

# Controlling Tensegrity Robots through Evolution

Atil Iscen  
Oregon State University  
Corvallis, OR, 97331, USA  
iscena@onid.orst.edu

Vytas SunSpiral  
SGT Inc. / NASA Ames  
MS 269-3, Moffett Field, CA 94035, USA  
vytas.sunspiral@nasa.gov

Adrian Agogino  
UC Santa Cruz / NASA Ames  
MS 269-3, Moffett Field, CA 94035, USA  
Adrian.K.Agogino@nasa.gov

Kagan Tumer  
Oregon State University  
Corvallis, OR, 97331, USA  
kagan.tumer@oregonstate.edu

## ABSTRACT

Tensegrity structures (built from interconnected rods and cables) have the potential to offer a revolutionary new robotic design that is light-weight, energy-efficient, robust to failures, capable of unique modes of locomotion, impact tolerant, and compliant (reducing damage between the robot and its environment). Unfortunately robots built from tensegrity structures are difficult to control with traditional methods due to their oscillatory nature, nonlinear coupling between components and overall complexity. Fortunately this formidable control challenge can be overcome through the use of evolutionary algorithms. In this paper we show that evolutionary algorithms can be used to efficiently control a ball shaped tensegrity robot. Experimental results performed with a variety of evolutionary algorithms in a detailed soft-body physics simulator show that a centralized evolutionary algorithm performs 400% better than a hand-coded solution, while the multiagent evolution performs 800% better. In addition, evolution 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. These successful controls will form the basis for building high-performance tensegrity robots in the near future.

## Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning

## General Terms

Algorithms, Design

## Keywords

Evolution, Robotics, Tensegrity

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.  
Copyright 2013 ACM 978-1-4503-1963-8/13/07 ...\$15.00.

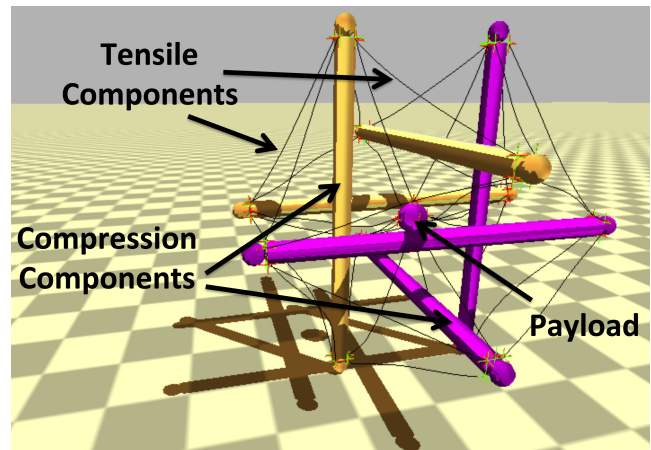


Figure 1: **Tensegrity Structure.** Tensegrities are composed of pure tension and pure compression elements (e.g. cables and rods). They can be light-weight, energy-efficient and robust to failures.

## 1. INTRODUCTION

Tensegrity robots are part of an exciting emerging field of soft-body robotics that are entirely composed of pure tension and compression elements (cables and rods - see Figure 1). These structures are made of axially loaded compression elements encompassed within a network of tensional elements, and thus each element experiences either pure linear compression or pure tension. As a result, individual elements can be extremely lightweight as there are no bending or shear forces that must be resisted. A unique property of tensegrity structures is how they can internally distribute forces. As there are no lever arms, forces do not magnify into joints or other common points of failure. Rather, externally applied forces distribute through the structure via multiple load paths, creating a system level robustness and tolerance to forces applied from any direction. Thus tensegrity structures can be easily reoriented and are ideally suited for operation in dynamic environments where contact forces cannot always be predicted.

Tensegrities have a number of beneficial properties including:

- **Light-weight:** 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.

- **Energy efficient:** Through the use of elastic tensile components and dynamical gaits, efficient movement is possible.
- **Robust to failures:** Tensegrities are naturally distributed systems and can gracefully degrade performance in the event of actuation or structural failure.
- **Capable of unique modes of locomotion:** Tensegrities can roll, crawl, gallop, swim or flap wings depending on construction and need.
- **Impact tolerant and compliant:** Since forces are distributed upon impact, they can fall or bump into things at moderate speed. In addition, their compliance ensures that they do minimal damage to objects they contact.
- **Naturally distributed control:** Characteristics of force propagation in tensegrities allows effective local controllers.

The last property is the most subtle but important. In “traditional” robots, distributed controls becomes messy due to the need to communicate global state information to all the controllers with high precision, and thus often undermines the very promise of distribution. Fundamentally, this stems from the fact that a rigidly connected structure will magnify forces internally through leverage, and will accumulate force into joints. Thus, the actions of a local distributed controller can have disproportionate global consequences. These consequences can require a certain amount of global coordination and state management, undermining the value of the local controller. Tensegrity structures are different, due to the tension network, there is no leverage in the structure. Thus, forces *diffuse* through the structure, rather than accumulate in joints. As a result, actions by a local controller diffuse through the structure, integrating with all the other local controllers. While any one local controller will impact the structure globally, that impact is locality relevant and not magnified via leverage. Thus, the structure enables true distributed control, because local actions stay (predominately) local.

Despite these desirable properties, tensegrity robots have remained mostly a novelty for many years due to difficult control properties that make them hard to control with traditional control algorithms such as:

1. **Complex oscillatory motions:** Tensegrity robots tend to have oscillatory motions influenced by their interactions with their environment.
2. **Elastic Nonlinear distributed interactions:** A force generated on one part of the tensegrity propagates in a nonlinear way through the entire tensegrity, causing shape changes, which further change force propagations.

Fortunately the combinatorial optimization capabilities of evolutionary algorithms combined with the distributed properties of multiagent systems are a natural match to these problems. Evolutionary algorithms can learn complex control policies that maximize a performance criterion without

needing to handle the oscillatory motions and distributed interactions explicitly. In addition, increased performance can be achieved by assigning evolving agents to different control points throughout the tensegrity. Then as a multi-agent system, the agents can co-evolve to create a unified control policy.

In this paper, we present how both centralized evolution as well as cooperative coevolutionary algorithms (CCEA) can be used to learn control policies that allow a six segment tensegrity to roll through its environment. We present different approaches for fitness and test the best one against different environmental conditions. 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 different approaches to evolving a control policy for the tensegrity robot. Section 5 presents experimental results. Section 6 discusses hardware details and Section 7 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 [6] and the artist Kenneth Snelson [19]. For the first few decades, the majority of tensegrity related research was concerned with form-finding techniques [25, 9, 20, 15, 26, 12, 14] and the design and analysis of static structures [1, 7, 18]. 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 [18, 10, 24]. The very properties that make tensegrities ideal for physical interaction with the environment (compliance, multi-path load distribution, nonlinear dynamics, etc.) also present significant challenges to traditional control approaches. A recent review [22] 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 [5]. Paul et al [13] 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 mainly due to shape constraints of the structure used.

Instead of evolving control policies for tensegrities, more recent work has been done on engineering control algorithms that leverage key features of locomotion [17, 2]. There has also been recent work involving hand tuning of controls for rolling tensegrity robots by body deformation [16, 8, 21, 4]. 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 collisions with other objects that occurs in many domains. Instead, with our evolutionary approach, these oscillatory complexities of the tensegrity are implicitly incorporated into the fitness evaluation 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

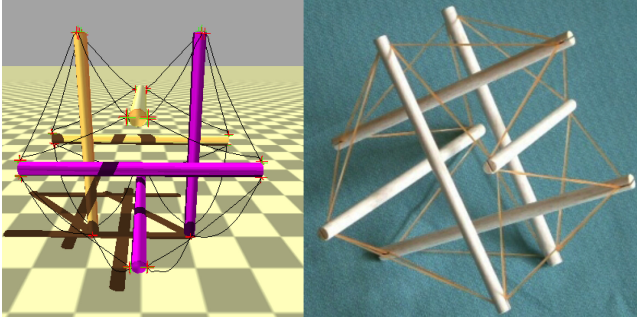


Figure 2: **Structure for Tensegrity Robot.** *This six-rod design is one of the simplest designs that can behave as a “ball.” It is capable of rolling, changing shapes, and can be robust against failures.*

In this paper we show how controls can be evolved 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. Rods do not connect directly with other rods, instead, rods are indirectly connected through cables, resulting in a continuous tension network as the primary load transfer system of the structure. In the orientation shown in Figure 2 (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. 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 structure.

In addition to the base tensegrity, we attached a ball shaped payload to the center of the tensegrity via an additional 8 payload cables. The payload represents the essential parts of the robot, such as computing, sensors, batteries, or other instruments. As opposed to the 24 outer cables, these 8 payload cables have constant length and are not actively controlled in this work (Figures 1,6).

### 3.2 Controls

The tensegrity is controlled by changing the lengths of the cables. Many hardware implementations do this by using a motor to wind the cable onto a spool that is either interior to the tensegrity or inside a rod. Other concepts involve using dynamic cable twisting or elastomers to change the length of the cable. In this paper we do not consider the hardware implementation, though we do limit our abstract model of actuation to reasonable performance characteristics for velocity, acceleration, and string elasticity.

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 the output of the evolutionary algorithm. The length of each cable is calculated with the formula:

$$y(t) = C + A * \sin(\omega t + \phi) \quad (1)$$

where,

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

By using 24 sinusoidal signals for 24 cables, overall control of the tensegrity is based on 96 ( $24 * 4$ ) parameters.

### 3.3 Simulation

Our tensegrity simulator is built on top of the open-source Bullet Physics Engine [3]. 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]. The simulated tensegrity structure has a size of 10 meters for each rod due to the fact that the physics engine is more precise for objects approximately that size. 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. EVOLUTIONARY ALGORITHMS

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 fitness evaluation to implement an evolutionary algorithm that can evolve a set of control parameters that leads to the desired behavior.

## 4.1 Evaluation Function

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

$$f = d(C_1, A_1, \omega_1, \phi_1, \dots, C_{24}, A_{24}, \omega_{24}, \phi_{24}), \quad (2)$$

where,  $d$  is the distance travelled, which is a function of the 96 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 fitness function tend to roll in fairly straight lines. Deviations from this pattern tend to hurt performance.

## 4.2 Centralized Evolutionary Algorithms

In this paper, we perform both centralized evolution and multiagent coevolution. In the centralized case, a single control policy is evolved for the entire tensegrity robot. This control policy sets the 96 parameters for the sinusoidal controllers. The algorithm is a simple evolutionary algorithm designed to maximize our fitness function. At the beginning of training, a population of  $n$  random policies is created and evaluated based on our fitness function  $f$ . After each round of evolution, the worst  $k$  policies are removed, and are replaced by mutated versions of the best  $k$  policies. Mutation is uniformly random for each parameter. For experiments that use crossover, we use a simplest basic single point crossover. For single point crossover algorithm, in addition to mutation, new  $k$  policies are assigned to teams of two, and for each team, those two policies exchange part of their parameters. The crossover point is a randomly selected number between 1 and 96 (number of parameters) and all the parameters after this point are exchanged. In both cases (crossover and no crossover), as evolution progresses, the population tends to converge to higher performance policies.

## 4.3 Cooperative Coevolutionary Algorithms

While centralized approaches often produce good results, evolving a centralized controller can be slow due to the size of the search space. If the control parameters are tightly coupled, then the centralized approach may be the best we can hope for. However, control of a tensegrity robot is not necessarily tightly coupled. While changing the length of a cable will strongly affect neighboring components, it will have less of an effect on components on the opposite side of the tensegrity. Therefore decentralized control is possible and can greatly reduce the search space for each component in the decentralized controller.

In this paper, we introduce such a decentralized controller as a cooperative coevolutionary system, using principles derived from multiagent systems. In this paradigm each of the 24 cables is controlled by an individual agent. The job of each agent  $i$  is to control the four parameters of the sinusoid controlling its cable:  $C_i$ ,  $A_i$ ,  $\omega_i$  and  $\phi_i$ . To do this, each agent  $i$  will have its own population of control parameters,  $p_i$ , where each member of the population,  $p_{i,j}$  specifies the four control parameters. Each agent then evolves its popu-

lation to produce good values of control parameters in the context of the control choices of the other agents. The goal of these coevolutionary agents is still the same as the centralized evolutionary algorithm: Maximize the global fitness evaluation,  $f$ .

The advantage of this paradigm is that each agent now only has to search through four parameters. The difficulty is the value of each agent's choice of control parameters now depends on the choices of the other agents. The exact same choice may be good or bad, depending on what other agents are doing. One way to address this is to evaluate every possible team existing at the current generation. In this paper, this is not possible due to the number of possible cooperators. We attempt to handle this issue by taking a fixed number of samples of the population at each generation of evolution. Between two generations, each agent's population is sampled  $s$  time and they are put together to form  $s$  complete control policies. Each of these control policies is then evaluated with respect to our global fitness evaluation function,  $f$ , producing  $s$  evaluations. Typically the number of samples  $s$  will be many times larger than the size of the population, therefore each individual member of a population will typically be part of multiple control policies. The main issue is how we evaluate a member of a population  $p_{i,j}$  which has participated in multiple control policies, when each control policy has received a different global fitness evaluation  $f$ .

In this paper, we address this issue in the following three ways:

1. Generational Average - Assign the mean value of the global fitness evaluations for control policies in which the member participated during this generation.
2. Leniency - Assign the highest value of the global fitness evaluations for control policies in which the member participated in.
3. Historical Average - Assign the mean value of global fitness evaluations, averaging across all the generations that the member survived.

Taking average and leniency are commonly used methods with cooperative coevolution [11]. To improve performance, we augment this approach by taking historical averaging. Historical averaging uses all the samples that the members had been part of, not only the current generation but also the ones with previous generations with previous teammates (Algorithm 1). Every surviving agent carries its history to the new generation and uses that information to calculate its average score.

## 4.4 Hand-Coded Solution

In addition to creating control policies through evolutionary algorithms, we explore how to hand-code a control solution using the same parameters available to the evolutionary system. The goal here is to explore the challenges of hand-coding a solution and to see how well our best effort compares to our learned solutions. It turns out that creating a control policy by hand using our 96 parameters is extremely difficult, so we created 8 control groups with 3 cables in each group. These 3 cables form a triangle that have the same length, making it easier to write a hand-coded controller. Even with this simpler approach, the best achieved solution barely moved. This problem will only get more difficult

---

**Algorithm 1:** Cooperative coevolutionary Algorithm with Historical Average

---

```
Data: Population of  $n$  elements for each agent
for  $i=1..15$  do
  random team  $\leftarrow \emptyset$ ;
  forall the Populations do
    random team  $\leftarrow$  random agent;
  end
  score = evaluate random team;
  forall the agents  $\in$  random team do
    agent.history  $\leftarrow$  agent.score;
  end
end
forall the Populations do
  forall the agents do
    agent.fitness = average(agent.history);
  end
  order the population;
  eliminate last  $k$ ;
  copy first  $k$  to last  $k$ ;
  set score of last  $k$  to  $MIN$ ;
  mutate last  $k$ ;
  clear history for last  $k$ ;
end
```

---

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 evolution-based methods to control tensegrity robots in the physics simulator described in Section 3.3. The goal of our experiments is to evaluate whether evolutionary 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.
- **Centralized Evolution** A single control policy is learned for the entire tensegrity robot.
- **Decentralized Evolution** Cooperative Coevolutionary Algorithms (CCEA) approach.

We then test different fitness methods for CCEAs: Average, Leniency, Historical Average. We test the robustness of our highest performance solution (historical average) in the following ways:

- **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.

- **Obstacles** We randomly place half sphere obstacles into the environment.

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 a 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 the 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. All our major results are  $p < 0.95$ .

### 5.1 Centralized, Decentralized and Hand-Coded

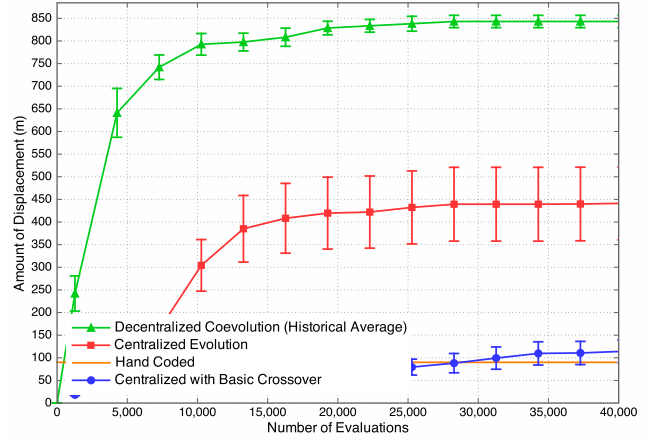


Figure 3: **Performance of Control Algorithms.** *Multiaгент coevolution performs significantly better than other methods since it is able to take advantage of distributed nature of tensegrity control.*

The first experiment compares three different control policies: Hand-coded, centralized evolution and multiagent coevolution. Figure 3 shows that both evolution-based approaches can easily outperform the hand coded solution. Looking at the converged policies, the multiagent evolution approach provides the best performance by moving 100% more quickly than the single agent and 400% more than our hand coded agent. Both centralized and decentralized evolution are able to achieve smooth rolling motions as shown in Figure 6. 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-evolving algorithms. In fact there have been several successful algorithms to do this [16, 8, 21, 4]. Instead we are illustrating that it is in fact quite difficult to create these controls, and that the centralized and multiagent evolutionary 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.

It can be seen that the tensegrity controlled by policies evolved from coevolution reaches a performance around 900 meters in 60 seconds. By observing the behavior, we confirmed that the movement is established by smooth rolling



motion as illustrated in Figure 6. This rate corresponds to the tensegrity moving at approximately 28 revolutions per minute.

## 5.2 Historical Average, Average and Leniency

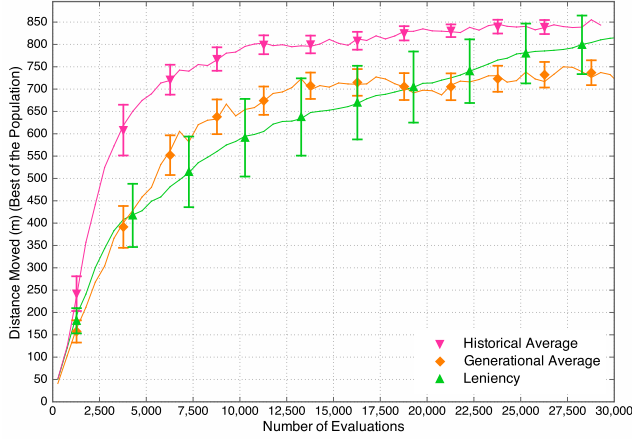


Figure 4: **Evaluation Methods.** Using a historical average reaches best score and it is consistent. Lenient learners have bigger error bars.

The second experiment compares different fitness assignment methods for CCEAs. As it can be seen, using a historical average performs better than every other method. Moreover, the small error bars signifies that using the historical average consistently provides similar performance. We are further analyzing historical average as a future research. Looking at the error bars of the lenient learners, it can be seen that standard deviation is high: It also reaches very good policies in some of the statistical runs, but the average success is lower.

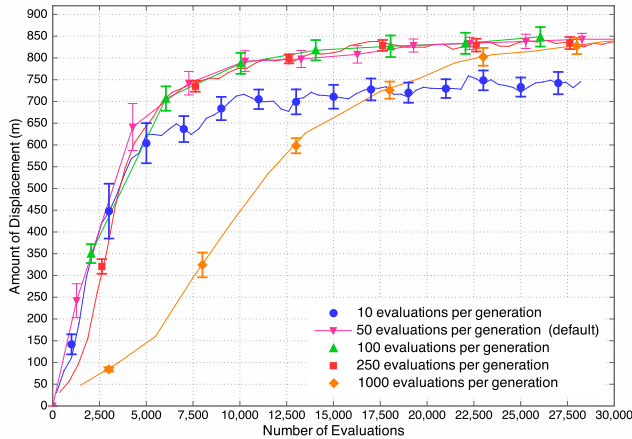


Figure 5: **Sample Size.** Taking more samples for each trial improves stability but takes more time reach better behavior

To make sure that the sample size that we chose does not significantly affect the results, we tested the historical average method with different sample size between each generation ( $s$ ). Figure 5 shows that the sample size affects performance when it is too low such as 10, on the other hand,

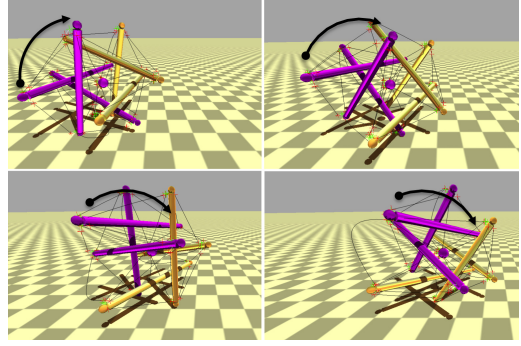


Figure 6: **Tensegrity Dynamics.** Tensegrity is able to achieve smooth rolling motion. This rolling is accomplished solely by changing the length of the cables. Our learned control policies produce rolling that is also dynamical as the tensegrity does not stop to setup next roll action. This type of rolling can be fast and highly efficient.

a sample size as high as 1000 decreases the learning speed. Considering this result, we used 50 as the default value for the rest of the experiments.

## 5.3 Actuation Noise

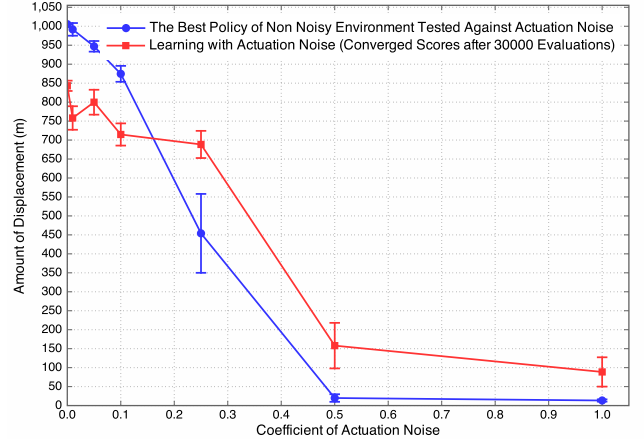


Figure 7: **Actuation Noise.** The best policy can take up to 25% noise, evolving in a noisy environment scores better in higher noise environments.

To measure the robustness of our evolutionary algorithm 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 evolutionary algorithm generate to control the cables. At every time step, a random value from a normal distribution is directly added to the value of the Equation 1. To test different levels of noise, we use different environments where the standard deviation of the noise 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 in our noisy environment: 1) Policies from a multiagent system that had learned in an environment without noise, and 2) Policies that are learned in an environment with the same

amount of noise they are tested in. Figure 7 shows that tensegrities trained both with and without noise can perform remarkably well when the level of noise is below 15%. This is an impressive result, as it shows that the solutions generated in a non-noisy environment are not highly specific to an exact model of a tensegrity and exact environmental conditions. Instead the solutions appear highly generalizable. Beyond this level of noise performance goes down significantly. However, while performance is low, a tensegrity trained with a high level of noise can still perform at a baseline level while subjected to high levels of noise, which could be very useful in many situations.

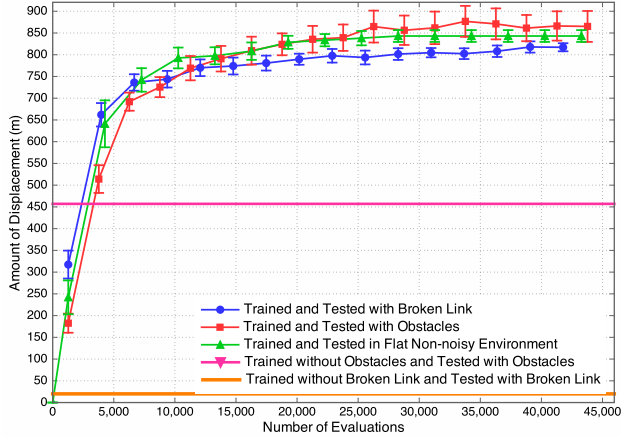


Figure 8: **Robustness Tests with Obstacles and Broken Link.** *Coevolutionary Algorithms can overcome both obstacles and broken link scenarios*

## 5.4 Broken Cable and Obstacles

The fourth experiment tests the robustness of the structure and the controller. In our first experiment we remove one of the cables, which decreases controllability and also disrupts the balance of the structure. With the cable removed, the structure is no longer symmetrical and it can not keep its ball shape by default. In our second experiment we place the normal tensegrity in an environment containing randomly placed half sphere obstacles. When we test our best policy trained in a perfect environment in these two conditions, the performance drops to 50% with obstacles and it cannot roll with a broken link (Figure 8). On the other hand, if we perform evolution in these conditions, the evolutionary algorithm can still find successful locomotion policies for a tensegrity with broken cable, or in an environment with obstacles.

This result shows that under adverse conditions we can evolve a controller that takes advantage of the large range of motion inherent in tensegrity robots to effectively maintain locomotion. Note that this result does not show that the evolved 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 evolution process and the ability to pre-evolve controls associated with potential failure modes.

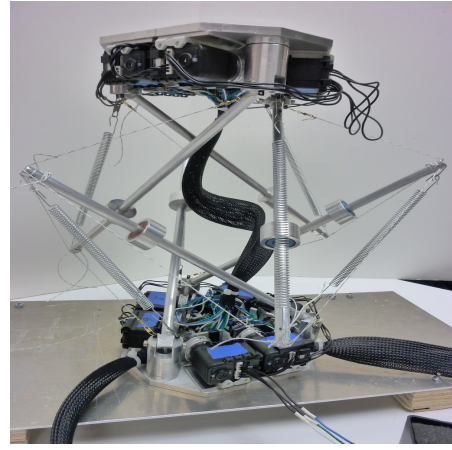


Figure 9: **Tensebot, our Experimental Tensegrity Robot Prototype.** *This 6-rod tensegrity robot is designed to test hardware implementation and shape-changing abilities of tensegrities. We are in the process of building 6-rod tensegrity that can roll.*

## 6. 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 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.

## 7. CONCLUSIONS AND FUTURE WORK

Tensegrity robotics matched with multiagent evolutionary 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, evolutionary algorithms are able to come up with an effective controller that allows a moderately complex tensegrity ball to roll. Then we show how performance can be improved by applying a multiagent coevolutionary system to this same tensegrity robot. To handle fitness assignment problems in coevolution, we use sampling, introduce a new method based on the historical average and compare it with average and lenient learners. For future work, we are working on a de-

tailed analysis of this method compared to averaging and leniency.

Not only is the coevolutionary approach 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 noise, obstacles and cable breakage. These results show that multiagent evolutionary systems are a strong candidate for tensegrity control. In addition, the high level of robustness may allow our framework now used in simulation to be used on our physical tensegrities now in development.

The multiagent evolutionary 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 fitness evaluation function. Their use of this global evaluation function can cause agents to take into account too much information and limit their ability to evolve quickly. In contrast, future research may show that it is possible to use agent-specific evaluation functions that are more relevant to an agent's particular action. Such changes could allow coevolving systems to be used for even more complex tensegrities and achieve more sophisticated control behaviors.

## Acknowledgments

This research was partially supported by the NASA Innovative Advanced Concepts (NIAC) Program. Tensebot constructed with support from Idaho Space Grant Consortium

## 8. REFERENCES

- [1] N. Bel Hadj Ali, L. Rhode-Barbarigos, A. Pascual Albi, and I. Smith. Design optimization and dynamic analysis of a tensegrity-based footbridge. *Engineering Structures*, 32(11):3650–3659, 2010.
- [2] V. Böhm, A. Jentzsch, T. Kaufhold, F. Schneider, F. Becker, and K. Zimmermann. 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*, pages 1–6. VDE, 2012.
- [3] BulletPhysicsEngine. <http://www.bulletphysics.org/>.
- [4] M. Calisti, A. Arienti, F. Renda, G. Levy, B. Hochner, B. Mazzolai, P. Dario, and C. Laschi. Design and development of a soft robot with crawling and grasping capabilities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4950–4955. IEEE, 2012.
- [5] M. Fujii, S. Yoshii, and Y. Kakazub. Movement control of tensegrity robot. *Intelligent Autonomous Systems 9: IAS-9*, 9:290, 2006.
- [6] B. Fuller. Tensegrity. *Portfolio and Art News Annual*, 4:112–127, 1961.
- [7] H. Klimke and S. Stephan. The making of a tensegrity tower. In *IASS Symposium*, Montpellier, 2004.
- [8] Y. Koizumi, M. Shibata, and S. Hirai. Rolling tensegrity driven by pneumatic soft actuators. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1988–1993. IEEE, 2012.
- [9] M. Masic and et al. Algebraic tensegrity form-finding. *International Journal of Solids and Structures*, 42:4833–4858, 2005.
- [10] R. Motro. *Tensegrity: structural systems for the future*. Butterworth-Heinemann, 2003.
- [11] L. Panait, K. Tuyls, S. Luke, and P. Kaelbling. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, page 2008.
- [12] C. Paul, H. Lipson, and F. J. V. Cuevas. Evolutionary form-finding of tensegrity structures. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 3–10, New York, NY, USA, 2005. ACM.
- [13] C. Paul, F. Valero-Cuevas, and H. Lipson. Design and control of tensegrity robots for locomotion. *Robotics, IEEE Transactions on*, 22(5):944–957, 2006.
- [14] A. Pugh. *An introduction to tensegrity*. Univ of California Press, 1976.
- [15] J. Rieffel, F. Valero-Cuevas, and H. Lipson. Automated discovery and optimization of large irregular tensegrity structures. *Comput. Struct.*, 87(5-6):368–379, Mar. 2009.
- [16] M. Shibata and S. Hirai. Moving strategy of tensegrity robots with semiregular polyhedral body. In *Proc. of the 13th Int. Conf. Climbing and Walking Robots (CLAWAR 2010), Nagoya*, pages 359–366, 2010.
- [17] M. Shibata, F. Saijyo, and S. Hirai. Crawling by body deformation of tensegrity structure robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4375–4380. IEEE, 2009.
- [18] R. E. Skelton and M. C. Oliveria. *Tensegrity Systems*. Springer, New York, 2009.
- [19] K. Snelson. Continuous tension, discontinuous compression structures. united states patent 3169611, February 1965.
- [20] Tibert and et al. Review of form-finding methods for tensegrity structures. *International Journal of Space Structures*, 18:209–223, 2003.
- [21] J. Tur. On the movement of tensegrity structures. *International Journal of Space Structures*, 25(1):1–14, 2010.
- [22] J. M. M. Tur and S. H. Juan. Tensegrity frameworks: Dynamic analysis review and open problems. *Mechanism and Machine Theory*, 44:1–18, 2009.
- [23] S. Wittmeier, J. Michael, K. Dalamagkidis, and M. Rickert. CALIPER : A Universal Robot Simulation Framework for Tendon-Driven Robots. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1063–1068, 2011.
- [24] A. Wroldsen, M. de Oliveira, and R. Skelton. A discussion on control of tensegrity systems. In *Decision and Control, 2006 45th IEEE Conference on*, pages 2307–2313. IEEE, 2006.
- [25] J. Y. Zhang and M. Ohsaki. Adaptive force density method for form-finding problem of tensegrity structures. *International Journal of Solids and Structures*, 43:5658–5673, 2006.
- [26] L. Zhang and et al. Form-finding of nonregular tensegrity systems. *Journal of Structural Engineering*, 132:1435–1440, 2006.