Design and Control of Modular Spine-Like Tensegrity Structures

Brian T. Mirletz[1], In-Won Park[2], Thomas E. Flemons[3], Adrian K. Agogino[4], Roger D. Quinn[1] and Vytas SunSpiral[2]

Abstract

We present a methodology enabled by the NASA Tensegrity Robotics Toolkit (NTRT) for the rapid structural design of tensegrity robots in simulation and an approach for developing control systems using central pattern generators, local impedance controllers, and parameter optimization techniques to determine effective locomotion strategies for the robot. Biomimetic tensegrity structures provide advantageous properties to robotic locomotion and manipulation tasks, such as their adaptability and force distribution properties, flexibility, energy efficiency, and access to extreme terrains. While strides have been made in designing insightful static biotensegrity structures, gaining a clear understanding of how a particular structure can efficiently move has been an open problem. The tools in the NTRT enable the rapid exploration of the dynamics of a given morphology, and the links between structure, controllability, and resulting gait efficiency. To highlight the effectiveness of the NTRT at this exploration of morphology and control, we will provide examples from the designs and locomotion of four different modular spine-like tensegrity robots.

I. INTRODUCTION

Most robots today rely on a combination of prismatic and rotary joints, both in their chassis and their limbs. Compliance is then built along and around these joints. In contrast, tensegrity robots are joined together in a tensile network of cables, and compliance extends to the entire body [21]. This paradigm makes a good fit for biomimetic structures, capturing the reality that bones are not rigidly hinged to each other, but have a ability to move relative to each other to some extent. Thus, we expect that tensegrity robots will eventually start improving the robots' abilities to adapt to their environment at the structural level. Of particular interest is a tensegrity model of the spine, since it could either function as a snake-like robot as in our prior work [27], or integrate the contact forces of legged locomotion [8].

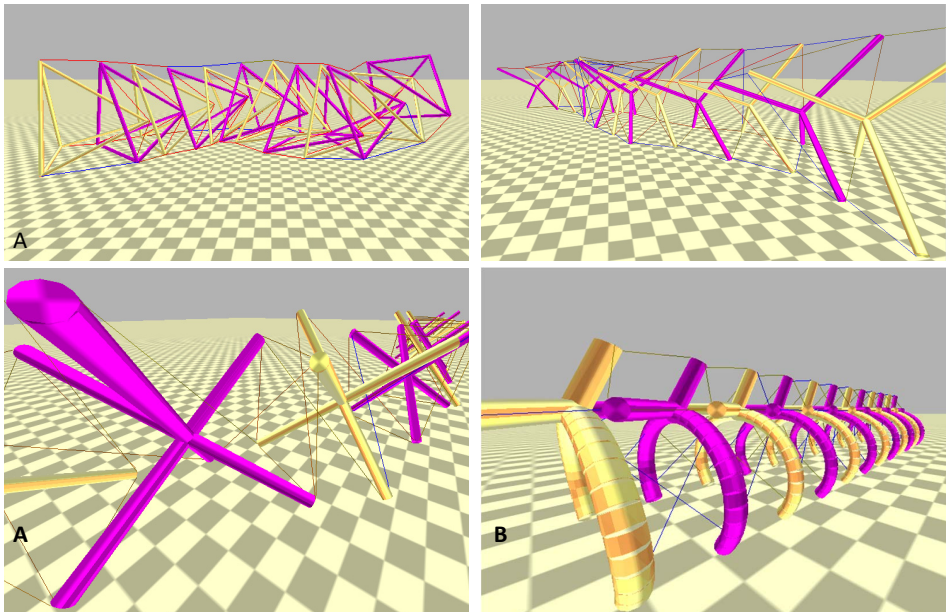

Fig. 1. Four different tensegrity spines tuned using our metholodolgy. A: Tetraspine, with tetrahedral segments and six strings between each segment. B: Tetrahedal complexes for segments, with eight strings connecting each segment. C: Octahedral complexes for segments, with four passive and four active strings between segments. D: A spine with rigidly connected ribs connected by seven active strings and four passive strings.

This work is supported by a NASA STMD Space Technology Research Fellowship, and NASA's Game Changing Development: Human Robotic Systems Program

[1]Brian T. Mirletz and Roger D. Quinn are with the Biologically Inspired Robotics Lab, Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44016, brt6@case.edu

[2]In-Won Park (Oak Ridge Universities) and Vytas SunSpiral (SGT Inc.) are members of the Intelligent Robotics Group, NASA Ames Research Center, Moffett Field, CA 94035, vytas.sunspiral@nasa.gov

[3]Thomas E. Flemons is the founder of Intension Designs Ltd, Saltspring, Island BC Canada,

[4]Adrian K. Agogino (UCSC) is with the Robust Software Engineering Group, NASA Ames Research Center, Moffett Field, CA 94035,

Published in Proceedings of The 6th World Conference of the International Association for Structural Control and Monitoring (6WCSCM), Barcelona, Spain, July 2014.

Tensegrity structures have already been proposed as models for biological organisms, on levels from the cell's cytoskeleton [15] to elements of human anatomy such as the spine [18], and the shoulder girdle [17]. With a few exceptions [3], [20], until now most biotensegrity structures have been unactuated physical models [9], [10]. In order to evaluate their viability as biomechanical models, we need a method to analyze their motion when actuated. Evaluation in a physics simulator is a natural choice since the morphology of a structure can be iteratively changed with a few keystrokes, but this still leaves an open question of how to determine a controller. Controllers have previously been developed by machine learning [22], [23], [24], [16]; to our knowledge thirty two actuators have been the most trained by prior methods [6]. Biological systems make use of far more actuators, such as the 640+ skeletal muscles in the human body [11]. We show that a central pattern generator (CPG) based on phase coupled oscillators can be used with random search algorithms to quickly generate effective locomotion in four different tensegrity spines, ranging from forty-four to eighty-eight actuators.

II. TETRASPINE

Our initial morphology, Tetraspine (Figure 1A), uses tetrahedrons as 'vertebrae', with six strings connecting each segment. Three strings extend from each outer vertex to the segment behind it, and three strings extend from the tip of each segment to the outer vertexes of the segment in front of it. Convex tetrahedrons for segments also allow a lot of room for mounting motors, sensors, and control boards for a hardware implementation. Twelve segments provided a good balance between computational complexity and flexibility of behavior, and was the number used on all simulated spines. With eleven connections, this results in sixty six actuators for this platform.

The arrangement of the strings allows for active control of pitch, yaw, and axial translation, and each string has a clear line of action within these degrees of freedom. Adjusting the tension in the outside strings allowed us to steer the direction of locomotion, and rear to cross obstacles [27]. Our prior work used hand tuned parameters for controllers which are discussed in Section III-B. Hand tuning these parameters provided for effective locomotion in simulation and hardware, but is too time intensive to rapidly develop and evaluate new structures. Therefore we have developed additional tools for our simulator, and have developed controllers for three more morphologies as examples.

III. SIMULATION

A. NASA Tensegrity Robotics Toolkit

Our simulator, the NASA Tensegrity Robotics Toolkit (NTRT), is built on the Bullet Physics Engine, version 2.82. NTRT is available as an open source platform for the design and control of tensegrity structures[1]. The toolkit includes a set of builder tools for specifying rods and strings as a set of points in Cartesian coordinates. Structures built out of these rods and strings can be specified as a tree of substructures, and can be rotated and moved as necessary. The NTRT also includes libraries for controllers such as Central Pattern Generators (Section III-B) and a machine learning framework, which allows users to specify their own learning algorithms, and is discussed more in Section III-C.

For this paper, simulations were run at a fixed timestep, typically 1000 hz. By decoupling physics updates from rendering, rendering can still be performed in real time (60 hz) on a modern computer (Intel Core^{TM}i7), or simulation can be faster than real time without rendering.

For strings, instead of the default Bullet softbodies as in our previous paper [27], we used a two point linear string model using Hooke's law forces with a linear damping term:

$$F = -kX - bV \tag{1}$$

Unless specified otherwise, strings were simulated with a stiffness of 1000 N/m, and a damping of 10 N-s/m. Our simulation with this string model was validated by our prior Planetary Lander work to be accurate within 15 mm for a robot with 1 m struts when performing a roll [6].

Strings were actuated by changing their rest length, changing the X term in equation 1 and exerting forces on the rods. In hardware, this typically means a DC motor spooling or unspooling the string. The actuators in this paper's simulations were limited kinematically to a maximum velocity of 24 cm/sec, and a maximum acceleration of 10,000 cm/sec2 (approx. 10 times gravity). This is possible for a motor with spool of a 1.15 cm radius, with a rotor mass of 200 grams, a max torque of 1.2 N-m, and a max speed of 200 RPM, which matches our prior hardware. Additionally, we did not allow the rest length of the strings to shorten further if the tension was above 70 Newtons. This is a fairly conservative set of assumptions, as our ongoing hardware work uses motors with 3N-m of torque [4]. The one disadvantage of this string model is it does not include contact with the environment. Future work will restore the contact dynamics.

[1]Additional information about NTRT can be found at `http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/ntrt/`

B. String Control

With a straightforward way to build and model physical systems, we now need a way to control their actuators. Biological systems tend to use hierarchical control schemes, with local reflexes and minimal descending commands [13]. We adopted this in our control structure by using impedance control as a reflex rule. Impedance control was first adapted for tensegrity by Orki et al. for their caterpillar-like tensegrity structure [20], and we made modifications to account for descending commands in the following equation:

$$T = T_0 + K(L - L_0) + B(V - V_0) \tag{2}$$

Where T is the output tension, T_0 is a tension offset, K is a position gain on the difference between the actual length L and a setpoint L_0, and B is the velocity gain on the difference between the string's velocity V and the desired velocity V_0. Similar to our previous paper, L_0 was constant, and V_0 was the descending output of the CPG. We did not modify T_0 for trajectory control as in our prior work.

At one level above reflexes, central pattern generators are groups of neurons involved in motor control capable of generating a rhythmic pattern [12]. When modeling CPGs for robotic use, it is important to consider the features of the neural circuits worth modeling. CPGs have advantages over more abstract sine wave approximations of the neural signals, such as the ability to smoothly integrate feedback and step changes in parameters [7]. For a detailed list of considerations see [5]. The specific CPG equations we chose are amplitude controlled phase coupled oscillators, also used in Ijspeert et al.'s salamander robot [14]:

$$\dot{\theta_i} = 2\pi v_i + \sum_j r_j w_{ij} sin(\theta_j - \theta_i - \phi_{ij}) \tag{3}$$

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

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

The phase of the CPG is determined by equation 3, where v_i is a frequency term, and coupling is determined by r_j, the amplitude of the coupled node, a coupling weight w_{ij}, and the phase difference between nodes, and a phase offset ϕ_{ij}. The amplitude is determined by equation 4, and is constant after it reaches the setpoint R_i. a_i is a positive constant. The final output is on the velocity of the impedance controller, as determined by equation 5. v_i and R_i can be specified as one parameter each, or as the combination of an offset, a gain on a descending command, and the command itself [14]. Equations were integrated using ODEInt, part of the Boost C++ libraries [1].

In our implementation, we assign one CPG node to each actuator. Node parameters, such as frequency and amplitude, were homogeneous throughout the structure. Therefore the gaits generated by this CPG depend on the coupling of nodes and the phase differences between them. However, letting machine learning decide the appropriate coupling out of "all to all" quickly becomes intractable in this continuous space, leading to $n(n-1)$ couplings, where n is the total number of nodes. If there are k segments with m nodes each, then this becomes $km(km-1)$. As a starting set of coupling rules, we decided that each node (actuator) would be coupled to nodes (actuators) with shared rigid bodies. This is similar to nearest neighbor coupling, which is the basis for models of swimming and walking networks in the salamander [2]. Since the rigid bodies are in a chain, this means that there are at most three rigid bodies worth of nodes in any repeated coupling set. For example, in Tetraspine with the outside strings forming the CPG, this leads to twenty four connections per segment ($m(3m-1)$), as opposed to the twelve in our previous paper. Couplings are bi-directional, and can have different sets of parameters for each direction. In the case of tetraspine, this would lead to forty eight parameters for the edges. So, in addition, we specified that symmetric couplings would be the same, consistent with our prior paper. This reduces the number of possible couplings to $m(3m+1)/2$. An example with a two node, three segment system is shown in Figure 2.

C. Parameter Estimation

In order to deal with this still fairly large parameter space (202 bounded continuous parameters in the largest case), we utilized Monte Carlo parameter estimation. Trials were scored by the distance the center of mass of the tensegrity spine moved in 30 or 60 seconds of simulation time, depending on computational intensity. Since the parameter estimation is random, trials can be run entirely in parallel, there is no need to synchronize for a mutation step. This allows for a rapid (though not exhaustive or optimal) search of a complex parameter space, allowing us to quickly identify parameters for effective locomotion.

After between 10,000 - 20,000 trials we took the best sets of parameters and ran Gaussian sampling on them with a standard deviation of 0.005. If a better set of parameters was found, sampling continued around the new best set. This allows for some hill climbing, without the need to compute a gradient. Additionally, there is little need for meta-optimization. The resulting best parameters have been included for analysis. On a desktop computer purchased in 2011 (Intel Core™i7 at 3.4 GHz), this search procedure takes 24 - 48 hours.

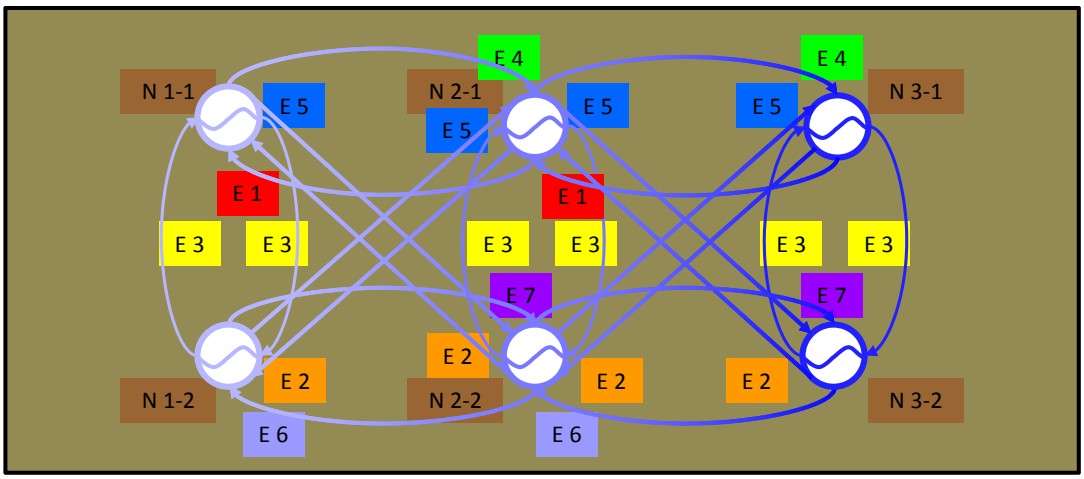

Fig. 2. The coupling rules used on the CPGs. This system leads to $m(3m+1)/2$ possible couplings, where m is the number of nodes. In this case, are a total of ten edges per node, but these are sorted by geometry into seven possible edges. This system scales at half the rate of specifying each coupling separately.

D. Cost of Transport

While distance traveled is a good metric for evaluating a new structure, it is not the only metric that should be considered. Therefore we also computed the efficiency metric cost of transport (COT) and included it in the results. Cost of transport is defined using Equation 6, as discussed in [28]:

$$COT = \frac{W}{mgd} \tag{6}$$

Where W is the work put in to the system, m is the mass, g is gravity, and d is the distance traveled. Cost of transport is unitless, and provides a means to compare any locomotor systems. Given equation 6, a lower COT indicates a more efficient system.

To compute the energy input to the tensegrity spines, we used the sum of the work used to shorten the strings: tension times distance shortened at each timestep. For energy calculations, we assumed that relaxation was passive, similar to biological muscles or a backdriven motor. The cost of transport for a system similar to our hardware implementation may be higher if the motors are actively unspooled during relaxation.

IV. NEW MORPHOLOGY AND LEARNED GAITS

A. Tetraspine Sidewinding

Our first attempt with this parameter estimation paradigm used the same CPG as our initial Tetraspine work, with four connections per segment. We also maintained the original coupling weights (0.5). This resulted in nine total parameters for estimation, as we left amplitude and frequency offsets and gains separate in this case since the equations were setup for descending commands, as in [14]. However, commands were held constant throughout the trials.

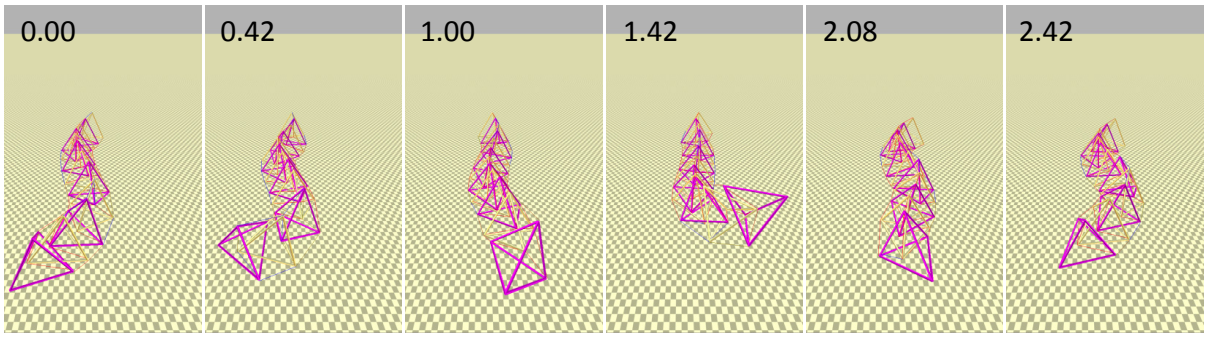

Fig. 3. The fastest gait overall, a form of sidewinding, was learned by a twelve segment Tetraspine. The vertex of the rear segment functions as a tail pushing the robot to the right, unlike our prior work where we treated it as the head [27]. Only the outside strings were included in the CPG, the velocity setpoint for the inside strings was zero. The time in seconds at each step is shown in the upper left corner.

The resulting sidewinding like gait[2] travels at 9.46 centimeters per second with a COT of 3.2. Scaling our prior results, this is 43% faster than the hand tuned gait of our prior work (6.6 cm/s).

Six key frames of video for this gait are shown in Figure 3. A second gait, similar to the hand tuned gait in our prior paper but traveling in the opposite direction, traveled at 7.78 cm/s with a COT of 2.4.

Given the significant improvements over our prior labor-intensive hand-tuning approach, these results were a strong proof of concept for applying random parameter estimation to tensegrity spines, allowing us to move on to larger parameter sets.

B. Tetrahedral Complex

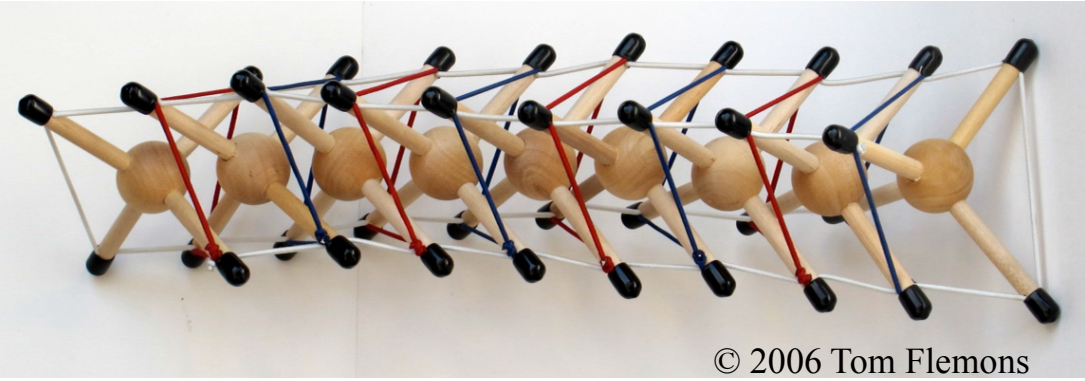

Fig. 4. A tensegrity spine model using a tetrahedral complex as vertebrae. Eight strings connect each segment, four of which form a tensegrity saddle joint [9]

Our second spine like tensegrity structure uses a tetrahedral complex as its vertebrae, with four strings composing a tensegrity saddle joint [9], and four strings running along the outside. All eight strings are actuated in this case. Changing the morphology of the structure from a convex tetrahedron to a tetrahedral complex and the addition of two strings adds two important elements to the structure. First, the center of mass of the structure is now over its line of support, which decreases the tension required to hold the structure's shape by a factor of 500 when compared to Tetraspine (10 N on Tetraspine vs 0.2 N on the tetrahedral complex). Additionally it allows control of all six degrees of freedom.

This connectivity had the highest number of parameters (202). We constrained the amplitude and frequency of the CPG to be single parameters at this point.

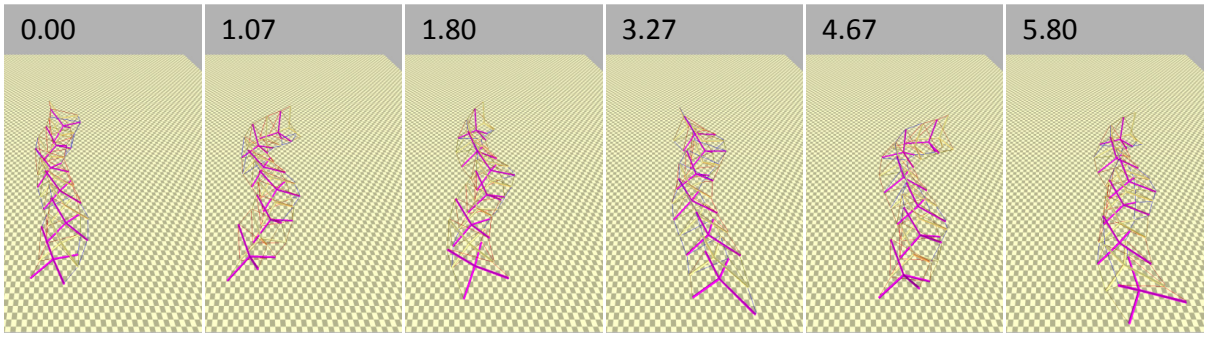

Fig. 5. The tetrahedral complex based spine also learned a gait similar to sidewinding. Large twists of the body are possible since all eight strings are actuated, moving the robot to the right and down. Other gaits (not shown) included rolling.

The best gait for the 'tetrahedral complex' spine, shown in Figure 5, was another form of sidewinding[3]. This gait moved at 6.13 centimeters per second, with a COT of 2.13, the lowest cost of transport of all four morphologies. A lower COT for sidewinding relative to other gaits matches results found in biological snakes [25].

C. Cross-linked Octahedral Complex

One possible disadvantage of the tetrahedral complex shape is the high number of actuators per segment. In order to reduce the number of actuators and maintain stability we decided to place passive strings at each of the tetrahedral saddle

[2]Video of the tetraspine sidewinding is available at `https://www.youtube.com/watch?v=I-7jCGT-WKk`

[3]Video of the tetrahedral complex sidewinding is available at `https://www.youtube.com/watch?v=Wyt7B7-OebI`

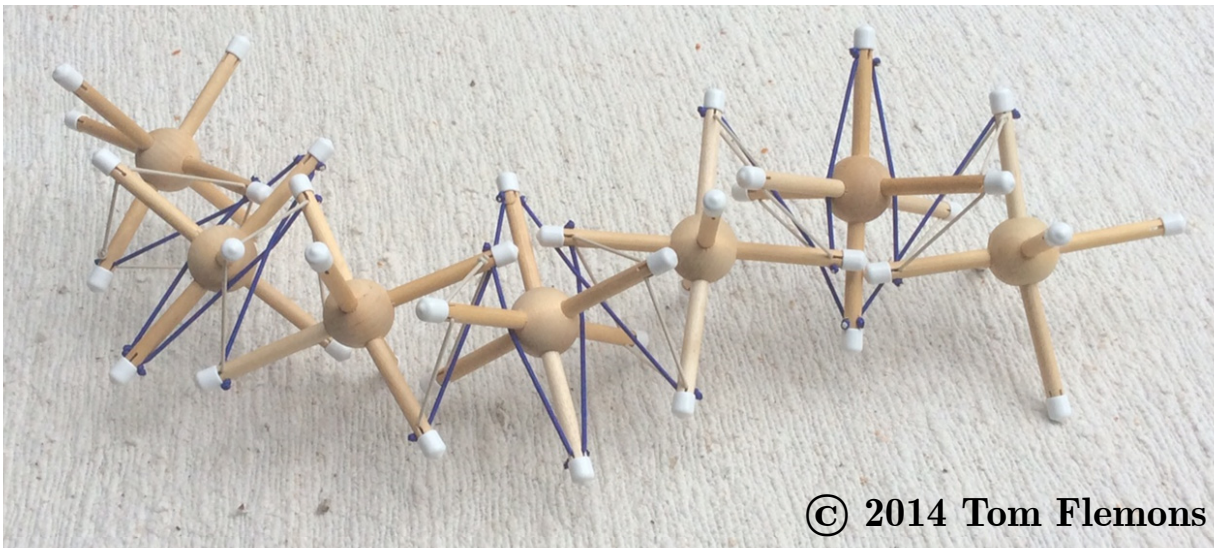

Fig. 6. Octahedral complexes form the vertebrae, with four passive (white) and four active (blue) connections between segments. The simulated spine has twice as many actuated strings. Every saddle joint has a pair of actuators on either side.

joints. Four strings compose the saddle joint, while another four cross link on either side of the saddle to provide antagonistic actuation. See Figure 6 for a passive model with half the number of actuators (blue strings). We also transitioned to an octehedral complex as the vertebrae shape, to avoid interference between the strings and vertebrae. This structure was our first attempt with unactuated strings, as opposed to the Tetraspine strings with their velocity setpoint of zero. In order to give the unactuated strings some pretension, we set their rest length 0.5 centimeters (5.4% of their length) shorter than the distance between their nodes. Since these strings were twice as stiff as the actuated strings (2000 N/m) the resulting pretension was 10 Newtons. These unactuated strings can be considered analogous to ligaments.

Reducing the number of actuators also dramatically reduces the number of parameters in learning. This morphology had to determine 90 control parameters.

When resting on the ground, only one rod from each vertebrae touches. The gait in Figure 7 utilizes these as legs in a walking like gait with a speed of 4.6 centimeters per second, and a cost of transport of 3.4. A similar gait[4] which was slightly slower (3.2 cm/sec) had a cost of transport of 2.6. This illustrates the common tradeoff between speed and efficiency in locomotor systems.

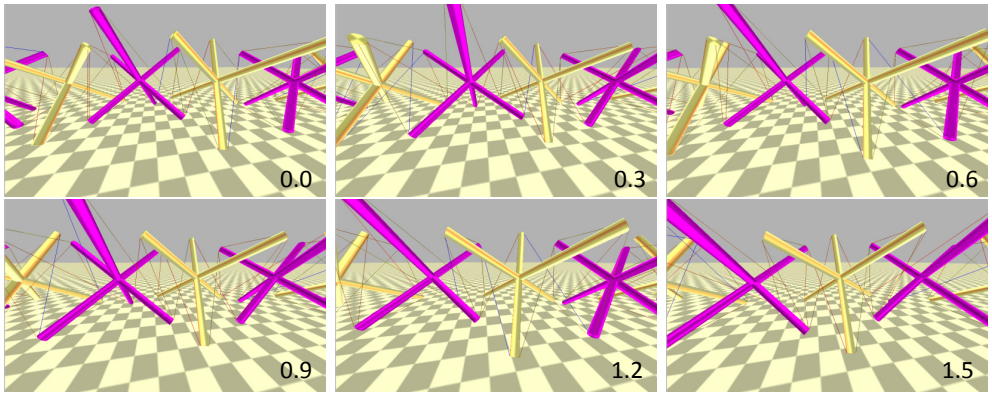

Fig. 7. When on a flat surface, the octahedral complexes have one contact per vertebrae. The learned gait uses these as legs and alternates sides in a walking pattern. While the robot mostly moves to the left, the motion is not purely two dimensional, the segments alternate slightly away from the central axis. In this plot phase of the CPG can be seen in the color of the strings, red indicates high tension strings, blue indicates low tension. This is also true in the previous plots, but the strings were typically too small to see. The time in seconds at each step is shown in the lower right corner.

[4]Video of the lowest COT gait for the octahedral complex based spine is available at `https://www.youtube.com/watch?v=Wk9x_40xWjM`

D. Spinal Column with Rib-like Attachments

Our final structure provided us with another opportunity to examine how the shape of the structure affects its motion. We hypothesized that snake like slithering motions would be easier on a closely packed vertebrae structure with longer attachments, like the ribs on a snake skeleton. We created the shape by attaching $3/4$ of an ellipse in place of one of the rods of the tetrahedral complex.

Muscle locations were chosen by examining anatomy textbooks [11], [19]. Five strings surrounding the spine mimic the Interspinales and Intertransversarii muscles. Our initial tests with these muscles proved unstable, so we added four passive strings between each segment's ribs, mimicking fascia in the thoraxic region. The passive strings were half of the stiffness of active strings (500 N/m), and were not pretensed. This was still insufficient, so we added two more muscles connecting the transverse processes to the inferior ribs. Upon reexamination of the anatomy books, we realized these may play a role similar to the Levatores costarum. With seven active strings, this morphology required 156 parameters.

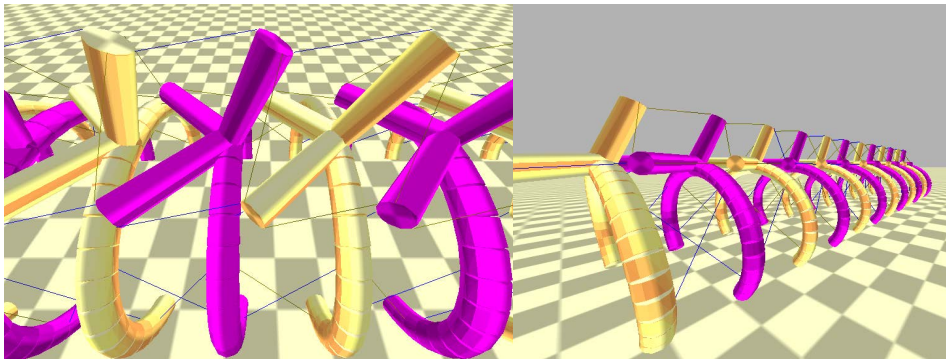

Fig. 8. Ribs are rigidly attached to three rods in a configuration that mimics three rods of the tetrahedral complex. Strings that only touch ribs are passive, the other seven are active.

Parameter estimation returned two gaits of interest. One, pictured in Figure 9 is a crawling motion similar to slithering.

The fastest gait[5] traveled at 5.37 centimeters per second, with a cost of transport of 6.0. While this gait is neither the fastest, nor most efficient, it has the advantage that the "head" segment stays relatively stable during locomotion. Future work will incorporate sensors, such as cameras, onto the robots, and terrain analysis will be greatly simplified with a stable gait compared to the challenges posed by the sidewinding and rolling gaits. Also, we expect that the speed and efficiency of this "ribbed" approach can be greatly improved through further bio-mimicry, such as optimized muscle routing, and use of compliant rib cages which provide better ground contact.

Thus our workflow returned seven gaits that merited discussion on four different morphologies. After an initial set of parameters is determined, optimization could also be run on efficiency, range of motion or other desired parameters.

V. HARDWARE PROTOTYPE

A. Sensors, Actuators

Given the tendency of machine learning to exploit physics engines, we want to ensure our results match reality. Therefore, in order verify additional elements of the NTRT's simulation results, we are constructing a three segment Tetraspine prototype in hardware.

Fig. 10 shows the recently developed hardware prototype, which consists of 12 DC motors, 12 vectran cables, and 12 load cells. The rod length and diameter of the current tetraspine prototype is equivalent to 38.10 cm and 0.64 cm, respectively. Compared to the previous two prototypes, the sensor resolution was significantly increased by replacing a knit stretch sensor with a load cell sensor. In addition, all inner and outer compression elements are made up of carbon fiber to reduce the mass of each segment from 1.84 kg in the previous prototype to 0.73 kg including sensor and control boards. One end of the vectran cable is connected to a DC motor (Faulhaber 1524 52:1) for rotational motion and the other end is fixed to a load cell (Load Cell Central LCC-CTD) for measuring cable tension. The prototype is controlled by a microcontroller (Arduino Nano) and a motor driver (Pololu MC33926). An instrumentation amplifier (TI INA114) is used to amplify the output voltage of the load cell because the maximum voltage swing is 10.0 mV. After the calibration process, the resolution of the load cell is 0.15 Newtons, where the maximum measurable tension is set to 110.0 Newtons.

[5]Video of the gait pictured in Figure 9 is available at `https://www.youtube.com/watch?v=j0UXr59C9mY`. The fastest gait can be viewed at `https://www.youtube.com/watch?v=PL22dswN2RA`

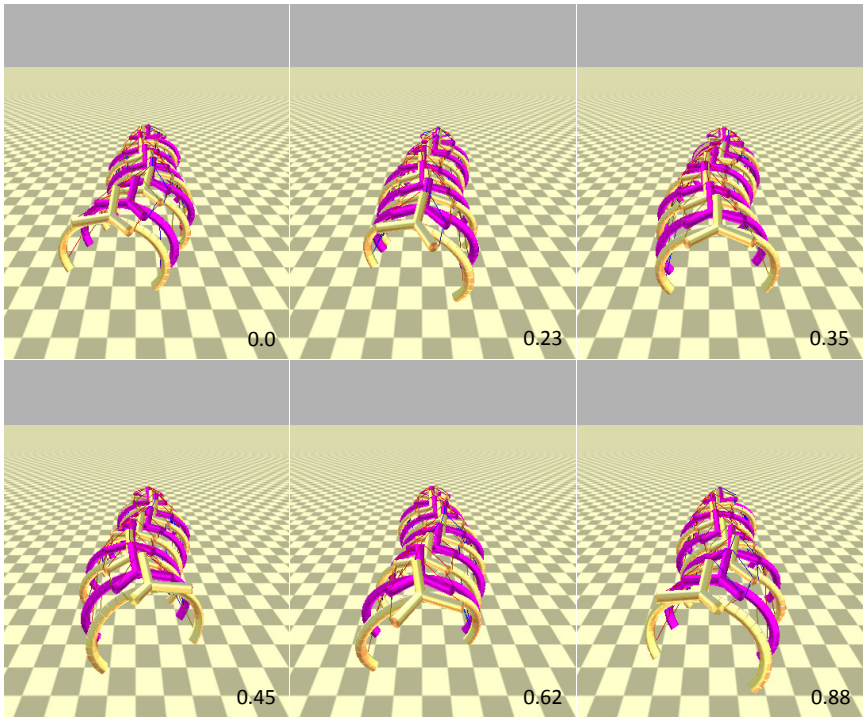

Fig. 9. The second fastest gait of the ribbed tensegrity is shown. This gait exploits turning the body left and right to move forward (down on the page), similar to lateral undulation.

B. Control

Both position control and impedance control are implemented in the firmware, where the control mode can be changed from the main pc. In control modes, the motor is controlled at every 0.001 sec, where the desired motor command is sent every 0.02 sec in real-time. Initially, the position control is used to set the initial length of each cable and to maintain the balance of all three segments, which is then switched to impedance control. An offset tension, the encoder information, and the desired velocity command as in equation (2) provide the target cable tension, and the cable tension is controlled with a PD controller.

C. Results

Our prior Tetraspine prototypes generated forward locomotion and qualitatively matched the the simulator[6]. While our simulator was already validated by our ReCTeR prototype [6], this new prototype will allow us to test elements such as sliding friction. Unlike our previous iterations of Tetraspine, we will now be able to use the exact control parameters used in the simulator. We are in the process of tuning the simulation and the hardware parameters to match their behavior. One major difference between our prior work and this new prototype is the absence of springs in-line with the actuation, making the strings approximately 600 times stiffer. Our tests will include whether our simulated actuators work at this stiffness, or if we need more detailed motor models.

VI. CONCLUSIONS

We have presented a method for the rapid design of tensegrity robots in simulation using the NASA Tensegrity Robotics Toolkit. Four examples, with between forty four and eighty eight actuators, were tuned for effective locomotion using a Monte Carlo parameter optimization procedure that requires little operator intervention. The tunably compliant motion resulting from the combination of tensegrity, central pattern generators, and impedance controllers was shown previously to provide effective locomotion over a variety of terrain [27], so we anticipate that these results will generalize to rough terrain conditions as well. While our primary interest is robotics, we hope that these approaches may eventually improve understanding of biomechanics and the neural control of locomotion.

Future work will extend this approach to additional morphologies, such as Snelson's spiral mast [26], and morphologies that include appendages similar to a salamander. Snake like morphologies may add a compliant soft belly for additional traction. We will also attempt optimization on metrics such as cost of transport, to more directly determine the most efficient

[6]Video of prior Tetraspine hardware compared to simulation `http://youtu.be/BA_YyUkCG6o`

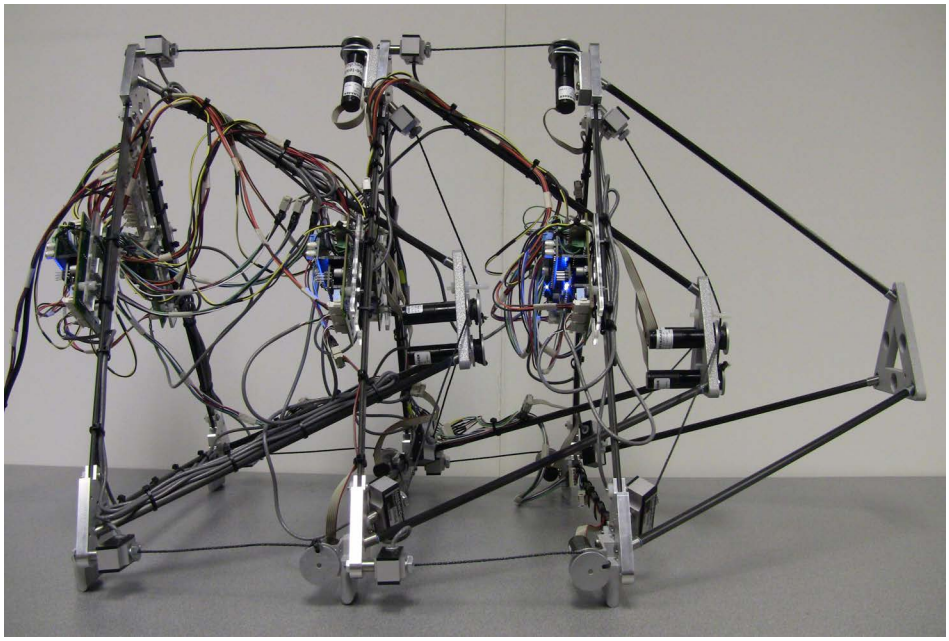

Fig. 10. Our hardware prototype of tetraspine, utilizing 12 DC motors, load cells and vectran®cable.

structures and controllers. Finally, we intend to fully utilize the dynamic properties of the CPGs in our control, which will allow for online parameter tuning, gait transitions and goal directed locomotion.

VII. ACKNOWLEDGMENTS

The authors thank Stephen Levin and Tom Meyers for their explorations of biotensegrity and anatomy. They would also like to thank Terry Fong, members of the NASA Ames Intelligent Robotics Group, and members of the CWRU Biologically Inspired Robotics Lab for all of 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] A. Bicanski, D. Ryczko, J. Knuesel, N. Harischandra, V. Charrier, Ö. Ekeberg, J. Cabelguen, and A. J. Ijspeert. Decoding the mechanisms of gait generation in salamanders by combining neurobiology, modeling and robotics. *Biological cybernetics*, 107(5):545–564, 2013.
[3] 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.
[4] J. Bruce, K. Caluwaerts, A. Iscen, A. P. Sabelhaus, and V. SunSpiral. Design and evolution of a modular tensegrity robot platform. In *IEEE International Conference on Robotics and Automation (ICRA-2014)*, 2014.
[5] J. Buchli, L. Righetti, and A. J. Ijspeert. Engineering entrainment and adaptation in limit cycle systems. *Biological Cybernetics*, 95(6):645–664, 2006.
[6] K. Caluwaerts, J. Despraz, A. Iscen, J. Bruce, A. P. Sabelhaus, B. Schrauwen, and V. SunSpiral. Design and control of compliant tensegrity robots through simulation and hardware validation. *Submitted*, 2014.
[7] A. Crespi and A. J. Ijspeert. Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator. In *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, volume 11, pages 19–27. Citeseer, 2006.
[8] M. S. Fischer, K. E. Lilje, J. Lauströer, and A. Andikfar. *Dogs in Motion*. VDH Service GmbH, 2011.
[9] T. Flemons. The geometry of anatomy. `http://www.intensiondesigns.com/geometry_of_anatomy.html`, 2007.
[10] T. Flemons. The bones of tensegrity. *http://www.intensiondesigns.com/bones_of_tensegrity*, 2012.
[11] H. Gray. *Anatomy of the human body*. Lea & Febiger, 1918.
[12] S. Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228(4696):143–149, 1985.
[13] S. Grillner, A. Kozlov, P. Dario, C. Stefanini, A. Menciassi, A. Lansner, and J. Hellgren Kotaleski. Modeling a vertebrate motor system: pattern generation, steering and control of body orientation. *Progress in brain research*, 165:221–234, 2007.
[14] 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.
[15] D. E. Ingber. The architecture of life. *Scientific American*, 278(1):48–57, 1998.
[16] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer. Robust distributed control of rolling tensegrity robot. In *The Autonomous Robots and Multirobot Systems (ARMS) workshop at AAMAS 2013*, 2013.
[17] S. M Levin. Putting the shoulder to the wheel: a new biomechanical model for the shoulder girdle. *Biomedical sciences instrumentation*, 33:412–417, 1997.
[18] S. M. Levin. The tensegrity-truss as a model for spine mechanics: biotensegrity. *Journal of Mechanics in Medicine and Biology*, 2:375–388, 2002.
[19] T. W. Myers, L. Chaitow, and D. Juhan. *Anatomy Trains: Myofascial Meridians for Manual and Movement Therapists, 1e*. Churchill Livingstone, 1 edition, October 2001.
[20] O. Orki, A. Ayali, O. Shai, and U. Ben-Hanan. Modeling of caterpillar crawl using novel tensegrity structures. *Bioinspiration & Biomimetics*, 7(4):046006, 2012.

[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] S. M. Secor, B. C. Jayne, and A. F. Bennett. Locomotor performance and energetic cost of sidewinding by the snake crotalus cerastes. *Journal of experimental biology*, 163(1):1–14, 1992.
[26] K. Snelson. Continuous tension, discontinuous compression structures. United States patent 3169611, February 1965.
[27] B. R. Tietz, R. W. Carnahan, R. J. Bachmann, R. D. Quinn, and V. Sunspiral. Tetraspine: Robust Terrain Handling on a Tensegrity Robot Using Central Pattern Generators. In *IEEE/ASME Advanced Intelligent Mechatronics*, pages 261–267, 2013.
[28] V. A. Tucker. Energetic cost of locomotion in animals. *Comparative Biochemistry and Physiology*, 34(4):841–846, 1970.