2013 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics (AIM)
Wollongong, Australia, July 9-12, 2013

Tetraspine: Robust Terrain Handling on a Tensegrity Robot Using Central Pattern Generators

Brian R. Tietz, Ross W. Carnahan, Richard J. Bachmann, Roger D. Quinn and Vytas SunSpiral

Abstract—**In order to produce a new mode of robust robotic locomotion and better understand how vertebrates coordinate motion with a compliant spine, we are developing a modular tensegrity robot inspired by the spine. The robot, called Tetraspine, is composed of rigid tetrahedron-shaped segments connected by six strings. Distributed impedance controllers coupled with central pattern generators (CPGs) generate tunable motion in the structure, making this the first mobile terrestrial tensegrity robot controlled by CPGs to the authors' knowledge. By eliminating rigid joints between segments and increasing compliance in the structure, Tetraspine is robust to perturbations; it traverses several types of irregular terrain successfully in simulation. Experiments in prototype hardware have proven the viability of the impedance controller and overall structure for locomotion.**

I. INTRODUCTION

The complex problem of agile locomotion of a robot can be greatly simplified if the structure and reactive controls of the robot provide a high level of locomotion competence. An inspiring goal is how decerebrated mammals can coordinate complex locomotion behavior without the involvement of their brains [29]. Due to the inherent uncertainty of operating in unstructured natural environments, modern robotic locomotion and manipulation research often focuses on compliant actuation. Tensegrity structures, which model the musculoskeletal system, extend this focus on compliance to the entire structure of the robot, providing desirable qualities such as variable stiffness, robustness to perturbations, and multi-path force distribution. Reactive controls draw inspiration from numerous biological studies showing significant locomotor computation below the brain (for examples and reviews see: cockroaches [31], stick insects [8], and cats [39], [29]), we focus on maximizing the reactive competence of our robots by exploring the combination of compliant tensegrity structures with central pattern generator (CPG) controls [14]. A motivating intuition for pairing tensegrity robots with CPG networks is the similarity in the dynamics of physical forces propagating through a tensegrity structure with the dynamics of control patterns propagating through CPG networks.

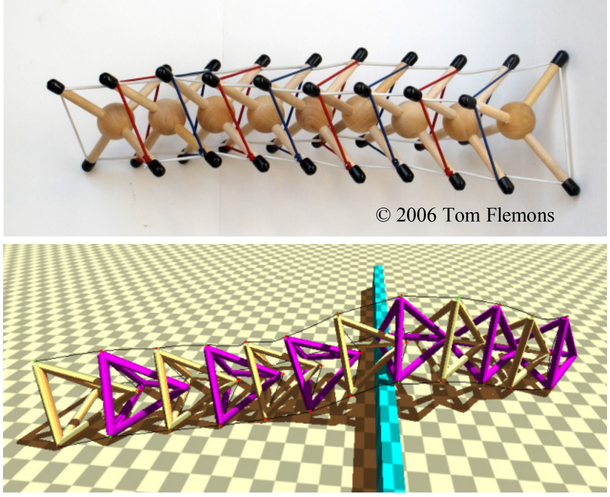
© 2006 Tom Flemons

Fig. 1. Top: A tensegrity spine with eight connections between segments, courtesy of Tom Flemons. Bottom: our abstracted spine model with tetrahedron segments, and six connections between segments. The simulated robot has ascended a wall and is in the process of crawling over it.

Use of tensegrity robots for mobility was initiated in 2004-6 by papers from Masic [20], Aldrich [2], and Paul [27], [28]. Masic's paper included an analytical study of tensegrity based locomotion via periodic waves in a worm-like tensegrity robot; Paul demonstrated mobility both in a physics based simulator and on a hardware prototype. As a result of studies showing the prevalence of tensegrity structures in nature such as cell structure [16] and anatomy [19], [30], and the challenges of controlling tensegrity structures using traditional approaches, the majority of the works in mobile tensegrity robotics have shown biological inspiration in their motivation, using evolutionary algorithms [26], [27], [28], [32], [33], [34], [17], neuroscience inspired CPGs [5], [3], [4], and biomimetic structures such as manta-ray wings [22], or caterpillars [33], [25], [23], [24]. (See also [35], [36], [18], [6], [21] for other works on the locomotion of tensegrity robots) While some work has continued in the analytical understanding of the dynamics of motion for tensegrity mobility [12], the dynamics of contact with the environment are not considered. Since contact dynamics greatly complicate the already difficult task of controller design, most work resulting in simulated or hardware demonstrations of mobility are using non-analytical approaches. This started with Paul's [26], [27], [28] and Rieffel's [32], [33], [34] work which used evolutionary algorithms to discover controllers that resulted in slow crawling and hopping motions. This was followed by Bliss's work using CPGs to control the oscillatory motion of a robotic tensegrity manta ray wing for swimming [5], [3], [4]. Additionally, CPG-like equations have been used by Boxerbaum et al. to control a soft robot

This work was supported by a NASA Office of the Chief Technologists Space Technology Research Fellowship, and NASA's Human Robotic Systems Program

B.R. Tietz student member, IEEE, R.W. Carnahan, R.J. Bachmann and R.D. Quinn, member IEEE, are with the Biologically Inspired Robotics Lab, Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44016, brt6@case.edu

Vytas SunSpiral, member IEEE, is a Senior Robotics Researcher with the Intelligent Robotics Group, NASA Ames Research Center, Moffett Field, CA 94035, vytas.sunspiral@nasa.gov

moving with peristalsis [7]. Boxerbaum's and Bliss' independent work both confirmed the validity of our approach to using CPGs to control mobile terrestrial tensegrity robots. Given that in our case the environmental dynamics are composed of discrete contact events with the ground instead of the continuous dynamics of swimming, we used the CPG network designs from Ijspeert's (non-tensegrity) work on CPG control of salamander walking [15]. Since Ijspeert's work used traditional rotary jointed modules, we modified it by including a lower level impedance controller for each string as developed by Orki's work on a caterpillar inspired tensegrity robot [25], [23], [24], which they have extensively modeled in 2D.

II. Structure Overview

Tetraspine is a tensegrity based on an abstracted model of the spine. Figure 1 compares our model with a more biologically accurate tensegrity spine designed by Tom Flemons [11], and Stephen Levin [19]. We have simplified the model to to use fewer connecting strings and a tetrahedral "vertebrae" which gives us a better ground contact surface and more surface area for mounting motors, controllers and sensors. Despite the change in shape and number of connecting strings, the general topology and dynamics of our design is similar to Flemon's original model. Thus we can investigate the control of a spine where the vertebrae are free to move relative to each other, as seen in animals.

Since the morphology involves three rods touching at each vertex of the tetrahedrons, the robot is a class three tensegrity structure [37]. The tetrahedrons are linked by six strings, three on the inside, three on the outside. The outside strings connect the vertices of each segment's base triangle to the one in front of and behind it, while the inside strings run from the tip of a segment to the base vertices of the one in front of it. Thus, individual strings can apply predictable rotations between segments, whereas strings moved in a coordinated fashion can generate predictable translations. We actively control all six strings using a combination of local controllers and central pattern generators, discussed in Sections III and IV, respectively. We have successfully controlled three degrees of freedom between two segments actively (axial translation, pitch and yaw), the other three degrees are passively compliant. The degree of compliance depends on the local controllers.

III. Local Control

We decided to use impedance control for local control of individual strings. Impedance control was initially designed for robotic manipulators, and has the advantage of being able to specify both a target trajectory and a stiffness along that trajectory [13]. More recently, Orki et al. used a formulation of impedance control in a two dimensional caterpillar tensegrity robot [25]. They added a tension offset to help prevent the strings from going slack, and stability of the control was proven in Orki's 2012 thesis [23]. Due to existing conventions within our simulator (Section V), we defined tension in the strings as positive. We also added a target velocity term similar to the original impedance controller, yielding the following scalar formula:

TABLE I. Parameters of Impedance Controllers on Flat Ground

Parameter	Inside	Outside
T_0	0.02	0.005
K	0.02	0.005
B	0.01	0.001
L_0	.55	.65

$$T = T_0 + K(L - L_0) + B(V - V_0) \tag{1}$$

where the output tension T is the sum of a tension offset T_0, a stiffness gain K multiplies the difference between the actual length L and the rest length L_0, and a velocity gain B multiplies the difference between the string's velocity V and the target velocity V_0, with lengthening defined as positive. Target velocities for each outside string are provided by the output of that string's central pattern generator, which is discussed in Section IV. The inside strings are compared to a zero target velocity, thus B is essentially a damping term. A block diagram is shown in Figure 2.

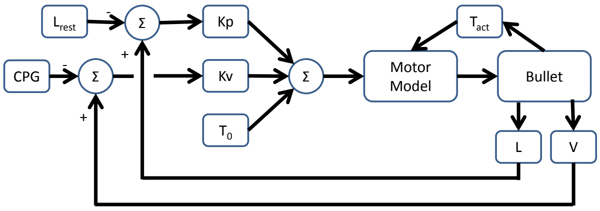

Fig. 2. A block diagram of the impedance controller. Inputs to the controller include L_{rest} as L_0, the CPG output as V_0 and the tension offset as T_0. A force setpoint is computed from the sum and sent to Bullet (our physics simulator), which uses a motor model to determine the change in tension for a specific timestep. In the hardware configuration we use a P controller to calculate motor torque based on the setpoint.

Typical, hand-tuned parameters for our impedance controller on flat ground are shown in Table I. All lengths and distances are given in ratios of lengths, the base length being one segment's rod (Ratio to Segment Length, or RSL). Since stiffness is defined by our simulator to be a unit-less restitution parameter from 0 to 1, our tensions are given in Δ RSL. Results with these parameters are shown in Figure 3. The strain on flat ground typically remains less than 4% of the rest length of the string, and the offset tension is sufficient to keep the string from going slack.

IV. CPG Control

We used a central pattern generator (CPG) network to generate the waveforms for Tetraspine's crawling motion. The CPGs we are using are based on the phase coupled oscillators of Ijspeert et al.'s salamander robot, and use following three differential equations at each node [15]:

$$\dot{\theta}_i = 2\pi v_i + \sum_j r_j w_{ij} sin(\theta_j - \theta_i - \phi_{ij}) \tag{2}$$

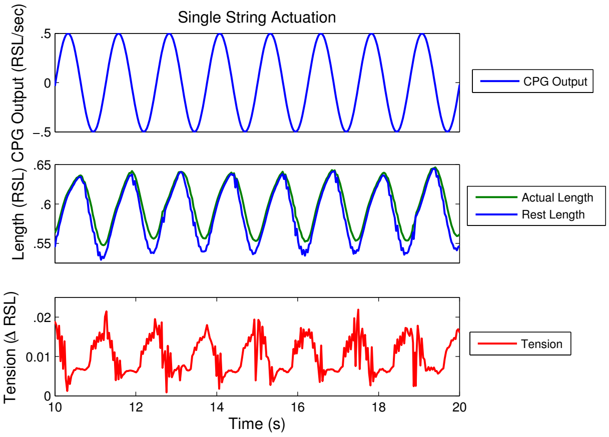

Fig. 3. The results of the impedance controller are shown. The reference velocity is the output of a CPG. The actual length tracks the rest length, and the difference between the two is the non-dimensional tension in the string. The tension is at a maximum when the string changes from shortening to lengthening. Lengths are divided by the length of the segment's rod, giving non-dimensional units of Ratio to Segment Length (RSL).

$$\ddot{r_i} = a_i(\frac{a_i}{4}(R_i - r_i) - \dot{r_i}) \tag{3}$$

$$v_i = r_i(cos(\theta_i)) \tag{4}$$

We then modified the morphology of the CPG network to match the morphology of the structure. We coupled these CPGs in groups of three, one for each outside string similar to Crespi et al.'s BoxyBot controller [10]. In our controller a phase difference of $-\pi/2$ is used between the top and bottom CPGs, and a phase difference of 0 between the two bottom CPGs. Groups are then connected linearly to their corresponding CPG in the ascending and descending segments with a phase difference of $\pi/2$ and $-\pi/2$, respectively. The CPGs at the front of the robot are connected to those at the back, with a phase difference of $\pm 3\pi/4$. A section of this morphology is shown in Figure 4. All parameter values were found by hand tuning, and were chosen for success over a variety of terrains. The resulting pattern of waves is shown in Figure 5. This produces a forward body wave for locomotion. The period of a wave is 1.3 seconds, and a full wave takes approximately 1.6 seconds to propagate through the body.

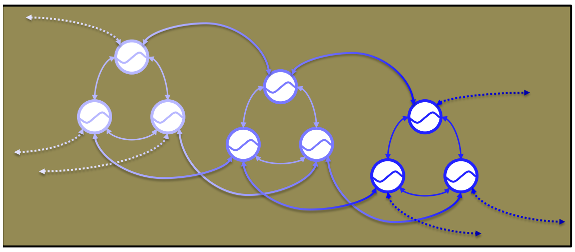

Fig. 4. The three CPG per segment model. Three segments are shown (grouped by color), we typically ran simulations with twelve. The CPGs form a ring, with a larger phase offset between the first and last segments.

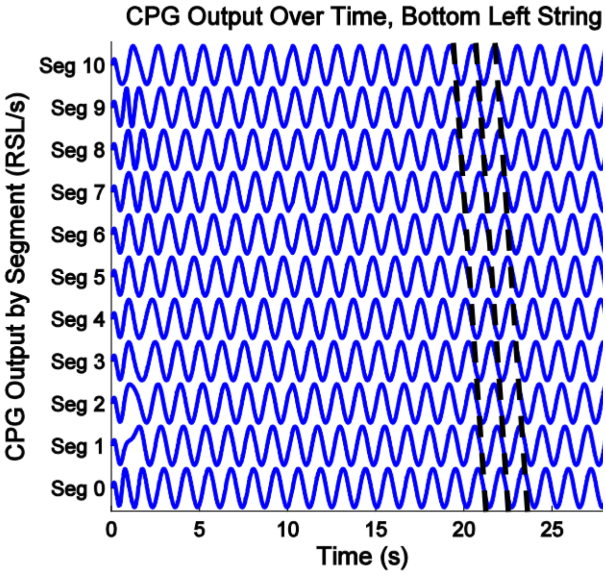

Fig. 5. The output of the CPGs for the bottom left segments. Zero is the head of the robot. The transient behavior can be seen between t=0 and t=5, with steady state waves propagating from back to front (shown by the dashed black lines) thereafter.

V. TENSEGRITY SIMULATOR

Since our primary objective with simulation was to determine control algorithms for tensegrity structures, we decided to build our simulator on top of the open-source real-time Bullet Physics Engine, implemented in C++ [9]. Bullet was chosen because of its built in support for soft-bodied physics, and has been used previously in tendon-driven robotics simulators such as Wittmeier et al.'s CALIPER software [40].

Strings are represented in Bullet as a set of nodes with Hooke's law like linear stiffness between them. We represented strings on our robot using 15 nodes. Overall string stiffness in Bullet is parametrized both through a coefficient of restitution and through the number of iterations that the soft-body solver runs - we found acceptable performance with maximum restitution (1) and 200 position solver iterations. Since we are designing based on the controls in this simulator, we chose to leave the actuation method abstract. Thus, we change the string lengths by changing the rest lengths of these nodes, and then interpolate the positions and overall mass proportionally. To enforce additional realism in the simulator, we prevent rest length of the strings being lengthened further when they are stretched more than 25%, as an upper limit on the hypothetical motor force. The actual length is still free to change based on the dynamics of the attached bodies. We also cap the actuation velocity and acceleration at a percentage of segment length in our motor model.

Impedance control is accomplished with the algebraic equations described in Section III, using the simulated values for rest length and actual length of the strings as sensory inputs. CPG output is generated through numerical integration using the C++ library ODEInt [1]. The simulator runs in real time (one clock second equals one simulation second) at 60 Hz. We collected data on a computer running Ubuntu 12.04 on an Intel Core i7$^{\text{TM}}$, running 8 cores at 3.4 GHz, and with 16 GB of ram, though those are beyond the minimum specifications required for real time simulation. As of this writing, we are in the process of open-sourcing the simulator. Please contact the authors for current information.

VI. SIMULATION RESULTS

A. Flat Ground

With the impedance controllers and CPGs implemented in Bullet, Tetraspine moves in a crawling gait similar to a rectilinear wave. Segments are picked up, moved forward and then placed on the ground each cycle. 1.5 cycles of this gait are shown in Figure 6. Similar motions have been produced in a number of non-tensegrity robots previously, see the section on crawling gaits in Transeth et al.'s review [38]. Each wave travels through the body from back to front, propelling the robot forward at an average of 2 body lengths per minute. The twelve segment robot has a rest length of 6.6 RSL, so the overall speed is 13.2 RSL/min. Thus, if the length of one rod was 30 cm, the speed would be 6.6 cm/sec.

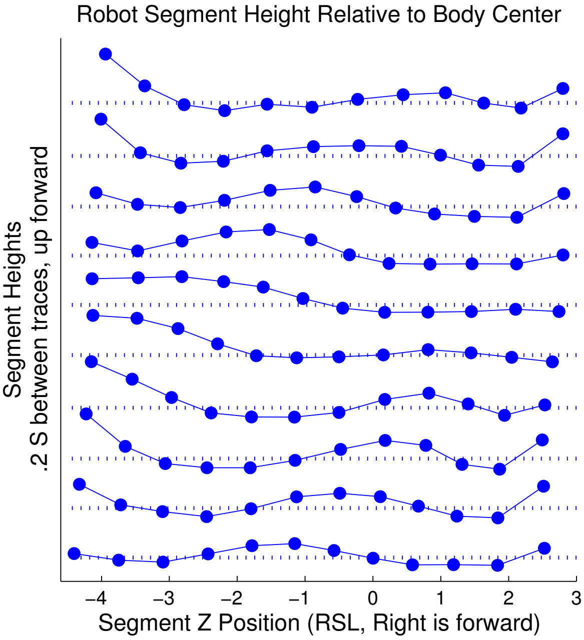

Fig. 6. The pattern of waves generated by the CPGs and impedance controllers over flat ground in the simulator. The body shape viewed from the right side is shown, with the front of the robot on the right of the diagram. The median height of the robot at each timestep is indicated by the dotted lines, and the large dots are individual segments. See the supplementary video for real time footage of simulated locomotion.

B. Steering

The direction of locomotion can be adjusted by tuning the forces in the tensegrity structure. Ijspeert et al. generated turning with a difference between the input parameter d along each side of the CPG network, changing the amplitude and frequency of the oscillations [15]. We found it was more effective in our structure to adjust the offset tensions of the impedance controllers. If the impedance controller's tension offset T_0 is higher on the left inside and outside strings than on the right side, we get a curve similar to the top picture in Figure 7. Tetraspine will then start turning toward the left. That approach can be reversed for right turns. Similarly, we can increase the tension in the top outside strings to get rearing behaviour, which is useful to prepare for climbing. Currently the tension is increased uniformly throughout the body, future work will explore increases for specific segments to produce more body shapes.

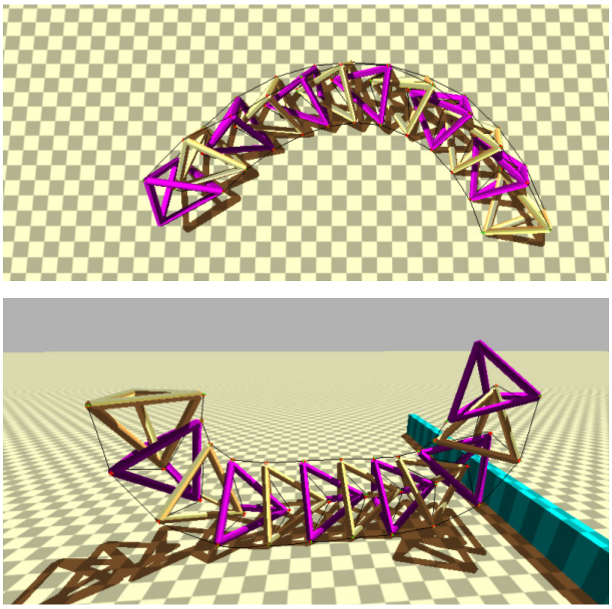

Fig. 7. Overall body shape is controlled controlled by adjusting string tensions. Top: Tetraspine executing a left turn. Bottom: The simulated robot rearing before climbing a wall.

C. Irregular Terrain

We tested Tetraspine on a variety of terrains in Bullet and it proved quite capable of traversing them with standard Coulomb friction. The only parameters that were changed between terrains were the tension offsets of the impedance controllers. With the default wave pattern and tension values the robot is capable of traversing a "bumpy" terrain composed of hills and valleys, and flat terrain that includes a 7.5 degree ramp. The average speed in the bumpy terrain is 0.81 body lengths per minute, the speed on the ramp largely depends on the robot's position on the ramp. These terrains are shown in the top and middle frames of Figure 8. With an operator changing the tension offsets as discussed previously, the robot can crawl over a wall 80% of the height of a single segment (Figure 1, and 7) and a series of randomly placed blocks (Figure 8 bottom picture). One limitation of our simulation is that we have uniform friction in both directions of movement, causing the robot to slip backwards when climbing slopes. Biology solves this by providing asymmetrical friction (via scales, fur, or feet) and we expect our performance to improve in hardware as we have the option to add directional friction to the segments. Footage of the simulated robot on these terrains, with the exception of the ramp (given its three minute duration), is available in the supplementary video.

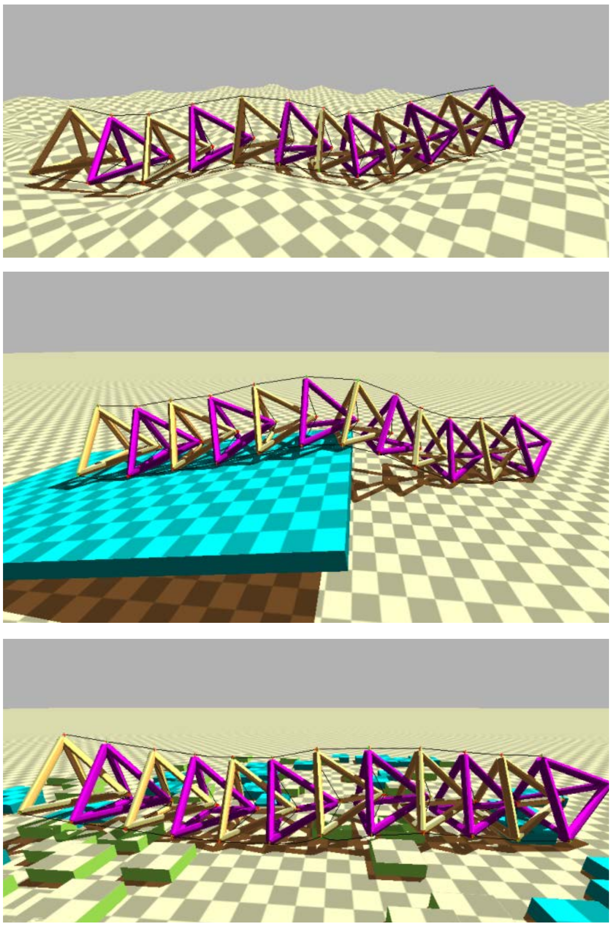

Fig. 8. Tetraspine moving across: Top: 0.4 RSL high hills, Middle: A 7.5 degree ramp, Bottom 0.1 RSL high blocks.

A major reason for the successful adaptations shown is the built-in compliance of the impedance controlled strings. Since the impedance controllers are tracking a velocity reference, they are able to continue oscillating through whatever rest length is required by the terrain, within the constraints specified by the controller. This is best illustrated by the performance on bumpy terrain, as shown in Figure 9. The top strings get longer when nearby segments are on top of a hill, and shorter when in a valley. In addition to making that adaptation to the terrain, the top outside strings are continuing to track the reference signal given by the CPGs. An inside string is also shown, reacting to the motions of the outside strings and the terrain.

VII. PROTOTYPE HARDWARE

In order to verify our simulation results, we are constructing a six segment Tetraspine prototype in hardware. Two segments have been completed and tested as of this writing. Our current prototype is made of carbon fiber segments, which measure the equivalent of a 28 cm rod. The segments are connected with strings that are spooled and unspooled with Pololu 12V 50:1 DC motors with encoders. All components are mounted directly to the carbon fiber using screws and nuts. The prototype is controlled by an Arduino Mega R3, and has a mass of 1.84 kg without batteries.

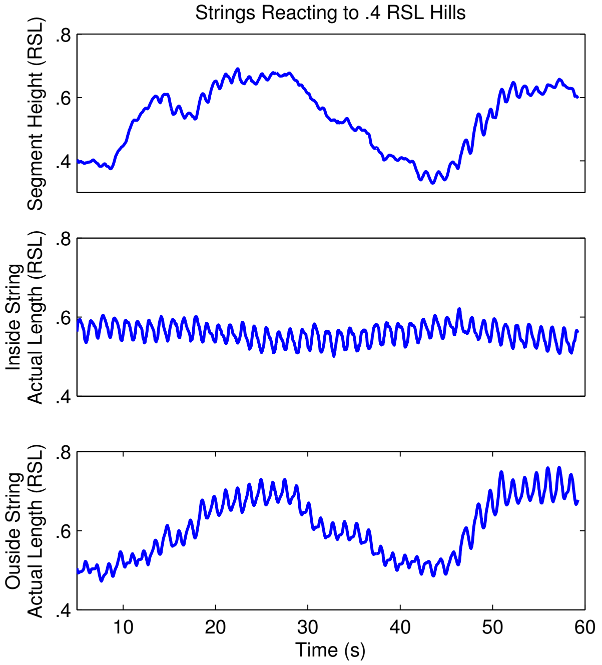

Fig. 9. Tetraspine crawls across a series of 0.4 RSL hills. The terrain can be seen in the center of mass position of a tetrahedron segment from the middle of the robot (top). The actual length traces of the top inside and outside strings from that segment show that the impedance controller can continue tracking a signal while the body conforms to the terrain.

A. *Stretch Sensors*

Tension in the strings was a major piece of sensory data in the simulator. However, many off the shelf stretch sensors are too elastic for the forces we need to apply through the strings, rubber sensors will easily stretch to 50 or 100% of their rest length. Thus, we opted to use in line stretch sensors by knitting conductive thread with yarn, an example is shown in Figure 10. This creates a resistance that changes non-linearly, for example, a 6 cm sensor changes from 47.6 Ohms to 25.4 Ohms over a 2 cm stretch. The stiffness was also non-linear; the first cm of stretch accumulated .7 N of force, while the second cm accumulated an additional 11.3 N. The length of the knitted portion was varied based on the application of the stretch sensor. We ran 5 Volts over these sensors and after the analog to digital conversion had a resolution of approximately 80 units over the 2 cm length change.

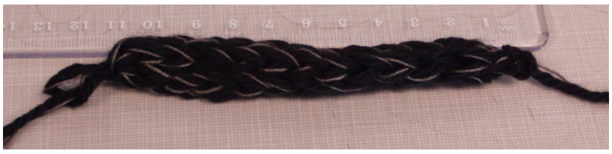

Fig. 10. A knit stretch sensor at slack length. The conductive yarn is silver, the actuator yarn is black. Units on the ruler are centimeters.

B. Control Code

The hardware control code is similar to the simulated controller. The stretch sensors provide an actual tension to reference. An offset tension, the encoder information and a sine wave then provide a setpoint, and the tension is controlled with a P controller, which also normalizes the output. This output is then translated into a motor speed. We use open source code provided by Pololu, the motor manufacturer, to provide a PWM signal to each motor's H-Bridge.

The major differences with the simulator are that the velocity reference is not yet used in the local controller, and a sine wave reference is used instead of a CPG. The major advantages of a CPG will become most apparent when more segments are added, until then a sine wave is a good approximation for two segments. In the prototype this sine wave is applied to the inside strings in addition to the outside, with a phase offset of π between them.

C. Results

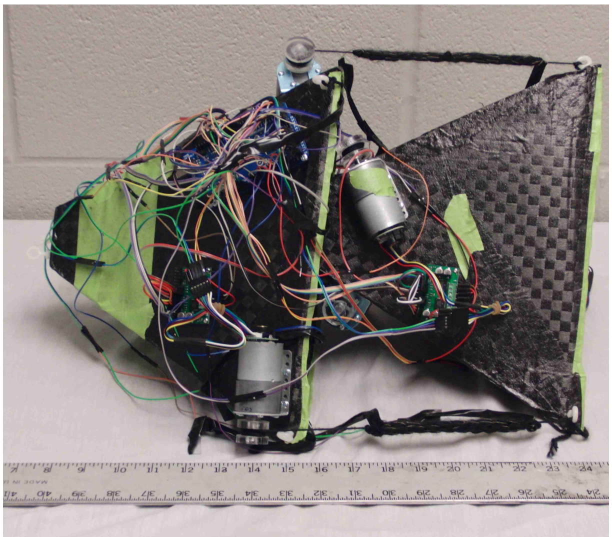

Fig. 11. Our current Tetraspine hardware with carbon-fiber tetrahedrons, six DC motors and a microcontroller. Units on the ruler are inches.

The current two segment Tetraspine is capable of moving the rear segment relative to the front segment in an oscillatory fashion, as shown in the supplementary video. As expected with only two segments, the rear segment has trouble anchoring the front segment, but we anticipate that the weight of additional segments will solve this issue. The important insight from this stage of fabrication is that the low level impedance control functions, and that all the required sensing and actuation, which was abstracted in the simulator, can be successfully implemented in hardware. The distributed controls we designed should help minimize complexity since we can use one Arduino per segment as we scale up. Each segment will then only need to communicate the phase information of the CPGs.

VIII. CONCLUSIONS AND FUTURE WORK

A. Conclusions

Tetraspine shows the power of combining a compliant tensegrity structure with a CPG-driven compliant control system for achieving a high level of reactive locomotion competence in a robot. This reactive competence enables robust locomotion in natural terrains by reducing the need for detailed sensor based modeling and world knowledge in order to plan and execute actuation decisions. Further, we expect that this approach will be robust to real world challenges such as unexpected soil slippages, rocks that roll or shift when traversed, and hidden terrain features obscured by grass or other natural soft coverings which limit pre-planning for locomotion.

While Tetraspine bears strong resemblance to a snake robot, our long term goal is legged locomotion. While most legged robotics research focuses on the design and control of legs, which are then bolted to a rigid central chassis, we are starting by understanding spine dynamics and control. As any athlete, dancer, or martial artist knows, powerful motion starts at the core (i.e. the spine), and moves out to the periphery (i.e. limbs). We expect that our compliant spine design will be an ideal system for reactively integrating the ground reaction forces experienced by limbs when traversing complex terrain. Current approaches, which use a large rigid box for the torso, result in large forces (magnified by leverage) to be reflected between attachment points of the limbs. While our work is leading towards more robust and competent robots, we also hope that it will result in a better understanding of the human spine and the neuromechanics of motion which are critical to our wellbeing.

B. Future Work

Our next step is to complete enough hardware segments to demonstrate locomotion over all the terrains explored in simulation. Future work on the simulator will include determining different sets of parameters for the CPGs that generate different gaits. We also hope to provide additional sensory information, such as ground contact, so that the CPG signals can actively adapt to the terrain or select an appropriate gait. We intend to look into more individual segment controls, like propagating a discrete deformation through the body to climb over objects more efficiently. In addition to being a viable robotic platform on its own, we also anticipate our findings with Tetraspine could be useful to future quadruped and biped robots, as rigid torsos could be replaced with a variably compliant, actuated spinal structure. Likewise, future "snake" versions may combine the spinal designs and controls explored here with a compliant soft belly for improved traction.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge Tom Flemons and Stephen Levin for their explorations of biotensegrity, many inspirational conversations, and support. They would also like to thank Heather Lemire for assistance with stretch sensor construction, and the members of the CWRU Biologically Inspired Robotics Lab for their continuous help and support. Finally, we would like to thank Terry Fong and members of the NASA Ames Intelligent Robotics Group for all their support and encouragement.

References

[1] K. Ahnert and M. Mulansky. Odeint - Solving ordinary differential equations in C++. In *AIP Conf. Proc. 1389*, pages 1586–1589, 2011.

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

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

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

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

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

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

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

[9] E. Coumans. Bullet physics library. http://bulletphysics.org/wordpress/, December 2012.

[10] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert. Controlling swimming and crawling in a fish robot using a central pattern generator. *Autonomous Robots*, 25(1-2):3–13, December 2007.

[11] T. Flemons. The geometry of anatomy. http://www.intensiondesigns.com/geometry_of_anatomy.html, 2007.

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

[13] N. Hogan. Impedance control: An approach to manipulation: Part I - Theory. *Transactions of the ASME*, 107(March 1985):1–7, 1985.

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

[15] A. J. Ijspeert, A. Crespi, D. Ryczko, and J. M. Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science (New York, N.Y.)*, 315(5817):1416–1420, March 2007.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[38] A. A. Transeth, K. Y. Pettersen, and P. Liljebäck. A survey on snake robot modeling and locomotion. *Robotica*, 27(07):999, March 2009.

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

[40] S. Wittmeier, J. Michael, K. Dalamagkidis, and M. Rickert. CALIPER : A Universal Robot Simulation Framework for Tendon-Driven Robots. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1063–1068, 2011.