SOFT ROBOTICS
Volume 2, Number 4, 2015
© Mary Ann Liebert, Inc.
DOI: 10.1089/soro.2015.0012

ORIGINAL ARTICLE

Goal-Directed CPG-Based Control for Tensegrity Spines with Many Degrees of Freedom Traversing Irregular Terrain

Brian T. Mirletz,[1] Perry Bhandal,[2] Ryan D. Adams,[3] Adrian K. Agogino,[4]
Roger D. Quinn,[1] and Vytas SunSpiral[5]

Abstract

To further the ability of robots to achieve goals in environments with irregular terrain, we have developed a series of tensegrity spines as an abstraction of the many degrees of freedom (DOF) compliant spines seen in nature, with full six DOF between vertebrae (constrained by a tensile network). This work provides insight into control strategies for such many DOF and compliant systems, which lack the rigidly connected segments needed by traditional control. Our Central Pattern Generator (CPG)-based controller receives both proprioceptive feedback and goal-directed input. We utilize artificial neural networks to process both the feedback and the input, and only use feedback available to our analogous robotic hardware. This approach seeks to maximize the low-level competence of the control system, by combining local reflexes with structural compliance. This is, to our knowledge, the first example of a robot controlled by CPGs that is simultaneously capable of goal-directed behavior and locomotion on irregular terrain. In addition, this is the first goal-directed controller for a tensegrity robot that can transition between different terrains.

Introduction

To achieve goals in applications such as search and rescue, exploration, medicine, and interaction with humans, robots need to adapt to their surroundings, utilizing compliant yet rugged bodies to execute those adaptations. Soft robots could provide this adaptability and revolutionize the role of robots in these areas.[1] However, one of the major challenges for soft robotic systems is the lack of control paradigms compatible with the many degrees of freedom (DOF) and whole body compliance of soft systems.[2] This control challenge is shared by tensegrity robots, which suspend rigid elements within a compliant tension network,[3] and tensegrity robots are often viewed as a soft–rigid hybrid.[2]

Studying animals highlights how a flexible, actuated spine could improve a robot's performance. In evolutionarily lower tetrapods such as lizards and crocodiles, the lateral bending (coronal plane) of the skeleton produces the advancement of the limbs for locomotion.[4] Mammals also have the ability to bend vertically (sagittal plane), which enables higher speed gaits such as galloping.[5] Galloping utilizes the elasticity of the spine to transfer energy between legs.[6] In addition, the dorsal muscles of a dog's spine appear to contribute to stabilization in all three planes of the body, with roles (mobility and stability) changing somewhat between gaits.[7]

Tensegrity structures are often proposed as models of biological systems. For example, tensegrity can model the nonlinear mechanical properties of the cell's cytoskeleton such as strain hardening.[8] These insights have implications for practical applications in tissue engineering, as they help determine how cells will attach to substrates,[9] and even change the genetic expression of the cell (known as cell fate).[10] Tensegrity properties can be observed in macroscopic biological structures, particularly through the connective tissue such as fascia.[11] Turvey and Fonseca hypothesize that tensegrity-based biomechanics would explain phenomena in haptic perception such as loss of limb awareness in microgravity and perception of phantom body parts.[12] Passive tensegrity models of the spine,[13,14] knee,[15] and shoulder girdle[16] can capture geometric features of anatomy without the simplifying assumption of a stable platform within the body or a large reduction in the DOF (as in more traditional models such as Ref.[17]). For review of

[1]Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, Ohio.
[2]School of Computing, Queen's University, Kingston, Ontario, Canada.
[3]Cyan Data Systems, Oakland, California.
[4]Robust Software Engineering Group and [5]Intelligent Robotics Group, NASA Ames Research Center, Moffett Field, California.

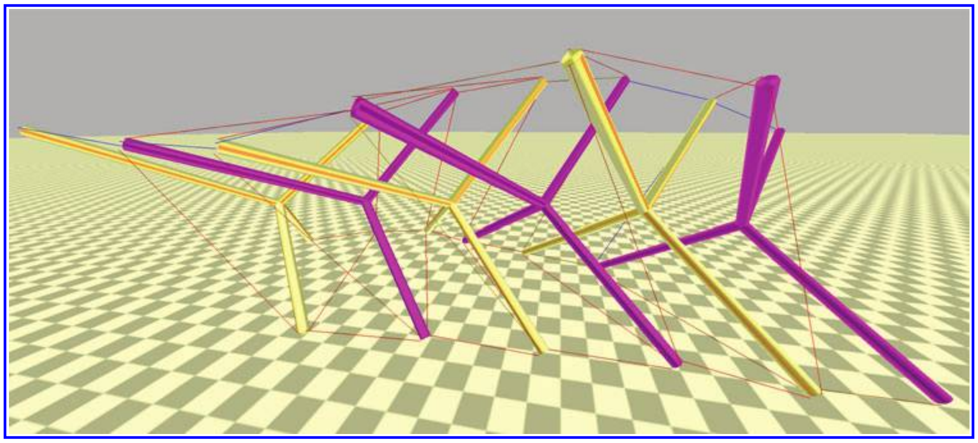

FIG. 1. A tensegrity spine model simulated in NASA Tensegrity Robotics Toolkit (NTRT) performing a crawling motion, using a tetrahedral complex as vertebrae. Eight strings connect each segment, four of which form a tensegrity saddle joint.

biotensegrity models, see Ref.[18] These biologically inspired designs motivated robotic tensegrity models of the caterpillar,[19] and Manta-ray's wings[20] or tail.[21,22]

While analytical methods exist for computing the dynamics of tensegrity structures,[23] they require detailed knowledge of the forces experienced by the robot, which quickly becomes intractable for dynamic locomotion in irregular terrain. Other types of controllers for tensegrity robots, such as a spiking neural network controlling an irregular (machine evolved) tensegrity structure[24,25] or a controller that exploits the vibrational modes of the structure,[26,27] have been demonstrated successfully on flat ground with minimal information about the environment. However, irregular terrain is likely to modify the direction of motion for these types of controllers. Similarly, machine learning can determine patterns of cable lengths that successfully traverse irregular terrain, but these need to be retuned for each terrain type.[28,29] Therefore, it is worth investigating controllers for soft and tensegrity robots that can produce goal-directed motion over multiple terrains without retuning, using sensory information that can be easily obtained in a tensegrity system.

Animals can control the many DOF of their compliant bodies across complex terrains, so it is natural to look for solutions in neuroscience and biology. Motor control in animals is distributed across neural centers, ranging from higher centers in the brain to local control circuits in the spine and body.[30] The biologically inspired robotics community has long known the value of distributed control, local reflexes, and passive dynamics for adaptive behavior.[31–33] More recently, engineers have applied the dynamics of neurons that control low-level locomotion to robots, through models of central pattern generators (CPGs).[34] Implementations range from using detailed models of individual neurons as a node[35,36] to neuron models based on firing frequency,[37] to abstract, oscillatory dynamical systems.[38,39] The abstracted mathematical behavior creates a stable limit cycle,[40] which entrains with the controlled system,[41] or provides stability in the presence of environmental perturbations.[42] Thus, CPGs are less likely to get stuck than reflexive state machine controllers, since the oscillatory dynamics drive locomotion. CPG-based controllers have formed the basis of bioinspired robotic behaviors, including gait transitions,[43] omnidirectional locomotion,[44] and adaptations to irregular terrain.[45] For tensegrity robots, CPG controllers have produced swimming in a tail-like tensegrity beam[22] and rolling and crawling motions in a six-strut icosahedron tensegrity robot.[46,47]

Our prior work developed CPG-based controllers for a variety of tensegrity robots as an abstraction of the many DOF-compliant spines seen in nature,[48,49] including online adaptations for traversing multiple terrains.[50] In this article, we add goal-directed input to the CPG control system through an artificial neural network (ANN), as an online, higher DOF generalization of manual body shape controls.[48] This addition makes the controller capable of goal-directed locomotion across multiple types of irregular terrain in simulation with a single set of tuned parameters.

Materials and Methods

Spine morphology

The tensegrity spine used in this work uses a three-dimensional tetrahedral complex for vertebrae. This spine was originally developed as a static model by Flemons[14] and is shown in Figure 1. The vertebrae consist of four rods that meet at a central point, as if they were drawn from the center of a tetrahedron to its vertices. When joined in a tension network, they make a stable yet flexible column through tetrahedral saddle joints. While the majority of man-made tensegrity models are from the small subset of possible structures one can make from simple straight rods and cables, the set of Class 1 tensegrity structures includes the use of any arbitrarily shaped rigid elements, as long as those rigid elements are not touching and are floating in the continuous tensile network of the cables.[23] Natural tensegrity structures take advantage of complexly shaped rigid structures, like the diverse morphologies of bones. An early example of engineered use of more complex rigid elements is Snelson's X-piece, which Skelton describes as a Class 1 structure.[23] The vertebrae of this spine model are related in form to the X-piece.

To create a tensegrity robot, all of the cables are individually actuated, although work to reduce the number of actuators is underway.[51] In hardware, actuation typically takes the form of a cable wrapped around a spool on a rotary DC motor.[48,49] In prior work, this tetrahedral complex morphology displayed the most efficient locomotion.[49] On additional tuning, this spine displayed the fastest locomotion of the tensegrity spine morphologies examined so far. For this work, a six-segment spine was used to reduce the computational time for contact dynamics and make the terrain features more difficult compared to a longer spine. Each rod was assumed to be 17.3 cm long, resulting in a 77.5 cm long robot.

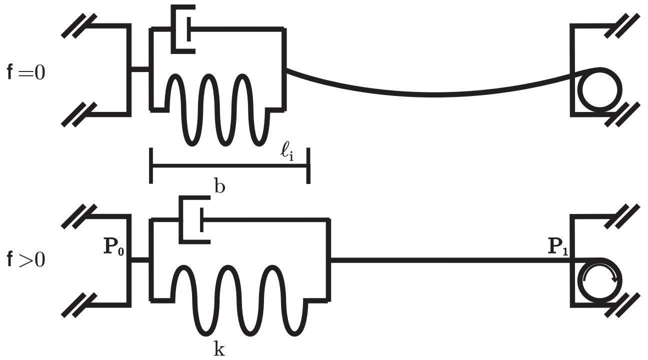

FIG. 2. A graphical representation of our Hooke's law-based spring model. The *top* picture represents a slack cable. In the *bottom*, the motor has spun clockwise, tightening the cable and applying equal and opposite forces to the rods.

NTRT

The NASA Tensegrity Robotics Toolkit (NTRT) is an open source software package with modules for the modeling, simulation, and control of tensegrity robots.* NTRT uses the Bullet Physics Engine's (version 2.82) mixed linear complementary problem solvers for rigid body dynamics and rigid–rigid contacts, and a custom soft body spring-cable model with contact dynamics that is, to our knowledge, the only open source cable model with both realistic forces and contact dynamics. This combination of rigid and soft interaction allows for the study of more complex terrain interactions than would be possible with an analytical model of the tensegrity structure. The internal dynamics of the cable model uses a Hooke's law spring model originally implemented in Ref.[47] and is shown in Figure 2. Prior work with NTRT has quantified the simulation error of this model with hardware trials, separately showing an average 1.3% error on position,[47] and 7.9% error on maximum force.[52] Actuation is accomplished by changing the spring's rest length, which is subject to the speed and force constraints of a DC motor model with a linear torque–speed curve. This motor model makes more accurate assumptions about energy use than prior work, such as including the applied torque when lengthening.

The cables' soft–rigid interaction is inspired by Servin and Lacoursiére[53] and Servin *et al.*[54] Collision detection is handled by ghost objects within Bullet, a soft body cable is represented as a massless cylinder with a small radius between the two rigid bodies. As additional contacts are accumulated, additional cylinders are added to account for bending, subject to a maximum resolution. Contacts are then subject to an overall cable length minimization similar to Ref.,[54] which accounts for sliding. Contacts are removed when they violate the resolution constraint, their force would be applied outward of the geometry (i.e., the string would push) or when contact is lost. Figure 3 provides an illustration of the forces computed by NTRT after Bullet has determined contacts, while Figure 4 shows an example implementation. Validation of the contact dynamics is future work.

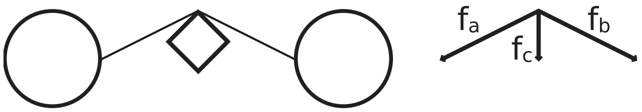

FIG. 3. *Left*: Two rods and a cable encounter a square block, creating a single point of contact. *Right*: A graphical representation of the force vectors in this situation. The contact force fc is applied to the block, while cable forces –fa and –fb are applied to the rods.

Tensegrity models in NTRT are constructed using a set of builder tools that use a tagged set of Cartesian coordinates (nodes) and their connectivity (pairs), and generates a tensegrity structure according to physical properties of the rods (radius, density, friction, restitution) and cables (stiffness, damping, motor properties). Accurate inertia matrices can be constructed by placing spheres at nodes as a concentrated mass along otherwise uniform density rods. Simulations were typically run at 1000 Hz. Simulations typically run in real time on an Intel Core™ i7 CPU, but simulation speed depends upon terrain.

Control algorithms

As demonstrated by prior work, it is possible to generate goal-directed locomotion with position-based controls of cable length[28] or open loop vibrational control.[26] In Figure 5, this is represented as the portion of the control system in brown. However, even with local distributed feedback, rough terrain alters the direction of motion or even prevents the robot from moving.[50] To adapt the controller to multiple terrains in real time, one needs to consider how to best utilize available sensory feedback. To achieve this, our controller incorporates distributed impedance controllers for rapid tension-based feedback (blue in Fig. 5), a CPG for adaptive trajectory generation (also blue), and a neural network for goal direction (orange).

Since locomotion primarily depends on the forces exerted on the environment, the control system for the robot is distributed at the lowest level to tension control for each individual cable. The tension set point is generated using a scalar form of impedance control, originally adapted to tensegrity by Orki *et al.*[19] and modified for dynamic locomotion by providing a trajectory to the velocity term.[48] Impedance control allows for tunable stiffness and incorporates position and velocity information into a tension set point. Figure 5 includes a block diagram of the impedance controller in relation to the entire control system. The scalar equation for an individual cable is as follows:

$$T = T_0 + K(L - L_0) + B(V - V_0), \qquad (1)$$

where T is the tension set point, T_0 is a tension offset, K is a position gain on the difference between the cable's current actual length L and desired length L_0. B is the analogous gain for V and V_0, where V_0 is a trajectory provided by the CPG. L_0 is typically the initial length of the cable, maintaining information about the robot's initial shape. If the resulting set point is less than zero, the controller will set $T = 0$. The tension set point is sent to a PD controller for low-level motor

*Information, source code, and documentation for NTRT and the controller presented here can be found at http://irg.arc.nasa.gov/tensegrity/NTRT

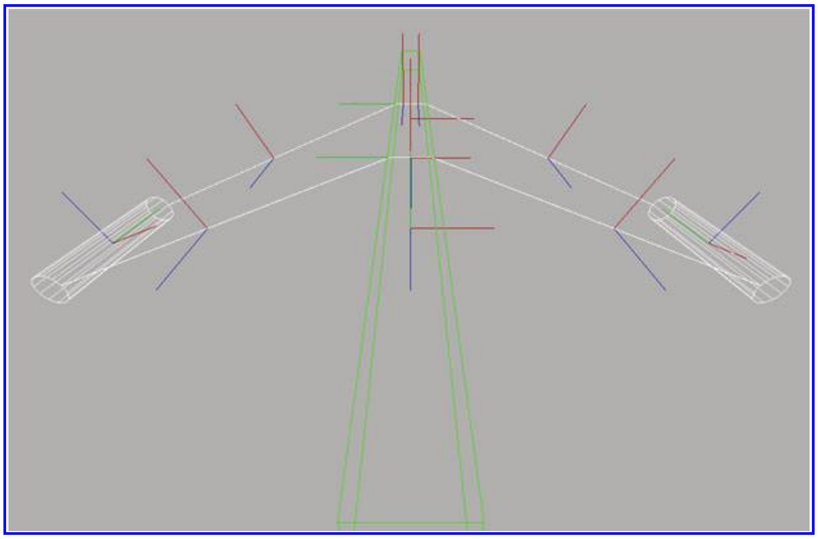

FIG. 4. An analogous situation to Figure 3 simulated in NTRT, except the cable contacts the rod in two places due to its rotation.

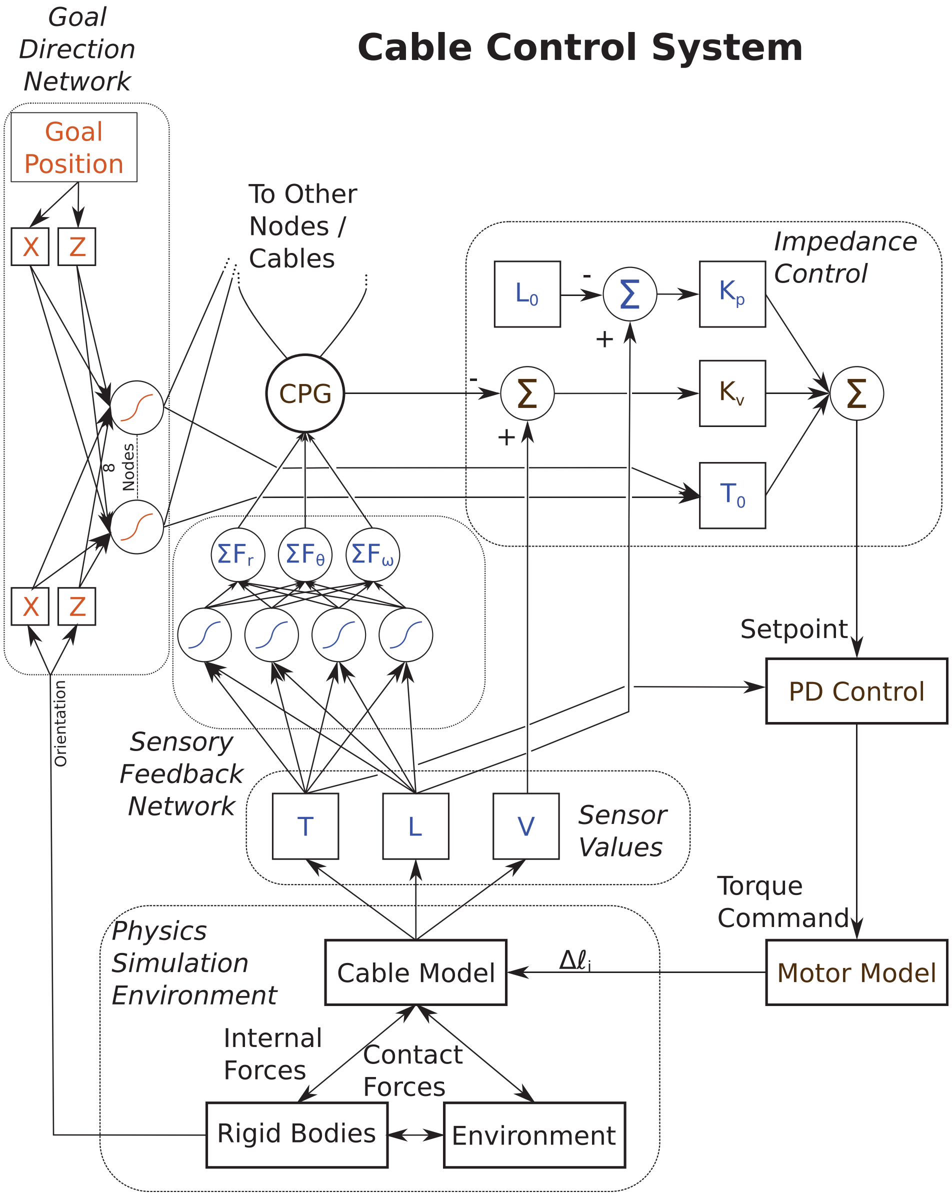

FIG. 5. One unit of the control system, applied to a single cable. Impedance control (*top right*) uses tension, length, and velocity to compute a tension set point. This is the input to a PD controller, which provides a desired torque to the motor model and affects the rest length of the cable. The components in *brown* are sufficient for locomotion, *blue* components provide environmental adaptations. Interactions with the environment are captured through tension and length of the cable, and provided as feedback about the cable's current length, and tension is provided to the central pattern generator (CPG) through an artificial neural network. The sensory feedback network parameters are applied to each cable. Another artificial neural network is used for goal direction. The goal direction network (*orange*) has four inputs, eight hidden nodes, and eight outputs: the tension set point of each cable's impedance controller. Connections between nodes are weighted linear functions, which sum into a sigmoid function. Each final output node is another sigmoid, which combines the outputs of the hidden nodes. The same network is used for each segment, resulting in the goal-directed feedback providing commands like "tense all of the top cables."

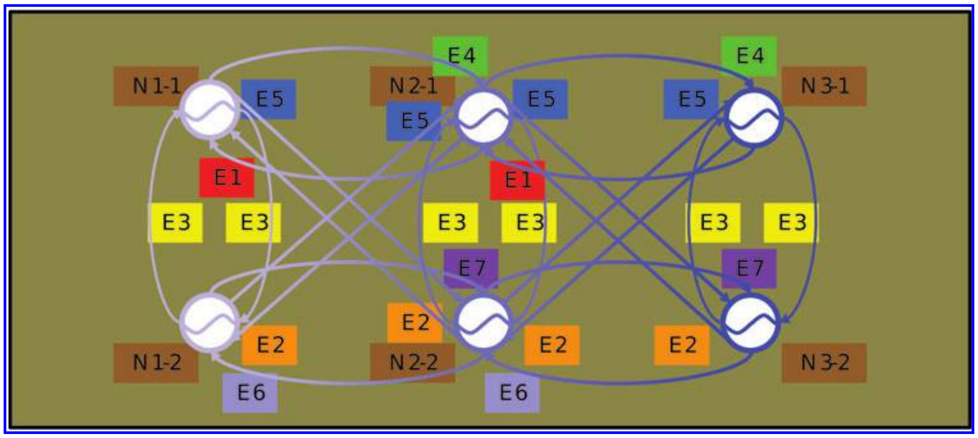

FIG. 6. The coupling rules used on the CPGs. This system leads to $m(3m+1)/2$ possible couplings, where m is the number of nodes. In this case, there is a maximum of 10 edges per node, but these are sorted by geometry into 7 possible edges. This system scales at half the rate of specifying each coupling separately. From Mirletz *et al.*[49]

control, with values $P = 20{,}000$, $D = 5$. These distributed controllers allow for some degree of morphological communication (as defined in Ref.[25]), as the tension of one cable affects other nearby cables.

The cables' motions are coordinated by a model of a central pattern generator, which provides an adaptive trajectory for locomotion. Each cable receives input from a single node of the CPG. The CPG equations used here are a combination of the adaptive phase-coupled oscillator equations of Righetti *et al.*[55] and Gay *et al.*[45]:

$$\dot{r}_l = \gamma(R_i + k_r F_r - r_i^2) r_i \tag{2}$$

$$\dot{\theta}_l = \omega_i + k_\theta F_\theta + \sum_j r_j W_{ij} sin(\theta_j - \theta_i - \phi_{ij}) \tag{3}$$

$$\dot{\omega}_l = k_\omega F_\omega sin(\theta_i) \tag{4}$$

$$V_i = r_i cos(\theta_i), \tag{5}$$

where r_{i} is the amplitude of the wave, θ_{i} is the phase, ω_{i} is the frequency, and V_{i} is the input to the impedance controller. The amplitude [Eq. (2)] is set by convergence parameter γ_{i} and set point R_{i}. The phase relates to connected nodes through weight w_{ij}, phase offset φ_{ij}, and the other node's amplitude r_{j} in Equation (3). The terms k_{r}, k_θ, and k_ω are scalar gains on feedback functions F_{r}, F_θ, and F_ω, similar to[45] providing sensory feedback on length and tension through an ANN (Fig. 5). An ANN was chosen since it can be efficiently tuned as multi-input, multioutput function approximator. Equations are integrated using ODEInt, part of the Boost C++ libraries.[56]

The parameters of the CPG nodes, such as frequency and amplitude, were homogeneous throughout the structure. For couplings that are specified by a weight (±1) and a phase offset ($\pm\pi$), each node is coupled to nodes with shared rigid bodies. Since the rigid bodies are in a chain, there are nodes associated with at most three rigid bodies in any coupling group, and these groups can be repeated for the length of the robot ($m[3m-1]$ couplings), where m is the number of nodes. In addition, symmetric couplings are the same, reducing the number of possible couplings by approximately half to $m(3m+1)/2$. An example with a two-node, three-segment system with seven possible couplings is shown in Figure 6. Since the couplings are defined over a three-segment subsystem, they can be scaled to a spine of any length, allowing parameters tuned on a shorter spine to be used immediately with a longer spine.

Each node of the CPG receives feedback that can modify its amplitude, phase, and frequency, based on the length and tension of the node's associated cable. This local feedback maintains the distributed nature of the control, uses sensors available to existing tensegrity hardware, and allows the trajectory to synchronize with the morphology and the terrain. The CPG will maintain its original amplitude, phase, and frequency without feedback.[45] Feedback is processed through an ANN with two inputs, four hidden nodes, and three outputs. Once tuned, the same parameters are reused for every distributed controller.

Prior work showed that changing the tension offset of the impedance controller can steer a tensegrity spine robot manually.[48] A second ANN generalizes this steering mechanism to the tetrahedral complex shape and updates the tensions automatically. For goal direction, inputs include the compass heading to the goal and the robot's current orientation. The input is processed through eight hidden nodes. The output is the tension offset for each cable, with eight outputs for each segment; output is duplicated for each segment of the robot. Commands to the CPG could also be used for steering,[43] but in practice, these commands conflicted with the rough terrain adaptations.

Machine learning

Tuning a controller for goal direction on rough terrain is a difficult problem, since the controller needs to learn basic locomotion, how to adapt locomotion to terrain, and how to steer toward a goal. While it may be possible to tune all of these features in a single algorithm, we found it effective to partition the problem into motion, terrain adaptation, and goal direction. Our objective function tested the tensegrity spine on three types of terrain: flat ground, sinusoidal hills with an amplitude of 2 cm, and a field of 500 randomly placed blocks. The blocks were 5 cm wide, 0.5 cm tall, and were fixed to the ground within a 200 cm by 200 cm area around the origin. The first trials were evaluated according to distance traveled ''as the crow flies'' in 60 simulation seconds. If multiple terrains were used, scores were averaged.

The first step is to generate basic locomotion. Finding good parameters for motion control is sometimes referred to as a ''needle in a haystack'' problem, as there are many more ways to fail than succeed, leading to a solution space without a clear

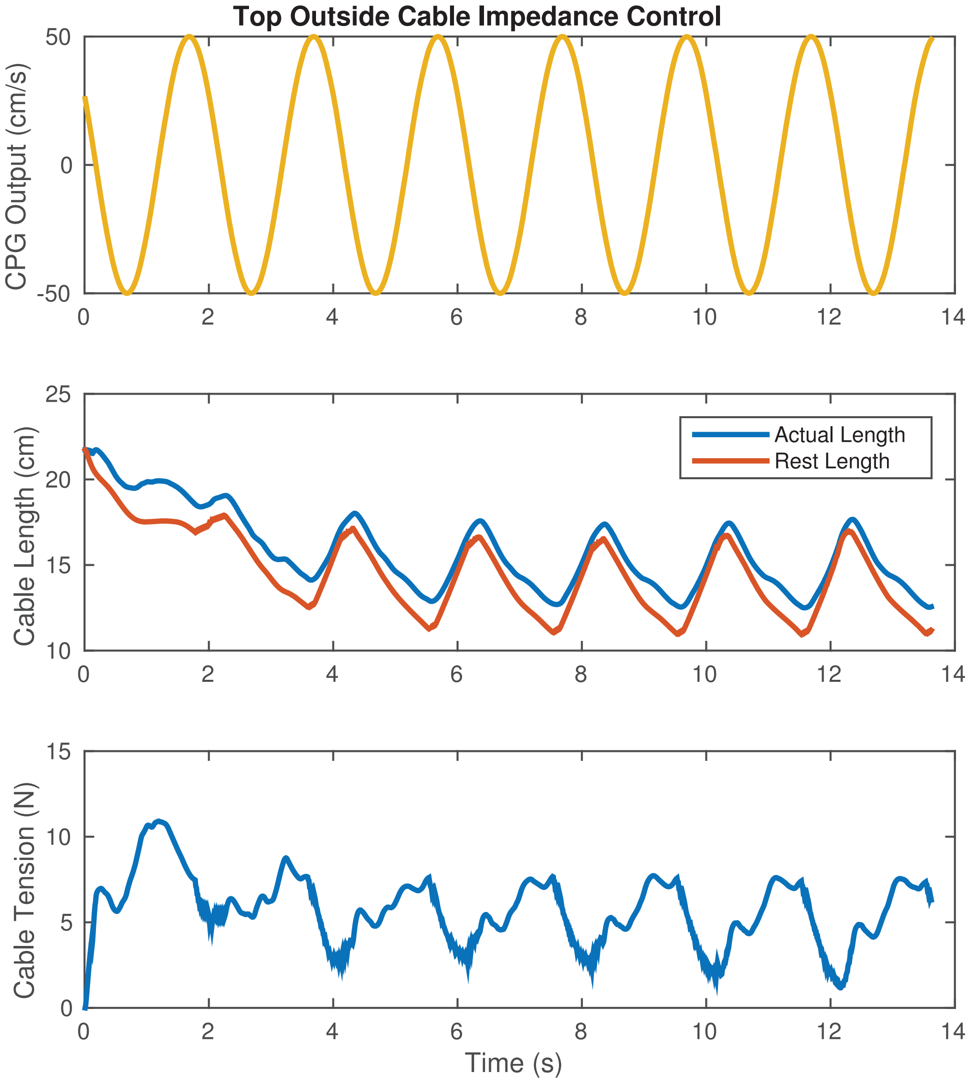

FIG. 7. Tetraspine (38 cm to a rod) crawling over flat ground. The top cable is shown responding to an impedance controller where $T_0 = 35$ N, $K = 300$ N/m, and $B = 50$ N s/m. The tension offset is high enough to prevent the cable from going slack, and the high amplitude of the velocity causes a 6.5 cm oscillation in the length of the cable. Color images available online at www.liebertpub.com/soro

gradient. Monte Carlo search is effective for exploring this type of space, since evaluations can be trivially parallelized. The initial CPG parameters were selected from 24,000 Monte Carlo trials. The eight best were then refined by a Gaussian sampling hill climbing optimization on all three terrains, where random samples are taken near the best results. All CPG parameters were scaled within the learning library to be between 0 and 1, the Gaussian had a mean of zero and a standard deviation of 0.005. The hill climbing step improved the results between 100% and 200% over Monte Carlo, but most of the improvement was on flat ground, indicating the need for feedback to the CPG. This two-part first stage was used for four different spine morphologies in prior work.[49]

The second stage parameterized the sensory feedback ANN for irregular terrain. In this case, the search algorithms are continuing to explore a near effective solution, so a genetic algorithm converges to a solution with fewer trials than Monte Carlo. The genetic algorithm had a population of 60, generating its members with crossover (40 per generation), mutation (5 per generation, standard deviation also 0.005), and elitism (15 per generation). The initial seed was random. Fitness was determined by average score between the three terrain types. The controller at this stage was discussed in Ref.[50]

To produce goal direction, the objective function needs to change to distance moved toward the goal. The final stage trained the initial goal-directed controller on a goal roughly to the left on the hilly terrain, 350 cm away (location B in Fig. 9). A genetic algorithm started tuning the goal-directed network from random parameters (population of 80, 15 generations, mutate 25 with standard deviation 0.03, crossover 15, elitism 40) and finished with fine tuning of the entire system (population 50, 45 generations, mutate 15 with standard deviation 0.01, crossover 10, elitism 25). Each trial took 120 simulation seconds to give the robot sufficient time to get to the goal on hilly terrain. Generalization to multiple goals (A, B, and C) took an additional 26 generations (same as full system evolution) of training on both hilly and flat ground, scores were averaged across these six trials. The final result was a single set of parameters tuned to reach multiple goals on multiple terrains.

Results

When properly tuned, the impedance controller functions similarly to reflexes. Figure 7 shows an example of the tension-based adaptations provided by the impedance controller. One

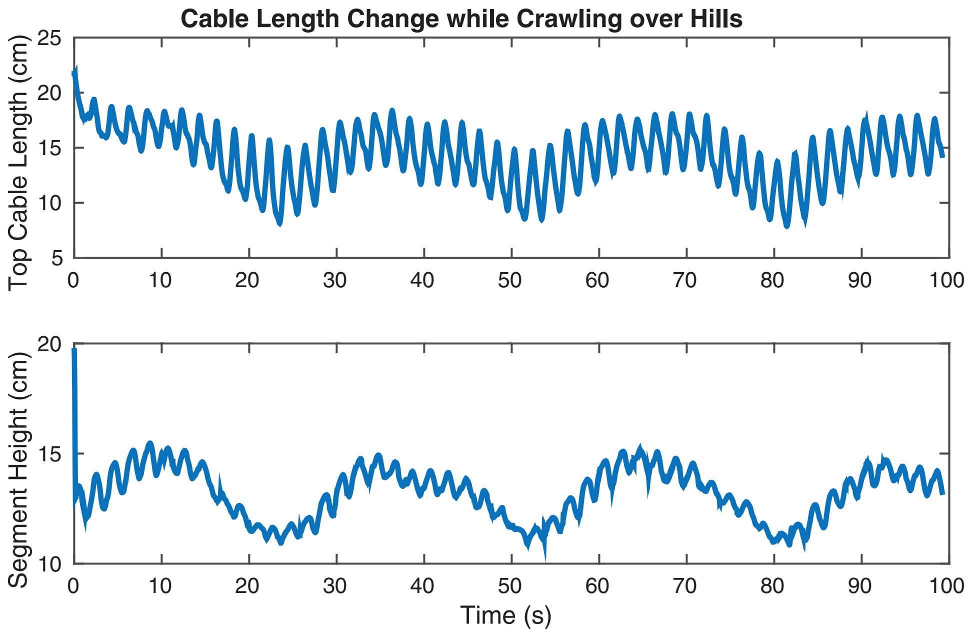

FIG. 8. Tetraspine (38 cm to a rod) crawling over 5 cm bumps. The impedance controller adapts the length of the cable (*top*) to the terrain (visible in the height of the segment, *bottom*), while still tracking the signal provided by the CPG and producing locomotion. Color images available online at www.liebertpub.com/soro

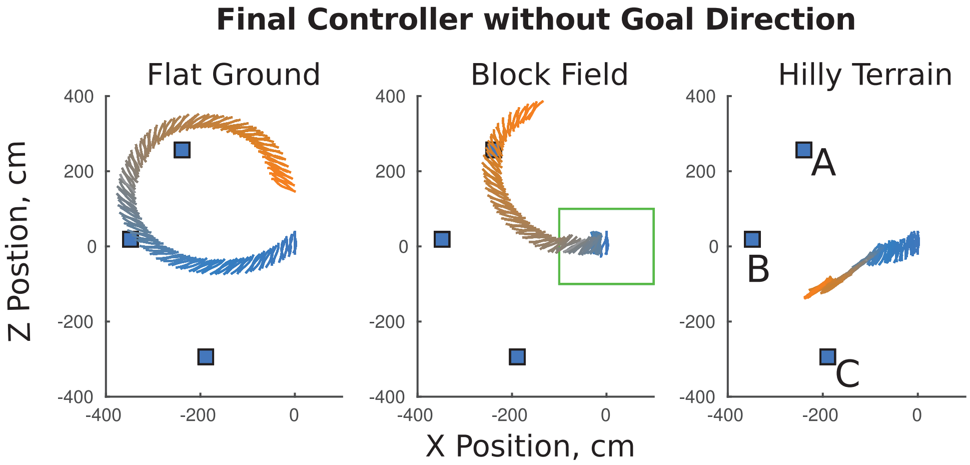

FIG. 9. The final controller on three terrain types without goal direction input. The three *square blocks* indicate the possible goal positions during training (the CPG, feedback function, and impedance controller were all optimized for this task during learning). The *small lines* indicate the positions of the center of mass of each segment of the robot. *Color* indicates the time step, with *blue* at 0 s fading to *red* by the end of the trial at 120 s. With no goal direction, the robot finishes in three different positions after 120 s. Color images available online at www.liebertpub.com/soro

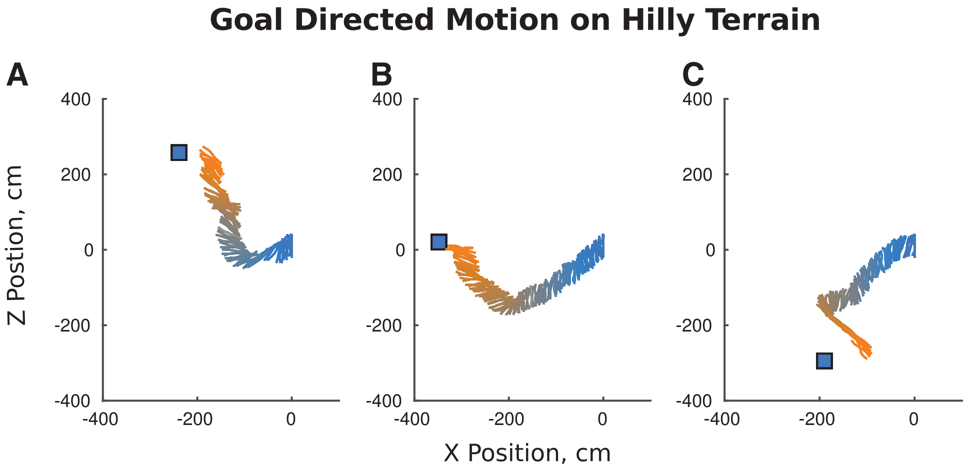

FIG. 10. Goal-directed motion on hilly terrain. The goal position is indicated with the *large blue box*, where **A**, **B**, and **C** indicate the goal positions as in Fig. 9, and the center of mass of each of the robot's segments is indicated with the *colored lines*. The color fade again occurs from 0 to 120 s. Color images available online at www.liebertpub.com/soro

of the top outside cables adapts its length while an analogous spine, Tetraspine, crawls directly forward across flat ground. The sine wave input simulates the output of a CPG and is sufficient for forward locomotion on flat ground and 5 cm hills (Fig. 8). As the robot crawls over the hills, the rest lengths around which the oscillations occur conform to the terrain.

More complex terrain types such as the block field required adaptations to the trajectory.[50] The final tuning of the CPG feedback resulted in the feedback decreasing the CPG frequency from 0.7 to 0.63 Hz and increasing the amplitude by 0.25 (cm/s). These changes align with the intuition that successful locomotion on irregular terrain requires slower, more precise motions. While feedback allows for motion across the terrain, the terrain interactions still change the direction of motion. Figure 9 shows the trajectory of the final controller with only the feedback controller active, no goal direction. The robot tends to move perpendicular to its length, in a clockwise arc.

After training the goal direction network on three goal positions, the robot is capable of directed motion on terrain. Figure 10 shows these results by tracing the positions of the robot's segments over time. If the center of mass of one of the robot's segments passes within 20 cm of the goal block's center of mass, the robot will contact the goal block and achieve the maximum score. The robot reaches the goal at the location B on the hilly terrain (Supplementary Video S1; Supplementary Data are available online at www.liebertpub.com/soro), for location A, the robot comes within 20 cm, and for the location C, it misses by 60 cm.

The geometry of the hills forces the robot to move differently than it would on flat ground. In particular, with the goal at location C, the robot spends a noticeable portion of the trial moving along its length, rather than perpendicular to its length. On hills, the goal-directed controller is about two times faster than the undirected controller,[50] covering 201.9 cm in the first 60 s. Tests on sloped terrains (global ramps) showed that with this controller the simulated robot was capable of climbing up to 7.5° slopes and can reach the goal on small slopes of 2°.

The goal direction is more precise on flat ground. Comparing Figure 9 with Figure 11, the shape changes yielded by the goal direction network change the radius of the arced path taken. To test the generality of the goal direction network, the robot was tested on an additional three goal locations on which it was not trained (locations D through F). In the flat ground trials, the robot hit the goal block in two trained locations (A, B) and two untrained locations (D, E) (Supplementary Video S2). The goal block is heavy enough that hitting the block stops the robot's continued motion, as in the trial with location B in Figure 11 (middle left) and location A (top left) in Figure 15. The longer run time of location F's trial (bottom right) shows that with the proper inputs, the controller can also produce counter clockwise turns. This controller displays its fastest motion on flat ground, moving up to 501.34 cm in 60 s. Images of this gait are shown in Figure 12. However, the optimization did not consider efficiency, so the behaviors designed for irregular terrain led to a high cost of transport of 11.2.

The goal-directed adaptations generalized effectively to the block field, the robot was able to escape in all 12 tests of the goal-directed controller (six locations, two block fields).

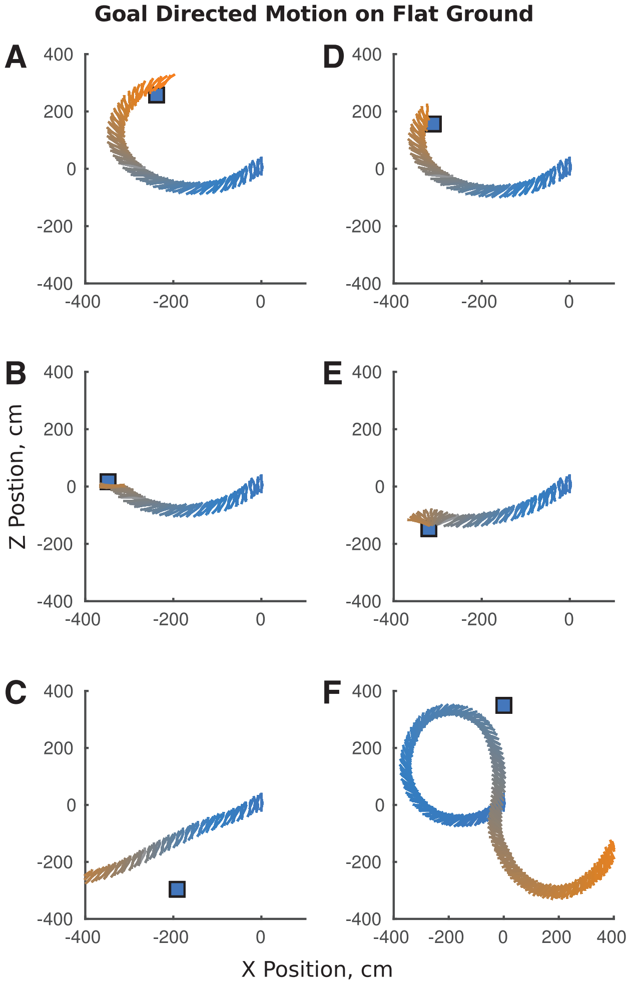

FIG. 11. Goal-directed motion on flat ground. Flat ground and the blocks were tested on three additional locations (**D–F**). For the first five plots (**A–E**), *blue* is 0 s and *red* is 82 s. For location F (*lower right*), *red* is 267 s. The robot contacts the goal block in four out of six locations, two of which were untrained. Color images available online at www.liebertpub.com/soro

Using the goal direction network enables the robot to escape the block field 28 s faster than the controller with no goal information. An escape is shown in Figure 13. Six locations with a single block field are shown in Figure 14 (Supplementary Video S3), and with a different block field in Figure 15. The robot contacts the goal block in six out of 12 trials. Each goal block set produces a success in a case where the controller missed on flat ground (C and F). Thus, the controller reaches all six goal locations in at least one trial.

Discussion

This work presents an adaptive, CPG-based controller for tensegrity spines with many DOF and demonstrated the first simulation of goal-directed motion of a CPG-controlled robot over irregular terrain. For tensegrity robots, this demonstrates the first controller that can cross multiple types of terrain in simulation with a single set of parameters. The advantages of this controller include the stable limit-cycle dynamics of the CPG for reliable yet adaptable cable trajectories, low-level

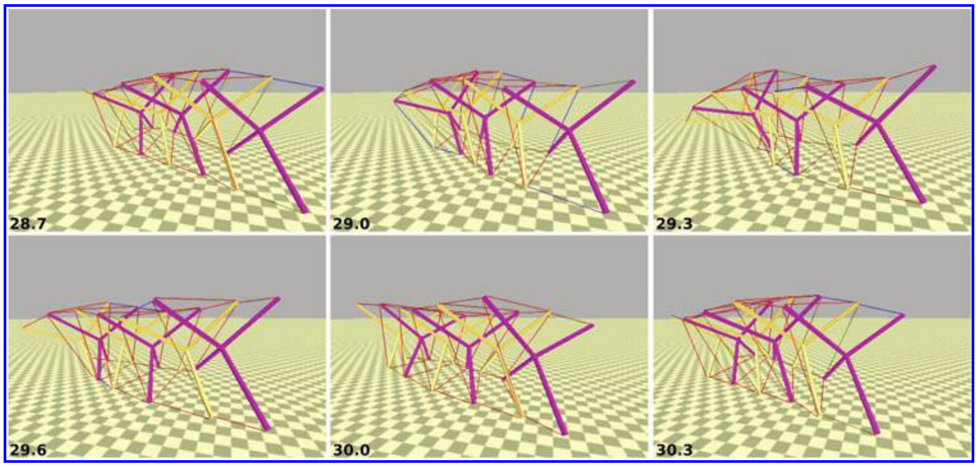

FIG. 12. Images from 28.7 to 30.3 s of the spine crawling on flat ground toward location B (off the image to the left). The gait anchors with the rear segment, lifting its right side, and scoots forward and to the left. This represents one cycle. Color images available online at www.liebertpub.com/soro

impedance control for handling fast perturbations, and control of whole-body behaviors through an ANN. The addition of a compass to our hardware implementation would enable implementation of this control scheme in hardware.[52]

The rhythmic motions produced by the CPG are analogous to the open-loop vibrational control schemes of Khazanov *et al.*[26] and Bohm and Zimmermann,[27] especially after feedback entrains the controller to the dynamics of the structure. As tuned, the CPG produces higher amplitude, lower frequency oscillations than these other controllers, which is expected for the more precise motions required to cross irregular terrain. Moreover, it would be possible to create a CPG through a spiking neural network, as in Rieffel *et al.*[25] We believe that the separation of behaviors (locomotion, adaptability, and goal direction) to different components of the controller clarifies the tuning procedure required to produce goal-directed motion.

Our machine learning procedure is similar in complexity to controllers tuned to have similar capabilities. For example, Gay *et al.* hand tuned their CPG parameters and then tuned feedback using particle swarm optimization.[45] Their controller did not demonstrate turning on terrain. The multistep machine learning procedure actually presents a few advantages. First, the stages provide a fallback position: if a subsequent stage is unsuccessful one does not have to start over from the beginning. Second, the procedure ensures the robustness of the lower level components: the CPG can produce motion on flat ground without feedback or goal direction, a single step learning procedure risks over-fitting. Ultimately, the direct encoding of parameters as used here may not be the most efficient way to learn this behavior. Adapting the learning methods used for continuous time recurrent neural networks to CPGs, such as Hyper-NEAT,[57] may accelerate learning for this type of controller.

With the current tuning, the coupling terms dominate the CPG equations, with feedback providing adjustments to synchronize with the structure and the terrain. Potentially, the

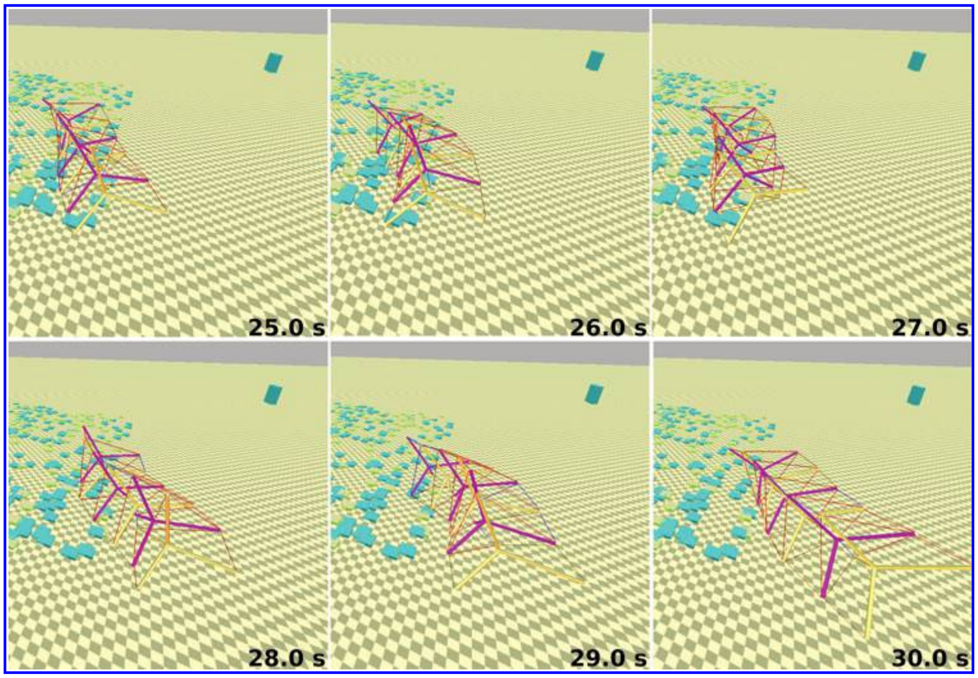

FIG. 13. Images from 25 to 30 s of the trial depicted in the *upper left corner* of Figure 14. The robot successfully departs the blocks and then crawls on flat ground to the goal (the large block in the *upper right corner* of each image). Color images available online at www.liebertpub.com/soro

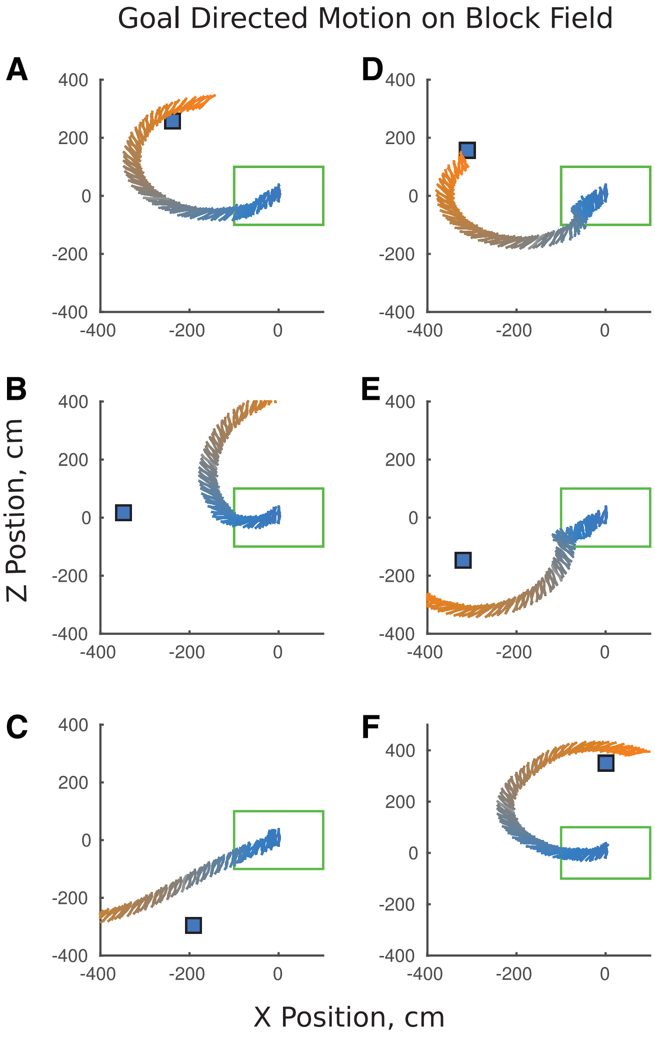

FIG. 14. Goal-directed motion on a block field. All six trials used a block field with the same random seed. The robot contacts the goal in three out of six trials. The trial length was 100 s in all cases. **(A–F)** indicate goal positions as in Fig. 11. Color images available online at www.liebertpub.com/soro

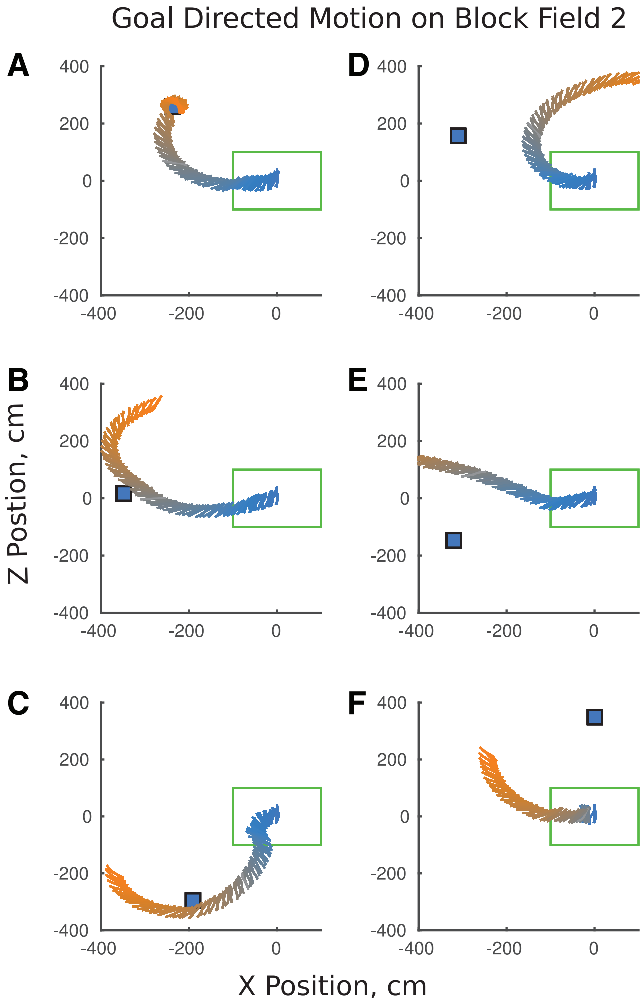

FIG. 15. Goal-directed motion on a second block field. All six trials use the same random seed, which was different from Figure 14. The goal positions **(A–F)**, remain the same. The robot reaches the location **A** again, as well as locations **B** and **C** (*left side*). Color images available online at www.liebertpub.com/soro

CPG could be retuned such that feedback plays a larger role for behaviors such as gait changes, which could be prompted by the tension set points of the goal direction controller. Additional sensors such as contact sensors should also improve performance.

Previously, we primarily conducted machine learning on 12-segment spines.[49] Longer spines displayed more complex behaviors and smoother gaits. This entire algorithm can control spines of that length and longer. When tested on a 30-segment spine (analogous to the number of articulating vertebrae in many vertebrates[58]), the controller is capable of controlling 232 actuators in one-half real time on a single core (Supplementary Video S4). The specific functions learned by the goal direction network are not tuned optimally for additional segments, but still show effective locomotion. Additional optimization could further improve performance, and numerical integration of the CPG could be paralleled for real time or faster computation.

This control algorithm can generalize with its existing symmetry rules to any tensegrity structure that is symmetric along a single axis. New symmetry rules could be devised to extend it to legged tensegrity robots or generally soft structures. The complexity of the controller would depend on the connectivity of the structure. With nearest neighbor coupling (based on cables that share rigid bodies), the work presented here had 23 edges per CPG node, whereas a structure as used in Ref.[24] would only have 10, and the icosahedron would have 14. Since these cases require fewer parameters, the learning algorithms should require fewer trials to converge to a solution. The complexity of the goal direction network would depend on what symmetry rules were used throughout the new shapes and the gaits produced by the CPG. However, the general principle of changing the tension offset of the impedance controller to adjust the shape of the robot and direction of the controller will apply to any soft or tensegrity robot. Thus, this work supports the continued development of flexible robots with many DOF.

Acknowledgments

The authors thank Tom Flemons and Stephen Levin for inspiration on biotensegrity, John Chris Adams for computational support, and Dorothea Blostein for comments on drafts. This work was supported by a NASA Space Technology Research Fellowship (No. NNX11AN15H) and the NASA Innovative Advanced Concepts Program.

Author Disclosure Statement

No competing financial interests exist.

References

1. Majidi C. Soft robotics: a perspective—current trends and prospects for the future. Soft Robot 2014;1:5–11.
2. Lipson H. Challenges and opportunities for design, simulation, and fabrication of soft robots. Soft Robot 2014;1:21–27.
3. Paul C, Valero-Cuevas FJ, Lipson H. Design and control of tensegrity robots for locomotion. IEEE Trans Robot 2006;22:944–957.
4. Gracovetsky S. An hypothesis for the role of the spine in human locomotion: a challenge to current thinking. J Biomed Eng 1985;7:205–216.
5. Schilling N, Hackert R. Sagittal spine movements of small therian mammals during asymmetrical gaits. J Exp Biol 2006;209:3925–3939.
6. Alexander RMN. Why mammals gallop. Am Zool 1988; 28:237–245.
7. Schilling N, Carrier DR. Function of the epaxial muscles in walking, trotting and galloping dogs: implications for the evolution of epaxial muscle function in tetrapods. J Exp Biol 2010;213:1490–1502.
8. Ingber DE, Wang N, Stamenovíc D. Tensegrity, cellular biophysics, and the mechanics of living systems. Rep Prog Phys 2014;77:046603.
9. Wang N, Tolíc-Nørrelykke IM, Chen J, Mijailovich SM, Butler JP, Fredberg J, *et al.* Cell prestress. i. stiffness and prestress are closely associated in adherent contractile cells. Am J Physiol Cell Physiol 2002;282:C606–C616.
10. Ingber DE. Tensegrity ii. how structural networks influence cellular information processing networks. J Cell Sci 2003; 116:1397–1408.
11. Myers TW. Anatomy Trains: Myofascial Meridians for Manual and Movement Therapists. London, UK: Churchill Livingstone, 2001.
12. Turvey MT, Fonseca ST. The medium of haptic perception: a tensegrity hypothesis. J Mot Behav 2014;46:143–187.
13. Levin SM. The tensegrity-truss as a model for spine mechanics: biotensegrity. J Mech Med Biol 2002;2:375–388.
14. Flemons T. The Geometry of Anatomy. www.intensiondesigns.com/geometry_of_anatomy.html (2007) (accessed July 27, 2015).
15. Flemons T. The Bones of Tensegrity. www.intensiondesigns.com/bones of tensegrity (2012) (accessed July 27, 2015).
16. Levin SM. Putting the shoulder to the wheel: a new biomechanical model for the shoulder girdle. Biomed Sci Instrum 1997;33:412–417.
17. Cholewicki J. McGill SM. Mechanical stability of the in vivo lumbar spine: implications for injury and chronic low back pain. Clin Biomech 1996;11:1–15.
18. Scarr G. Biotensegrity: The Structural Basis of Life. Pencaitland, United Kingdom: Handspring Publishing, 2014.
19. Orki O, Ayali A, Shai O, Ben-Hanan U. Modeling of caterpillar crawl using novel tensegrity structures. Bioinspir Biomim 2012;7:046006.
20. Moored KW, Taylor SA, Bart-Smith H. Optimization of a Tensegrity Wing for Biomimetic Applications. Proc. SPIE 2011;6173:617313.
21. Bliss TK, Iwasaki T, Bart-Smith H. CPG control of a tensegrity morphing structure for biomimetic applications. Adv Sci Technol 2008;58:137–142.
22. Bliss T, Werly J, Iwasaki T, Bart-Smith H. Experimental validation of robust resonance entrainment for cpg-controlled tensegrity structures. IEEE Trans Control Syst Technol 2013; 21:666–678.
23. Skelton RE and De Oliveira MC. Tensegrity Systems. New York, NY: Springer, 2009.
24. Rieffel J, Valero-Cuevas F, Lipson H. Automated discovery and optimization of large irregular tensegrity structures. Comput Struct 2009;87:368–379.
25. Rieffel JA, Valero-Cuevas FJ, Lipson H. Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. J R Soc Interface 2010;7:613–621.
26. Khazanov M, Humphreys B, Keat W, Rieffel J. Exploiting dynamical complexity in a physical tensegrity robot to achieve locomotion. Adv Artif Life ECAL 2013;12:965–972.
27. Bohm V, Zimmermann K. Vibration-driven mobile robots based on single actuated tensegrity structures. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013, pp. 5475–5480.
28. Iscen A, Agogino A, SunSpiral V, Tumer K. Flop and roll: learning robust goal-directed locomotion for a tensegrity robot. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), 2014. Chicago, IL; pp. 2236–2243.
29. Iscen A, Caluwaerts K, Bruce J, Agogino A, SunSpiral V, Tumer K. Learning tensegrity locomotion using open-loop control signals and coevolutionary algorithms. Artif Life 2015;21:119–140.
30. Grillner S, Kozlov A, Dario P, Stefanini C, Menciassi A, Lansner A, *et al.* Modeling a vertebrate motor system: pattern generation, steering and control of body orientation. Prog Brain Res 2007;165:221–234.
31. Brooks RA. A robot that walks; emergent behaviors from a carefully evolved network. Neural Comput 1989;1:253–262.
32. Espenschied KS, Quinn RD, Beer RD, Chiel HJ. Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot. Robot Auton Syst 1996;18:59–64.
33. Reeve RE, Webb BH. New neural circuits for robot phonotaxis. Philos Trans A Math Phys Eng Sci 2003;361:2245–2266.
34. Ijspeert AJ. Central pattern generators for locomotion control in animals and robots: a review. Neural Networks 2008;21:642–653.
35. Hellgren J, Grillner S, Lansner A. Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons. Biol Cybern 1992;68:1–13.
36. Daun-Gruhn S, Tóth TI. An inter-segmental network model and its use in elucidating gait-switches in the stick insect. J Comput Neurosci 2011;31:43–60.
37. Ekeberg O. A combined neuronal and mechanical model of fish swimming. Biol Cybern 1993;69:363–374.

38. Matsuoka K. Sustained oscillations generated by mutually inhibiting neurons with adaptation, Biol Cybern, 1985;52: 367–376.
39. Ijspeert AJ, Crespi A. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In: Proceedings of 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 2007, pp. 262–268.
40. Buchli J, Righetti L, Ijspeert AJ. Engineering entrainment and adaptation in limit cycle systems. Biol Cybern 2006; 95:645–664.
41. Righetti L, Buchli J, Ijspeert AJ. Adaptive frequency oscillators and applications. Open Cybern Systemics J 2009; 3:64–69.
42. Taga G, Yamaguchi Y, Shimizu H. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biol Cybern 1991;65:147–159.
43. Ijspeert AJ, Crespi A, Ryczko D, Cabelguen JM. From swimming to walking with a salamander robot driven by a spinal cord model. Science 2007;315:1416–1420.
44. Santos CP, Matos V. CPG modulation for navigation and omnidirectional quadruped locomotion. Robot Auton Syst 2012;60:912–927.
45. Gay S, Santos-Victor J, Ijspeert AJ. Learning robot gait stability using neural networks as sensory feedback function for central pattern generators. In: Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 194–201.
46. Caluwaerts K, D'Haene M, Verstraeten D, Schrauwen, B. Locomotion without a brain: physical reservoir computing in tensegrity structures. Artif Life 2013;19:35–66.
47. Caluwaerts K, Despraz J, Iscen A, Sabelhaus AP, Bruce J, Schrauwen B, *et al.* Design and control of compliant tensegrity robots through simulation and hardware validation. J R Soc Interface 2014;11:20140520.
48. Tietz BR, Carnahan RW, Bachmann RJ, Quinn RD, Sunspiral V. Tetraspine: robust terrain handling on a tensegrity robot using central pattern generators. In: Proceedings of 2013 IEEE/ASME International Conferences of Advanced Intelligent Mechatronics (AIM), Wollongong, Australia, 2013, pp. 261–267.
49. Mirletz BT, Park IW, Flemons TE, Agogino AK, Quinn RD, SunSpiral V. Design and control of modular spine-like tensegrity structures. In: Proceedings of 6th World Conference of International Association Structural Control and Monitoring, Barcelona, Spain, 2014.
50. Mirletz BT, Quinn RD, SunSpiral V. CPGs for adaptive control of spine-like tensegrity structures. In: Proceedings of International Conference on Robotics and Automation (ICRA2015) Workshop on Central Pattern Generators for Locomotion Control: Pros, Cons & Alternatives, Seattle, WA, USA, 2015.
51. Sabelhaus AP, Ji H, Hylton P, Madaan Y, Yang CW, Agogino AM, *et al.* Mechanism design and simulation of the ultra spine, a tensegrity robot. In: Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Boston, MA, USA, August 2015.
52. Mirletz BT, Park IW, Quinn RD, SunSpiral V. Towards bridging the reality gap between tensegrity simulation and robotic hardware. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots Systems, Hamburg, Germany, 2015, pp. 5357–5363.
53. Servin M, Lacoursiére C. Massless cable for real-time simulation. Computer Graphics Forum 2007;26:172–184.
54. Servin M, Lacoursiére C, Nordfelth F, Bodin K. Hybrid, multiresolution wires with massless frictional contacts. IEEE Trans Visual Comp Graphics 2011;17:970–982.
55. Righetti L, Buchli J, Ijspeert AJ. Dynamic hebbian learning in adaptive frequency oscillators. Physica D 2006;216: 269–281.
56. Ahnert K, Mulansky M. Odeint—solving ordinary differential equations in C++. AIP Conf Proc 2011;1389:1586–1589.
57. Auerbach JE, Bongard JC. On the relationship between environmental and mechanical complexity in evolved robots. Artif Life 2012;13:309–316.
58. Dorland WAN. Dorland's Medical Dictionary, Philadelphia, PA; Saunders Press, 1980.

Address correspondence to:
Vytas SunSpiral
NASA Ames Research Center
MS N269-3
Moffett Field, CA 94035

E-mail: vytas.sunspiral@nasa.gov