TJHSST Senior Research Project
Evolving Motor Techniques for Artificial Life
2007-2008

Kelley Hecker

June 10, 2008

Abstract

I have created a program for simulating unique creatures in a 3D environment using co-evolution of the creatures' mental and physical structures. Creature data is stored in a tree-like genome consisting of various nodes for each physical body segment. The brain of the creature is controlled by neuron modification of sensor inputs. There is a system for converting genomes to physical representations to allow for physical simulation in the environment, and eventual selection of prime candidates through a genetic algorithm.

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 catagorized 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 creaturs learn on their own and develope 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

Research done by Karl Sims is very similar to what I wish to accomplish. His work with evolving creatures led to a variety of organisms specialized in different areas and had very organic movements. The creatures were controlled by a system of neurons which produced joint-angle effectors.

Yoon-Sik Shim and Chang-Hun Kim continued Sims' research and explored the possibiliy of flying creatures. They developed and explained a system of storing genomes as one-dimensional arrays and further expanded on some of Sims' ideas for neural networks and genetic selection.

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-then, 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. "Generating Flying Creatures Using Body-Brain Co-Evolution"

   In this paper, Yoon-Sik Shim and Chang-Hun Kim continue Sims' research and simulate evolving flying creatures. Based on their ideas I modified my program to store creatures as genomes. Each genomes is a one-dimensional array of nodes, and each node represents a body segment. Data for the physical dimensions of the body part, as well as the neurons that control the joints, are stored in the node. By storing physical data in the node, creatures should easily be able to evolve physically for better performance. [2]

3. 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. [3]

4. 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 opimization of techniques, but can also spontaneous evolution and ecosystem modeling. [4]

5. 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). [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 needed 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

The entire simulation will be run by the Controller object. At the beginning of each simulation the Controller will create an array of genomes, and maintain this array throughout the generation. It also displays the creatures in the physical environment and measures their fitness levels. After the fitness levels are compared, the Controller will manage the reproduction of the creatures and update the genome array with the next generation.

Each genome is make up of several nodes, each representing a body segment in the physical creature. The nodes store physical dimensions for the limb, a list of connected limbs or children, points where the segment connects to its parent and children segments, and the neurons which will control the joints.

While the Controller generations the initial population of genomes, the creature itself must create the object which will be physically represented in the simulation. The creatures have a recursive method which goes through the genome and creates a leg object for each node, and add all of these legs to the root. This creates the Multibody which will be simulated.

Each creature is also responsible for sending its neurons to the CreatureGA, calculating its final fitness value, and for removing itself at the end of the generation.

At each time-step the joint-angle values for each node are measured and passed to the Creature Genetic Algorithm. This algorithm passes the sensor values through the neurons for that node and produces an effector, which will be the joint velocity. Possible neuron functions are sin, cos, atan, sumthreshold,sign-of, min, max, if, mem, saw-wave, log, expt, devide, interpolate, and differentiate.

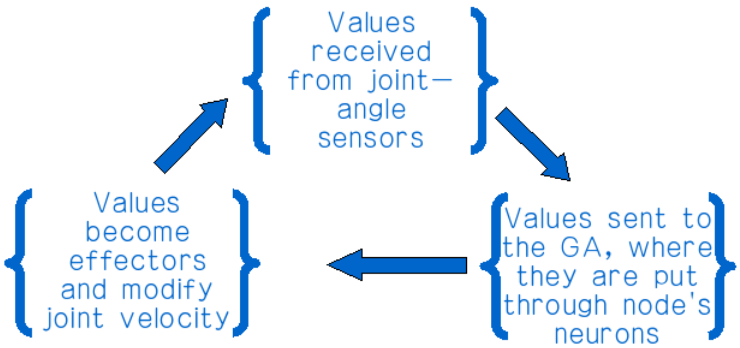
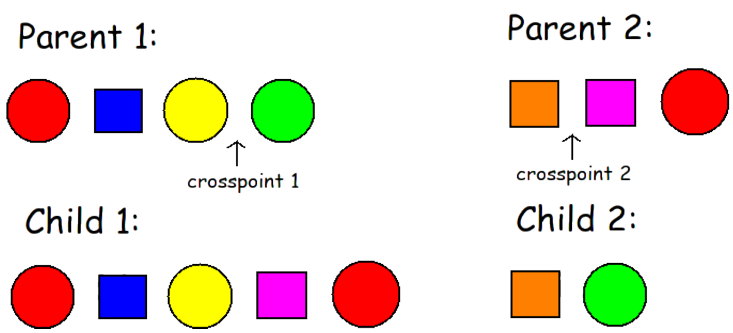


Image representing the circulation of data. Data originates from sensors measuring interaction with the environment, and then is data is modified by neurons within each node. Finally, the modified data is passed to the joint as velocity values. The cycle starts over again when sensors receive new data after the joints move.

After the creature has had its chance to move, its fitness value is evaluated and compared with the rest of the population. At this point the Controller initiates reproduction. The top 50 percent of creatures are copied directly to the next generation via asexual reproduction, while the remaining 50 percent reproduce through genetic cross-over with another creature. Mutation is also possible at this point, occurring in about 10



A diagram of the crossover process. Each shape represents a different node.

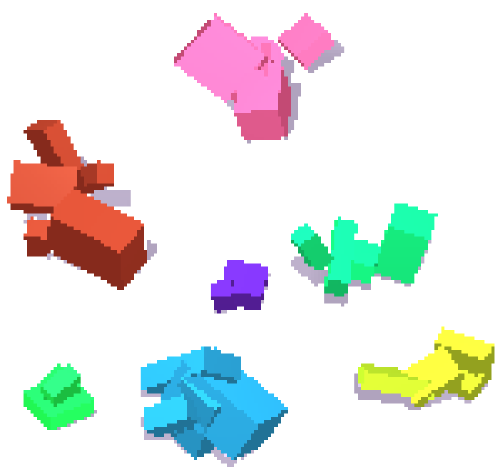
This cycle is then repeated with the next generation for a given number of generations. By the time the simulation is complete, a variety of creatures with different physical and mental characteristics should exist.

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. This can be done by creating a graph of the average fitness values as the generations progress, and performing a linear regression on this data. A linear regression line with a positive slope shows that the average fitness value is increasing.

3.4 Visual Representation



Examples of starting creatures.

4 Results

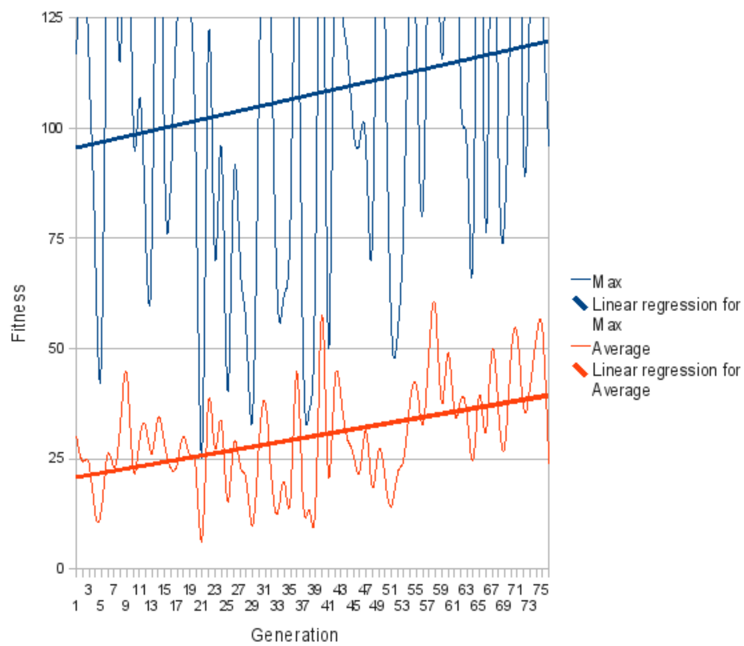
4.1 Summary

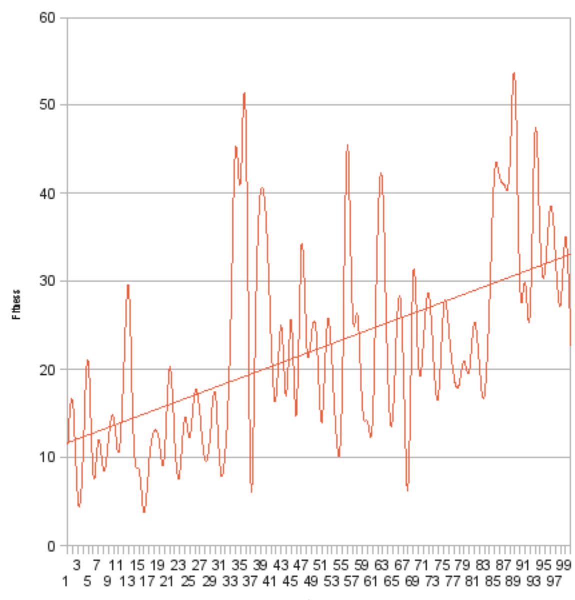
My goal was to create a simulation of virtual creatures who could evolve and develop different methods for moving, while at the same time evolving physically. If an organism's fitness level is steadily improving throughout the simulation, then my project will be a success.

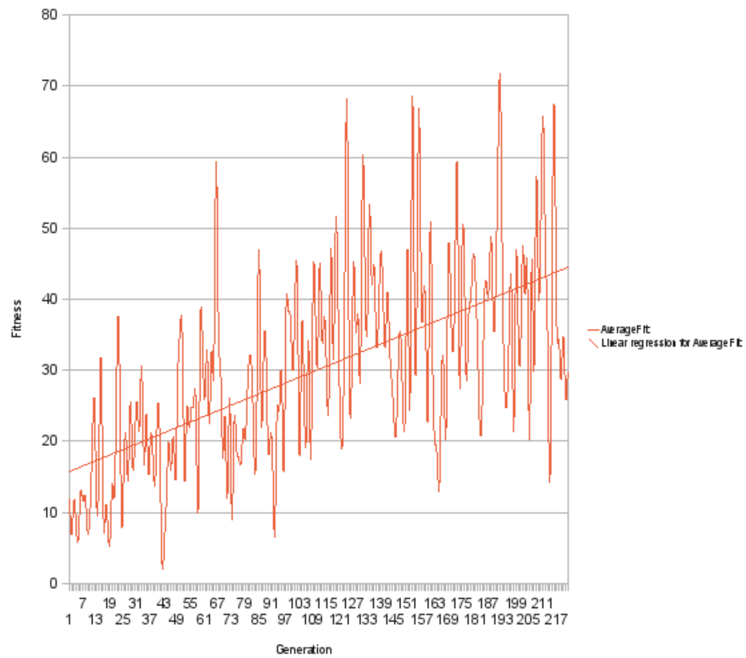
4.2 Results

Results were fairly consistent over several runs of the simulation. Each generation would be three or four creatures who failed to move at all, probably due to an unfavorable combination of neurons in the nodes. Usually one to three creatures would appear as being superior, while the rest managed to move but not very efficiently. Even in later generations the motionless creatures were present.

Average fitness values increased for most simulations. Some improved only by a fitness point or two over the course of the simulation, while others increased by over twenty points.







Graphs showing the increase in average fitness value over time.

Some of the simulations with a duration longer than 100 generations started to show a leveling off or decrease in the average fitness value, although after adding mutation this result rarely occurred. This would suggest that the simulation has a maximum fitness point, after which the values begin to decrease. It could also be a sign of a bust and boom cyclical pattern.

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

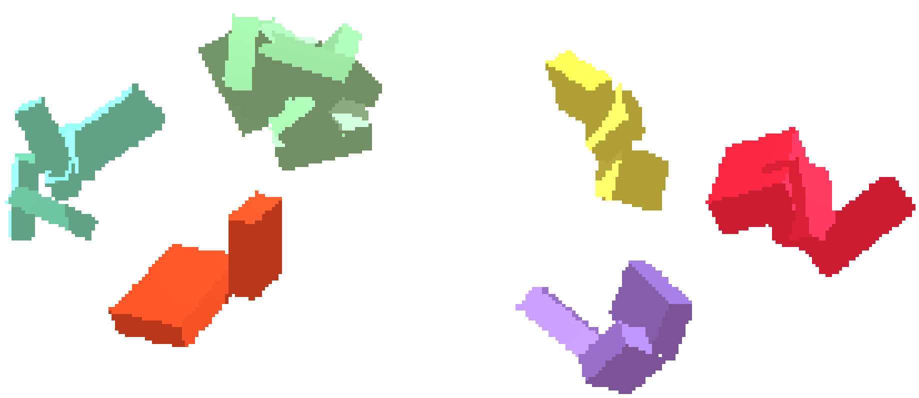
I believe that my theoretical evolution procedure will be effective not only in producing creatures that develop more advanced motion techniques, but also in creating unique methods. Since my method takes data both from the last generation and from randomly generated creatures, it should allow for a 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 Conclusion

My program appears to be a success and produces creatures with increasing fitness values. The number of creatures that could possible be generated is nearly limitless. The genome structure allows creatures with the same body structure to move very differently and vice versa. However, after running many simulation it seems that creatures who are lightweight with long, thin limbs usually produce the highest fitness values. Larger creatures with many boxy limbs were often unable to move.

Along with different physical structures, many different methods for movement appeared. Some creatures would rotate their legs around a central body and "roll" around. Others had one or more limbs that they flapped up and down, allowing them to bounce along the ground.



Examples of potential creaturs. The creatures on the left used their legs to roll, while those on the right hopped.

References

[1] Sims, Karl. “Evolving Virtual Creatures.” 1994. Thinking Machines Corporation. 24 Oct. 2007 < http://www.genarts.com/karl/>.

[2] Shim, Yoon-Sik, and Chang-Hun Kim. “Generating Flying Creatures Using Body-Brain Co-Evolution.” Department of Computer Science, Korea University. 2003. 22 Jan. 2008 <http://portal.acm.org/citation.cfm?id=846316>.

[3] 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/>.

[4] Komosinski, Maciej, Szymon Ulatowski. “Framsticks.” Online. <http://www.frams.alife.pl/> November 02, 2007.

[5] Lassabe, Nicolas. ”Research.” Nicolas Lassabe’s Homepage. Online. <http://www.irit.fr/ Nicolas.Lassabe/research.php> 2 Novemeber 2007.