

TJHSST Senior Research Project

Evolving Motor Techniques for Artificial Life

2007-2008

Kelley Hecker

November 2, 2007

Abstract

This project simulates evolving virtual creatures in a 3D environment, based on ideas proposed by Karl Sims in his research paper "Evolving Virtual Creatures." Creatures' movements are calculated via an internal neuron system which produces effector output values. Generations of creatures are evaluated based on specific fitness tests and the most successful instances are used in the next generation. This eventually allows for the evolution of sophisticated motor skills and unique movement techniques. Creatures should also have the ability to evolve physically to allow for more sophisticated movement with the addition of specialized body parts.

Keywords: genetic algorithms, artificial life, evolving creatures

1 Introduction

1.1 Rational

Creatures that can learn to do seemingly simple tasks, such as walking, are the first step to even more complex simulated organisms. While it is easy to tell a program to do something, having it learn on its own and produce unique results is much more difficult. If we can begin to understand and program the beginnings of evolution, then this opens up the possibility of creating and simulating much more complex creatures.

1.2 Purpose and Goal

My purpose is to program a simulation which can model evolving creatures. Starting with random motions, new and more complex patterns for movement should develop. Different methods would be compared through various fitness tests, and the most successful creatures would be reused in the next generation.

Ideally, creatures should be able to develop unique and efficient methods of moving, as well as evolve physically to allow for even more complex movements. Specialized limbs could evolve to further allow creatures to improve their techniques. Unique evolution would allow for creatures specialized in specific areas to also be feasible.

While categorized under the wide scope of Artificial Intelligence programming, my simulation more specifically addresses the idea of simulating creatures that can learn. Rather than hard-coding and simulating basic AI functions, such as telling a creature to move to a colder area when it becomes too hot or find water when it is thirsty, my simulation attempts to have creatures learn on their own and develop their own forms of walking. This should allow for unique creatures and specialization.

This project creates a platform for much more complex simulated evolution. Future creatures could evolve and perfect more sophisticated techniques, such as learning to hunt, create tools, and communicate.

My project addresses the relatively simple problem of creating creatures that can learn to walk. Although compared to more complex functions such as communication and organization this may seem basic, I want to explore walking creatures to its fullest extent. I would like the simulation to produce many unique creatures when run multiple times. Also, creatures should be able to develop specialized limbs and extra body parts to allow for more complex techniques. The ability for specialized creatures, such as for running or swimming, should also be present.

2 Background

2.1 Similar Projects

This project is based on research proposed in Karl Sims' paper "Evolving Virtual Creatures." His paper introduces the idea of virtual creatures compared with various fitness evaluation functions to simulate creatures specialized to

walk, swim, jump, and follow. He proposes a simulation that can create an infinite number of unique possible creatures with interesting movement strategies.

Another project similar to my own is also an implementation of Karl Sims' research. Nicolas Lassabe created a number of virtual creatures using Sims' method. Rather than generating creatures that can swim, jump, and follow, some of his creatures are able to climb stairs, work together to push blocks, and push a skateboard.

2.2 Relevant Theory

1. "Evolving Virtual Creatures," Karl Sims

The most relevant theories are those proposed by Karl Sims in "Evolving Virtual Creatures," since that is what my research is based on. Sims' creature morphology is based on a node with children representing the connected parts. A node contains information for a body part as well as directions for implementing the parts. A body part can create instances of its children, as well as recursively call itself.

The control structures for these creatures is based on a series of input sensors, neurons, and output effectors. Sensors take in information from the simulated world, pass it to neurons within the creature's "brain," and this in turn produces effector values with control the creature's next movement. A creature's brain can have many neurons, and neurons can also be associated with a specific body part. Neurons can perform functions that are completely based on their inputs, such as divide and greater-than, or they can have functions that are also dependent on variables such as time, like oscillate-wave. Neurons can also have a constant value.

Behavior selection is performed by comparing the results of various fitness evaluations. Creatures with high fitness levels are bred, and their offspring replace those that did not survive. Breeding creatures together allows for genetic mutation as well as genetic crossover and grafting.[1]

2. Robotic Introspection: Self Modeling

Researchers from Cornell University created a starfish-like robot that learns to walk through self-modeling. The robot moves its limbs and

creates a 3D model of itself, which is then used to test different possible walking techniques. After the best movement model is chosen, the creature then executes the given motion sequence. This process can be repeated to further refine the creature's movement or to create a new sequence.[2]

3. Framsticks

Framsticks is very similar to Karl Sims' research. It models both creature bodies and their control systems. However, Framsticks allows for the capacity to simulate more complex evolution. It can do simple optimization of techniques, but can also spontaneous evolution and ecosystem modeling.[3]

4. Nicolas Lassabe

Nicolas Lassabe, a doctorate student at the Universite Toulouse, has taken Sims' work and created a simulation of it in breve. This is more or less what I want to achieve, although my creatures may not be able to reach the same level of complexity as Lassabe's. His website includes videos of his simulated creatures, but no code or description of his work (he simply references Karl Sims).[4]

5. Sugarscape

Sugarscape is a program much more focused on community development, rather than the evolution of a single creature. The creatures of Sugarscape gather sugar. This simple task, however, can later lead to much more complex simulation. Sugarscape can model migration, grouping, and competition.[5]

3 Development

3.1 Overview

My simulation will be written in a 3D simulator software called breve. The language used by the simulator is steve.

In order for my simulation to be successful, a creature will have to develop a better method for walking than its predecessors. The success of a

specific method will be measured by how far the creature moves in a given amount of time: the faster it moves, the more successful the technique.

Many different systems need to be evaluated before evolution can occur. Basic physical interactions are handled by breve, but may need to be tested and modified for specific cases (such as an object intersecting with itself). The method for moving the creature was also tested. Movement can occur either through using the move command on an object, or the set-joint-velocity command on a joint. Testing this was as simple as assigning some arbitrary velocity to the joints and seeing how the creature moved. I tested the system of neurons by creating a creature with some random initial velocity, and then having the sensor data go through the neurons to produce output data. I made sure that each creature produced walked randomly.

3.2 Research Theory

My project will use a combination of Sims' neurons and gene-guided creatures. The simulation will begin by allowing a neuron-controlled creature to move in the simulation for a short period of time. It will save the creature's velocity at each time step in an array, and the Controller object will calculate the creature's fitness level and add it to an array of creatures along with the velocity. This process will be repeated for ten creatures.

After that, the Controller will compare the fitness values of the ten creatures. The two highest scoring ones will "survive," and pass their genes on to the next generation. The simulation averages the velocity arrays of the survivors, and passes it on to a creature in the next generation. This creature will not move by neuron-modified data, but instead will use the velocities from its parents. All the creatures that did not survive are replaced with more neuron-controlled creatures. This process is repeated until a stable method of movement is reached that has a higher fitness level than any of the neuron-guided organisms.

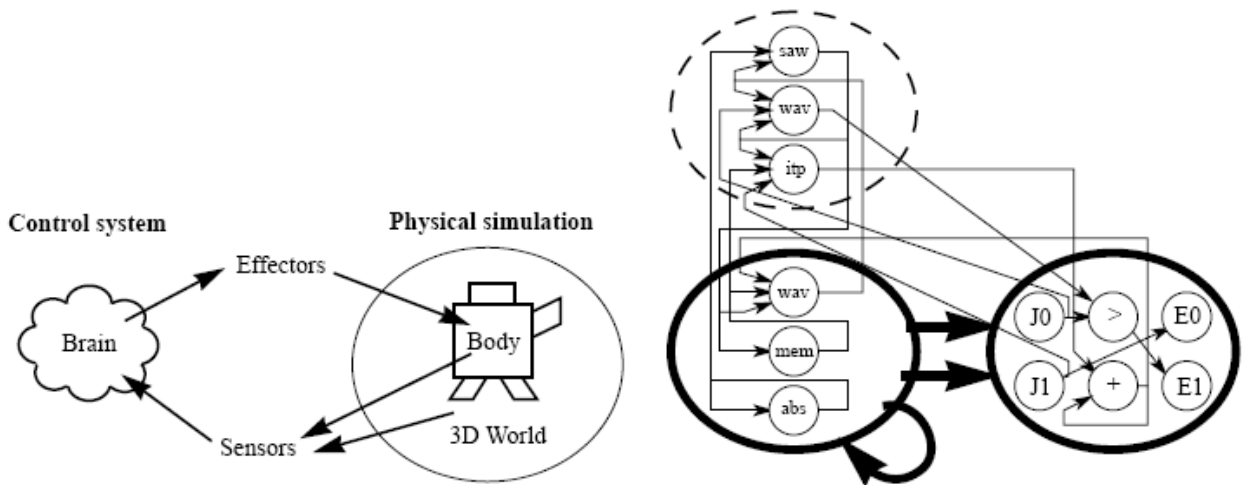
If I decide to add the ability for specialized creatures, this would be done with different fitness algorithms. For example, to create a creature that is specialized for jumping, the fitness evaluator would check for vertical distance rather than horizontal. This allows for creatures with the best traits for a certain specialization to pass their genes on.

3.3 Testing and Analysis

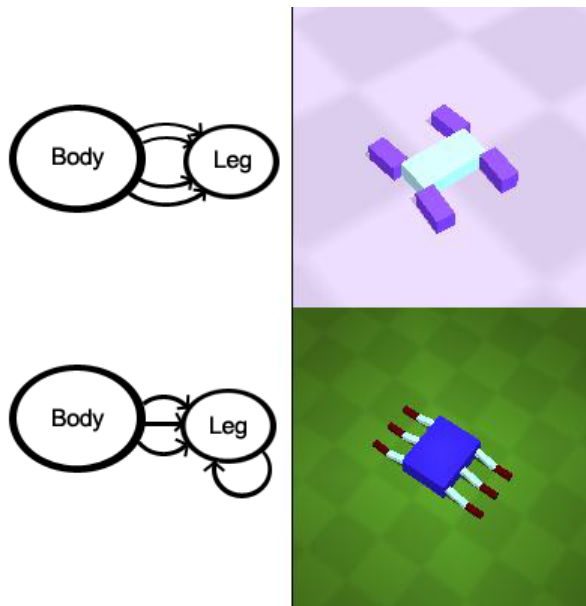
The same testing done within the program to select the best organisms can also be used to test the overall success of the simulation. For example, a fitness test used to compare walking creatures could be used in the same method to compare creatures at various timesteps in the program.

The purpose of conducting a fitness test outside of the program would be to check if creatures are actually evolving to better fitness levels. If every 10 generations the fitness level increases, then it can be assumed that the creatures' motion techniques are improving.

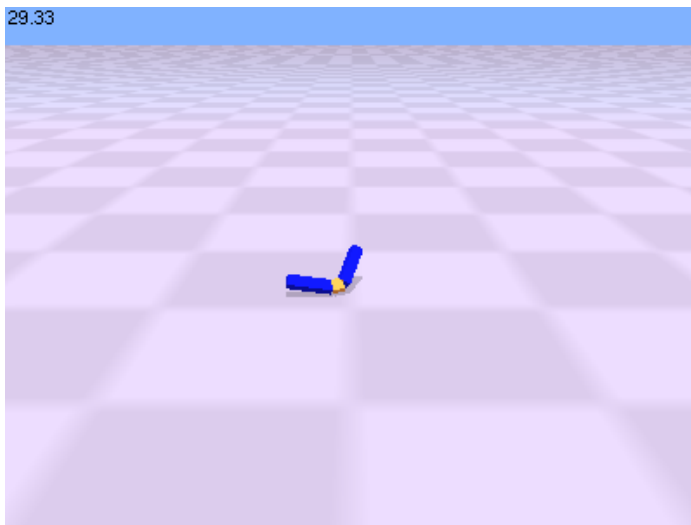
3.4 Visual Representation



On the left, a representation of the control struction described by Sims. The input values are modified by neurons in the creature's "brain," and these values effect the creature's joint angle velocities. The picture on the right shows the flow of input data (J0 and J1) through the neurons and into the effectors (E0 and E1).



On the left, the genotype of a creature. The resulting phenotype is shown on the right.



An example of a neuron-guided creature.

3.5 Developmental Procedures

4 Results

4.1 Summary

My goal was to create a simulation of virtual creatures who could evolve and develop different methods for moving. If an organism's fitness level is steadily improving throughout the simulation, then my project will be a success. So far I have not met these requirements, but I still have half a school year.

4.2 Results

Right now my program has few results, since evolution is not yet occurring. It can produce creatures with different walking methods, but does not yet select them based on fitness levels.

My project can be used on a larger scale to simulate more complex functions. For example, the behavior of a deep-sea animal that little is known about could be predicted using this simulation. It could also perhaps be applied in reverse, and used to simulate reverse evolution of creatures back to their forms thousands of years ago.

4.3 Procedure Analysis

I believe that my theoretical evolution procedure will be effective not only in producing creatures that develop more advanced motion techniques, but also in creating unique methods. Since my method takes data both from the last generation and from randomly generated creatures, it should allow for a lot of possibilities. Using data from parent creatures will assure that the best genes are maintained, while the random genes add uniqueness and allow deviance from parent genes.

My method of testing using fitness levels provides an accurate estimate of the level of evolution, as well as provides the possibility of customization. Creatures can be tested using different fitness equations to evolve certain traits.

4.4 Findings

I found that the idea of evolution is not as easy as simply picking the best traits out of a group of creatures. I quickly realized that if the two best creatures are selected and bred, only one creature is created in the next generation. Also, I learned that because the neuron structure is so random, it can't be used in all situations. Creatures who have received genes from the last generation will not implement neurons, because this would more or less revert them back to a completely random walking method with no evolution.

4.5 Conclusion

Although it has few concrete results now, I believe my project will be able to successfully simulate the evolution of artificial life. I've begun to better understand the methods behind simulated evolution, and I think that is the hardest step. Originally I wanted to base my project solely off of Karl Sims' research, but I realize that his methods do not have enough detail for me to implement them completely. Instead, I hope to incorporate his ideas but still formulate my own methods for evolution.

References

- [1] Sims, Karl. "Evolving Virtual Creatures." 1994. Thinking Machines Corporation. 24 Oct. 2007 < <http://www.genarts.com/karl/>>.
- [2] Bongard, Josh, Victor Zykov, and Hod Lipson. "Robotic Introspection: Self Modeling." 21 Nov. 2006. Cornell Computational Synthesis Lab. 24 Oct. 2007 <<http://ccsl.mae.cornell.edu/research/selfmodels/>>.
- [3] Komosinski, Maciej, Szymon Ulatowski. "Framsticks." Online. <<http://www.frams.alife.pl/>> November 02, 2007.
- [4] Lassabe, Nicolas. "Research." Nicolas Lassabe's Homepage. Online. <<http://www.irit.fr/Nicolas.Lassabe/research.php>> 2 November 2007.
- [5] Schelling, Thomas. Sugarscape. Online. <<http://sugarscape.sourceforge.net/>> 2 November 2007.