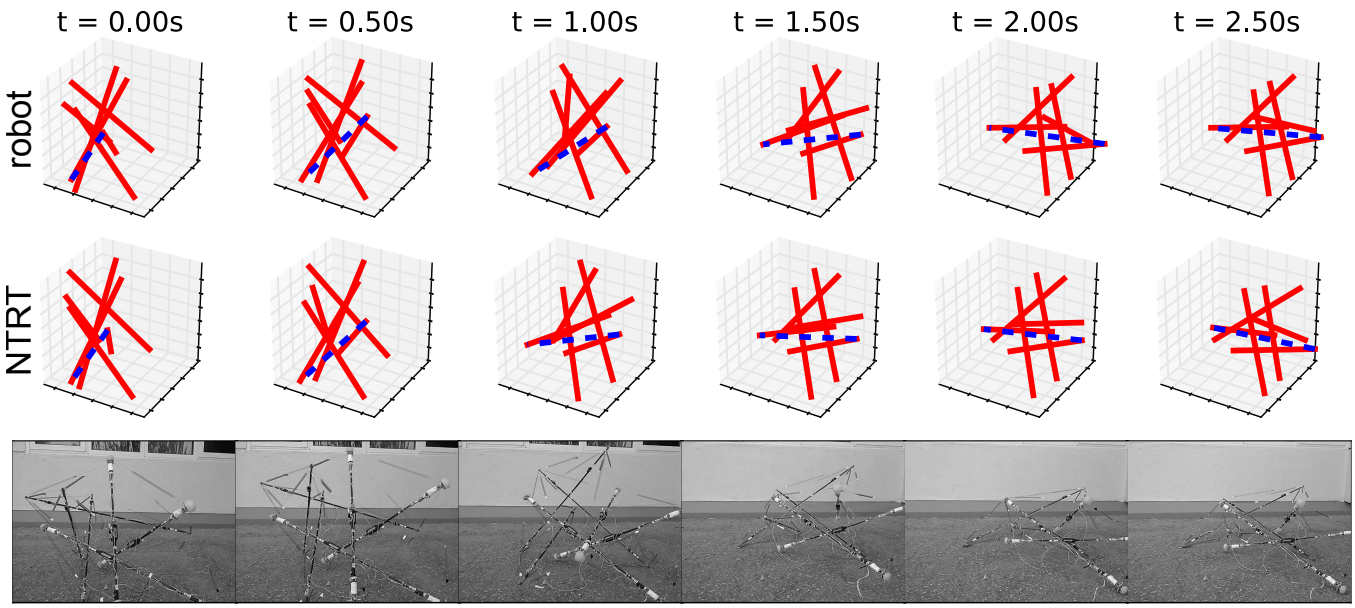

Figure 7: Comparing the dynamics of the robot and NTRT. The tensioned spring-cable assembly indicated by the dashed line is released (0.32m to 0.535m at 0.6m/s), causing the robot to topple. Two other actuated members are also tensioned, while the other three actuated springs are at their initial lengths, resulting in two slack springs. We observed a time-averaged error of the end caps' vertical positions of less than 5% of ReCTeR's diameter for all end caps.

interactions; and second, that it can simulate conversion of potential energy into kinematic energy when a spring is released. The experimental setup is shown in Figure 7. The robot initially has a non-minimal ground contact (four end caps on the ground instead of just three), and three actuated springs are tensioned. Next, one of the tensioned, actuated springs is loosened by its actuator, causing the robot to roll over.

Since the experiment also depends on ReCTeR's initial state, 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. Recorded motor positions from the physical test were then applied into the simulator, causing a similar roll-over motion.

IV. Locomotion Control

Once it had been determined that the NTRT simulator modeled these robotic dynamics reasonably well, locomotion experiments were performed with tensegrity robots in both simulation and hardware. The algorithms described in section IV.1 apply to various configurations, but are presented here in simulation for the first configuration, as shown in Figure 5 with 1.5 meter rods, weighing 15kg. The controls in section IV.2 were applied to simulations of the second configuration. Section IV.3 presents hardware results on ReCTeR and a comparison of those results to previously-simulated implementations. Appendix A provides a summary of the control methods in this work and an overview of related work.

IV.1 Coevolutionary Control

The first control method from our group is based on coevolutionary algorithms [25]. We demonstrated successful rolling locomotion of a tensegrity icosahedron with this technique in the NTRT [29]. In this scheme, each spring-cable assembly is active and has a controller that evolves independently from the other controllers (i.e., in independent pools), but cooperation is used to optimize behavior of the entire robot. The objective function for this maximization was set to be ReCTeR's distance traveled during a fixed amount of time. The simplest implementation of this technique is an optimization of open-loop control signals that are only a function of time; sinusoidal functions performed well.

After this method was explored, the effects of different complexities and frequencies of these open-loop signals were analyzed. More precisely, we optimized step-wise functions with varying numbers of via points. This enables the study of computational load and scaling properties needed to estimate power consumption of various controllers, as well as to investigate effects of actuator failure. Figure 8 shows the learning curve of this process. In this case, optimized rest length signals had four via points. An analysis of practical aspects of these results (power consumption, actuator failure, etc.) is under way.

While these open-loop controllers demonstrated basic

rolling behavior, they commonly failed in the presence of external forces or unexpected terrain conditions. To solve this problem, we developed a simple rolling algorithm that uses ground contact sensors located on the simulated end caps. Preliminary results have shown steerable rolling on various terrains.

This brief analysis of coevolutionary learning for icosahedral tensegrity locomotion demonstrates that learning-based controls can provide robust rolling locomotion without analytical knowledge of the robot's dynamics.

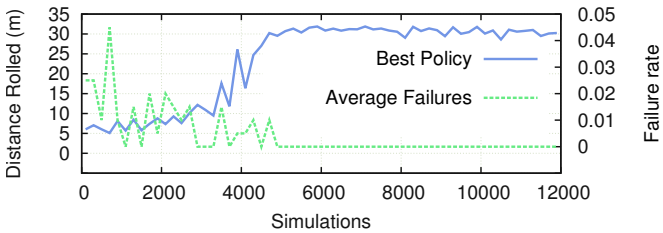

Figure 8: Distance covered by the robot in 60s with distributed learning of open-loop controllers based on coevolutionary algorithms. Each of the 24 outer shell spring-cable assembly controllers has a different evolution pool, but their combined behavior is optimized.

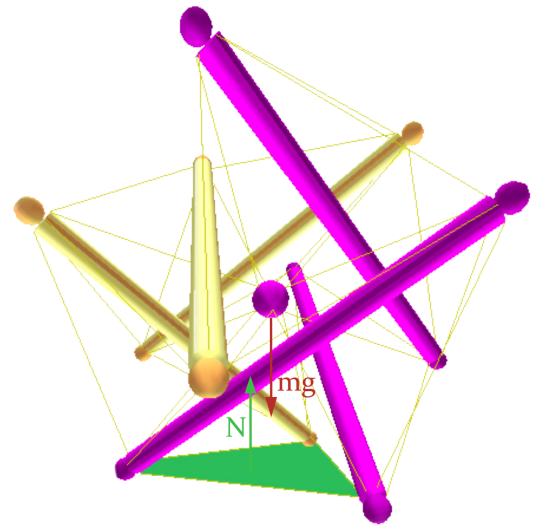

Figure 9: Regular icosahedron tensegrity shape with central payload (see Figure 5 center). The triangular contact surface with the ground, highlighted in green, creates a reaction force N that, at rest, balances the weight of the structure, represented on the figure by the red arrow mg. Torque is created on the whole structure when displacement of the center of mass from its rest position occurs.

IV.2 Bio-Inspired Control

In contrast to the direct learning technique presented above, our second set of approaches is more designer-involved and specific to this structure. State feedback was used to increase rolling performance of the tensegrity robot with payload simulated with the NTRT (Figure 5).

The idea behind these control laws is to create torque by moving the robot's center of mass with respect to ground contact surface to cause the robot to roll, as illustrated in Figure 9. This motion is achieved with a two-layer control architecture: the robot's heading and speed are controlled by displacement of the central payload using the inner spring-cable assemblies, and motion is simplified by actuating the outer shell.

Three control approaches test inner spring-cable assemblies: reactive controls, Inverse Kinematics-based controls (IK), and CPG-based controls. Outer spring-cable assemblies are controlled with hand-tuning. Actuation of the outer shell reduces ground contacts, not directly influencing heading or speed. This affects motion in several ways. First, it allows for creation of greater torques with the same payload displacement. Second, it smoothens rolling behavior of the structure by preventing discontinuities due to excessive ground contact.

For each control approach, inputs were taken as functions of the robotic state. The height of each strut was computed using simulated omnidirectional distance sensors located at the end of each rod. The height assigned to each spring-cable assembly was computed as the average of the two end points' height. We understand 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 [32], is to estimate system states from other sources, such as accelerometer 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 [20]. Finally, one could use a CPG to emulate rhythmic activation of sensors, similar to the approach in section IV.2.2.

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, requiring less force to induce a roll. Typically in the presence of a slope, reduction of ground contact surface is sufficient to cause a 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, speed is a scalar number and its sign depends on the robot's heading(positive in the desired direction and negative otherwise). Speed can then be used as feedback to influence the spring actuator command. Rest lengths of the *shell* spring-

cable assemblies are computed using the following actuation rule:

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

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 where length corrections occur. ℓ_0 and h_0 represent the offset-rest length of the spring and the maximum height measurement. 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. Input and output parameters of this control law are updated continuously through feedback control. Impedance control, which was adapted to tensegrities previously [24, 33], is used to modify spring-cable rest lengths.

IV.2.1 Reactive Controls

The first technique for actuation of inner payload spring-cable assemblies was the use of reactive controllers.We note that the only controllable parameter is cable length. The variables ℓ_i here are the rest length of the inner springs. 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, 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:

$$\dot{\ell}_i = (\ell_0 + d_i\gamma - \|\mathbf{p}_{i,0} - \mathbf{p}_{i,1}\|)w_r \tag{3}$$

$$\ell_i(0) = \ell_0 \tag{4}$$

where the weight w_r determines reactivity of the system and $\gamma < 0$ is a fixed parameter. Thus, without any external perturbation, the system has a stable equilibrium position at $\ell_0 + d_i\gamma$. Rest length of the spring-cable assemblies where the orientation aligns with the global heading is reduced. Vice-versa, springs pointing in the opposite direction are elongated. The global result is displacement of the payload in the direction of the heading vector, as shown in Figure 10. Note that the heading direction $\mathbf{v}$ can be chosen arbitrarily and adjusted dynamically. This method resulted in stable and smooth rolling gaits, allowing a roll of up to 1m/s (≈1 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.

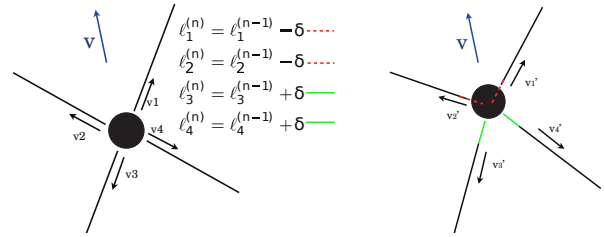

Figure 10: Computation of new rest lengths according to the spring-cable assemblies' individual orientations $\vec{v}_i$ (time $t^{(n-1)}$). Length modification is indicated by colored lines, dashed red if reduced and green if elongated. The resulting effect is displacement of the central payload in the desired direction $\vec{v}$ (time $t^{(n)} = t^{(n-1)} + dt$).

IV.2.2 CPG Controls

CPGs have been successfully used in past tensegrity systems [23]. Such controls are a feasible alternative to reactive controllers that enable generation of regular motion patterns. For this control, full-state information is used to generate smooth motion under reactive controls. Then, resulting periodic commands were stored as a stable limit cycle of a CPG. Once this process completes, the tensegrity can be driven by CPG output with much less feedback. We used an adaptive frequency Hopf oscillator [34] during the learning phase where the tensegrity was reactively control-driven. The underlying dynamical system reads:

$$\dot{u} = \gamma(\mu - (u^2 + g^2))u - \omega g + \epsilon b(t) \tag{5}$$

$$\dot{g} = \gamma(\mu - (u^2 + g^2))g + \omega u \tag{6}$$

$$\dot{\omega} = -\epsilon b(t)\frac{g}{u^2 + g^2} \tag{7}$$

where u, g, and ω are 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 to simplify notation: u designates $u_i(t)$, with i the index of the spring-cable assembly. The output u can synchronize to any periodic input signal $b(t)$ and replaces the feedback signal from the previous section.

$$d_i(t) = u_i(t). \tag{8}$$

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

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

$$\dot{g} = \gamma(\mu - (u^2 + g^2))g + \omega u \tag{10}$$

$$\dot{\omega} = 0 \tag{11}$$

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, minimal computations. However, it is important to note that the dynamical system runs on a much larger time scale than perturbations disturbing the system. A tensegrity driven only by a CPG would then, in the best case, only have a stable rolling gait on a flat, obstacle-free terrain. Consequently, it is necessary to include a second control method that works on this smaller time scale and gives an appropriate response to external perturbations.

IV.2.3 Hybrid CPG - Inverse Kinematics Controls

The final control method we tested 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 write this as a small displacement $\delta\mathbf{p}$ of the payload:

$$\begin{aligned} \delta p_i \approx p_i(\boldsymbol{\ell}^{(0)}) \quad &+ \quad \sum_{j=1}^{n} \frac{\partial p_i(\boldsymbol{\ell}^{(0)})}{\partial \ell_j} \delta \ell_j \\ &+ \quad \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 p_i(\boldsymbol{\ell}^{(0)})}{\partial \ell_j \partial \ell_k} \delta \ell_j \delta \ell_k \end{aligned} \quad (12)$$

or

$$\begin{aligned} \delta p_i \approx p_i(\boldsymbol{\ell}^{(0)}) \quad &+ \quad \left(J(\mathbf{p}^{(0)}) \delta \boldsymbol{\ell} \right)_i \\ &+ \quad \frac{1}{2} \delta \boldsymbol{\ell}^T H \left(p_i(\boldsymbol{\ell}^{(0)}) \right) \delta \boldsymbol{\ell} \end{aligned} \quad (13)$$

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

Considering Eq. 13, we additionally define $\Delta\mathbf{p} = \delta\mathbf{p} - \mathbf{p}(\boldsymbol{\ell}^{(0)})$ and $\mathbf{f}(\delta\boldsymbol{\ell}; \Delta\mathbf{p})$ as

$$f_i(\delta\boldsymbol{\ell}; \Delta\mathbf{p}) = \left(J(\mathbf{p}^{(0)}) \delta \boldsymbol{\ell} \right)_i + \frac{1}{2} \delta \boldsymbol{\ell}^T H \left(p_i(\boldsymbol{\ell}^{(0)}) \right) \delta \boldsymbol{\ell} - \Delta p_i. \quad (14)$$

Computation of the spring-cable rest length changes $\delta\boldsymbol{\ell}$ for a desired payload displacement $\Delta\mathbf{p}$ corresponds to finding the $\delta\boldsymbol{\ell}$ that cause $\mathbf{f} = \mathbf{0}$. Since this last equation is over-determined, nonlinear, 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:

$$\delta\boldsymbol{\ell}_{k+1} = \delta\boldsymbol{\ell}_k - J_k^{-1} \mathbf{f}(\delta\boldsymbol{\ell}_k; \Delta\mathbf{p}) \quad (15)$$

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

$$J = \left[\frac{\partial f_i}{\partial (\delta \ell_j)} \right]_{ij}. \quad (16)$$

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 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 IV.2.2. This approach is inspired by two previous works by Ajallooeian et al. [35] and Gay et al. [36]. We update ξ if and only if the payload position lies on the opposite side of the robot's center of mass, and we continuously adjust ξ with time according to the following evolution rule:

$$\frac{d}{dt} \xi(t) = -\alpha \xi(t) \quad (17)$$

with $\alpha \in \mathbb{R}_+$. In this way, 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:

$$\dot{u} = \gamma(\mu - (u^2 + g^2))(u - \xi(t)) - \omega g - \eta h(t) \quad (18)$$

$$\dot{g} = \gamma(\mu - (u^2 + g^2)) g + \omega(u - \xi(t)), \quad (19)$$

If the value of $\xi(t)$ is constant over time, the dynamical system converges asymptotically to $u(t) = \xi$ [36]. While the pure CPG implementation does not allow any steering control, this implementation enables guidance on a desired trajectory on flat terrain (see Figure 11).

Table 1 provides a summary of results obtained with different control strategies over regular, flat terrain.

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

Table 1: Bio-inspired control strategies summary

Note that results do not take the trajectory of the path into account and, consequently, even if the distance traveled using the CPG controller without any trajectory control is larger than with hybrid control, the "quality" of the path is worse (e.g., see Figure 11). Interestingly, we observe the stable gait pattern obtained in simulation is a sequence of contacts defined as energetically optimal by Koizumi et al. [20] for a tensegrity icosahedron. With the current implementation, only the reactive controller manages to induce rolling efficiently over

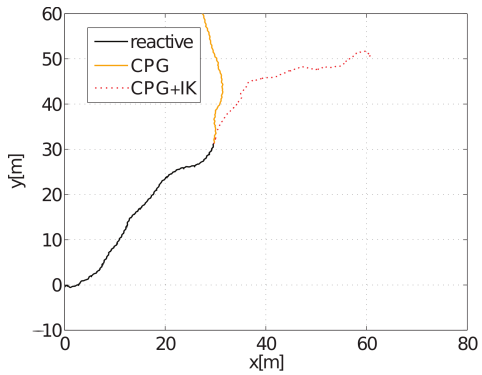

Figure 11: Trajectory of the tensegrity (top view). The black curve represents the trajectory while the robot is driven by the reactive control algorithm and the CPG is in the learning mode (50s). Motion is regular and the heading is maintained throughout the entire period. Yellow and red trajectories 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.

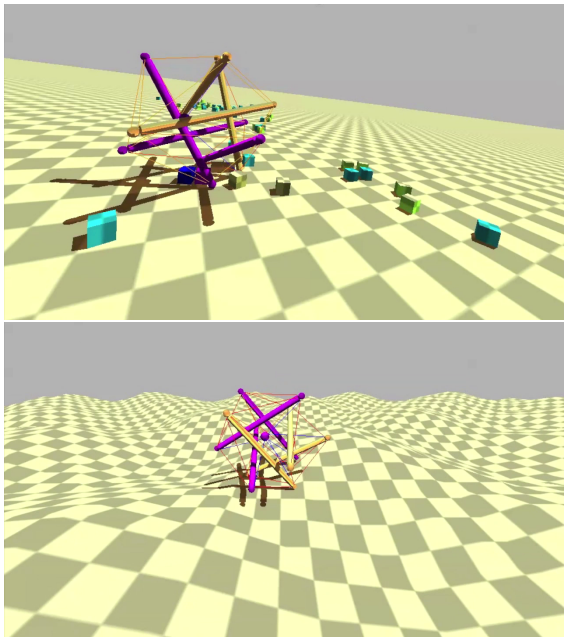

Figure 12: Examples of successful locomotion over complex terrains, such as slopes, bumps and obstacles

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. Figure 12 presents such results from within the NTRT simulator.

Experimentation demonstrated that the hybrid controller's performance is highly sensitive to some parameters appearing in the CPG equations, such as the ones presented in Eq. 18 and 19. As a result, future work will incorporate other methods to optimize feedback data and compute corrections to more accurately navigate complex environments. A good example of such an improvement can be found in Gay et al. [37], 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, Reservoir Computing can also be a suitable tool for feedback computation, as detailed in the following section.

IV.3 Learning a Matsuoka Oscillator With Physical Reservoir Computing

This section presents the final controls results of this work: an implementation of the Physical Reservoir Computing (PRC) principle on the ReCTeR hardware prototype. These results validate use of the NTRT in an untethered, underactuated feedback control of tensegrity icosahedra with string force sensors [38, section 5.1.1]. Here, closed-loop feedback control is used when motor signals are generated by a Matsuoka oscillator. This demonstrates a successful adaptation of our simulation results to a physical platform (ReCTeR), 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 [39] (see [38, section 3.1] for oscillator parameters and motivation). These signals represent desired actuator signals. We manually scaled target signals so 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 that allowed motion-capture tracking of the full experiment.

The algorithm proceeds by first applying target rest lengths to actuators in an open-loop setup, inducing the robot to start moving. Next, online learning is applied to approximate desired signals based on sensor readings. These approximations are the feedback signals. The ratio of open-loop versus feedback signals is gradually decreased until signals are generated by the feedback loop alone. At this point, the robot will robustly maintain oscillatory patterns. The precise equations and parameters used in the experiment are provided in [38] and the supplementary material.

In our prior simulation work, we used the term PRC to describe how nonlinear computations, which are inherently performed by a physical system, can be easily exploited to simplify control of tensegrities [38]. PRC extends the Reservoir Computing (RC) concept that, at its origin, is a simple

technique to train recurrent neural networks [40, 41]. The common idea is that a system with complex dynamics is perturbed externally, but is otherwise left untouched. 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 [42, 43].

Controller feedback signals are obtained from ReCTeR's 24 force transducers. Since these sensors are mounted perpendicularly to the robot's struts, output values depend on the angle of attack and tension of the attached spring-cable assembly. Thus, the sensors provide a readout of the robot's state, similar to state observations in RC. The robot's behavior was evaluated using the motion capture setup described in section III.1.

Figure 13 shows the result of an experiment where we first outsourced motor-signal generation to the feedback loop by online learning of the feedback weights. After training, we disturbed the system (lifting and constraining the robot). In this case, the robot stops moving and switches back to its original oscillatory mode when released, demonstrating 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 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 that allow mitigation of stringent sensor requirements. We discussed many variations and extensions on the hardware experiment presented here in our prior simulation work [38].

V. Future Work

Current prototype hardware allows for multiple verification levels of NTRT simulations. However, a more capable robot design is required to implement fully dynamic controls from the CPG systems and related work. ReCTeR has a maximum tension-force limit in its cables, as well as with the number of cables actuated. We are currently working on a redesigned six-strut robot with twice the number of actuators, with torque and velocity capabilities an order of magnitude higher than ReCTeR [44]. This robot will be able to implement the more advanced control schemes described in this paper. Design of this new robot will also target payload protection, a crucial feature for space exploration.

On the control side, one of our future goals is further integration of CPG and RC approaches, to maximally exploit the advantages of both.

VI. Conclusion

Tensegrity is a curious design concept, spanning art, science, and biology. This work presented the tensegrity workflow developed at the NASA Ames Research Center. Our simulator setups were described and demonstrated, and a new, highly compliant, untethered tensegrity robot - ReCTeR - was used to validate simulator setups in both dynamic and kinematic situations. Next, various control strategies were presented, based on evolutionary algorithms and CPGs, and a feedback controller was implemented on the hardware platform to demonstrate sensor capabilities. The biologically-inspired control approaches we are exploring appear naturally suited for biologically-inspired tensegrity structures, due to their matching nonlinear and oscillatory qualities.

An important aspect of this work is the creation of an Open Source simulation environment (the NTRT) for tensegrity-based mobility and manipulation controls research that has now been validated against hardware. This simulation environment enables us to develop an understanding of the structure and qualities of successful control approaches. Using evolutionary exploration of parameters for different structural and biologically-inspired control approaches, this system can be used to develop performance-driven hardware requirements, such as the forces experienced in the rods, speed and torque requirements for actuators, elasticity constants for springs, and sensor requirements and placements. Developing the right toolset and design workflow enables progress beyond tensegrity robots that merely move, and into a realm where tensegrity systems purposefully interact with the environment and execute tasks.

Acknowledgments

This research was supported by the European Commission's FP7 program under grant agreement No 248311 - AMARSi and the NASA Innovative Advanced Concepts program. Ken Caluwaerts was supported by a Ph.D. fellowship of the Research Foundation - Flanders (FWO). Support also came from NSF Graduate Research Fellowship No. DGE1106400, and by NASA Prime Contract Number NAS2-03144 awarded to the University of California, Santa Cruz, University Affiliated Research Center.

The authors would like to thank Ryan Adams, Adrian Agogino, Alice Agogino, Mostafa Ajallooeian, Lee Brownston, Michiel D'Haene, Stephen R. Ellis, Tom Flemons, Terry Fong, Jeffrey Friesen, Auke Jan Ijspeert, George Korbel, Stephen Levin, Sophie Milam, Kyle Morse, Greg Orzech, In Won Park, Alexandra Pogue, Brian Tietz, Kagan Tumer, Tim Vets, Tim Waegeman, Kyunam Kim, and the NASA Ames Intelligent Robotics Group.

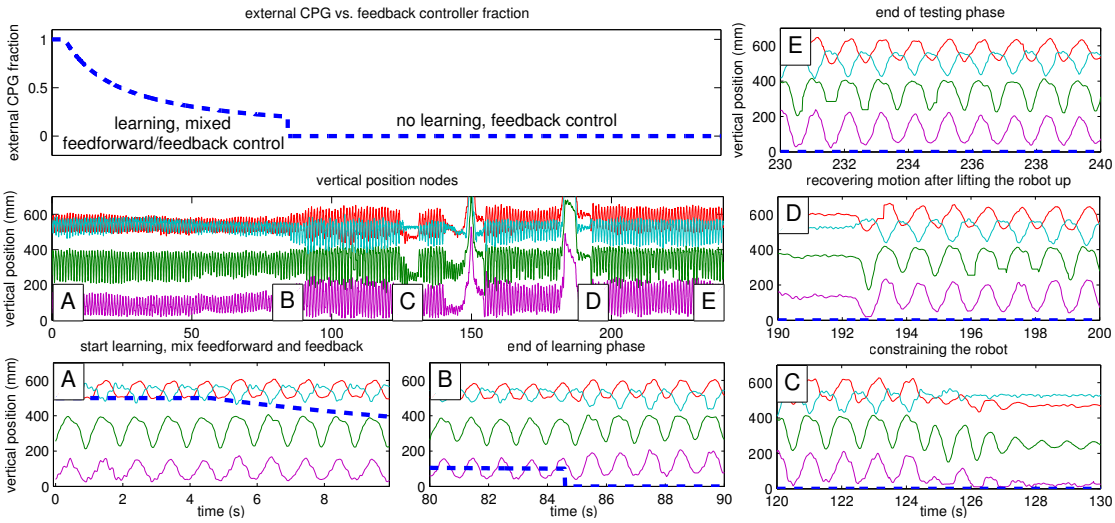

Figure 13: Fast online learning of a static feedback controller for a Matsuoka oscillator on ReCTeR based on uncalibrated strain-gauge sensors. The top left plot shows the fraction of feedforward versus feedback control. During learning, both feedback and feedforward controllers (training signals) are 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 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 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 from the ground. E) Behavior of the robot after 250s (170s closed loop).

controller type	main features			type of locomotion	
robot	HW/SIM	loop	actuators	sensors	references
motion capture experiments	untethered, underactuated control, and simulator validation			rolling, single flop, forward kinematics	
ReCTeR	HW & SIM	open	6	(motion capture)	Section III
reactive controller with coevolution	robust feedback controller with minimal assumptions			steerable rolling over unknown terrain	
icosahedron with/without payload	SIM	closed	24	touch sensors	[45] & Section IV.1
Bio-inspired strategies (CPG)	robust and bioinspired			steerable rolling over unknown terrain	
icosahedron with payload	SIM	closed	36	distance sensors	Section IV.2
physical reservoir computing	robust controller with uncalibrated sensors, link with CPGs			crawling (HW), various (SIM)	
ReCTeR (HW), icosahedron (SIM)	HW & SIM	closed	6 (HW), variable (SIM)	tension sensors	[38] & Section IV.3
sine waves with coevolution	simple, distributed implementation			forward rolling on flat terrain	
icosahedron with/without payload	SIM	open	24/36	-	[29]
stepwise functions with coevolution	HW constraints and power consumption information			forward rolling	
icosahedron with/without payload	SIM	open	24	-	Section IV.1
morphological communication	communication through body dynamics, distributed control			crawling like	
15 bar tensegrity tower	SIM	closed	30	tension sensors	[27]
genetic algorithms	first dynamic locomotion			crawling	
3 and 4 bar prisms	SIM	open	9/12	-	[1]
pneumatic actuators	insightful control, original hardware implementation			rolling	
icosahedron with pneumatic actuators	HW	open	24	-	[19, 20]
feedback non-linear control	control theory approach			position/trajectory control	
any tensegrity	SIM	closed	all	full state	[15]
vibration driven	cheap hardware, exploits body dynamics			various	
various	HW	open	1	-	[46]
kinematic controllers	tested on HW platforms and well studied			none	
various (constrained)	HW & SIM	both	variable	-	[47, 48]
CPG resonance entrainment	demonstration of HW CPG control			none	
class two tensegrity beam	HW & SIM	both	2 linear actuators	tension & position	[23]

Table 2: Overview of various types of controllers for tensegrity robots and our experiments.

A. Tensegrity Control Methods Overview

Table 2 provides an overview of various control methods for tensegrity structures. We list their main features and the type of locomotion.

References

[1] Paul, C., Valero-Cuevas, F. J., and Lipson, H., 2006 Design and control of tensegrity robots for locomotion. *IEEE Transactions on Robotics* **22**(5). doi:10.1109/TRO.2006.878980.

[2] Wang, N., Tytell, J. D., and Ingber, D. E., 2009 Mechanotransduction at a distance: mechanically coupling the extracellular matrix with the nucleus. *Nature reviews Molecular cell biology* **10**(1):75–82. doi:doi:10.1038/nrm2594.

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

[4] Wang, N., Naruse, K., Stamenović, D., Fredberg, J. J., Mijailovich, S. M., Tolić-Nørrelykke, I. M., Polte, T., Mannix, R., and Ingber, D. E., 2001 Mechanical behavior in living cells consistent with the tensegrity model. *PNAS* **98**(14):7765–7770. doi:10.1073/pnas.141199598.

[5] Levin, S., 2002 The tensegrity-truss as a model for spine mechanics: Biotensegrity. *Journal of Mechanics in Medicine and Biology* **2**:375–388. doi:10.1142/S0219519402000472.

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

[7] Flemons, T., 2012 The bones of tensegrity. *http://www.intensiondesigns.com/bones_of_tensegrity* .

[8] Tibert, G., 2002 *Deployable Tensegrity Structures for Space Applications*. Ph.D. thesis, Royal Institute of Technology.

[9] SunSpiral, V., Gorospe, G., Bruce, J., Iscen, A., Korbel, G., Milam, S., Agogino, A., and Atkinson, D., 2013 Tensegrity based probes for planetary exploration: Entry, descent and landing (EDL) and surface mobility analysis. In *10th International Planetary Probe Workshop (IPPW)*.

[10] Agogino, A., SunSpiral, V., and Atkinson, D., 2013 Super Ball Bot - structures for planetary landing and exploration. *NASA Innovative Advanced Concepts (NIAC) Program, Final Report* .

[11] Fuller, R. B., 1975 *Synergetics: Explorations in the Geometry of Thinking*. Scribner. ISBN 002541870X.

[12] Snelson, K., February 1965. Continuous tension, discontinuous compression structures. United States patent 3169611.

[13] Juan, S. H. and Tur, J. M. M., 2008 Tensegrity frameworks: Static analysis review. *Mechanism and Machine Theory* **43**(7):859 – 881. doi:10.1016/j.mechmachtheory.2007.06.010.

[14] Bel Hadj Ali, N., Rhode-Barbarigos, L., Pascual Albi, A., and Smith, I., 2010 Design optimization and dynamic analysis of a tensegrity-based footbridge. *Engineering Structures* **32**(11):3650–3659. doi:10.1016/j.engstruct.2010.08.009.

[15] Skelton, R. E. and De Oliveira, M. C., 2009 *Tensegrity Systems*. Springer, 2009 edition. ISBN 0387742417.

[16] Tur, J. M. M. and Juan, S. H., 2009 Tensegrity frameworks: Dynamic analysis review and open problems. *Mechanism and Machine Theory* **44**:1–18. doi:10.1016/j.mechmachtheory.2008.06.008.

[17] Graells Rovira, A. and Mirats-Tur, J. M., 2009 Control and simulation of a tensegrity-based mobile robot. *Robotics and Autonomous Systems* **57**(5):526–535. doi:10.1016/j.robot.2008.10.010.

[18] Paul, C., Roberts, J. W., Lipson, H., and Valero Cuevas, F. J., 2005 Gait production in a tensegrity based robot. 216–222. doi:10.1109/ICAR.2005.1507415.

[19] Shibata, M., Saijyo, F., and Hirai, S., 2009 Crawling by body deformation of tensegrity structure robots. In *ICRA*, 4375 –4380. doi:10.1109/ROBOT.2009.5152752.

[20] Koizumi, Y., Shibata, M., and Hirai, S., 2012 Rolling tensegrity driven by pneumatic soft actuators. In *ICRA*, 1988–1993. doi:10.1109/ICRA.2012.6224834.

[21] Ijspeert, A. J., 2008 Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks* **21**(4):642–53. doi:10.1016/j.neunet.2008.03.014.

[22] Degallier, S. and Ijspeert, A. J., 2010 Modeling discrete and rhythmic movements through motor primitives: a review. *Biological Cybernetics* **103**(4):319–338. doi:10.1007/s00422-010-0403-9.

[23] Bliss, T., Werly, J., Iwasaki, T., and Bart-Smith, H., 2012 Experimental validation of robust resonance entrainment for CPG-controlled tensegrity structures. *IEEE Transactions On Control Systems Technology* **21**:666–678. doi:10.1109/TCST.2012.2189400.

[24] Orki, O., Ayali, A., Shai, O., and Ben-Hanan, U., 2012 Modeling of caterpillar crawl using novel tensegrity structures. *Bioinspiration & Biomimetics* **7**(4):046006. doi:10.1088/1748-3182/7/4/046006.

[25] Panait, L., 2010 Theoretical convergence guarantees for cooperative coevolutionary algorithms. *Evolutionary Computation* **18**:581–615. doi:10.1162/EVCO_a_00004.

[26] Tumer, K. and Agogino, A., 2007 Distributed agent-based air traffic flow management. In *AAMAS*, 330–337. doi:10.1145/1329125.1329434.

[27] Rieffel, J., Valero-Cuevas, F. J., and Lipson, H., 2010 Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of the Royal Society Interface* **7**(45):613–21. doi:10.1098/rsif.2009.0240.

[28] Iscen, A., Agogino, A., SunSpiral, V., and Tumer, K., 2013 Learning to control complex tensegrity robots. In *AAMAS*, 1193–1194.

[29] Iscen, A., Agogino, A., SunSpiral, V., and Tumer, K., 2013 Controlling tensegrity robots through evolution. In *GECCO*. doi:10.1145/2463372.2463525.

[30] Coumans, E., 2005. Bullet physics engine.

[31] Calladine, C. R., 1978 Buckminster Fuller's Tensegrity structures and Clerk Maxwell's rules for the construction of stiff frames. *Int. Journal of Solids and Structures* **14**(2):161–172. doi:10.1016/0020-7683(78)90052-5.

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

[33] Tietz, B. R., Carnahan, R. W., Bachmann, R. J., Quinn, R. D., and SunSpiral, V., 2013 Tetraspine: Robust terrain handling on a tensegrity robot using central pattern generators. In *AIM*, 261–267. doi:10.1109/AIM.2013.6584102.

[34] Righetti, L., Buchli, J., and Ijspeert, A. J., 2005 From dynamic hebbian learning for oscillators to adaptive central pattern generators. In *AMAM*, 45.

[35] Ajallooeian, M., Gay, S., Tuleu, A., Sproewitz, A., and Ispeert, A., 2013 Modular control of limit cycle locomotion over unperceived rough terrain. In *IROS*, 3390–3397. doi:10.1109/IROS.2013.6696839.

[36] Gay, S., Dégallier, S., Pattacini, U., Ijspeert, A., and Victor, J. S., 2010 Integration of vision and central pattern generator based locomotion for path planning of a nonholonomic crawling humanoid robot. In *IROS*, 183–189. doi:10.1109/IROS.2010.5648788.

[37] Gay, S., Santos-Victor, J., and Ijspeert, A., 2013 Learning robot gait stability using neural networks as sensory feedback function for central pattern generators. In *IROS*, 192–201. doi:10.1109/IROS.2013.6696353.

[38] Caluwaerts, K., D'Haene, M., Verstraeten, D., and Schrauwen, B., 2013 Locomotion without a brain: physical reservoir computing in tensegrity structures. *Artificial Life* **19**(1):35–66. doi:10.1162/ARTL_a_00080.

[39] Matsuoka, K., 1987 Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics* **56**(5-6):345–353. doi:10.1007/bf00319514.

[40] Verstraeten, D., Schrauwen, B., D'Haene, M., and Stroobandt, D., 2007 An experimental unification of reservoir computing methods. *Neural Networks* **20**:391–403. doi:10.1016/j.neunet.2007.04.003.

[41] Jaeger, H. and Haas, H., 2004 Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**(5667):78–80. doi:10.1126/science.1091277.

[42] Nakajima, K., Hauser, H., Kang, R., Guglielmino, E., Caldwell, D. G., and Pfeifer, R., 2013 A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience* **7**(91). doi:10.3389/fncom.2013.00091.

[43] Nakajima, K., Hauser, H., Kang, R., Guglielmino, E., Caldwell, D. G., and Pfeifer, R., 2013 Computing with a muscular-hydrostat system. In *ICRA*, 1504–1511. doi:10.1109/ICRA.2013.6630770.

[44] Bruce, J., Caluwaerts, K., Iscen, A., Sabelhaus, A. P., and SunSpiral, V., 2014 Design and evolution of a modular tensegrity robot platform. In *ICRA*, 3483–3489.

[45] Iscen, A., Agogino, A., SunSpiral, V., and Tumer, K., 2014 Flop and roll: Learning robust navigation for rolling tensegrity robot. In *IROS*. To appear.

[46] Bohm, V. and Zimmermann, K., 2013 Vibration-driven mobile robots based on single actuated tensegrity structures. In *ICRA*, 5475–5480. doi:10.1109/ICRA.2013.6631362.

[47] Fest, E., Shea, K., and Smith, I. F., 2004 Active tensegrity structure. *Journal of Structural Engineering* **130**(10):1454–1465.

[48] Mirats-Tur, J. and Camps, J., 2011 A three-dof actuated robot. *Robotics Automation Magazine, IEEE* **18**(3):96–103. doi:10.1109/MRA.2011.940991.