Growing and Evolving Vibrationally Actuated Soft Robots

Benjamin Berger
Union College
807 Union Street
Schenectady, New York

Alvin Andino
Union College
807 Union Street
Schenectady, New York

Andrew Danise
Union College
807 Union Street
Schenectady, New York

John Rieffel
Union College
807 Union Street
Schenectady, New York
rieffelj@union.edu

ABSTRACT

Designing soft robots is difficult, time-consuming, and non-intuitive. Soft robot design faces two main challenges: structure and control. This research uses generative encodings to grow structures and a vibrational mechanism to control locomotion. In this paper, we demonstrate the ability to successfully evolve soft robots that can move when vibrated. Soft bodies are grown through a grammatical process and simulated in the Bullet physics engine. We also briefly outline a method of evolving scalable solutions that we are currently investigating. It should be capable of generating soft robots of various sizes that can move when vibrated.

CCS Concepts

•**Computing methodologies** → **Genetic algorithms;**

Keywords

Soft robot; genetic algorithm; vibrationally actuated; generative encoding

1. INTRODUCTION

Robots are becoming increasingly useful: automating physical tasks and performing beyond human capabilities of strength and precision. Robots allow us to operate in environments that can be hazardous for humans, such as performing search and rescue missions at a failing nuclear facility. Typical modern robots contain motors and are made of metal or are comprised of rigid structures. However, soft robots can be made entirely out of plastic, rubber, silicone, or a variety of materials that allow them to be flexible and change shape [16]. For example, a snake-like soft robot can change shape to slither its way in between rubble of a collapsed building to look for potential survivors [10]. Some robots are being designed to grasp objects of unusual shape or of a delicate nature (like internal organs during surgery) [8, 2].

GECCO'15, July 11-15, 2015, Madrid, Spain.

ISBN 978-1-4503-3488-4/15/07...$15.00

DOI: http://dx.doi.org/10.1145/2739482.2768485

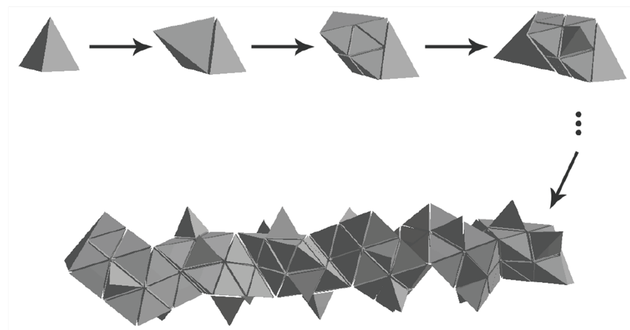

Figure 1: A tetrahedral mesh that is grown using a generative encoding. The robot grows a little larger and more complex each time the rules of the encoding are applied. One application of the ruleset can also be referred to as a "face rewrite". Picture from [14].

Designing soft robots is incredibly difficult. Designers face two main challenges in soft robotics: structure and control. The structure of a robot is highly dependent on its purpose, and the control mechanisms are dependent on the structure. Modify the structure of the robot and the control mechanisms are useless. Modify the control mechanisms and the structure may behave differently, rendering it useless. This creates a "chicken and egg" problem that makes engineering soft robots complex and time-consuming for humans [13]. This research uses generative encodings to tackle the structure problem and a simple vibrational mechanism to tackle the control problem. We evolve the structure of the robot to fit the vibrational mechanism, kept consistent between individuals. Our goal is to evolve the right structure to leverage the vibrational mechanism as a source of locomotion.

2. BACKGROUND AND RELATED WORK

Researchers have investigated automated methods for designing soft robots with genetic algorithms [13, 15, 4, 12]. To automate the selection process, researchers test designs in computer simulation to avoid physically constructing every generated robot.

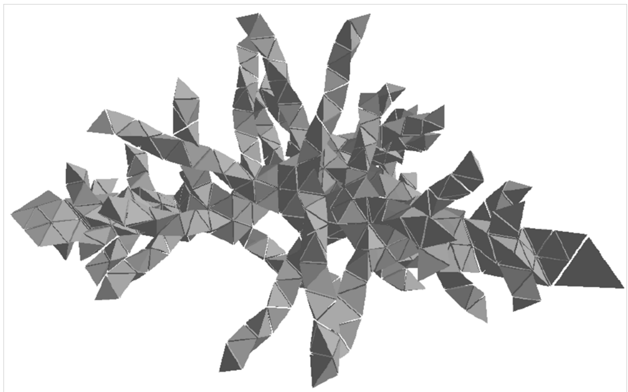

Figure 2: An example of a tetrahedral mesh grown using a generative encoding. Picture from [14].

2.1 Generative Encodings

A generative encoding is a set of rules that specify how a robot should be constructed. Each generative encoding represents a soft robot in a similar way that DNA can represent an organism. When an organism's DNA is changed, the traits that are expressed are changed. Similarly, when the rules of a generative encoding are altered, the resulting structure of the soft robot will be different. In this sense, the generative encoding is the genotype, while the resulting 3D model is the phenotype.

Generative encodings have existed for awhile in the computer science research community. Also known as formal grammars and L-systems, generative encodings have been used from natural language processing [6] to algorithmically generating plant structures [11]. Hornby and Pollack used generative encodings in combination with genetic algorithms to evolve tables (like the furniture) [7]. They showed that generative encodings produced better results at a faster rate than its non-generative counterpart. Additionally, Cheney *et al* used both generative encodings and direct encodings to evolve soft-bodies [3]. They concluded that the generative encodings outperformed the direct encodings, noting that the generative encodings evolved bodies with "homogeneous patches of materials akin to tissues" while the direct encodings produced seemingly random designs.

An interesting property of generative encodings is that they can create robots of various sizes from one ruleset. This is akin to building a brick wall. If you have a set of rules for laying bricks, and follow these rules for 10 bricks, you will build a small wall. Using the same rules, you can lay 100 bricks to build a larger wall, or 1000 bricks to build an even larger wall. Similarly with generative encodings, the more times the rules are applied, the larger and more complex the robot will grow. In this sense, one genotype can create numerous phenotypes. Figure 1 illustrates this principle well with a tetrahedral mesh. Researchers typically fix a set number of times to apply the rules of the generative encoding *a priori* when using them in conjunction with genetic algorithms. However, this can evolve solutions that are not scalable, a situation which is discussed further in Section 4.1.

2.2 Actuation Through Vibration

Since soft robots are made of flexible materials, the structures naturally bend and deform slightly under a stress. Conventional engineering of rigid structures tries to minimize vibration and mitigate the effect of resonance frequencies. For example, if a bridge vibrates and hits its resonance frequency, the structure can collapse (like the infamous Tacoma Narrows Bridge). However, researchers want to exploit the flexible material properties of soft robots. For example, researchers have successfully implemented vibrationally based locomotion systems in tensegrity structures [9, 1]. Tensegrity structures are comprised of tensile and rigid components and are capable of deforming and changing shape. Vibrating the structures at certain frequencies can cause them to move. Changing the frequency can alter the direction or velocity that the tensegrity moves.

This inspires our work with soft robots.

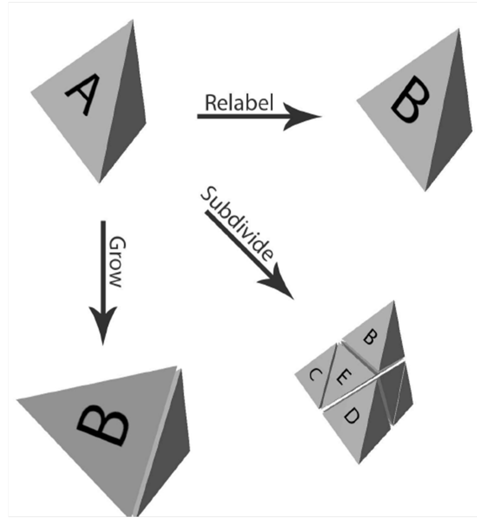

Figure 3: Example rules used in a generative encoding for tetrahedral meshes. Each open face of a tetrahedron is labeled and can have a rule applied to it according to what the generative encoding specifies. The process of evolving a generative encoding modifies the set of rules to produce new designs. Picture from [14].

3. PRELIMINARY RESULTS

This research uses generative encodings to represent soft robot designs like the one in Figure 2. The generative encoding is designed to produce tetrahedral meshes [14]. This was chosen because physics engines (PhysX and Bullet) represent soft bodies as tetrahedral meshes in simulation. Additionally, STL files used for 3D printing represent structures as tetrahedral meshes, making it convenient to print and test robots in the real world (see Figure 5). Each generative encoding is comprised of a different combination of the rules depicted in Figure 3. Robots are grown by applying the rules of the generative encoding multiple times (fixed *a priori*), illustrated in Figure 1.

Robots are simulated in the Bullet physics engine (chosen for its support of soft-bodies, extensive documentation, and an active online community). Every robot is embedded with a vibrational mechanism, represented in the simulation as a mass that rotates around an axis. The vibrational mecha-

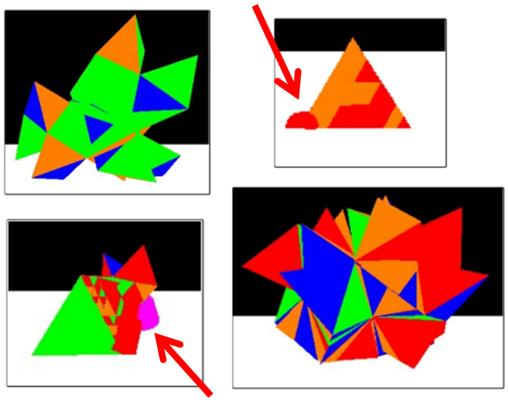

Figure 4: These are some robots that were created and tested in simulation as part of the evolutionary process. Each individual is represented by a different generative encoding, each of which is comprised of a different combination of the rules depicted in Figure 3. Every robot is embedded with a vibrational mechanism, represented in the simulation as a pink mass that rotates around a red cylinder. The red arrows indicate the position of the vibrational mechanism, most clearly visible for the robot on the bottom left.

nism is attached to the root tetrahedron and is set to not collide with the soft-body. Some examples of the robots in simulation are shown in Figure 4.

The fitness of each robot is determined by its displacement in the XZ plane. This was chosen to promote the development of designs that can travel large distances when vibrated. To keep things simple, the path that the robot takes is ignored; only start and end positions are considered. In order to discourage individuals from exploiting the fitness function by simply falling over or rolling, we begin to measure displacement after a brief time delay, allowing structures to settle before evaluation. Parents for subsequent generations are chosen via fitness proportional selection. New individuals are created through mutation 60% of the time and crossover 40% of the time. We have not tested other rates of mutation and crossover at this time.

After running the experiments, we have successfully evolved soft-robot designs that move when vibrated. The progress of the evolutionary process is depicted in Figure 6.

4. FUTURE WORK

4.1 Scalable Generative Encodings

The current methods used to evolve generative encodings do not create scalable solutions. Fixing a set number of times to apply a ruleset *a priori* creates soft robots that are only guaranteed to be fit for one size. Using the brick wall analogy mentioned in Section 2.1, this would be like creating a ruleset that only works well if you lay 1000 bricks. If you use the same ruleset to lay 100 bricks, your wall is not guaranteed to be as good. Viswanathan and Pollack discussed how preselecting a fixed size of a generative encoding can retard evolutionary progress [17]. By not evaluating multiple

Figure 5: A 3D printed robot designed by our generative encoding. A pager motor can be embedded into the structure to vibrate it, testing how far the design moves outside of simulation. Picture from [14]

.

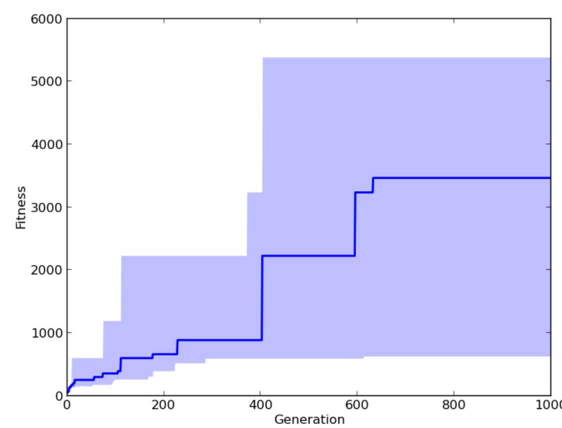

Figure 6: Maximum, minimum, and median fitnesses of best individuals per generation across five runs of the experiment. For this set of experiments, each generative encoding was expanded 40 times. We used a population of 10 individuals for each run.

phenotypes produced by the genotype, it can take more generations to achieve a specified level of fitness. Devert *et al* discuss how evolving generative encodings is directly linked to the Halting Problem [5]. The stopping criteria of the developmental process is key for certain types of problems, affecting the robustness and scalability of solutions.

We are currently investigating a method of creating scalable generative encodings, capable of producing robots of various sizes that can move when vibrated. This method involves assigning designs to categories (small, medium, large, etc). Each generative encoding is evaluated by how far the robots it produces for each category can move when vibrated. One design is deemed better than another if it can dominate across all categories (*pareto dominance*). We hypothesize that this method will allow us to evolve scalable generative encodings.

4.2 Physical Testing

Another avenue for future research is applying this process to physical robots. Evolved designs can be 3D printed and embedded with a pager motor. Researchers can experiment with different materials, or combinations of materials, and see if they can accurately model these materials in simulation. If printing and testing physical designs is streamlined enough, we could start the process of "rapid prototyping". This could allow us to replace some of the simulation with real-world experiments, providing a more accurate evaluation for our genetic algorithms.

5. CONCLUSION

In this paper, we have described a method to evolve soft robots that can move when vibrated, along with a generative encodings to grow soft robots in an open-ended fashion. We have also discussed how pareto evolution over developmental time-scales may help address issues of scalability.

In upcoming work, we are looking forward to fabricating soft robots using flexible materials like silicone and rubber, as well as 3D printing them. These physical prototypes can then be used to validate our experimental results.

Soft robots have incredible appeal in domains such as urban search-and-rescue. However, the coupled problems of design and control need to be more fully addressed before soft robots can have a tangible impact on society. Our research is helping to address these issues.

6. REFERENCES

[1] V Bohm and Klaus Zimmermann. Vibration-driven mobile robots based on single actuated tensegrity structures. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5475–5480. IEEE, 2013.

[2] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.

[3] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 167–174. ACM, 2013.

[4] Andrew Danise. *Evolving Soft Robots with Vibration Based Movement*. Undergraduate thesis, Union College, Schenectady, NY, 2014.

[5] Alexandre Devert, Nicolas Bredeche, and Marc Schoenauer. Robustness and the halting problem for multicellular artificial ontogeny. *Evolutionary Computation, IEEE Transactions on*, 15(3):387–404, 2011.

[6] Eli Goldberg, Norbert Driedger, and Richard I Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53, 1994.

[7] Gregory S Hornby and Jordan B Pollack. The advantages of generative grammatical encodings for physical design. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 600–607. IEEE, 2001.

[8] Filip Ilievski, Aaron D Mazzeo, Robert F Shepherd, Xin Chen, and George M Whitesides. Soft robotics for chemists. *Angewandte Chemie*, 123(8):1930–1935, 2011.

[9] Mark Khazanov, Julian Jocque, and John Rieffel. Developing morphological computation in tensegrity robots for controllable actuation. In *Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion*, pages 1049–1052. ACM, 2014.

[10] Ming Luo, Weijia Tao, Fuchen Chen, Tri K Khuu, Selim Ozel, and Cagdas D Onal. Design improvements and dynamic characterization on fluidic elastomer actuators for a soft robotic snake. In *2014 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2014.

[11] Przemyslaw Prusinkiewicz, Aristid Lindenmayer, and James Hanan. The algorithmic beauty of plants. *The virtual laboratory (USA)*, 1990.

[12] John Rieffel. Heterochronic scaling of developmental durations in evolved soft robots. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 743–750. ACM, 2013.

[13] John Rieffel, Davis Knox, Schuyler Smith, and Barry Trimmer. Growing and evolving soft robots. *Artificial life*, 20(1):143–162, 2014.

[14] John Rieffel and Schuyler Smith. A face-encoding grammar for the generation of tetrahedral-mesh soft bodies. In *ALIFE*, pages 414–420, 2010.

[15] John Rieffel, Francisco Valero-Cuevas, and Hod Lipson. Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures*, 87(5):368–379, 2009.

[16] Robert F Shepherd, Filip Ilievski, Wonjae Choi, Stephen A Morin, Adam A Stokes, Aaron D Mazzeo, Xin Chen, Michael Wang, and George M Whitesides. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 108(51):20400–20403, 2011.

[17] Shivakumar Viswanathan and Jordan Pollack. How artificial ontogenies can retard evolution. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 273–280. ACM, 2005.