

A virtual creatures model for studies in artificial evolution

Thomas Miconi

School of Computer Science
University of Birmingham
Birmingham B15 2TT
United Kingdom
t.miconi@cs.bham.ac.uk

Alastair Channon

School of Computer Science
University of Birmingham
Birmingham B15 2TT
United Kingdom
a.d.channon@cs.bham.ac.uk

Abstract-

We present the results of our replication of Karl Sims' work on the evolution of artificial creatures in a physically realistic 3D environment. We used standard McCulloch-Pitts neurons instead of a more complex set of ad hoc neurons, which we believe makes our model a more general tool for future experiments in artificial (co-)evolution. We provide a detailed description of our model and freely accessible source code. We describe our results both qualitatively and quantitatively, including an analysis of some evolved neural controllers. To the best of our knowledge, our work is the first replication of Sims' efforts to achieve results comparable to Sims' in efficiency and complexity, with standard neurons and realistic Newtonian physics.

1 Introduction

1.1 Motivation

Experiments involving autonomous articulated structures ("creatures") in a physically realistic 3D environment can be an important experimental tool for artificial evolution. First, the challenge presented by their very complexity offers a stark contrast with more usual test problems such as function optimisation, and is arguably closer to what an evolutionary algorithm would be confronted with in many real-world applications (especially in the field of intelligent robotics). Second, the realism of situated, embodied creatures allows these creatures to *construct* their own behaviours, as opposed to choosing from a pre-defined repertoire of behaviours through action selection mechanisms or acting on a few parameters (e.g. wheel speeds) for pre-defined behaviours. Because the morphologies and controllers of the creatures evolve simultaneously, evolution can not only select or modify pre-existing behaviours defined by the user: it can construct *novel* behaviours and features as it progresses. For example, in many predator-prey experiments, co-evolving individuals can only act on the speed of their displacements. It is conceivable that, by using a physically realistic environment and allowing evolution to control not only parameters, but complete morphologies and behaviours of evolving creatures, a wealth of new possibilities would be created (such as allowing the prey to defend or protect itself). The freedom of creating new features and behaviours, besides opening new avenues of exploration in evolution and co-evolution, may prove a decisive advantage in the quest for open-ended evolution: the perpetual inven-

tion of novel behaviours within a given population¹

We are therefore interested in building a very specific type of system: a physically realistic 3D environment in which articulated creature can evolve towards potentially arbitrary levels of complexity². We add the requirement that this system must be powerful enough to allow for reliable, consistent success, at least for simple tasks such as locomotion. Finally we want to do this with very general architecture, without any ad hoc machinery to favour a specific type of behaviour. To the best of our knowledge, the closest thing to such a system so far has been Karl Sims' model of evolving creatures.

1.2 Related work

It has now been more than a decade since Karl Sims presented the results of his experiments on the evolution of virtual creatures in a three-dimensional (3D), physically realistic environment [12, 11]. Taylor & Massey have attempted to replicate this work [15], but we believe it is fair to say that by visual inspection, their results do not compare favourably to Sims' (or to ours). While any such appreciation necessarily contains a subjective component, in section 6, we attempt to identify more objective factors for evaluating efficiency in mobile creatures.

While there has been a significant amount of work in projects related to the simulation of 3D creatures, most of it has concentrated on specific areas of research such as developmental systems or modular robotics. For example, Bongard & Pfeifer [1] used such a system to build a model of genetic regulation in development, based on genetic regulatory networks. The study of these regulation networks, and the emergence of modularity in the resulting phenotype, was the central subject of this work. The resulting creatures were rather limited in their behaviours and did not exhibit the efficiency and nature-like aspect of Sims' - which is unsurprising since neither efficiency nor natural aspect were objectives of this study.

Hornby & Pollack [3] offered a more computational model, in which development was directed by the application of a genetic grammar based on L-systems. The system involved creatures made of sticks arranged in complex 3D shapes. The author reported a significant improvement by using their developmental method instead of purely declara-

¹Clearly other conditions may be necessary for opening evolution in such a way, such as the introduction of co-evolution in the system.

²Note that we are *not* asserting that evolution will mechanically impose an increase in complexity.

tive genomes. Their success came at the price of a severe restriction on the physics engine, which assumes that the simulation is stable at each timestep, thereby preventing any dynamic behaviour such as running or jumping. The same can be said of the GOLEM project [7], in which creatures made of rigid cylinders were first evolved in simulation, then built in the real world using 3D printing.

Applications to modular robotics were also reported. Marbach and Ijspeert [8] and Mesot [9] simulated modular robotics systems, in which identical elementary modules were assembled into larger robots. In these two systems, either the controllers or the morphology (or both) were strictly constrained, with controllers being typically reduced to coupled oscillators.

Framsticks [6] is perhaps the most successful example of artificial evolution in a 3D environment. Framsticks involves creatures made of flexible sticks, connected by joints and controlled with neural network based on functional neurons. These creatures are controlled by neural networks, with special neurons. Framsticks emphasises interactivity between the world and the user, and the quality of the software and the ease with which experiments can be set up has attracted a significant community of users worldwide. From our viewpoint, the main drawback of Framsticks is that the (simplified) physics, as well as the shape of agents, may limit the possibilities of interaction between these agents.

Besides physics simplification, another common feature in the simulation of 3D creatures is the widespread use of high-level, ad hoc elements in the controllers. Sims' creatures, in particular, were controlled by functional networks, including arithmetic functions, tunable oscillators and logic operators (among others) as elementary building blocks. This represents a significant amount of *a priori* knowledge given to the system. Sims [12] acknowledged the ad hoc aspect of this choice in which, in his own words, led "a creature's brain [to] resemble a dataflow computer program more than a typical neural network". Hornby & Pollack [3] and Framsticks [6], among others, also resort to high-level functions such as oscillators³. High-level, ad hoc elements have the obvious advantage of facilitating the emergence of certain types of behaviours, especially with regard to locomotion; however this may come at the expense of generality. We believe that using simple McCulloch & Pitts neuron may provide more freedom to the system in the type of behaviours it generates. At any rate, it would reduce the amount of *a priori* knowledge in the system, and thus offer a higher evolutionary challenge.

1.3 The system

In this article we describe our own model for the evolution of artificial creatures in a physically realistic 3D environment. This model is broadly similar to Sims', but with important differences. Our work brings three contributions

³Oscillators in general are ubiquitous in physics and biology, and might be argued to be fundamental elements in their own right; however the type of oscillators referred to here are high-level oscillating 'neurons', with a tunable frequency calculated by temporal integration of inputs, arguably a high-level mechanism.

with respect to Sims':

- Our creatures are controlled by standard neural networks, based on classical McCulloch & Pitts neurons with sigmoid or radial activation functions. While Sims' approach was entirely justified given the seminal aspect of his work, we believe that using standard neurons provides a higher level of generality to our model.
- We provide a complete description of our system, as well as the original source code. In the course of our description, we also point out some issues with Sims' model which, unless they are addressed, might hinder its application to more complex experiments. The lack of information on crucial aspects of Sims' system has been an obstacle to replication. The program described in this paper is freely available under the terms of the GNU General Public License (GPL) at <http://www.cs.bham.ac.uk/~txm/creatures/> together with video samples.
- We present experiments performed with this model with both qualitative and quantitative results, in contrast to a purely qualitative, screenshot-based account as found in e.g. Sims [12], Taylor & Massey [15], Marbach [8] and others. We also describe and analyse the neural networks of two evolved creatures.

2 Creature morphology

In the following sections we provide a broad description of our system, stressing both similarities and differences with Sims' model. In order to facilitate comparisons, our description deliberately follows the same organisation as Sims [12], section by section. At the end of this paper, we provide a table which summarises the important numerical constants used for our experiments (see Table 1).

As in Sims' model, the creatures are branching structures composed of rigid 3D blocks. The blocks (or "limbs") are connected to their parent limb by a hinge joint - except for the first, "root" limb which has no parent. The genetic specification of a creature is given as a graph of nodes. Each of these nodes contain morphologic and neural information about one limb. In contrast with Sims' model, our genetic graph may not contain any loop, including self-loops. Each node is responsible for storing the description of its limb's physical connection with its parent node's limb, removing the need for connections to carry their own information, as is the case in Sims' model.

The morphologic information in each genetic node specifies the *dimensions* of the limb (i.e. width, length and height), the *orientation* of this limb with regard to its parent (in the form of two parameters indicating polar angles with the xz and the xy planes, i.e. longitude and latitude, in the frame of reference of the parent limb; these two parameters are discrete multiples of $\pi/8$), the *axial direction* of the hinge joint which may be either horizontal or vertical (i.e. aligned either with the y or with the z axis of the limb), and a boolean flag for *reflection* which governs symmetric replication along the xz plane of its parent (see section 4.1).

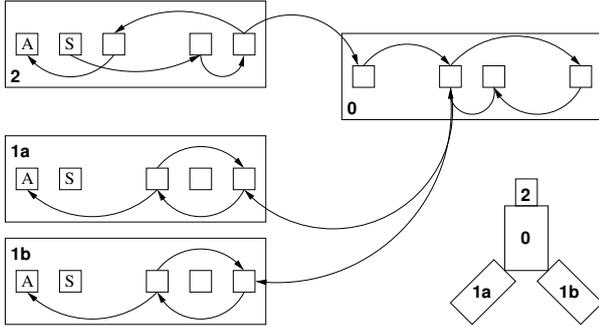


Figure 1: Organisation of a fictional creature pictured in the bottom-right corner. Limb 0 has no sensor (S) or actuator (A). Limb 1 is reflected into two symmetric limbs 1a and 1b, which share the same morphologic and neural information.

A limb also contains *neural information*, as described in the next section.

3 Creature control and neural organisation

Our creatures are controlled by neural networks. As in Sims’ model, each limb contains a set of neurons. Genetic information about a given neuron specifies the *activation function* for this neuron, a threshold/bias parameter θ taken in the $[-3/2, 3/2]$ range, and connection information. The activation function may be either a sigmoid ($\frac{1}{1+\exp^{-(\sigma+\theta)}}$) or the hyperbolic tangent $\tanh(\sigma + \theta)$ where σ is the weighted sum of inputs (the difference between sigmoid and tanh is that the first has values in $[0, 1]$ while the latter has values in $[-1, 1]$). Connection information specifies, for each connection, the source of this connection (i.e. the neuron whose output is received through this connection) and a weight in the $[-1, 1]$ range. Neurons can only be connected with other neurons from adjacent limbs, or from the root limb. Each neuron may receive a variable number of connections, up to a maximum value (3 in the present experiments).

The most important difference with Sims’ model lies in the choice of standard neurons with traditional activation functions, in contrast to Sims’ large set of functions (including arithmetic operations and oscillators). An important consequence of this simpler set of functions is that there is no trivial way for evolution to generate oscillators or other cyclic forms of behaviour, which are necessary for any sustained locomotion to take place. Such behaviours have to emerge out of the interaction between several neurons, assembled together under the guidance of evolution. A more practical consequence is that in our model, each neuron may have an arbitrary number of inputs (up to a maximum value), by contrast to Sims’ neurons which had a fixed number of inputs, dependent on their function.

3.1 Sensors and actuators

Sensor neurons and actuator neurons are handled specially. The only type of sensor neuron used in the present experiments is a proprioceptive neuron, which measures the current angle formed by the hinge joint to which this neuron’s

limb is attached, scaled within the $[-1, 1]$ range. Actuator neurons command the movement of each limb, that is, its rotation around its joint. Their inputs are defined similarly as other neurons, but their activation function is always a scaled hyperbolic tangent of the form $MaxSpeed * \tanh(\sigma + threshold)$, where $MaxSpeed$ is a system constant. In our model, sensor neurons do not receive any connection from any other neuron, and no neuron may receive a connection from an actuator neuron. Sensors and actuators are, respectively, pure sources and pure sinks of data.

A difference with Sims’ model is that an actuator does not specify a force or a torque, but a *desired speed*. The physics simulator implements a motor at each joint, which will constantly attempt to reach the desired speed, with the constraint that the total torque it exerts cannot be larger than a specified maximum. This maximum is a system constant. This mechanism corresponds to a very simple model of servomotors.

Note that while each limb has a sensor and an actuator, there is no requirement that they should receive or send connections from or to other neurons: connections are established in a random manner and no connection toward these special neurons is explicitly enforced. Thus each limb is free to use its sensor and actuator, or not, depending on how its network evolves. This is equivalent to Sims’ model.

4 Combining morphology and control: expression of the genome

The creatures are constructed according to the information contained in the genetic nodes. A simple developmental system translates the genotype into a corresponding phenotype, and may introduce additional complexity if the genetic information dictates it.

A developmental system corresponds to the introduction of a bias in the search space: while the total number of different individuals accessible with a given N-bits genome is obviously bounded by 2^N in any case, using a developmental system will allow the system to obtain different (usually more complex) creatures, at the expense of making others impossible. Quite often raw, declarative genomes describe a compact search space of very similar creatures, and the developmental system has the effect of “stretching” the search space, creating a globally wider, but sparser, range of possibilities. Developmental systems must of course be chosen with care so that the newly obtainable individuals can be expected to perform globally better than the “sacrificed” ones. A common feature of developmental systems is a stress on modularity and replication, with the expectation that non-linear interactions between similar elements may bring interesting behaviours with minimal additional information; Hornby and Pollack [3], for example, provide noticeable examples of modularity introduced by a developmental system⁴. Our system uses only one developmental

⁴Developmental systems in artificial evolution have been used for a relatively long time, but rigorous analysis of their effects and implications has only recently started to attract interest. Stanley and Miikkulainen [14] provide an enlightening review and discussion of developmental systems

feature, adapted from Sims: bilateral symmetry.

4.1 Reflection

Symmetry in our model is implemented somewhat differently than in Sims'. In our model, each genetic node (corresponding to a limb) may possess a "reflection" flag, which means that when this node is read and the corresponding limb attached to its parent, a symmetric copy of this limb will also be created. Any further sub-limbs will similarly be duplicated in a symmetric fashion, which leads to the appearance of bilaterally symmetric branches. Our present design allows for only one type of symmetry, namely symmetry along the parent's xz plane. When a given limb is randomly generated, its reflection flag is set with probability P_{ref} (for this paper, $P_{ref} = 0.1$).

4.2 A limitation of symmetry

Symmetry in Sims' model (and in ours, under its current implementation) has a limitation which is not discussed in his papers. When a limb is duplicated by reflection, all genetic information is duplicated in the process, including neural information. A consequence of this duplication is that *a given limb cannot distinguish information it sends to, or receive from, either of its symmetric sub-limbs*. Because neurons from both symmetric sub-limbs share the same connection information, they will receive identical connections (and information) from the same neurons in the parent. Similarly, any connection that the sub-limbs send to the parent will point to the same neuron in the parent, and information from both sub-limbs will be merged at that point. Thus, although both limbs may behave in different manners due to their distinct inner neural networks, they will not be able to send distinct information to the parent, or to receive distinct information from it. While several solutions to this problem may be considered (e.g. assigning input from symmetric limbs to symmetric neurons within the parent's array, and assigning input from the parent to symmetric neurons within the arrays of the two symmetric limbs), for the present task, we decided that such refinements would be unnecessary.

4.3 Absence of segmentation

In Sims' model, a loop in the genetic graph corresponds to a set of limbs which is repeated a certain number of times. This, in essence, is a simple and effective model of *segmentation*, that is, the repetition of homologous modules arranged sequentially, as apparent in many animals (vertebrates, arthropods, annellidae, etc.). Because our genetic graphs are organised in a tree hierarchy (i.e. no loops are allowed within the graph), this feature of Sims' model is not present in our system in its current implementation. By preventing loops in our genetic graphs, we thus remove one "trick" from our evolutionary toolbox. However our results indicate that this particular feature is not necessary for the reliable emergence of meaningful and efficient behaviours.

in artificial evolution.

5 Creature evolution

5.1 Genetic operators

We use three genetic operators, broadly similar to those used by Sims. *Crossover* is performed by simply aligning the genetic nodes of both parents in two rows, then building a new list of genetic nodes by concatenating the left part of one parent with the right part of the other. *Grafting* corresponds to the removal of a branch (i.e. a limb and all its sub-limbs), and its replacement by a branch taken from another individual. Connectivity information is adapted and maintained: the neurons of the trunk establish the same connections with the new branch as they had with the old one, and similarly the new branch has the same connection with its new trunk as it had with its previous trunk.

Mutation occurs by modifying each parameter of the genome with a certain probability P_{mut} (in this paper, $P_{mut} = 0.04$). The mutation operator proceeds in a sequential manner. First, with probability P_{mut} , a random limb may be deleted (along with its sub-limbs) from the creature (with the restriction that no creature may have less than two limbs). Then a new randomly generated limb (with randomly generated neural information) may be created. Then, each slots in the neural array may be "flipped" (i.e. empty slots are filled with a new neuron with randomly assigned connections, existing neurons are deleted). Then, the threshold value of each existing neuron may be modified by a random amount between -0.33 and 0.33 (threshold values are taken from the $[-3/2, 3/2]$ torus). The output function may be changed. Each connection of every existing neuron may be "flipped", i.e. created (and randomly assigned) if it is unassigned, or deleted otherwise. The weight of each existing connection may then be modified by a random amount between -0.33 and +0.33 (weight values are taken from the $[-1, 1]$ torus). The source of each connection (i.e. the neuron from which it originates) may be randomly reassigned. Finally, with probability P_{mut} , morphological information for each node is mutated. Morphological mutation performs one randomly selected operation out of seven possibilities: reassigning a given limb to a different "ancestor" limb (which amounts to moving a whole branch along the organism), randomly assigning a new length, width or height to the limb, modifying either of its orientation angles (possible orientations are discrete multiples of $\pi/4$; mutation occurs by choosing a new value within the range $[-\pi/2, \pi/2]$ around the current value), switching the orientation of its joint (horizontal or vertical), and flipping its "reflection" flag. Again, each of these modifications is applied with probability P_{mut} for each parameter.

5.2 The evolutionary algorithm

We chose to use a simple steady-state genetic algorithm (SSGA, as described in e.g. Mitchell [10]) with "triple tournament" selection. After a whole population of individuals is randomly generated, and each individual is evaluated, the algorithm goes through the following cycle:

1. Three individuals are randomly selected, and ordered

according to their score: let us call them A, B and C, where A has the highest score and C has the lowest.

2. C is deleted and replaced with a newly created offspring. This offspring may be created in one of three ways, with equal probability:

- Crossover between A and B, followed by application of the mutation operator.

- Grafting between A and B (A providing the trunk, B the new branch), followed by application of the mutation operator.

- Four sequential applications of the mutation operator to A - this corresponds to Sims' asexual reproduction.

If the resulting offspring is non-viable (i.e. two non-adjacent limbs intersect, or the developed phenotype contains too many limbs), this step is repeated as often as necessary until a viable creature is produced.

3. The resulting offspring is evaluated, and the cycle starts again from step 1.

Evaluation simply occurs by letting the creature move for a fixed period of time, and measuring the distance it has covered in the x direction. In order to penalise trivial, inelegant behaviours (such as an animat simply falling down and using its sheer length to obtain a good score), actual evaluation only starts after $1/8$ of the total evaluation time has elapsed.

6 Results and interpretation

6.1 Description of results

Our experiments involved populations of 500 individuals, which were evaluated for 10000 timesteps, each timestep corresponding to 0.01 second of simulated time. Fig. 2 describes the progression of the best individual in each of 15 different runs, over 500000 cycles (each cycle corresponding to one reproduction and one evaluation). Each curve plots the score of the current best individual in this particular run after a given number of evaluation-reproduction cycles.

From this data it is clear that in this system, the probability of basic success (i.e. the probability that a given run manages to come up with an efficient locomotive behaviour at all) is high, although not one. Only one run fails to produce any interesting behaviour. Another one starts to progress very late in the course of evolution, and when the experiment ends the resulting behaviour is still clearly sub-optimal. Note that in this work, the main defining criterion for deciding whether a creature exhibits an efficient locomotive behaviour is the presence of cyclical or at least oscillatory propulsive movement, without which no sustained locomotion can take place: in the absence of any oscillation, all limbs simply revolve around their joints until they either stop moving or reach the angular limit of their joints, and no further movement occurs. The distinction between limb-level oscillation and creature-level cycle is emphasised in the following section.

A visible feature of these curves is that those that stabilise do so on noticeably different values. This is related to the fact that creatures converge towards different behaviours, which may be divided into several categories: "Crawlers" use two symmetric limbs to propel themselves forward by direct contact with the ground, "snakes" undulate in a coordinated way to obtain efficient locomotion, "bouncers" use very quick oscillations at one or several joints and exploit the resulting momentum to jump forward, etc.

Several variants of these behaviours exist. The most important feature is proper exploitation of dynamics: the most efficient creatures are able to remain in a constantly dynamic, unstable state, in other words, they are constantly "jumping". Jumping can be obtained by direct propulsion and/or exploitation of inertia and momentum.

Another feature of these curves is that they mostly share the same general shape, divided in three domains: an initial low regime, a rapid increase in performance, and a stabilisation at a high plateau (though there is much variety in the length of each phase). This corresponds to actual phases in the evolution of creatures. The initial low regime corresponds to creatures that cannot perform any locomotion at all: they do not have any oscillating or cyclical movement, and stop moving after a short time when all joints have reached their angular limits. The increase in performance starts with the discovery of a first oscillatory mechanism which allows for locomotion of some sort. After a short phase of experimentation, the population tends to settle fairly early on a specific type of creature (which happens to crudely implement one of the behaviours mentioned above). From there on, evolution essentially optimises this particular plan. When the resulting creature has been thoroughly optimised, leaving little room for further progress, the curve stabilises on a plateau, corresponding to the maximum fitness of this creature.

6.2 Optimisation and efficiency

An interesting aspect of these creatures is their high degree of optimisation. Movements of limbs around joints are both ample and well-coordinated. The movement is not only oscillatory, but *cyclical* (i.e. the complete organism goes through a cycle with identical initial and final configuration), and this cycle is often synchronised with the displacement of the creature: for example, "bouncers'" movements are tuned so that the propelling part of the body performs exactly a full cycle during the time of a full jump, which ensures sustained, efficient locomotion. The coordination of movements ensures that a high proportion of motile activity in the creature is transformed into overall forward motion: there is no dispendious "jittering" (disordered activity which incidentally results in slow overall locomotion due to asymmetry in the distribution of movements) or "parasite" movement (movements that do not contribute to the overall motion of the creature, or even contradict and hinder it). This leads to a criterion for efficiency which can be evaluated by comparing the speed and activity of individual limb displacements with the overall velocity of the crea-

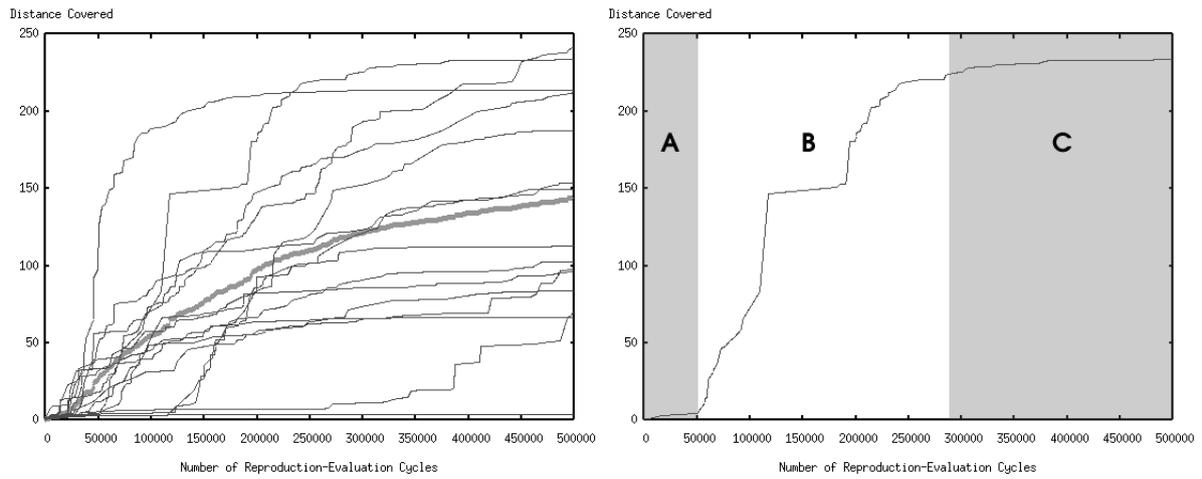


Figure 2: *Left*: Performance of the best individual, in 15 different runs. The average over the 15 runs is shown shown as a thicker light line. *Right*: A typical run, divided into three phases - see text for details

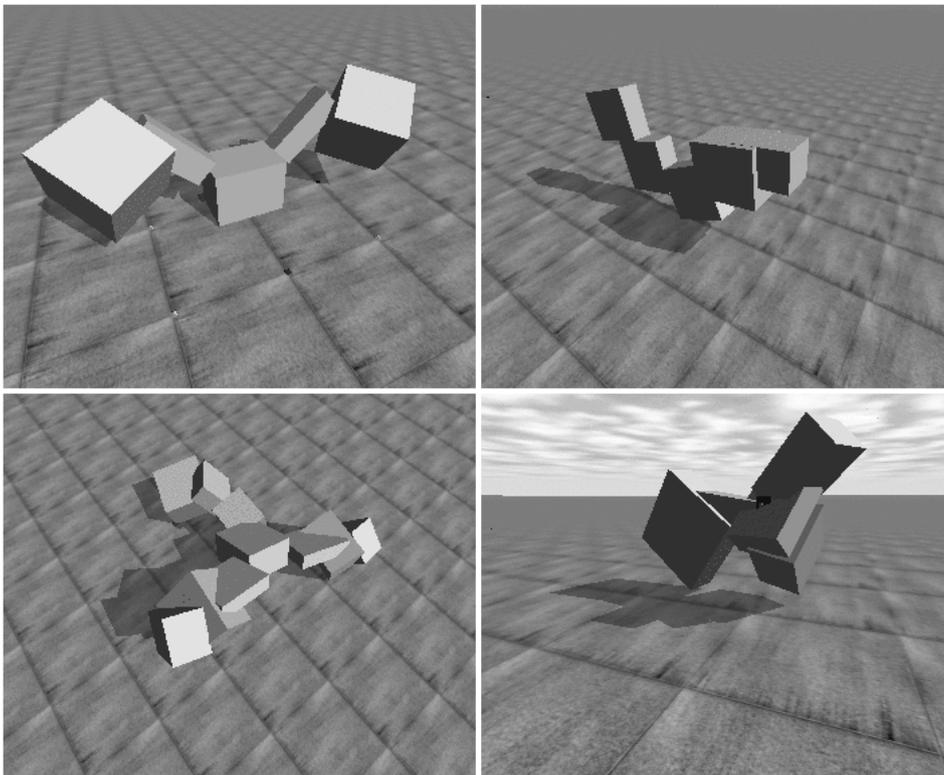


Figure 3: Four creatures from four different runs: a simple crawler, a "snake", a tailed crawler and a bouncer.

ture: if individual limbs move significantly faster than the whole creature (as is often observed in “jittering” creatures), the conversion of general activity into overall locomotion is poor, and hence the creature can be called inefficient. To be successful, a creature must be both active (sustained, ample motion must be observed in individual limbs) and efficient (this local activity must be efficiently converted into overall locomotion through the global organisation and coordination of movement).

The efficient use of symmetry completes the impression of optimisation and, we believe, gives a natural “touch” to our creatures (there is a strong temptation to classify bilaterally symmetric, cyclically mobile, propulsive appendices as “arms” or “legs”). We believe that this degree of optimisation, also apparent in Sims’ results, is a distinctive feature of our work with comparison to other replications of Sims’ experiments. We encourage readers to consult video samples (available at the URL specified in section 1.3) to observe evolved creatures in action.

6.3 Neural network analysis

Through observation of values and artificial “lesions” (suppression of selected neurons in order to determine which neurons have an influence on the behaviour of the creature), we can determine the functional parts of the neural network of any creature. Fig. 4 represents the functional part of the neural network (as obtained by observation of values and artificial lesions) for two simple creatures: respectively, one arm of the simple crawler and the “snake” shown in Fig. 3. In these two cases, the “root” limb did not participate in the functional network, which makes the structure clearer. For each neuron, the activation function is shown with the threshold represented as a constant bias to the sum of inputs x . In the first case, both actuators spontaneously start with a negative speed, because of their negative bias; however cross-connections between the lower and upper limbs mean that the upper limb will strongly tend to have an angular velocity opposed to the current value of the joint between these two limbs, while the lower limb will tend to have an angular velocity positively correlated with the angle between the upper limb and the trunk. The overall result is that both limbs move in opposite directions, and in phase opposition. Because the joint orientations are respectively vertical and horizontal, the upper limb will alternatively lower and raise the “arm”, and the lower limb will act as a “foot” which will oscillate orthogonally to this arm, in the direction of the movement of the creature. The foot will reach its highest speed backward (resp. forward) when the arm will be in its lowest (resp. highest) position, notwithstanding shifts induced by damping and thresholds. The connections with the interneuron are not necessary for locomotion to occur, but stabilise the creature in the initial phase of the evaluation.

Similar considerations apply to the second creature. The snake-like oscillations occur through a cascade of cross-connections between the sensors and actuators of different limbs. The lowest limb, having a speed positively correlated with the angle between its parent and its grandparent, with-

Simulation constants	
Max. number of neurons per limb	6
Max. number of connections per neuron	3
Initial proportion of assigned connections	75 %
Max. number of genes	7
Max. number of limbs (after development)	11
Activation functions	$\frac{1}{1+e^{-3x}}, \tanh(2x)$
Joint angle range	$[-3\pi/4, 3\pi/4]$
Range of connection weights	$[-1, 1]$
Range of neuron thresholds	$[-3/2, 3/2]$
P_{ref} (see text, section 4.1)	0.1
P_{mut} (see text, section 5.1)	0.04
Max. force applicable by each motor	4 N
Max. angular speed at each joint	4 rad / s
Actual mass of a limb of dimensions (x, y, z)	$0.8 * \tanh(x * y * z)$
Physics engine parameters	
Step size	0.01 s
ERP	0.015
CFM	0.01

Table 1: Simulation constants

out bias, acts as a “whiptail” which amplifies the undulation at this extremity of the creature. These two creatures exemplify successful joint evolution between morphology and neural information (both connection weights and thresholds).

7 Future work

While evolving creatures for simple tasks such as locomotion is interesting in itself and may have applications in related fields such as evolutionary robotics, we intend to use this model for further evolutionary experiments involving more complex tasks and co-evolution. Our first objective is to complete replication of Sims’ results by implementing a co-evolutionary, “box grabbing” task [11]. A longer term objective would be to study more complex tasks, such as physical fight.

The neural organisation of our creatures also opens interesting routes for investigation in other domains. Our networks offer similarities with Kauffman’s Random Boolean Networks (RBN) [5]: our neurons specify their *incoming* connections, and the number of incoming connections for each neuron may either be specified a priori (corresponding to the K constant of a RBN), or left under evolutionary control. It would be interesting to see whether Kauffman’s results on RBN apply to our networks as well, especially regarding the presence of a phase transition between a chaotic and a frozen regime, at which a transitory regime exhibits properties which make the system suitable for dynamic op-

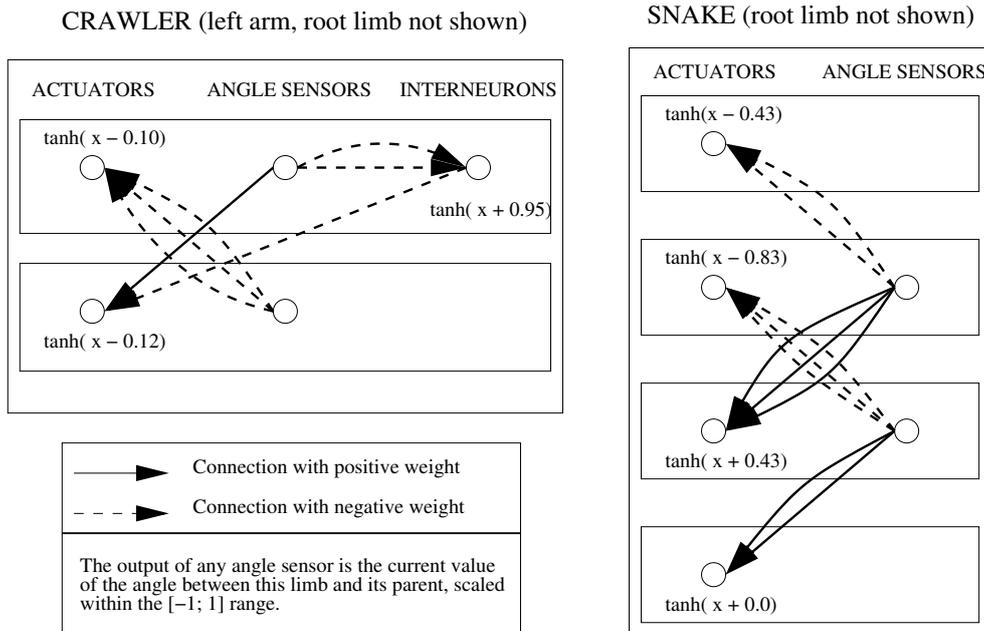


Figure 4: Functional subnets extracted from the neural networks of two creatures shown in Fig. 3. See text for details.

eration and evolution.

Acknowledgements

This research is now being funded by the School of Computer Science at the University of Birmingham, and was previously funded by the Intelligent Systems Group in the Department of Electronic and Computer Engineering at the University of Portsmouth, under the direction of David Brown.

Bibliography

- [1] J. C. Bongard and R. Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In Spector et al. [13], pages 829–836.
- [2] F. Gruau. Automatic definition of modular neural networks. *Adaptive Behavior*, 3(2):151–183, 1995.
- [3] G. S. Hornby and J. B. Pollack. Body-brain co-evolution using L-systems as a generative encoding. In Spector et al. [13], pages 868–875.
- [4] A. J. Ijspeert, J. Hallam, and D. Willshaw. From lampreys to salamanders: evolving neural controllers for swimming and walking. In R. Pfeifer, B. Blumberg, J.-A. Meyer, and S. Wilson, editors, *Proceedings of the Fifth Conference on the Simulation of Adaptive Behavior (SAB '98)*, pages 390–399. MIT Press, 1998.
- [5] S. Kauffman. *The Origins of Order*. Oxford University Press, 1993.
- [6] M. Komosinski. The world of framsticks: simulation, evolution, interaction. In *Proceedings of 2nd International Conference on Virtual Worlds (VW2000), Paris*, pages 214–224. Springer-Verlag (LNAI 1834), 2000.
- [7] H. Lipson and J. Pollack. Automatic design and manufacture of artificial lifeforms. *Nature*, 406:974–978, 2000. GOLEM.
- [8] D. Marbach and A. Ijspeert. Co-evolution of configuration and control for homogenous modular robots. In F. G. et al., editor, *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS8)*, pages 712–719. IOS Press, 2004.
- [9] B. Mesot. Self-organisation of locomotion in modular robots: A case study. Master’s thesis, EPFL, Lausanne, feb 2004.
- [10] M. Mitchell. *An introduction to genetic algorithms*. The MIT Press, 1996.
- [11] K. Sims. Evolving 3d morphology and behavior by competition. In *ALife IV : Proceedings of the 4th Conference on Artificial Life*, pages 28–39. MIT Press, 1994.
- [12] K. Sims. Evolving virtual creatures. In *SIGGRAPH 94*, pages 15–22. ACM Press, 1994.
- [13] L. Spector, E. D. Goodman, A. Wu, and W. B. Langdon, editors. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. Morgan Kaufmann, 2001.
- [14] K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.
- [15] T. Taylor and C. Massey. Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7(1):77–87, 2001.