Simulated Robotic Autonomous Agents with Motion Evolution

Neil Vaughan[1]

[1]Bournemouth University, Data Science Institute,
Talbot Campus, Fern Barrow, Poole, Dorset, BH12 5BB
nvaughan@bmth.ac.uk

Abstract

This research implemented autonomous control of robotic agents. The movement controls are simulated within a virtual environment. The control system algorithms were subjected to evolution. Genetic algorithms were implemented to enable the robotic agents to adapt in response to objects within the virtual environment. Additionally, each robot's physical characteristics were subjected to evolution through a survival of the fittest system based on crossover with random mutations. Survival of the fittest was simulated by a shortage of food causing competition. When the food quantity was increased the evolution rate decreased. With increased food, there was reduced competition and average fitness stopped increasing over time. Removing the food bottleneck stopped the survival of the fittest mechanism.

Introduction

The aim of artificial life is to exhibit characteristics of living organisms. Artificial life fits into three categories: (i) Microscopic systems such as chemical, cellular and tissues. (ii) Mesoscopic systems such as whole organisms. (iii) Macroscopic systems such as entire species and collective societal group (Bedau et al. 2003).

This research fits in category (ii) looking at the evolution of algorithms to control movement of a robotic autonomous agent. Also within category (iii) this research covers the group dynamic evolution by measuring changes in evolutionary rate within the group (species) of robots in a competitive environment and when bottlenecks are removed or reduced.

Types of Robotic Motion Control

Several recent research efforts have focussed on simulating evolution. Programs like Avida can model evolution of simulated organisms, made up of algorithms which re-produce. With Avida, organisms are represented as a string of commands which are executed. Programs like Avida can autonomously evolve basic computer software containing logic and comparison functions, for use in controlling physical robots. This digital evolution enables a fundamentally new approach to software design. Evolved program DNA can be cross-compiled into other languages and executed directly within physical robots or other systems. Avida was used to demonstrate that evolution can produce complex features by combining previously evolved building blocks (Lenski et al. 2003).

However, for representation of genetics within digital evolution various methods are interchangeably used. With Avida, the animal DNA is a computer algorithm, which directly represents executed behaviour. However this is not the case in natural organisms, as DNA describes the layout and physical structure of the body and brain, which in turn enables an individual to learn behaviour through their experience. In Avida certain commands are rewarded with increased computing power, again this method may not accurately represent methods within natural organisms.

Darwin Pond (2005) is an evolution simulator for 'Swimmers'. Food grows at a predefined growth and spread rate throughout the environment (Fig. 1). Swimmers have characteristics setting their hunger level, maximum lifespan, energy from food and energy from mating. Individual swimmers can be saved as .dna files which contain the genes, saved as around 16 ten digit numbers. Each swimmer is differentiated from others by individual settings for variables including: number of limbs, segments per limb, rate, amplitude, straddle, turning, joint angle, motion genes, base colour and colour shift. A swimmer may only mate with swimmers of the same colour.

Other evolution simulators include corewars, a game in which algorithms compete for the core system memory, which causes a survival of the fittest mechanism.

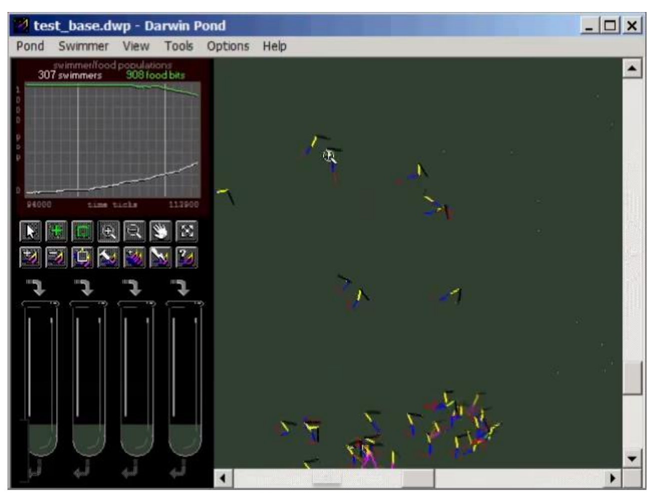

Fig. 1. Simulator of autonomous evolution in Darwin pond.

Physics Based Mobile Robot Simulators

There has been recent research focussing on 3D geometric model based creatures whose movement is governed by simulated laws of physics. Examples include artificial flying creatures created by modelling the air drag forces (Furukawa et al, 2010). Other virtual creatures have been able to use movement of their body parts, to drag themselves across a surface. Simulators may need to model several physical laws including gravity, momentum, energy transfer and others.

Evolution of group or co-operative mobility control and the co-ordination of movements has been a focus of recent research. These dynamics are critical to group behaviours such as flocking, avoiding obstacles, and eluding enemies.

Evolving motion strategies with a simulator could enable mobile robots to adapt to conditions in ways not otherwise apparent to human designers.

Recent studies suggest that when an evolving group is subjected to a catastrophic population bottleneck, the group often rebounds and that the result on average fitness and genetics can be comparable to the difference the system would have accumulated if it had been left untouched (Olson et al. 2013). Simulated robotic evolution may provide a method by which the effects of population rebounds on group genetics could be measured.

Results of Robot Motion Simulator

A simulator was developed to model autonomous robots which were mobile within a limited area on a flat surface.

The robotic agents were programmed to walk in random directions until food was within a certain distance (Fig. 2). Once food was seen the agents would walk towards the food and eat it. If the agents do not obtain food within a certain time the agents die and are removed from the simulation.

Breeding was implemented by genetic algorithms to enable the robotic agents to adapt in response to objects within the virtual environment. When breeding, each new robot's physical characteristics were based on their parents using crossover with random mutations. Physical characteristics include leg length and the distance of eyesight.

This system resulted in evolution of longer legs and better eyesight over time, due to survival of the fittest. A shortage of food eliminated the worst performing agents.

Finally when food quantity was increased to an amount where all agents could survive. When the amount of food was increased, the survival of the fittest mechanism no longer occurred and average fitness stopped increasing over time since there was no longer a survival bottleneck. To assist the reader in visualising the simulator a short video was uploaded here: https://www.youtube.com/watch?v=BNtT98MkYzM

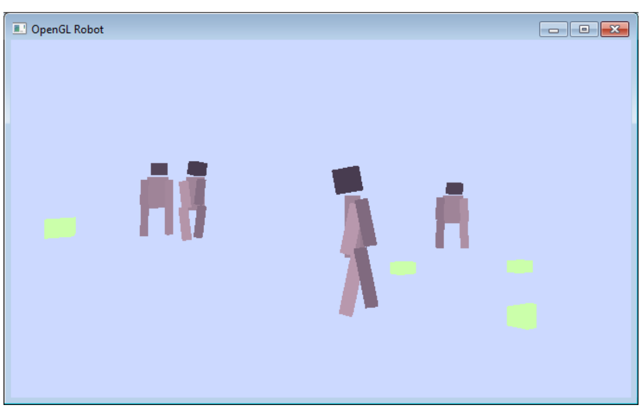

Fig. 2. Developed robot motion simulator.

Discussion

The simulated evolution showed that increased fitness over time occurs only in the presence of survival bottlenecks, causing only the fittest to survive. In this simulation, the fittest robots with the fastest movement rate due to their larger leg components were more suited to reach food quickly and survive. When food was increased to a rate at which all robots were surviving, the fittest were no longer at an advantage.

Survival bottlenecks decimate the number of individuals in a species. Whilst many individuals suffer, the destruction can work in favour of the species as a whole, leaving the fittest and best adapted to survive. The average fitness will have increased. However when a species has access to more than enough food for all individuals, bottlenecks are eliminated. This enables the weakest individuals to survive with an equal chance of reproducing as the fittest and evolution stops.

This research has aimed to measure the effect of food bottleneck on the rate of evolution and average fitness.

Future evolution simulators may benefit from developing a standard method for storing and interpreting genetics as currently various methods are being used. In some cases genes represent physical characteristics, such as the number of limbs and limb length. Alternatively genes can executable code, representing behaviour as in Avida. In this case, the genes serve as an algorithm for behaviour which could include motion and logic based decision making.

Conclusions and Future Work

The next stage of research could be to implement computer vision for the autonomous agents. For each robot agent, the robotic field of vision can be calculated from the robot's position by projecting the view from their head, as 45 degree on the x and y axis. The resulting field of view can be calculated on each iteration of the software and used as input to computer vision algorithms. A variety of image processing algorithms can be applied. This may result in the robots evolving to respond to visual perceived colours, objects.

This could address a central problem to robotics, to appropriately respond to sensory information in real-time.

References

Avida, Devosoft (2009), http://avida.devosoft.org.

Bedau, M. A., Packard, N. H. (2003). Evolution of evolvability via adaptation of mutation rates. *Biosystems*, 69(2):143-162.

Darwin Pond, Ventrella J, Dodd B (2005). http://www.ventrella.com/darwin/darwin.html.

Furukawa, M., Morinaga, M., Watanbe, M., Suzuki, I., Yamamoto, M. (2010). Artificial flying creature by flapping. *IEEE International Conference on Systems Man and Cybernetics (SMC)*:624-629.

Lenski R.E., (2003). The Evolutionary Origin of Complex Features. *Nature*, 423:139-144.

Olson, R., Mirmomeni, M., Brom, T., Bruger, E., Hintze, A., Knoester, D., & Adami, C. (2013). Evolved digital ecosystems: Dynamic steady state, not optimal fixed point. *Advances in Artificial Life, ECAL*. 12:126-133.