DuCTT: a Tensegrity Robot for Exploring Duct Systems

Jeffrey Friesen[1], Alexandra Pogue[1], Thomas Bewley[1], Mauricio de Oliveira[2], Robert Skelton[2], Vytas Sunspiral[3]

Abstract— A robot with the ability to traverse complex duct systems requires a large range of controllable motions as well as the ability to grip the duct walls in vertical shafts. We present a tensegrity robot with two linked tetrahedral frames, each containing a linear actuator, connected by a system of eight actuated cables. The robot climbs by alternately wedging each tetrahedron within the duct and moving one tetrahedron relative to the other. We first introduce our physical prototype, called DuCTT (Duct Climbing Tetrahedral Tensegrity). We next discuss the inverse kinematic control strategy used to actuate the robot and analyze the controller's capabilities within a physics simulation. Finally, we discuss the hardware prototype and compare its performance with simulation.

I. INTRODUCTION

Many duct systems are confined or otherwise inaccessible for human inspection and repair, especially in the event of emergencies (radiation leaks, etc). Lightweight mobile robotic vehicles are well suited to navigate, inspect, and repair such systems. Some existing vehicles use wheels to press against the walls while driving through ducts; such vehicles can move quickly, but have difficulty overcoming sharp corners or other irregularities. Other vehicles handle such irregularities by implementing mechanical limbs, but their rigidly jointed legs generally require high power and heavy actuators to achieve vertical ascent [1] [2]. We propose a new form of robotic vehicle for such applications that circumvents these problems via tensegrity design principles.

Tensegrity is a structural design paradigm in which compression members are suspended in a network of tension elements. Tensegrity structures use tensile forces to distribute external loads throughout a structure in such a manner that no individual member within the structure experiences a bending moment, with pretension of the structure used to keep any tensile member from going slack. In a mobile robot this tension network can be beneficial, as it distributes forces globally and helps to prevent the force concentrations which can occur near the joints of traditional designs.

Several tensegrity robots have been designed which capitalize on the oscillatory dynamics of tensegrity structures

[1]Jeffrey Friesen, Alexandra Pogue, and Thomas Bewley are with the UC San Diego Coordinated Robotics Lab, MC 0411, La Jolla CA, 92093 USA `jfriesen@ucsd.edu`, `alexie.pogue@gmail.com`, `bewley@ucsd.edu`

[2]Mauricio de Oliviera and Robert Skelton are with the Dept of MAE, MC 0411, UC San Diego, La Jolla CA, 92093 USA `mauricio@ucsd.edu`, `bobskelton@ucsd.edu`

[3]Vytas Sunspiral is a Senior Robotics Researcher, SGT Inc., with the Intelligent Robotics Group, NASA Ames Research Center, Moffett Field CA 94035 USA `vytas.sunspiral@nasa.gov`
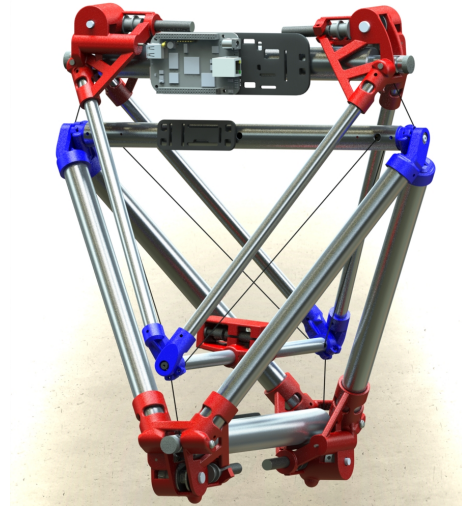
Fig. 1. DuCTT Prototype

to produce efficient gait patterns. Paul et al created three and four strut class 1 tensegrity structures which were capable of locomotion [3]. Others have explored the use of evolutionary algorithms to optimize gaits by altering the structural characteristics and control strategies [4] [5] . To the authors' knowledge no tensegrity robots have been developed with the primary objective of planned climbing.

A tensegrity robot has characteristics which are ideal when faced with the task of climbing. Tensegrity structures have already been shown to be highly mass efficient [6] and it is often the case that actuators within a well-designed tensegrity robot act in parallel with each other, and often with significant mechanical advantage, which reduces the power requirements and thus weight of each actuator. Furthermore the inherent passive compliance within a tensegrity robot mitigates damage to the robot and environment in the event of unplanned collisions.

The dual tetrahedral class 3 tensegrity structure of DuCTT is shown in Fig. 1; this structure is inspired, in part, by a biotensegrity model of a vertebrate spinal column [7]. If all eight tension elements of this structure are actuated independently, one tetrahedron's position can be controlled in all six degrees of freedom relative to the other tetrahedron. Additionally, because there are redundant actuators in our system design, the pretension in the structure can be adjusted during operation; this adjustable pretensioning can be used to modify the external stiffness of the structure [8].

Another benefit is the robot's minimalist design which reduces its cross-sectional area (see Fig. 2) that would otherwise impede flow through the duct. If it is properly
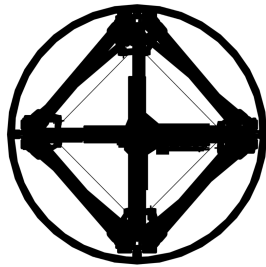
Fig. 2. Top view of DuCTT in a round duct. Its minimal cross-sectional area enables DuCTT to move through air ventilation systems while they are operating, without significantly impeding the flow in the duct.

insulated, a more robust DuCTT prototype might even be able to move through pipe systems filled with stationary or flowing water or oil.

II. PROTOTYPE HARDWARE

A. Topology

DuCTT is composed of two tetrahedral frames with six rigid members linked by hinge joints. Linear actuators are placed between nodes A-B and H-G, as labeled in Fig. 4. The actuators move between a length of 31.5 cm, which produces an approximately regular tetrahedron, and 41.5 cm which causes the tetrahedron angles opposite the linear actuator to increase.

Cables one through four in Fig. 4 are dubbed *vertical cables*. The tension forces in the vertical cables serve to pull the two tetrahedra together. Cables five through eight serve to oppose these forces, and are dubbed *saddle cables*. In the current prototype, all four vertical cables are independently actuated. The saddle cables are actuated in adjacent pairs such that cables eight and six as well as cables seven and five always have an equal rest length.

Because of this coupled saddle actuation, our current DuCTT prototype is capable of controlled motion in four degrees of freedom. The robot can translate one tetrahedron relative the other in the x-z plane, and can rotate one tetrahedron about the axes passing through C-D and F-E.

B. Cable Actuators

Each vertical cable is driven by a 1.4 watt Maxon motor which is passed through the power transmission depicted in Fig. 3. This drive train serves three purposes: it aligns the motor axis with the linear actuator to reduce needed space, it increases mechanical advantage, and it prevents back-driving of the motor, allowing the vertical cable actuators to maintain a static pretensioned state without expending energy.

Each set of saddle cables is actuated by a 1.4 watt Faulhaber motor with a large gear reduction, placed within the tube of the bottom tetrahedron, and directly linked to a cable pulley. The saddle cables have a simpler power transmission which is dynamically efficient, but is back drivable (thus, energy consumption is required to hold a position).
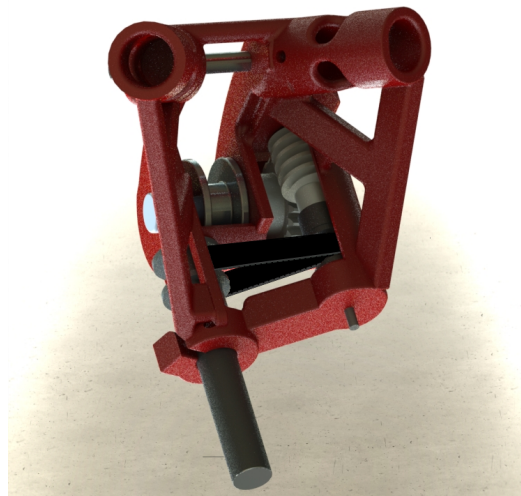

Fig. 3. Power transmission for cables. The motor first passes through a timing belt assembly, then through the worm into the worm gear before applying force to the cable via the reel.
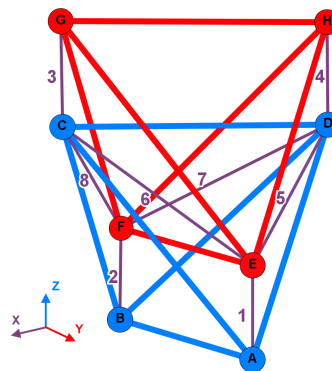

Fig. 4. Labeling of nodes and cables.

C. Electronics

DuCTT is currently powered by eight 18650 lithium ion cells placed in series and stored within the tubes of the bottom tetrahedron. The top tetrahedron has a mass of 1.1 kg, which is significantly less than the 2.0 kg mass of the bottom tetrahedron; this is ideal, since the saddle cables offer less mechanical advantage than the vertical cables for lifting during the climbing process described in the next section.

A Beaglebone Black is used to perform high level control, while three Arduino microcontrollers are used to perform position control of the six cable actuators. A single I2C bus is used to communicate between all the microcontrollers with the Beaglebone set as master. Power and communication are fed between the tetrahedra through a single ribbon cable. Quadrature encoders are used to read motor position and conductive rubber cord stretch sensors are placed in parallel with the linear actuators to get rough length estimates. The Arduino also monitors the current being fed into the linear actuator to determine when adequate force has been exerted on the duct walls.
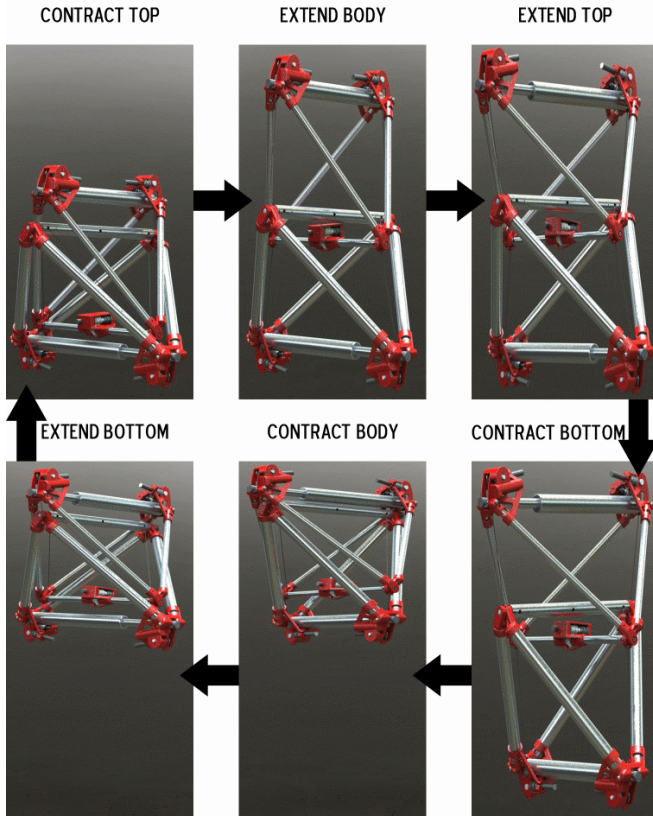
Fig. 5. Sequence of steps used for climbing.

D. Climbing Process

Fig. 5 shows the sequence of motions used to climb a duct. The linear actuators are used alternately to expand and wedge against the duct walls, thus bracing and securing one of the segments. The other segment is released, and the actuated cables are used to lift one tetrahedron or the other to the next position. This sequence can easily be implemented to climb autonomously within a duct at any orientation.

When DuCTT encounters a corner in the duct system, it can rotate the top tetrahedron to orient it for wedging in the new direction (see Fig. 10) and then translate the tetrahedron into the new section. It then extends the top linear actuator to fix the top tetrahedron in place. Then it retracts the bottom linear actuator and pulls it into the new duct before reverting to its standard climbing procedure.

III. CONTROL AND SIMULATION

A. Force Density Method for Inverse Kinematics

We use the force density method to solve the inverse kinematics of the robot, with the basic assumptions being:

- Desired x, y, and z coordinates for all nodes are known.
- All connections between rigid members are frictionless ball joints.
- All forces are known and exerted at nodes, no mid-member forces.
- Bars are perfectly rigid and do not experience deformation.

As described in [9] [10] [6], for a tensegrity with r bars, s cables and n nodes the topology can be expressed by a connectivity matrix, $\boldsymbol{C}(\in \mathbb{R}^{(s+r) \times n})$, where the first s rows of $\boldsymbol{C}$ correspond to cable members and the last r rows of $\boldsymbol{C}$ correspond to bar members. If member k connects nodes i and j $(i < j)$ then the ith and jth elements of the kth row of $\boldsymbol{C}$ are set to 1 and -1, respectively, as

$$\boldsymbol{C}_{(k,h)} = \begin{cases} 1 & \text{if } h = i, \\ -1 & \text{if } h = j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Let $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}(\in \mathbb{R}^n)$ denote the nodal coordinate vectors in the x, y and z directions, respectively. As seen in [9] the static equilibrium equations can then be stated as

$$\boldsymbol{C}^T \boldsymbol{Q} \boldsymbol{C} \boldsymbol{x} = \boldsymbol{p}_x, \tag{2a}$$

$$\boldsymbol{C}^T \boldsymbol{Q} \boldsymbol{C} \boldsymbol{y} = \boldsymbol{p}_y, \tag{2b}$$

$$\boldsymbol{C}^T \boldsymbol{Q} \boldsymbol{C} \boldsymbol{z} = \boldsymbol{p}_z, \tag{2c}$$

Where $\boldsymbol{p}_x$, $\boldsymbol{p}_y$ and $\boldsymbol{p}_z$ are the vectors of external loads applied to the nodes in the x, y and z directions, respectively, $(\cdot)^T$ denotes a matrix transpose, and $\boldsymbol{Q}(\in \mathbb{R}^{(s+r) \times (s+r)})$ is the diagonal square matrix described as

$$\boldsymbol{Q} = diag(\boldsymbol{q}), \tag{3}$$

such that $\boldsymbol{q}$ is the force density vector, defined in [9] as $\boldsymbol{q} = \{q_1, q_2, q_3, ...q_{s+r}\}^T$, where each entry q_i is defined as the ratio between the force, f_i, and the length, l_i, such that

$$q_i = f_i/l_i, \tag{4}$$

or the force density in the member. Substituting (3) into (2) yields

$$\boldsymbol{C}^T diag(\boldsymbol{q}) \boldsymbol{C} \boldsymbol{x} = \boldsymbol{p}_x, \tag{5a}$$

$$\boldsymbol{C}^T diag(\boldsymbol{q}) \boldsymbol{C} \boldsymbol{y} = \boldsymbol{p}_y, \tag{5b}$$

$$\boldsymbol{C}^T diag(\boldsymbol{q}) \boldsymbol{C} \boldsymbol{z} = \boldsymbol{p}_z. \tag{5c}$$

Equation (5) can then be reordered as

$$\boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{x}) \boldsymbol{q} = \boldsymbol{p}_x, \tag{6a}$$

$$\boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{y}) \boldsymbol{q} = \boldsymbol{p}_y, \tag{6b}$$

$$\boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{z}) \boldsymbol{q} = \boldsymbol{p}_z. \tag{6c}$$

Thus, letting

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{x}) \\ \boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{y}) \\ \boldsymbol{C}^T diag(\boldsymbol{C} \boldsymbol{z}) \end{bmatrix}, \tag{7}$$

and

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_x \\ \boldsymbol{p}_y \\ \boldsymbol{p}_z \end{bmatrix}, \tag{8}$$

allows (6) to be written as

$$\boldsymbol{A} \boldsymbol{q} = \boldsymbol{p}. \tag{9}$$

Equation (9) can then be easily solved for q using the Moore-Penrose psuedoinverse, with the general solution being

$$q = A^+ p + (I - A^+ A)w, \qquad (10)$$

where $(\cdot)^+$ denotes the psuedoinverse of a matrix, I is the identity matrix, and w is any vector in $\mathbb{R}^{3n}$. We select w by minimizing $q_s^T q_s$ with respect to $q_s \geq 0$ where q_s is defined as the first s entries of q. This will minimize the amount of elastic potential energy contained in the cables while the constraint serves to ensure all cables remain in tension. Equation (9) can then be substituted into this equation and the optimization problem can be written as

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & (A_s^+ p + V w)^T (A_s^+ p + V w) \\ \text{subject to} \quad & A_s^+ p + V w \geq 0, \end{aligned} \qquad (11)$$

Where $A_s (\in \mathbb{R}^{3n \times s})$ is the first s columns of A and $V (\in \mathbb{R}^{3n \times 3n})$ is defined as $V = I - A_s^+ A_s$. Expanding the minimization function in (11) and dropping terms which don't contain w gives

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & w^T V^T V w + 2 w^T V^T A_s^+ p \\ \text{subject to} \quad & A_s^+ p + V w - c \geq 0, \end{aligned} \qquad (12)$$

additionally the vector $c \ (\in \mathbb{R}^{3n})$ added to the constraint condition allows for minimum force densities to be set for each cable. A quadratic program can then be implemented to solve (12) and determine appropriate values for q. Once q is known, appropriate rest lengths can be calculated using a linear spring model for each cable. Implementing the linear spring equation as

$$f_i = K_i (l_i - l_{0i}), \qquad (13)$$

substituting (13) into (4) and solving for l_{0i} gives

$$l_{0i} = l_i (1 - q_i / K_i) \qquad (14)$$

The rest lengths from (14) can then be fed into the robot to achieve any possible static configuration.

B. Tensegrity Robotics Simulator

The simulation environment chosen was the NASA Tensegrity Robotics Toolkit (NTRT), built on top of the discrete time, open source Bullet Physics Engine. An advantage of Bullet is that it is designed to work in real time, and thus simulates collisions without using excessive processing power [11]. All forces, inertias, and constraints are automatically integrated, and can be measured throughout the simulation. Furthermore, the physics engine is able to provide an updated world transform for all the objects, making the extraction of data for evaluation possible [12]. These capabilities enable the NTRT to be used for both hardware and controller design.

The NTRT provides tensional elements with realistic material properties and tension numbers. Cables were modeled as force pairs that apply tensional forces to the vertices of the tetrahedra. Because they are not Bullet soft-bodies, their physical contact with other elements in the simulation
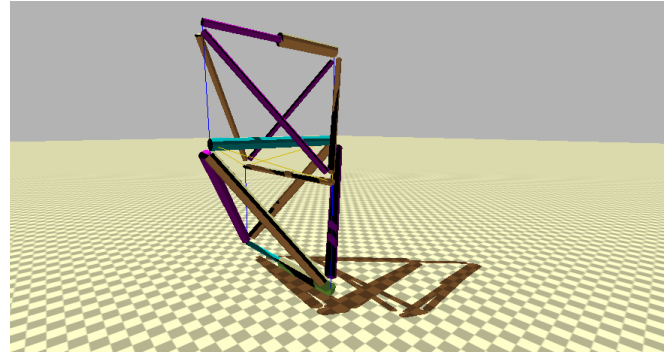

Fig. 6. NASA Tensegrity Robotics Toolkit (NTRT) simulated DuCTT model

is not computed, but the geometry of DuCTT is such that string collisions do not occur during intended maneuvers. The accuracy of the NTRT simulator for compliant tensegrity structures has been extensively studied by comparing to both an Euler-Lagrange simulation, and a real hardware tensegrity robot motion experiment tracked with an active marker motion capture system [11]. NTRT is currently in the process of being open sourced, please contact the authors for further information.

In simulating DuCTT (Fig. 6), 40 % of the mass distribution for the top tetrahedron was divided equally among the five structural tubes, with 60 % attributed to the linear actuator. Each component of the linear actuator is of equal weight, with the center of mass of the object located a varying distance from the axis of symmetry, depending on actuator configuration. Components of the linear actuator were constrained to translational movement along the axis of the cylinder. All structural tubes were attached using hinges to inhibit movement to one degree of freedom.

Tensioning elements between tetrahedral vertices were modeled as weightless, linear springs. For saddle cables an elasticity of 2000 N/m was used, a property calculated for similar off-the-shelf ropes such as Kevlar. The simulation aided in determining what spring hardware would be best to use for vertical cable springs; elasticity was varied until dynamic behavior was deemed acceptable. A spring constant of 600 N/m provided the best results. Any actuation by motors was simulated by inputting rest lengths specified by the inverse kinematics control policy (IKCP) to the simulation controller.

The main objective of the simulation was to verify that rest lengths determined by the IKCP were effective in positioning the robot in a desired configuration. Conditions for implementation of the static controller were optimized by instituting a .2 second pause to allow the system to settle before each data point was extracted, and a motor speed function was utilized in order to increase the time taken to achieve a change in rest length.

Position control was tested for all four degrees of freedom, first by commanding the tetrahedron to translate linearly in the x-z plane then by rotating the tetrahedron about axes coincident to bars C-D and E-F, referenced in Fig. 4, at

a given z position. For translational tests, minimum force densities, or the entries of c in Equation (12) were first set to 1 N/m then compared to results observed when minimum force densities were set to 16 N/m. The rotational test was performed with a set of minimum force densities equal to 0.1 N/m.

Tests of the maximum force necessary to attain certain static equilibrium positions were also conducted for both the translational and rotational motions in order to assess feasibility of hardware implentation.

Plots for translational mobility in the x-z plane are depicted in Fig. 7 and Fig. 8. In order to construct the plot, the tetrahedron traveled twice along a 39,801 point mesh. In order to eliminate small directional biases, position differences at each coordinate for a forward and a reverse direction were averaged, then all nodal errors were averaged.

When the entries of c were set to 1 N/m, position error ranged from .26 cm to 1.2 cm, indicating that the control policy is highly effective in position control. The maximum force ranged from 3.8 N to 31.1 N. Error depicted in plots was likely due to imperfections in physical modeling or mathematical approximations made in inverse kinematics. The force plot indicates that in these positions, if the tetrahedron is not rotating, directing it to upper corners of the base tetrahedron will be impossible with current hardware.

Error for minimum force densities of 16 N/m ranged from .13 cm to 0.4 cm, representing a 67 % decrease in maximum error. A large portion of the data exhibits smaller error when compared to data represented in Fig. 7, indicating the increase in minimum force density largely attenuated the error.
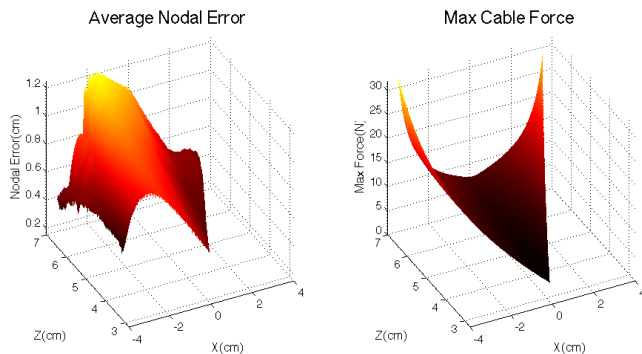

Fig. 7. Average error and maximum force plots for force density of 1 N/m.

Results indicate that higher minimum force densities result in better position control given the limitations of static tensegrity analysis. Varying pretension numbers in hardware testing should be conducted in the future to confirm results.

The average nodal error and maximum force plot for rotational degrees of freedom are depicted in Fig. 9. The position domain represents possible static configurations defined as having force density vector, q, greater than a positive c (Eq. 4). Thus c was decreased to a value of 0.1 N/m, to increase the number of possible static configurations. Because a higher minimum force density significantly decreased
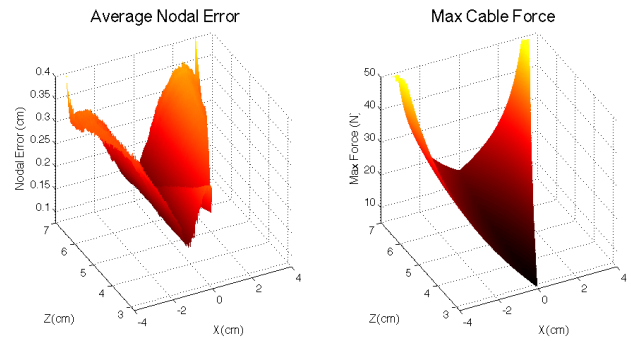

Fig. 8. Average error and maximum force plots for force density of 16 N/m.
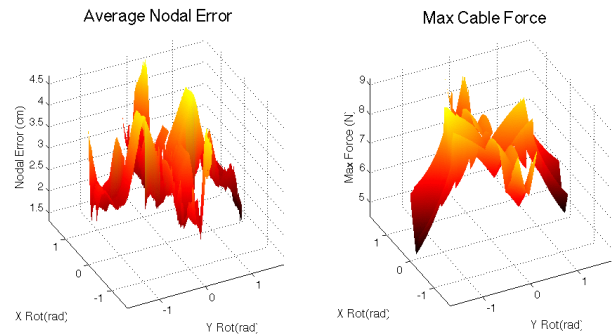

Fig. 9. Average error and maximum force plots for force density of 0.1 N/m.

range of motion by severing connections between stable regions, a comparative test was not conducted. However, it is likely that in regions where higher minimum force densities are possible, higher average nodal error occurred due to a decrease in magnitudes of c. Average nodal error ranged from 1.4 cm to 4.7 cm, while max force ranged from 4.8 N to 9.2 N, well within the range of the hardware capabilities.

C. Hardware Implementation

A preliminary test was performed to evaluate controller performance on the hardware. As seen in Fig. 10, the robot was fixed at its base while oriented vertically, as it would be when climbing in a vertical tube. A series of successive static configurations were then sent to the robot every 0.6 seconds to translate the top tetrahedron along the z axis while also rotating about the axis passing through nodes E-F. Data was collected by measuring pixel distances in frames within a video and thus has some inherent error but is adequate as an initial evaluation of the robot's performance.

The results of this test are shown in Fig. 11. As seen from the plot, the robot tracks targeted angular position well, with worst case errors in the range of 0.1 radians, but has worse performance along the z axis, with maximum errors around 4 cm which occurred toward the end of the test. The two impulsive errors around 60 and 70 seconds were likely caused by errors in communication between the microcontroller which supervised the actuation of the saddle cables and the Beaglebone. If the microcontroller were to have missed an intermediate position set-point, a large jump
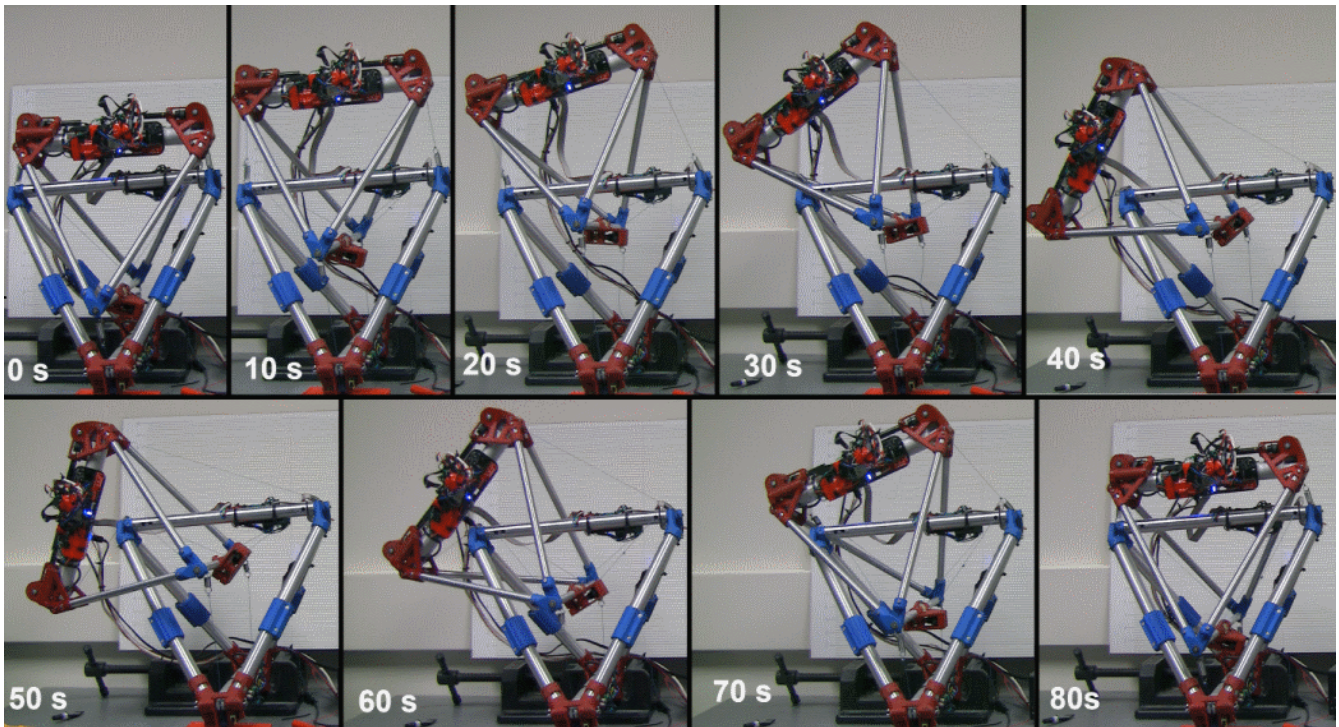
Fig. 10.   Frames taken in ten second intervals of the hardware test

would ensue when the set-point experienced a sudden change associated with delayed update, as is seen in the plot. A more robust communication protocol should be implemented in future iterations.

Regardless of these small errors, the test demonstrates the abilities of the robot. It should be noted that the robot is capable of moving an order of magnitude faster than within the test, but precision is reduced due to the static nature of the controller. Thus a slow command sequence was implemented to get an idea of how the controller performs in semi-static circumstances.

Fig. 11.   Z position and vertical angle about axis E-F for hardware test

IV. FUTURE WORK

In order to fully investigate the capabilities of the hardware, additional testing and tuning of the DuCTT prototype will need to be conducted. An overhaul of the communication between microcontrollers as well as installation of printed circuit boards to reduce wiring will yield large improvements to stability. Additionally, integration of two more actuators within the design will allow independent actuation of all cables. This will improve the mobility of the robot, allowing it to orient one tetrahedron in all six degrees of freedom relative to the other. Rubberized feet will also be added to the ends of the linear actuator to aid in gripping duct walls.

These additions will allow us to begin testing the prototype in duct systems. We will first test within a straight section and measure the speed and energy consumption of the robot during this process. Next we will optimize a set of commands within the NTRT simulator to round right angle corners in ducting, one of the most common obstacles this type of robot will encounter. These commands will then be sent to the robot in an appropriate test section and results will be compared.

We will also analyze the robots ability to move past small irregularities with a focus on how compliance aids in the process. If the mechanical compliance of the robot proves inneffective in responding appropriatley to irregularites, we will add force sensing within the cables. We can then implement impedance control, which will allow the robot to be much more responsive to its environment and improve behavior in the event of unplanned obstacles.

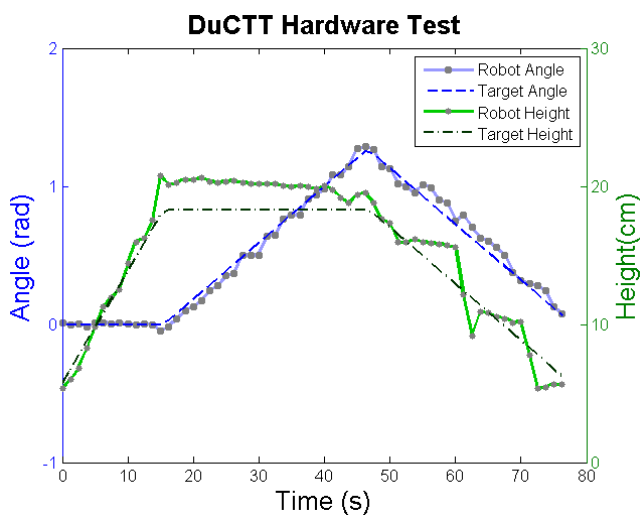Research of other control techniques, with a focus on

dynamic controllers, also needs to be conducted. This will allow for smoother motion as well as the ability to dynamically move through regions which may not have stable static configurations, further increasing the workspace of the robot.

While the design of the robot was inspired by the topology of a human spine, we were surprised to note that the form, workspace, and mobility of the two tetrahedra joint has a striking resemblance to that of a human shoulder, as seen in Fig. 12. The shoulder serves as a highly effective six degree of freedom joint in the human arm, and is a key component in many high-precision, compliant, dynamic tasks that are currently difficult for robots to accomplish such as writing, painting, and throwing. While traditional robotic manipulators simulate a spherical joint by placing multiple rotary motors in series, which introduces many complications and challenges, this approach keeps the actuators in parallel as we see in real shoulders, and allows for a compliant motion around an actual center of rotation. Thus if it were possible to duplicate such abilities with similar architecture to what is proposed here, this topology could be exploited as a sub-component of many different limbed robots. It also suggests that further investigation into the benefits of non-regular tetrahedra within the structure needs to be conducted.

V. CONCLUSIONS

In this paper we have presented DuCTT, a lightweight, compliant robot which is capable of exploring duct systems. We have discussed the numerous advantages of using tensegrity concepts to distribute forces and have shown that it is possible to overcome problems with highly coupled control by using inverse kinematics. Results from physics simulation have demonstrated that the inverse kinematics controller can be accurate to within a few millimeters when tuned correctly. Implementation of the controller in hardware demonstrated the initial feasibility of the controller, however more work needs to be done to improve overall performance of the prototype.

ACKNOWLEDGMENT

REFERENCES

[1] W. Neubauer, "A spider-like robot that climbs vertically in ducts or pipes," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 2, 1994, pp. 1178–1185 vol.2.

[2] T. White, N. Hewer, B. Luk, and J. Hazel, "The design and operational performance of a climbing robot used for weld inspection in hazardous environments," in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, 1998, pp. 451–455 vol.1.

[3] C. Paul, F. J. Valero-Cuevas, and H. Lipson, "Design and control of tensegrity robots for locomotion," *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 944–957, 2006.
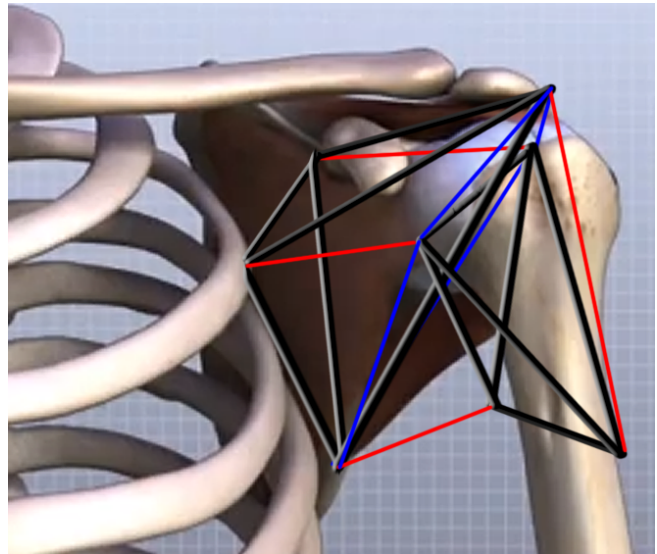
Fig. 12.   Two tetrahedron topology overlayed on human shoulder

[4] 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*, vol. 7, no. 45, pp. 613–621, 2010.

[5] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer, "Learning to control complex tensegrity robots," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1193–1194.

[6] R. E. Skelton and M. C. Oliveira, *Tensegrity systems*. Springer, 2009.

[7] T. Flemons, "The geometry of anatomy," 2007. [Online]. Available: http://www.intensiondesigns. com/geometry of anatomy.html, 2007

[8] M. Adli, K. Nagai, K. Miyata, and H. Hanafusa, "Analysis of internal force effect in parallel manipulators," *Trans. Society of Instrument and Control Engineers*, vol. 27, pp. 27–34, 1991.

[9] H.-J. Schek, "The force density method for form finding and computation of general networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 3, no. 1, pp. 115 – 134, 1974. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0045782574900450

[10] H. C. Tran and J. Lee, "Advanced form-finding of tensegrity structures," *Computers and Structures*, vol. 88, no. 3â"4, pp. 237 – 246, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045794909002636

[11] K. Caluwaerts, J. Despraz, A. Iscen, A. Sabelhaus, J. Bruce, B. Schrauwen, and V. SunSpiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *Journal of the Royal Society Interface (submitted for publication)*, August 2013.

[12] E. Coumans, *Bullet 2.80 Physics SDK Manual*, 2012.