Robust Distributed Control of Rolling Tensegrity Robot

Atil Iscen[1], Adrian Agogino[2], Vytas SunSpiral[3], and Kagan Tumer[4]

[1] Oregon State University, Corvallis, OR, 97331, USA
iscena@onid.orst.edu
[2] UC Santa Cruz / NASA Ames, MS 269-3, Moffett Field, CA 94035, USA
Adrian.K.Agogino@nasa.gov
[3] SGT Inc. / NASA Ames, MS 269-3, Moffett Field, CA 94035, USA
vytas.sunspiral@nasa.gov
[4] Oregon State University, Corvallis, OR, 97331, USA
kagan.tumer@oregonstate.edu

Abstract. Tensegrity structures are based on the idea of balanced structures composed of rigid bodies that are connected only by using tension elements. Robots that are constructed using tensegrity principle can offer many advantages such as being lightweight, impact tolerant or offering unique modes of locomotion. Controlling tensegrity robots has many challenges due to their overall complexity and nonlinear coupling between components. In this paper we overcome these challenges by using multiagent learning methods to control a ball shaped tensegrity robot. Experimental results performed in a soft-body physics simulator show that the single-agent learning system performs 80% better than a hand-coded solution, while the multiagent learning systems performs 100% better. In addition, learning is able to discover diverse control solutions (both crawling and rolling) that are robust against structural failures and can be adapted to a wide range of energy and actuation constraints. We also discuss the different control strategies and hardware implementation of the simulated robot.

Keywords: Robotics, Tensegrity, Multiagent Systems

1 Introduction

Tensegrity structures is based on the idea of building balanced structures that are composed of pure tension and compression elements (cables and rods - see Figure 1). Since there are no bending or shear forces, each component of the structure and overall structure can be lightweight. Moreover, since the rigid components are connected via cables, any external force is internally distributed to the structure. As there are no lever arms, forces do not magnify into joints or other common points of failure. These facts increase system level robustness of tensegrity structures and make them ideally suited to dynamic environments with unpredicted contact forces.

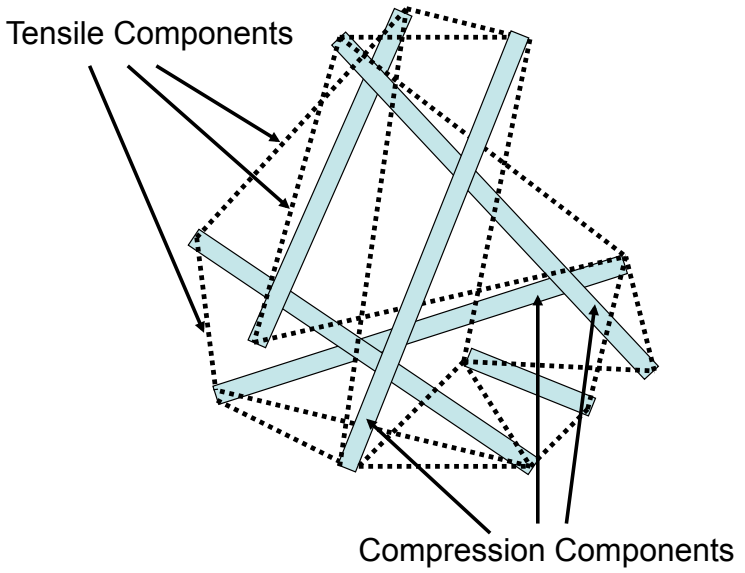

Fig. 1: **Tensegrity Structure.** *Tensegrities are composed of pure tension and pure compression elements (e.g. cables and rods). They can be light-weight, energy-efficient and robust to failures.*

Structures built from tensegrities are well-suited for many robotic tasks where robustness, low-weight and efficiency are desired. Tensegrities tend to be energy efficient as the use of elastic tensile components and dynamical gaits, enable efficient movement . Tensegrities tend to be light-weight as forces align axially with components and shocks distribute through the tensegrity, allowing tensegrities to be made of light-weight tubes/rods and cables/elastic lines. Also tensegrities are naturally distributed systems and can gracefully degrade performance in the event of actuation or structural failure. In addition to these structural properties, tensegrities are also capable of unique modes of locomotion, as they can roll, crawl, gallop, swim or flap wings depending on construction and need.

Despite these desirable properties, tensegrity concept were not used in robotics for many years due to difficult control properties. There are two main properties of tensegrities that make them hard to control with traditional control algorithms. First, a force generated on one part of the tensegrity propagates in a non-linear way through the entire tensegrity. Second, tensegrity robots tend to have oscillatory motions influenced by their interactions with their environment. With many elements connected to each other, these two properties challenge traditional controls. Fortunately these issues can be overcome through the use of a centralized learning algorithm, and performance can be improved further through multiagent learning. Multiagent learning is a natural match for tensegrity control, as the forces in the tensegrity tend to propagate in a distributed way.

By assigning agents to control different portions of the tensegrity, an unified, yet distributed control policy can be achieved.

In this paper, we present how a direct policy search based learning algorithm and a multiagent system can be used to learn control policies that allow a six segment tensegrity to roll through its environment. This paper is organized as follows: Section 2 gives background on tensegrity robots and previous work. Section 3 gives details about the tensegrity robot used in this paper. Section 4 shows how a learning algorithm can be used to create a control policy for our tensegrity robot. Section 5 presents experimental results. Section 6 discusses alternative control strategies and different activation approaches. Section 7 discusses hardware details. Section 8 ends the paper with conclusions and future work.

2 Background and Previous Work

Tensegrity structures are a fairly modern concept, having been initially explored in the 1960's by Buckminster Fuller [7] and the artist Kenneth Snelson [19, 18]. For the first few decades, the majority of tensegrity related research was concerned with form-finding techniques [25, 10, 20, 26, 13, 14] and the design and analysis of static structures [1, 8, 17]. Research into control of tensegrity structures was initiated in the mid-1990's, with initial efforts at formalizing the dynamics of tensegrity structures only recently emerging [17, 11, 24]. The very properties that make tensegrities ideal for physical interaction with the environment (compliance, multi-path load distribution, non-linear dynamics, etc) also present significant challenges to traditional control approaches. A recent review [21] shows that there are still many open problems in actively controlling tensegrities.

There are several approaches that have been taken to control tensegrity robots. Most related to the work in this paper are approaches to locomotion of tensegrity robots using evolutionary algorithms [6]. Paul et al [12] shows two different tensegrity robots that can perform a locomotion movement. These robots perform motion mostly by alternating between different configurations and doing small hops and crawling. Being able to successfully evolve these gaits is impressive given that one of the tensegrities uses only three rods, while the other uses four. However, such simple tensegrities are not able to achieve efficient rolling motion or complex dynamical movements, which is the main goal of this paper.

Instead of learning control policies for tensegrities, more recent work has been done on engineering control algorithms that leverage key features of locomotion [16, 2, 3]. There has also been recent work involving hand tuning of controls for rolling tensegrity robots by body deformation [15, 9, 22, 5]. While this work is able to produce stable smooth dynamics, they are not designed to address the oscillatory nature of tensegrities that come up at high speeds, on uneven terrain, or upon contacts with other objects that occurs in many domains. Instead, with our learning approach, these oscillatory complexities of the tensegrity are implicitly incorporated into the reward function generated from

the physics simulations, and therefore we are able to create dynamical control that can incorporate complexities of the domain as they arise.

3 Target Tensegrity Platform

In this paper we show how controls can be learned on a ball-shaped tensegrity capable of a large range of movement. To do this we choose as our experimental platform, a 6-rod, 24-cable tensegrity as shown in Figure 2. It is chosen since it is one of the simplest tensegrity platforms that can exhibit the following complex behaviors:

- **Many modes of locomotion**: They can crawl, "gallop" and roll, with rolling being an especially efficient and fast mode of locomotion.
- **Robust against failures**: They exhibit enough redundancy that they can recover from hardware failure.
- **Shape changing**: They can change shape to "peer" over things, get unstuck or to move sensors located on tensegrity structure.

These "ball" tensegrities can be useful in many domains, especially those in which a tensegrity has to navigate rugged terrains that can be difficult for wheeled vehicles.

3.1 Structure

The structure of the tensegrity used in this paper is shown in Figure 2. As with all tensegrities, rods never connect directly with other rods. Instead rods are indirectly connected though cables. In the orientation shown in Figure 10 (left) one pair of the rods are parallel to x-axis, another pair is parallel to y axis and the last pair is parallel to z axis. Both ends of the rods are connected via cables. Each end of a rod is connected to the ends of other non parallel rods via 4 different cables. When the structure is in balance, it is symmetrical and convenient for a rolling motion. On the other hand, when an external force is applied, it easily deforms and distributes the force to every component of the structures.

3.2 Controls

The tensegrity is controlled by changing the lengths of the cables. Many physical designs do this by using a motor to pull the cable around a spool that is either interior to the tensegrity or inside a rod. Other concepts involve ways of using dynamic cable twisting or elastomers to change the shape of the cable. In this paper, we discuss possible actuation implementations in Section 7. Also, we analyze possible range limitations of control implementations in Section 5.1.

In principle it would be possible to provide individual controls to each of the 24 cables in our tensegrity. However, to simplify our control problem, the 24 cables are put into 8 groups according to the symmetry of the structure

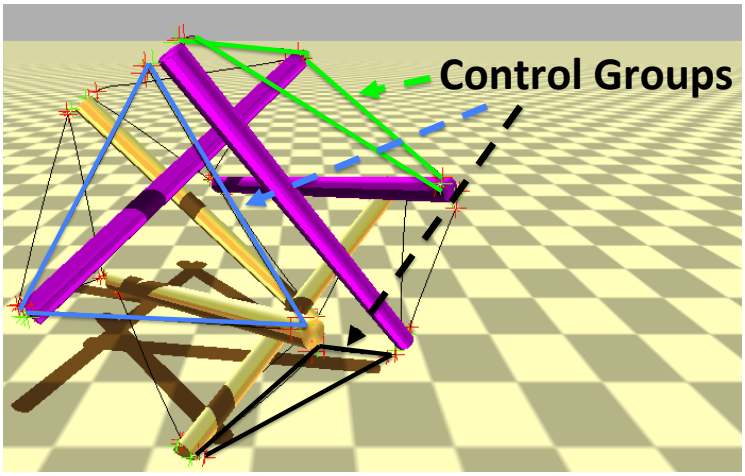

Fig. 2: **Controls**. *Tensegrity controls are broken down into eight groups containing three cables each (three of the groups are shown here). All of the three cables in a group are set to the same target length. Using groups reduces complexity over having to control 24 cables individually.*

(see Figure 2). The structure is symmetrical according to x plane, y plane and z plane, which divides the structure into front - back, left-right and top-down segments. Each group contains 3 cables forming a triangle. Each of these groups is controlled as a whole, with the control algorithm always setting the target length of each of the three cables within a group to be the same.

The control of the robot is done via sinusoidal control of the lengths of the cables. The lengths of the cables change over time according to a sinusoidal signal, and the parameters of the signal are controlled by the agents. The value of the cable is calculated with the formula:

$$y(t) = C + A * sin(\omega t + \phi) \tag{1}$$

where,

- C, represents the center position of the sine wave.
- A, the amplitude, is the peak deviation of the function from its center position.
- ω, the angular frequency, is how many oscillations occur in a unit time interval
- ϕ, the phase, is specifies where in its cycle the oscillation begins at t = 0.

3.3 Simulation

Our tensegrity simulator is built on top of the open-source Bullet Physics Engine [4]. 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 [23]. Cables are represented as nodes with Hooke's-law-like stiffness between them. Therefore our "cables" are actually somewhat elastic and exert a force dependent on their length. We keep our model of actuation abstract in order to explore the best control solutions and then drive requirements back into real hardware design requirements. To enforce additional realism, we prevent the cables being actuated when stretched more than 25%, as an upper limit on the hypothetical motor force. This approach allows us to find the types of control and requirements that will be driven into actuation selection.

4 Learning Controls

While the control parameters of our tensegrity platform are relatively straightforward, the relationship between these parameters is highly complex. In this section we explore how we can use the simulation combined with a reward evaluation to implement a learning algorithm that can learn a set of control parameters that leads to high performance.

4.1 Reward Evaluation

We measure the performance of a simulated tensegrity based on how far it can travel from a starting location within 60 seconds:

$$r = d(C_1, A_1, \omega_1, \phi_1, \cdots, C_8, A_8, \omega_8, \phi_8) \ , \tag{2}$$

where, d is the distance travelled, which is a function of the 32 parameters of the control policy. Note that the decomposition of the distance function d is not readily obtainable in closed form. Instead it must be computed from observing simulations or measured from a physical implementation. Also note that our evaluation does not explicitly take any behavior into account besides distance moved. Tensegrities can exhibit many different gaits, ranging from hopping to rolling, and many different paths, ranging from spirals to straight lines. However, tensegrities that maximize our reward function tend to roll in fairly straight lines. Deviations from this pattern tend to hurt performance.

4.2 Single Agent Learning

In this paper, we perform both single agent learning and multiagent learning. In the single agent case, a single control policy is learned for the entire tensegrity robot. This control policy sets the 32 parameters for the sinusoidal controllers for the eight groups of cables. The algorithm is a simple population-based direct policy search that tries to learn a policy that maximizes our reward function. At the beginning of training, a population of n random policies is created and evaluated based on our reward function r. After each round of learning, the

Algorithm 1: Multiagent learning algorithm for tensegrity control.

```
Data: Population of n elements for each agent
for i=1..15 do
    random team ← ∅ ;
    forall the Populations do
        random team ← random agent;
    end
    score = evaluate random team ;
    forall the agents ∈ random team do
        if score > agent.score then
            agent.score = score ;
        end
    end
end
forall the Populations do
    order the population;
    eliminate last k;
    copy first k to last k;
    set score of last k to MIN;
    mutate last k;
end
```

worst k policies are removed, and are replaced by mutated versions of the best k policies [5]. As learning progresses, the population tends to converge to higher performance policies.

4.3 Multiagent Learning

In addition to single agent learning, we perform multiagent learning, where one agent is assigned to each control group. Therefore there are 8 agents total, and each agent is responsible for setting the values for the 4 parameters of the sinusoidal controller used for that group. These 4 parameters represent the control policy of the agent. The goal of each agent is to create a control policy that helps maximize the overall system reward function r defined in Equation 2 when combined with the control policies of all the other agents.

In our multiagent learning system, each agent has a population of n policies. First the performance of individual agent policies is assessed. This is done by first creating 15 full system policies by sampling the agent policies uniformly. Each system policy is then evaluated according to the full system reward r. Each agent policy is then given the evaluation of the full system policy that it participated in that received the highest reward. After the evaluation step, for each agent the lowest performing k policies are removed, and are replaced by

[5] This can be seen as a simple form of evolution with no cross-over. In related experiments, no large performance differences are seen between this and non-population based reinforcement learning

the mutated versions of the best k policies. The pseudo-code for this algorithm is as follows:

4.4 Hand-Coded Solution

In addition to creating control policies through learning, we explore how to hand-code a control solution using the same parameters available to the learning systems. The goal here is to explore the challenges of hand-coding a solution and to see how well or best effort compares to our learned solutions. It turns out that creating a control policy by hand using our 32 parameters is very difficult, and the best achieved solution barely moved. This problem will only get more difficult as we scale the tensegrity robots to more complex versions with more elements. To improve performance, we reduced the parameter space by hand coding the amplitudes of each group and making the oscillation frequency the same for all groups. The results shown later in this paper are for this second, better-performing hand-coded solution.

5 Experimental Results

In this section, we present experiments evaluating the performance of our learning methods to control tensegrity robots in the physics simulator described in Section 3.3. The goal of our experiments is to evaluate whether learning systems can be successfully applied to tensegrity robots under nominal conditions, and how robust these solutions are to limitations in the range of actuation, to actuator noise and to a physical breakage in a cable of the tensegrity. For the nominal condition case we test the following methods of creating the controller:

- **Hand Coded** Control policy is developed by hand to try to achieve maximum performance.
- **Single Agent Learning** A single control policy is learned for the entire tensegrity robot.
- **Multiagent Learning** A multiagent system learns the control policy for tensegrity robot, with one agent assigned to each of the 8 control groups.

We then test the robustness of our highest performance solution (multiagent learning as shown below) in the following ways:

- **Actuation Range** We limit how far the cables are allowed to contract, to simulate designs where range may be limited and to simulate control modes where low-power locomotion is needed.
- **Actuation Noise** We add noise to how far cables are actually moved as compared to how far they are being requested to move.
- **Cable Failure** We test performance when a single cable in the robot breaks.

All experiments start with a stationary tensegrity robot on the ground. For each experiment, the robots are created on a flat surface, and after 5 seconds of

stabilization time, active control of the cables starts. The agents are given fixed amount of time (60 seconds) to move the robot as far as possible. The evaluation function is the distance between the starting position and the position at the end of given time period. The population size in the policy search is set to $n = 10$ and the selection parameter is set to $k = 5$. We perform 10 statistical runs for each type of experiment. Using a t-test we confirm that our conclusions are statistically significant.

5.1 Nominal Conditions

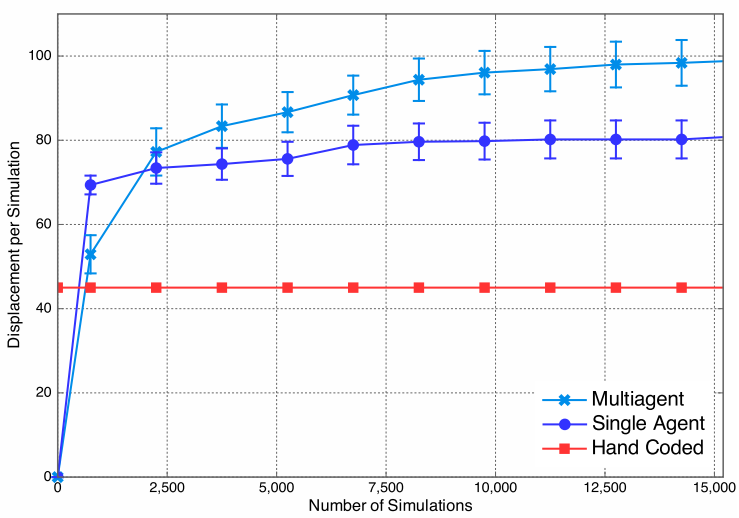

Fig. 3: **Evolutionary Approach vs. Hand Coded Agorithm**. *The policies are evaluated according to how far the tensegrity can move in 60 seconds. Single agent approach performs better than the hand coded policy. Multiagent approach performs the best.*

The first experiment compares three different control policies: Hand-coded, single agent learning and multiagent learning. Figure 3 shows that both learning approaches can easily outperform the hand coded solution. The multiagent learning approach provides the best performance by moving 20% more quickly than the single agent and 100% more than our hand coded agent. Both single agent and multiagent algorithms are able to achieve smooth rolling motions as shown in Figure 4. Note that while our hand coded tensegrity is not able to achieve a rolling motion, we are not trying to imply that this problem is impossibly complex for non-learning algorithms. In fact there have been several successful algorithms to do this [15, 9, 22, 5]. Instead we are illustrating that it is

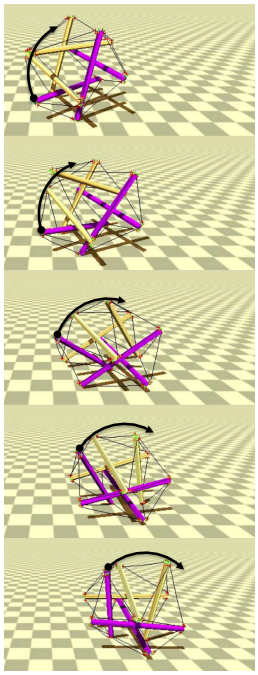

Fig. 4: **Tensegrity Dynamics**. *Tensegrity is able to achieve smooth rolling motion. This rolling is accomplished solely by changing the length of the cables. Our learned control policies produce rolling that is also dynamical as the tensegrity does not stop to setup next roll action. This type of rolling can be fast and highly efficient.*

in fact quite difficult to create these controls, and that the single agent and multiagent learning algorithms are creating complex, non-trivial control solutions. In addition a multiagent framework has the potential to be adapted to many different complex tensegrities with less effort than hand coding an algorithm for each new tensegrity.

5.2 Actuation Range Limitations

In the next experiment, we test different maximum actuation ranges for the controller. The maximum change in the rest length of a cable length is varied from 1% of the size of a tensegrity rod to 40%. Limiting the actuation range is done to both simulate situations where our actual hardware has limited actuation range (i.e. long range pulley/cable actuators, vs. short range electro-elastomers), and to simulate situations where we want to reduce power requirements by limiting actuation. Figure 5 shows that for multiagent controllers, after a 10% maximum actuation range, additional range does not gain any more advantage. On the other hand, decreasing these parameters results in robots that move less quickly.

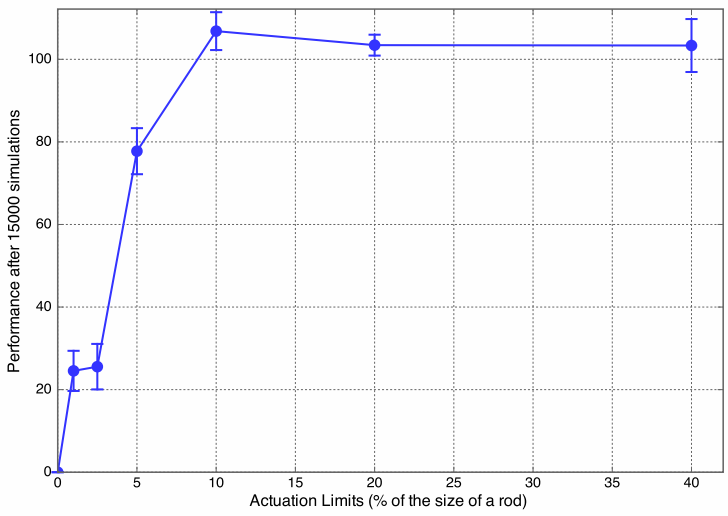

Fig. 5: **Performances of Robot under Actuation Limitations**. *Limits on the range that a cable can be contracted and expanded. The performance tops with an actuation range equal to 10% of the length of a tensegrity rod. Lower actuation range reduces speed, yet increasing actuation range beyond 10% does not increase performance.*

A controller that can only change its cable length 5% can only move the tensegrity at 75% of the speed compared to a controller that can change the cable length 10%. If we further decrease the range of actuation, performance declines even more.

5.3 Actuation Noise

To measure the robustness of our learning approach against noise, we test the multiagent tensegrity robot in an environment with different levels of actuation noise. Actuation noise is applied at every time step to the sinewave that the agents generate to control the cables. At every time step, noise is directly added to the value of the Equation 1. To test different levels of noise, we use different environments where the standard deviation is set to $1\%, 2\%, 5\%, 10\%, 25\%, 50\%, 100\%$ of the amplitude of the sine wave for each cable.

In this experiment, we test two different policies: 1) A policy derived from a multiagent system that had learned in an environment without noise, and 2) A policy derived from a multiagent system that had learned in the noisy environment. For each level of noise, agents that are tested are trained in an environment with that specific amount of noise. Figure 6 shows that the tensegrity that is trained without noise still has tolerable performance, but its performance is significantly lower than what is is in a non-noisy environment. When we train

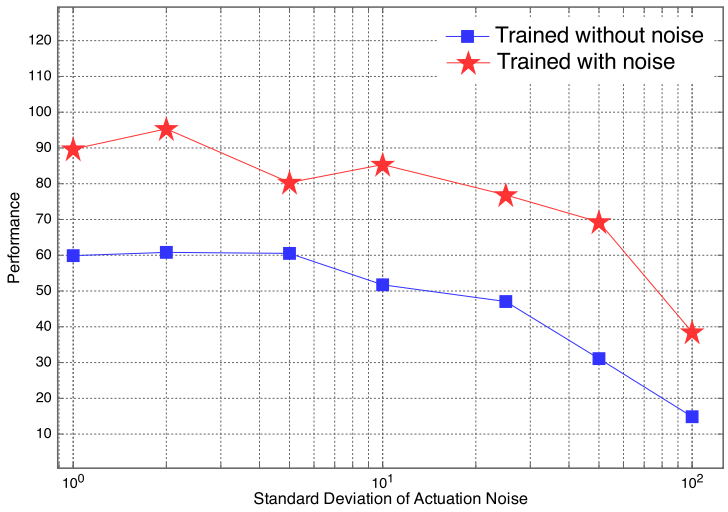

Fig. 6: **Performances of Robot under Actuation Noise**. *Robots are tested in environments having gaussian noise added to the control amplitude with standard deviations of 1, 2, 5, 10, 25, 50 and 100 % of the maximum control range. The agents that are trained in a noisy environment are able perform smooth rolling motion even in the presence of high actuation noise.*

the agents with noise, it can be seen that they can perform 50% better in low-noise environments ($1\% - 10\%$) and 100% better in high-noise environments ($50\% - 100\%$) than the agents that are trained without noise. This is an impressive result, as it shows that the solutions generated are not highly specific to an exact model of a tensegrity and exact environmental conditions. Instead the solutions appear highly generalizable.

5.4 Broken Cable

The fourth experiment tests the robustness of the structure and the controller. We take the same tensegrity structure, but remove one of the cables. The removal of one cable not only decreases our ability to control the tensegrity, but also disrupts the balance of the structure. With the cable removed, the structure is not symmetrical anymore and it can not keep its ball shape by default. To be able to compensate for the broken cable, we trained a controller with a high range of actuation (40%) as well as a controller with a medium range of actuation (10%). Although these two controllers score the same when used on an unbroken tensegrity, the results change when they control the tensegrity with a broken cable (Figure 8). While the controller with a medium activation range can no longer perform well, the controller with a high range of activation is able to still perform decently with a broken cable. This result shows that while having a larger range of available motion may not be valuable under nominal conditions,

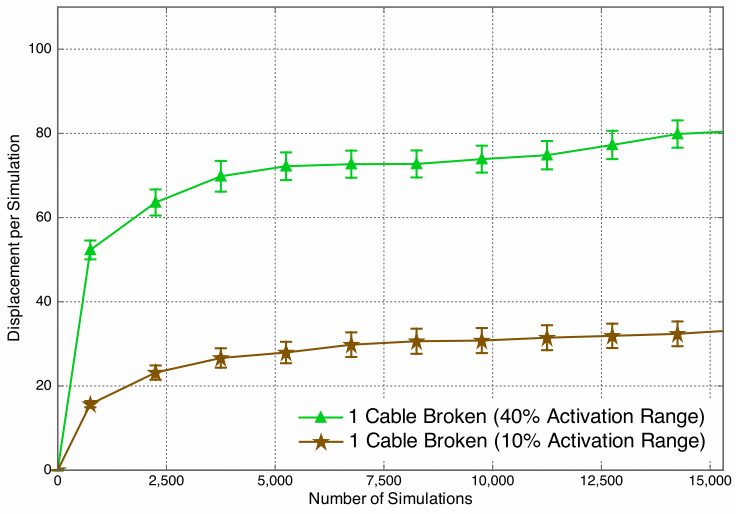

Fig. 7: **Multiagent Learning with a Tensegrity Robot with Broken Cable**. *The robot with 40% activation range can still move the robot despite the missing cable.*

under adverse conditions, we can learn a controller that takes advantage of the larger range of motion to effectively move the tensegrity robot. Note that this result does not show that the learned control policy dynamically adapts to problems, since in this experiment we retrain our policy after the breakage. However, it does show the flexibility of the learning process. In addition, in many situations a may be possible to upload solutions derived in simulation to disabled robots in the field. This could be especially useful when the robot is highly inaccessible.

6 Alternative Control Strategies

In our simulations we took a specific approach to control our 6 rods 24 muscles tensegrity. We first grouped 24 muscles into 8 groups of 3 and assigned each group to a different agent. This approach simplifies the problem, but there are different control approaches that can be taken. Currently, we are working on two different approaches to control the same tensegrity. First one is assigning different agents for each cable. With this approach, the problem becomes harder to learn with 24 agents learning to cooperate to find a solution. On the other hand, search space is much bigger allowing different policies that cannot be created using 8 control groups. To be able to use such an approach, we are working on different multiagent learning and coevolutionary algorithms methods.

On the other hand, a completely different controls approach is using payload muscles as illustrated in Figure 10. In addition to the base structure, we add a payload to the center of the tensegrity and connect the payload to the end of

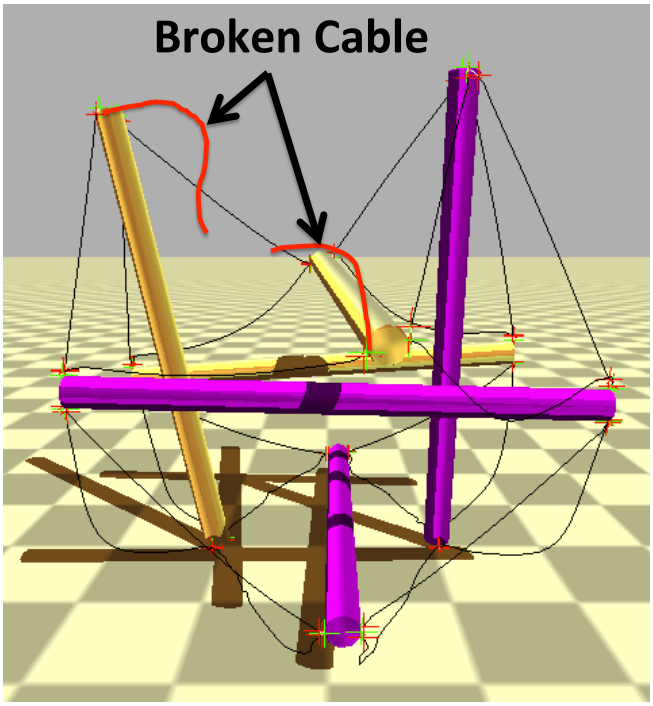

Fig. 8: **Tensegrity Robot with Broken Cable**

the rods with 12 cables. At this point, we still have the base tensegrity and its nice properties, and we have the payload in the middle protected from external impacts by the rods and elasticity of the cables. Considering this structure with 36 muscles (24 shell and 12 payload muscles) it allows a completely different controls approach. While keeping 24 shell muscles uncontrolled but stiff, one can control the tensegrity using 12 payload muscles. One advantage of this approach is relevant to the hardware design. Instead of placing actuators to the rods, they can be placed at one central location. Our early experimentation with this approach shows that multiagent learning can provide smooth rolling behavior by controlling payload muscles. Comparing two approaches visually, we were able to determine that the way the tensegrity deforms itself is much different, but both methods provide rolling motion.

7 Hardware Robot

With the actuation requirements explored in simulation, and building on our experience with prior prototype tensegrity robots, we will be spending this year researching appropriate actuation technologies and building a prototype of the rolling tensegrity robot discussed in this paper. Our existing prototype tensegrity robot uses position controlled spooled-cable actuation, and we will explore

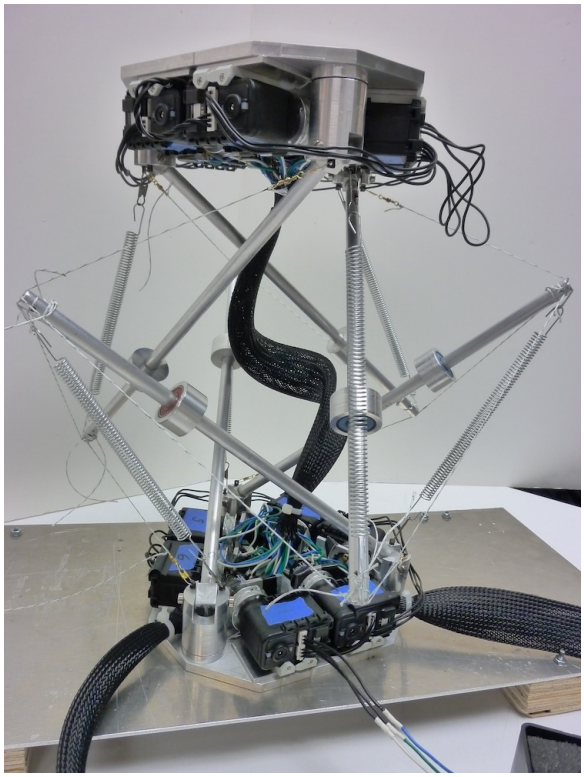

Fig. 9: **Experimental Tensegrity Robot Prototype.** *This 6-rod tensegrity robot is designed to test hardware implementation and shape-changing abilities of tensegrities. We are in the process of building 6-rod tensegrity that can roll.*

two new approaches: Impedance Controlled (Tension and Position) Spooled Cable actuation, and Twisted Cable Actuation. Our existing prototype robot is already designed for spooled cable actuation and we will retrofit it with new sensors and controls to support Impedance Control. In parallel we will evaluate a novel "twisted cable" actuation approach that we believe will allow for the use of significantly smaller and energy efficient motors due to the decoupling of motor torque output from actuator tension output. Finally these two actuator approaches will be evaluated for design simplicity, power efficiency, and total system mass, and the best approach will be used on our new rolling tensegrity robot. This new robot is designed to validate the controls approaches explored here and to show that these tensegrity robots can be used as landing and mobility systems.

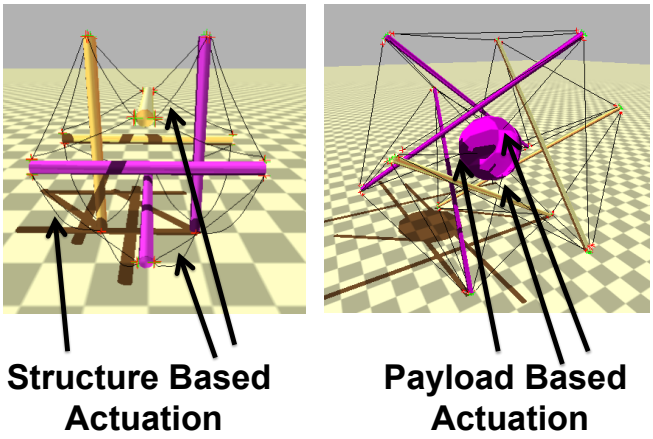

Fig. 10: *Payload based actuation concentrates the actuators at the center and controls payload muscles instead of structure muscles.*

8 Conclusions and Future Work

Tensegrity robotics matched with multiagent learning systems have a promising future. The structural properties of tensegrities give them many beneficial properties, while their distributed nature makes them a perfect match for multiagent systems. In this paper, we introduce a first step to this promise. We first show that in simulation a direct policy search algorithm is able to learn an effective controller that allows a moderately complex tensegrity ball to roll. Then we show how performance can be improved by applying a multiagent learning system to this same tensegrity robot. Not only is the multiagent system able to produce a smooth rolling motion for the tensegrity robot, it is able to do so under a wide range of adverse conditions, including actuation limitations, actuation noise and cable breakage. These results show that multiagent learning systems are a strong candidate for tensegrity control. In addition, the high level of robustness may allow our multiagent framework now used in simulation to be used on our physical tensegrities now in development.

The multiagent learning system used in this paper represents just a glimpse of what may be possible for tensegrity control. While the distributed nature of a tensegrity makes it a natural match to the distributed nature of a multiagent system, the multiagent system we use in this paper is actually not as distributed as it could be. While all the agents take independent actions, they all try to maximize the same global system reward. Their use of this global reward can cause agents to take into account too much information and limit their ability to learn quickly. In contrast, future research may show that it is possible to use agent-specific rewards that are more relevant to an agent's particular action. In addition, it may be possible to partition agents into more distributed sets. Such changes could allow multiagent systems to be used for even more complex tensegrities and achieve more sophisticated control behaviors.

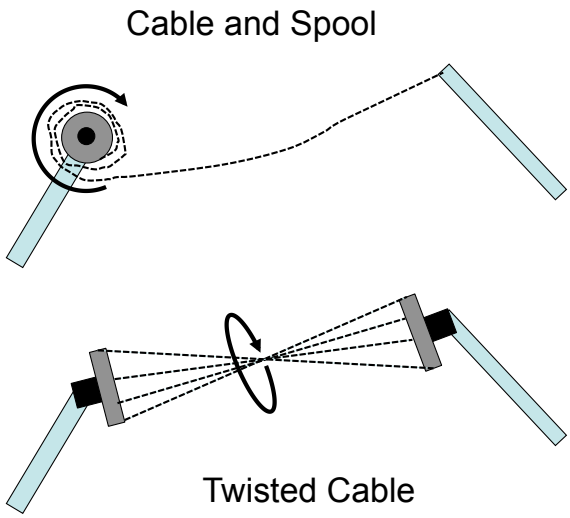

Fig. 11: *Actuation can be performed by pulling in a cable with a spool. Alternatively a bundle of cables can be twisted and the action of twisting and untwisting the cable will change the length of the bundle.*

References

1. Bel Hadj Ali, N., Rhode-Barbarigos, L., Pascual Albi, A., Smith, I.: Design optimization and dynamic analysis of a tensegrity-based footbridge. Engineering Structures 32(11), 3650–3659 (2010)
2. Böhm, V., Jentzsch, A., Kaufhold, T., Schneider, F., Becker, F., Zimmermann, K.: An approach to locomotion systems based on 3d tensegrity structures with a minimal number of struts. In: Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on. pp. 1–6. VDE (2012)
3. Böhm, V., Jentzsch, A., Kaufhold, T., Schneider, F., Zimmermann, K.: An approach to compliant locomotion systems based on tensegrity structures. Proc. of the 56th IWK, TU Ilmenau (2011)
4. BulletPhysicsEngine: http://www.bulletphysics.org/
5. Calisti, M., Arienti, A., Renda, F., Levy, G., Hochner, B., Mazzolai, B., Dario, P., Laschi, C.: Design and development of a soft robot with crawling and grasping capabilities. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 4950–4955. IEEE (2012)
6. Fujiia, M., Yoshiia, S., Kakazub, Y.: Movement control of tensegrity robot. Intelligent Autonomous Systems 9: IAS-9 9, 290 (2006)
7. Fuller, B.: Tensegrity. Portfolio and Art News Annual 4, 112–127 (1961)
8. Klimke, H., Stephan, S.: The making of a tensegrity tower. In: IASS Symposium. Montpellier (2004)
9. Koizumi, Y., Shibata, M., Hirai, S.: Rolling tensegrity driven by pneumatic soft actuators. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 1988–1993. IEEE (2012)
10. Masic, M., et al.: Algebraic tensegrity form-finding. International Journal of Solids and Structures 42, 4833–4858 (2005)

11. Motro, R.: Tensegrity: structural systems for the future. Butterworth-Heinemann (2003)
12. Paul, C., Valero-Cuevas, F., Lipson, H.: Design and control of tensegrity robots for locomotion. Robotics, IEEE Transactions on 22(5), 944–957 (2006)
13. Paul, C., Lipson, H., Cuevas, F.J.V.: Evolutionary form-finding of tensegrity structures. In: Proceedings of the 2005 conference on Genetic and evolutionary computation. pp. 3–10. GECCO '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1068009.1068011
14. Pugh, A.: An introduction to tensegrity. Univ of California Press (1976)
15. Shibata, M., Hirai, S.: Moving strategy of tensegrity robots with semiregular polyhedral body. In: Proc. of the 13th Int. Conf. Climbing and Walking Robots (CLAWAR 2010), Nagoya. pp. 359–366 (2010)
16. Shibata, M., Saijyo, F., Hirai, S.: Crawling by body deformation of tensegrity structure robots. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. pp. 4375–4380. IEEE (2009)
17. Skelton, R.E., Oliveria, M.C.: Tensegrity Systems. Springer, New York (2009)
18. Snelson, K.: http://www.kennethsnelson.net/
19. Snelson, K.: Continuous tension, discontinuous compression structures. united states patent 3169611 (Feburary 1965)
20. Tibert, et al.: Review of form-finding methods for tensegrity structures. International Journal of Space Structures 18, 209–223 (2003)
21. Tur, J.M.M., Juan, S.H.: Tensegrity frameworks: Dynamic analysis review and open problems. Mechanism and Machine Theory 44, 1–18 (2009)
22. Tur, J.: On the movement of tensegrity structures. International Journal of Space Structures 25(1), 1–14 (2010)
23. Wittmeier, S., Michael, J., Dalamagkidis, K., Rickert, M.: CALIPER : A Universal Robot Simulation Framework for Tendon-Driven Robots. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1063–1068 (2011)
24. Wroldsen, A., de Oliveira, M., Skelton, R.: A discussion on control of tensegrity systems. In: Decision and Control, 2006 45th IEEE Conference on. pp. 2307–2313. IEEE (2006)
25. Zhang, J.Y., Ohsaki, M.: Adaptive force density method for form-finding problem of tensegrity structures. International Journal of Solids and Structures 43, 5658–5673 (2006)
26. Zhang, L., et al.: Form-finding of nonregular tensegrity systems. Journal of Structural Engineering 132, 1435–1440 (2006)