Incremental Neuroevolution of Reactive and Deliberative 3D Agents

Adam Stanton[1] and Alastair Channon[1]

[1]School of Computing and Mathematics, Keele University, ST5 5BG, UK
{a.stanton,a.d.channon}@keele.ac.uk

Abstract

Following earlier work on the neuroevolution of deliberative behaviour to solve increasingly challenging tasks in a two-dimensional dynamic world, this paper presents the results of extending the original system to a three-dimensional rigid body simulation. The 3D physically based setting requires that a successful agent continually and deliberately adjust its gait, turning and other motor control over the many stages and sub-stages of these tasks, within its individual evaluation. Achieving such complex interplay between motor control and deliberative control, within a neuroevolutionary framework, is the focus of this work. To this end, a novel neural architecture is presented and an incremental evolutionary approach used to bootstrap the locomotive behaviour of the agents. Agent morphology is fixed as a quadruped with three degrees of freedom per limb. Agent populations have no initial knowledge of the problem domain, and evolve to move around and then solve progressively more difficult challenges in the environment using a tournament-based co-evolutionary algorithm. The results demonstrate not only success at the tasks but also a variety of intricate lifelike behaviours being used, separately and in combination, to achieve this success. Given the problem-agnostic controller architecture, these results indicate a potential for discovering yet more advanced behaviours in yet more complex environments.

Introduction

Living systems exhibit a large variety of coordinated activities at many different scales. We find homeostasis, locomotion, learning, group and social behaviours throughout the natural world. Since the earliest days of Artificial Life, a defining ambition has been to understand how to engineer systems that exhibit some of these complex behaviours, either to solve problems or to understand the underlying principles that gave rise to them in nature (Langton, 1989).

The specification of a model requires assumptions to be made concerning the degree to which its most basic units and the rules governing their behaviour are able to act as reliable proxies for their natural analogues. The granularity of a system has a direct impact on both its speed and its potential to accurately mimic nature, and on the strength of conclusions about the natural world based on phenomena observed to emerge from interactions within it.

The Dimensionality of Virtual Environments

Simulations of living systems have covered a broad range of abstraction but typically aim to exhibit behaviours at the level above that specified in the model's design. When building animat simulations that focus on interactions recognisable at the human scale, such as moving around, fighting and environmental manipulation, one of the key distinctions between designs is the physicality in which agents operate, specifically the choice between 2D and 3D environments. A two-dimensional world abstracts simulations away from the natural physical domain. Agents in these flat environments generally do not have to solve any complex physical control problems (Channon and Damper, 1998), as controllers are able simply to signal directions in which to move or turn the agent. Such models can encourage early emergence of more complex composite behaviours but preclude the development of novel motor control which may later allow for a richer interaction between agents and their environments. Two-dimensional simulations have not tended toward clearly *displaying* the prodigal physical interaction observed in nature, whether or not complex (simulated) non-physical interactions have evolved. This can be attributed, at least in part, to the fundamental rigidity and paucity of physical actions in such environments.

By contrast, having three-dimensional articulated bodies in a 3D world provides for much greater intricacy in how agents can interact with their environment and each other. Agents must begin to construct a coordinated motor pattern that results in basic directional motion before richer behaviours can develop as composites of these lower-level patterns. The specific characteristics of the environment are implicitly included in the performance of these motor patterns, and this couples agents to their environment. This coupling is crucial, together with the coupling of brain and body, to two key principals of embodied cognition: "first that cognition depends upon the kinds of experience that come from having a body with various sensorimotor capacities, and second, that these individual sensorimotor capacities are themselves embedded in a more encompassing biological, psychological and cultural context" (Rosch et al., 1991). Re-

cent trends reinforce this point of view, highlighting the importance of morphology and soft materials in the embodied loop (Pfeifer et al., 2014).

In terms of the ongoing ambition to evolve advanced life-like behaviour, both 2D and 3D approaches have been fruitful. For example, using 2D non-articulated agent bodies, early work by Yaeger showed (in a 3D environment) the emergence of complex collective behaviour (Yaeger, 1993); Channon demonstrated the first candidate synthetic open-ended evolutionary system using an agent-based (2D) world (Channon and Damper, 1998); and Robinson et al. (2007) evolved agents capable of reactive and deliberative behaviours in novel and dynamic environments.

In 3D the inherent complexities of articulated 3D physical form refocused work on the problems of motor control and locomotion: difficulties that had been largely abstracted away in 2D models. The seminal work by Sims (1994) remains an exemplar to the present day. Subsequent research has made incremental steps from this point, including demonstrating realistic co-adapted behaviours using just general purpose neurons (Miconi and Channon, 2006), making use of a human-specified syllabus of reactive locomotion-based tasks (Lessin et al., 2013) and using Novely Search (Lehman and Stanley, 2008) to evolve a range of gaits for a fixed morphology robot (Cully and Mouret, 2015), but continues to focus primarily on locomotion alone, leaving more complex behaviours aside.

General Approach of this Work

This work constitutes a first attempt to combine the incremental neuroevolution of reactive and deliberative behaviours with the neuroevolution of a 3D agent's motor control. Our overarching aim is the incremental evolution of sophisticated behaviours, for the population to overcome increasingly complex challenges in the agents' environment over evolutionary time.

The challenge is difficult because deliberative behaviour will be limited by necessary performance in motor control. An incremental approach can take this subtask-interdependency into account and prevent loss or lack of evolutionary gradient early in evolution. However, Stanton and Channon (2013) found that care is required when designing such incremental steps, as changing selection pressures too rapidly or too slowly can, respectively, cause evolution to lose gradient or over-fit to the current challenge. That work also demonstrated that it is necessary to revisit earlier incremental steps in order to prevent the loss of evolved abilities and therefore to find general solutions.

There is then a question of how to implement deliberative processing alongside physical control in a single controller. Deliberative planning systems learn a state-based action policy in order to select the best next state given a set of available actions. In contrast, flexible control of 3D motion requires a continuous-time closed-loop control system to keep physical variables within operational parameters. Also, for locomotive behaviours, a self-generating oscillation within the controller or body–controller action loop is necessary to achieve a reliable gait.

The requirements of each of these control systems is fundamentally different; it is difficult to design an architecture that can effectively learn the two different problems. The choice is between either an architecture that is general enough to be capable of both episodic categorisation and time-based close-coupled motor control, or a combination of the two architectures each tailored to a specific part of the problem and integrated elsewhere. In this work we opt for the latter, as a pragmatic step toward a more general architecture.

Hypothesis

The present work examines the following hypothesis: *that it is possible to produce reactive, deliberative behaviours in three-dimensional virtual creatures using a general evolutionary paradigm to optimise an implementation of the hybrid neural architecture detailed below*. The "River Crossing" (RC) task devised by Robinson et al. (2007) is used as the baseline reactive–deliberative problem. This task is adapted by the addition of a requirement of physical motor control in 3D, and the complete problem against which agents are tested is hereafter referred to as the 3D River Crossing or 3D RC task.

The remainder of this paper presents details of the 3D RC task, the agent and its hybrid neural architecture, and the evolutionary system, before reporting qualitative and quantitative results and our conclusions. It provides an existence proof that demonstrates the sufficiency and overall success of the design.

Experimental Design

The main contribution of this paper is the novel fusion of multiple neural architectures, each addressing different aspects of the 3D RC task, in order to enable the incremental evolution of agents that achieve the full task. This section of the paper introduces the environment and physical model and then describes the hybrid neurocontroller in detail, making reference to the inputs and outputs defined by the agent–environment relationship. Finally, the evolutionary algorithm is described in terms of the parameters of the neural architecture, and the experimental set-up is outlined.

Environment and Physical Model

The environment for the evolutionary problem is a modified version of the RC task first used in Robinson et al. (2007). In this task, agents exist and move around in a discrete, 20×20 bounded grid world. Each grid cell has attributes which can affect the agent: *traps* kill it, as does *water* (drowning); *grass* is neutral and *stones* can be picked up and put down. Stones can be placed on water, enabling bridges to

be built. The final attribute, *resource*, is the agent's goal. The RC task is an incrementally difficult challenge, with a staged introduction of difficulties. By collecting the resource, agents progress through more complicated environments, eventually arriving at a 20×n-cell river, where n is the increasing width of the river and thus the difficulty of the bridge-building task.
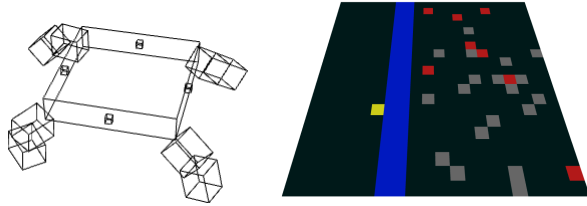

Figure 1: Agent morphology and environment, showing resource in yellow, river in blue, traps in red and stones in grey.

The 3D RC environment used in this work extends the 2D RC environment. Agents have a symmetrical quadruped body plan (figure 1) comprised of a torso (dimensions 1.0×1.0×0.2 cell-widths), four upper limbs (0.5×0.2×0.2), four lower limbs (0.5×0.2×0.2) and four small sensors (0.05×0.05×0.05). The upper limbs are attached to the torso at each lower corner with a 2-axis constraint. The constraint limits the range of motion of the upper limb relative to the torso, to $\frac{\pi}{2}$ radians around the vertical axis, and π radians around the line lying tangent to the agent's torso in the plane of the torso. Lower limbs are connected to upper limbs via a knee constraint which limits the range of motion between the two parts to $\frac{\pi}{2}$ radians around the y-axis. The sensors are attached with fixed constraints to the centre of each of the four faces of the agent's torso perpendicular to the ground plane. The physical simulator used was Open Dynamics Engine (ODE) version 0.13.1, with friction pyramid approximation for contact response ($\mu = 10.0$) between agent and the ground plane, universal ERP of 0.2 and CFM of 5×10^{-5}.

In order to bootstrap the evolution of locomotive behaviour, two additional levels were added at the start of the incremental RC task. The first level distributes "food" around the RC world. This confers additional fitness on agents once collected. The second level ("dash") has only one occupied cell, containing the resource. These levels together promote locomotive behaviour, and ultimately optimise the behaviour for speed of movement.

The difficulty of the RC environment is increased incrementally across six progressively more challenging levels. An agent's fitness is incremented from zero by 100 each time it successfully finds the resource, a requirement to progresses to the next level.

- Level 1: *Food*. The RC environment contains only cells with the resource (one cell) and food (probability 1/20 per

cell). Interaction with a food cell removes the food from the environment and increments the agent's fitness by 1.

- Level 2: *Dash*. This level contains only a single resource cell which agents must discover.

- Level 3: *Stones and Traps*. This level contains eight traps and twenty stones, as well as the target resource.

- Level 4: *Easy bridge*. This level is as level three but with a river of width 1 crossing the terrain.

- Level 5: *Medium bridge*. As level four, but width 2.

- Level 6: *Hard bridge*. As level five, but width 4.

On completion of level 6, agents are returned to level 1 and can continue to accumulate fitness until the time limit of 10 simulated minutes is reached, when evaluation is terminated.

Neural Architecture

A neural architecture capable of solving the 2D RC task was a major contribution of Robinson et al. (2007) and is extended in the present work. In the 3D RC task, an agent's neurocontroller transforms sensory inputs into torque values for motor control, which gives rise to behaviour in the physically simulated environment. The control system must produce directed locomotive behaviour in the quadruped, and change locomotive behaviour over the stages and sub-stages of the RC task, according to external (sensory) and internal (neural) state.

The hybrid neural architecture (figure 2) integrates the outputs of the RC world *decision network* (DN) and the diffusive *shunting model* (SM) with the inputs of the *physical network* (PN), and then use this information to pilot the agent through the world by affecting the operation of the agents' *pattern generator* (PG) neurons.

The Decision Network. The DN architecture follows the design laid out in Robinson et al. (2007). The DN is a standard feedforward neural network which takes inputs representing the attributes of the agent's current location in the RC world, and an input indicating whether or not the agent is currently carrying a stone. The hidden layer contains four neurons which sum over the inputs and apply a hyperbolic tangent activation function. The output layer sums over the hidden layer, applies a hyperbolic tangent activation function and tests at the thresholds -0.3 and 0.3; output neurons have three possible values: -1, 0 or 1, and determine the *iota values* used in the SM. These iota values indicate the saliency of the attributes in the environment, so the DN outputs iota values for each attribute (resource, stone, water and trap) except grass (which has an iota value of zero).
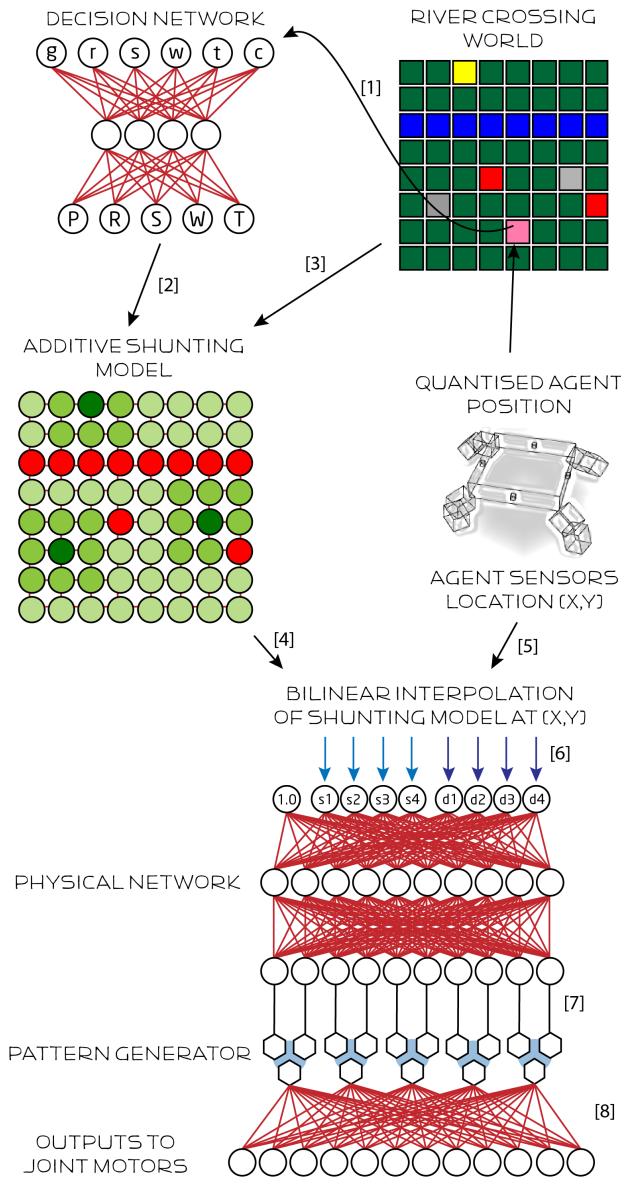
Figure 2: Neural architecture. Attributes at the agent's position (g=grass, r=resource, s=stone, w=water, t=trap, c=carrying flag) determine inputs to the Decision Network [1]. The Shunting Model constructs a landscape using iota values output by the DN [2] (P=pickup action, R=resource, S=stone, W=water, T=trap) and the locations of objects [3]. The SM activity landscape is interpolated [4] at the positions of the animat's four sensors [5], and these values fed to the Physical Network [6]. PN outputs are fed to the Pattern Generator Network [7], which outputs to neuromotor controllers. Links in red are genetically specified.

The Shunting Model. The SM was first used as a novel approach to motion planning by Meng and Yang (1998). The approach uses the homomorphism between the varying external environment and the intrinsic dynamics of the

architecture to achieve route generation (planning) without explicitly searching over possible paths. It is a generalisation of the potential field approach of Glasius et al. (1995), historically an evolution of the model of neural connectivity first proposed in Hodgkin and Huxley (1952). The SM uses a locally-connected, topologically-organised network of neurons to propagate desirable states across the entire network of transitions in the space. This produces an *activity landscape* with peaks at target states and valleys at configurations to avoid. One of the most common implementations of the SM is the *additive model* (Grossberg, 1988), which sacrifices gain control (and thus, stability) for simplicity. This model defines the following differential equation to model the diffusion of input values across the state landscape:

$$\frac{dx_i}{dt} = -Ax_i + \sum_{j \in N_i} w_{ij}[x_j]^+ + I_i \qquad (1)$$

where each neuron in the SM corresponds to one discrete cell in the environment; x_i is the activation of neuron i, taken to be zero outside of the environment; A is a passive decay rate; N_i is the receptive field of i; w_{ij} is the connection strength or weight from neuron j to neuron i, specified to be set by a monotonically decreasing function of the Euclidean distance between cells i and j (zero outside of the neighbourhood); the function $[x]^+$ is $max(0, x)$; and I_i is the external input to neuron i.

This technique was used in Robinson et al. (2007) to model the state space of the RC problem by directly representing the discrete RC world in the configuration of the SM, with each cell's receptive field set to be the eight cells in its Moore neighbourhood, within which all $w_{ij} = w$, and external input I_i determined by the attributes present in cell i and the saliency (*iota value*) for those attributes as computed by the DN. Neural activations propagate from external input I according to the local connectivity of the neurons, and the entire network can be considered a diffusive model that produces landscapes in which following positive gradients leads to target states. With well-chosen constant multipliers, this method exhibits no undesirable dynamics and has been found to be considerably versatile in a variety of subsequent works, including those of Borg et al. (2011) and Luo et al. (2014).

In this work, we simplify and clarify the setting of of decay rate and scales for distance (or weights) and iota values. A stable solution ($x_i^{new} = x_i$ for all i) to equation 2 is a stable solution ($\dot{x} = \mathbf{0}$) to equation 1. We absorb the constant A into the scales for iota values and distances, and set and limit weights and activation according to neighbourhood size (8) and maximum iota value ($maxI$=15), resulting in equation 3.

$$x_i^{new} = \frac{1}{A}\left(\sum_{j \in N_i} w_{ij}[x_j]^+ + I_i\right) \quad (2)$$

Following the computation of external inputs I by the DN, we zero SM activations and then iterate equation 3 fifty times to allow activity to propagate and stabilise across the 20×20 array of SM neurons.

$$x_i^{new} = min\left(\frac{1}{8}\sum_{j \in N_i}[x_j]^+ + I_i,\ maxI\right) \quad (3)$$

The Physical Network. The PN controls the agent's behaviour in the world. It receives as inputs the SM activations (interpolated) at the positions of the four sensors located on the four sides of the agent's torso. Since the SM represents a neural quantisation of the continuous landscape in which the sensors move, a single value is calculated for each sensor using a bilinear interpolation of the SM's activity values at the four points around the relevant sensor:

$$\begin{aligned} a(x,y) &= f[\lfloor x\rfloor,\lfloor y\rfloor](1-\{x\})(1-\{y\}) + \quad (4)\\ & f[\lceil x\rceil,\lfloor y\rfloor]\{x\}(1-\{y\}) +\\ & f[\lfloor x\rfloor,\lceil y\rceil](1-\{x\})\{y\} +\\ & f[\lceil x\rceil,\lceil y\rceil]\{x\}\{y\} \end{aligned}$$

where $a(x,y)$ is the interpolated activity at $(x,y) \in \mathbb{R}^2$, $f[i,j]$ is the SM activation at the discrete point $(i,j) \in \mathbb{Z}^2$ and $\{x\}$ denotes the fractional part of x.

These four sensor values are normalised (divided by $maxI$) and then fed into the PN, together with four values that indicate which sensor has the maximum value. The PN operates as a standard feedforward neural network where hidden nodes receive a weighted sum of the inputs. The hidden layer uses a hyperbolic tangent activation function in order to maintain negative values. The output layer uses a sigmoid activation function.

The Pattern Generator Network. The PG is a set of pre-evolved oscillatory neural circuits which are modelled on the networks of leaky integrators presented in Beer and Gallagher (1992) and used for locomotor pattern generation in many subsequent works, including Reil and Husbands (2002) and Stanton and Channon (2013). The circuits themselves are three-neuron motifs evolved to produce 1Hz sinusoidal oscillations from an output node in the presence of an input signal, and to be quiescent otherwise. Each complete PG network has a set of five identical motifs, initially isolated, which receive input from the PN via a set of weights and send their outputs to the final stage of the agent's controller. The neurons comprising these motifs are simple

continuous-time leaky integrators, with behaviour governed by the following equations:

$$\tau_i \frac{dA_i}{dt} = -A_i + \sum_{j=0}^{n} w_{ij}O_j \quad (5)$$

$$O_i = tanh(\frac{\alpha_i - A_i}{2}) \quad (6)$$

where A_i is the activation of a neuron i, O_i is the output of neuron i, w_{ij} is the weight from neuron j to neuron i, α_i is the bias of neuron i and τ_i is the time-constant of neuron i. At each iteration of the update algorithm ($dt = 0.01s$), equation 5 computes the change in the activity of the ith neuron for all neurons, and then equation 6 computes the output value for all neurons. It is this output value that is used by the neuromotor controllers.

To generate the original motif, a population of 1000 randomly initialised three-neuron networks was created with weights, time-constants and biases defined by a real-valued genotype. These networks were evaluated against a fitness function which measured the match between the desired frequency and the output response by summation of the undesirable (non-target) frequencies found in the frequency domain after application of Fourier transform. Networks were simulated for 10 seconds, twice. Once with a high input and a target frequency of 1Hz, and once with no input and a target quiescent state. Through three-genome tournament selection, strong candidates were used to generate new, mutated members of the population using the same evolutionary parameters as the general system described below.

Neuromotor Controllers. In the final stage, 12 motor controllers (one for each degree of freedom in the agent's morphology) receive the outputs of the PG network via a weighted sum and sigmoid activation function. These motor controllers implement a proportional-derivative (PD) controller, as used by Reil and Husbands (2002), which takes network outputs to be target angles within each joint's range of motion and applies a torque to the joint according to the following formula:

$$T = k_s(\theta_d - \theta) - k_d\dot{\theta} \quad (7)$$

where T is the torque applied to the joint, k_s is the spring constant, k_d is the damping constant, θ_d is the target angle and θ is the current angle. In this work, $k_s = 0.25$ and $k_d = 0.175$ were found to produce stable action at joints. This method has the advantage of relieving the neurocontroller of the problem of balancing an agent's weight against the force of gravity.

Evolutionary System

A steady-state evolutionary algorithm was used, in which a population of 150 agents are evaluated in groups of three and the least-fit individual replaced by a mutated single-point crossover of the fitter two.

Adam Stanton, Alastair Channon (2015) Incremental Neuroevolution of Reactive and Deliberative 3D Agents. Proceedings of the European Conference on Artificial Life 2015, pp. 341-348
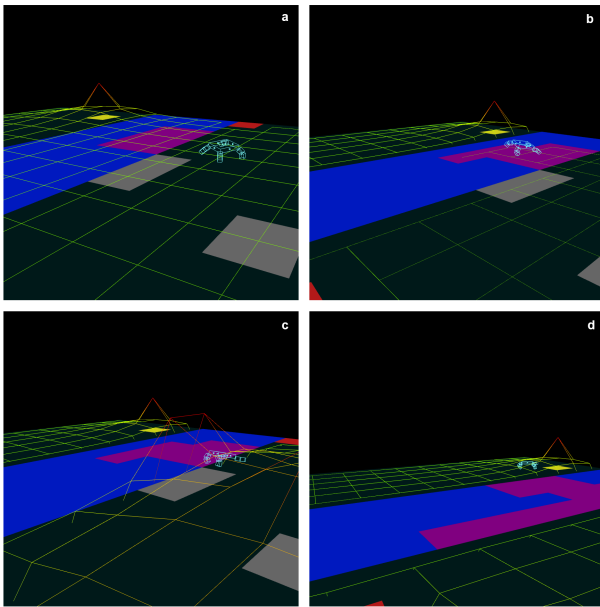
Figure 4: Bridge building in action. In (a) the agent has already started to build a bridge and is returning to collect another stone. In (b) the agent has just dropped a stone and is beginning to turn around. In (c) the agent is carrying a stone to drop on the water. In (d) the agent has completed the bridge and is about to reach the resource. The figure also illustrates the SM activity landscape superimposed on the 3D RC world and shows the changes to this landscape due to the updated iota values that occur as the agent's state, and thus DN inputs, vary.

Genetic Representation. Individuals' neurocontrollers are represented as an array of floating-point values. The sections are laid out as arrays of weights for each network stage as outlined above: the DN input–hidden and hidden–output weights, the PN input–hidden and hidden–output weights, the PG interneuron weights and the PG–motor weights.

Results

Twenty runs were carried out, each for 10^6 tournaments.

Qualitative Results. In those runs scoring highly on the final level of the task, intricate and diverse behaviours can be observed as the agents progress through their environmental challenges. In any single species, several different locomotive strategies can be observed depending on whether the agent is near or far from its target, and whether there are obstacles in the way. In the case of a "clear run", agents often gallop (figure 3) toward the target, whereas if more careful movement is required agents will progress more slowly, making time to avoid unexpected sensory conditions (i.e. traps and water). In both cases, directed control is observed as agents update their heading whilst engaging in locomo-

tion to remain aligned with the target. Agents also often display a distinct "turning" behaviour which will engage if the agent is beyond some angular threshold away from facing its target. Figure 4 shows an example evolved agent solving 3D RC task.

One of the most lifelike behaviours to be observed is avoidance: due to the non-spreading negative values in the activity landscape agents can unexpectedly encounter a highly negative region. In this case, agents will often crouch and spring back from the hazard, minimising the chance of falling on it due to imprecise control or previous momentum. Finally, in the case where no activation is present on the landscape around the agent, i.e. all directions are of equal saliency, agents engage in a form of random walk reminiscent of similar exploratory behaviour that can be seen in many simple animals. The temptation to interpret these actions in a human or animal context is ever present–agents can seem to exhibit surprise on encountering an unexpected danger, confusion if trapped in a mediocre part of the landscape and even happiness as they gallop toward the resource. The reader is encouraged to view example behaviours by watching the video at http://eprints.keele.ac.uk/rt4eprints/file/2093/.

Quantitative Results. The fitness scores of the three agents in each tournament were collected. Figure 5 shows the progress of the population from a typical run, in solving each level of the 3D RC task. Table 1 shows an overview of the performance of the entire system by aggregating and examining the results of the final 1000 tournaments from each run. From this table, it can be seen that every run was able to complete levels one and two in at least 80% of the final 1000 tournaments, and 95% of runs were able to complete level three to this standard too. Performance fell sharply against the bridge-building challenges, although 10% of runs were still able to complete level four in at least 80% of evaluations. At the hardest level of the task, 65% of runs achieved at least 1 evaluation which was able to complete level 6, and 20% of runs achieved at least 20% evaluations able to complete level 6. Figure 6 shows this aggregate data for all runs and levels and makes clear the spread of success across the whole problem in the experiment; a clear divide can be seen between the first half and latter half of the problem.

When examining the progression of the evolutionary algorithm in individual runs, it can be seen that the first level of the problem is solved early on in the search–typically after only 10000 tournaments. Success at level two soon follows as the problems are similar. Success at the third level (traps and stones, but no river) also occurs early on, in most runs. Levels four, five and six cause a longer delay in the search, and solutions do not appear at all in some runs even though the earlier levels have been solved in similar time to other, successful runs. When solutions do occur, there is often a delay between the solution for level four and later levels.
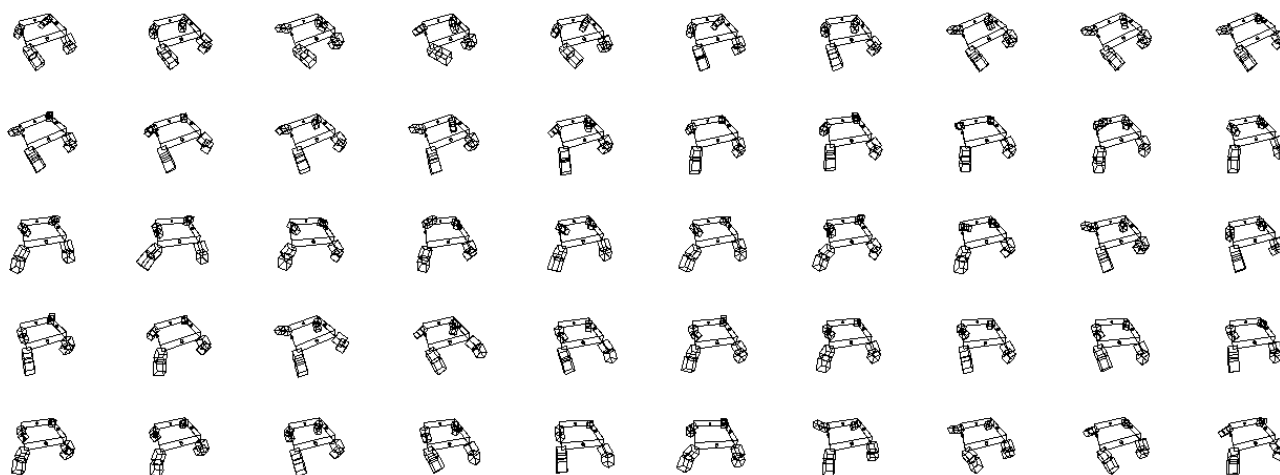
Figure 3: Example of a "galloping" locomotive behaviour. Time axis is left to right, top to bottom.
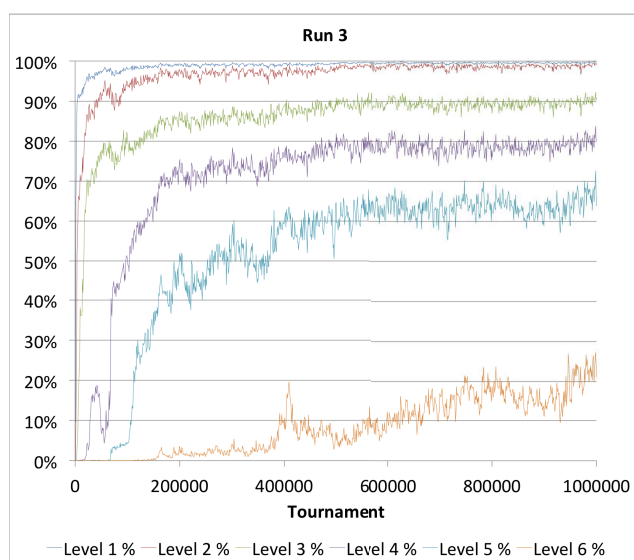

Figure 5: Progress of a typical run over one million tournaments. The graph shows the percentage of evaluations successful at completing each level of the 3D RC task, averaged over 1000 tournaments.

Cover Level	>0%	20%	40%	60%	80%
1 (Food)	100%	100%	100%	100%	100%
2 (Dash)	100%	100%	100%	100%	100%
3 (Traps)	100%	100%	100%	100%	95%
4 (River 1)	85%	85%	85%	30%	10%
5 (River 2)	85%	65%	50%	20%	-
6 (River 4)	65%	20%	-	-	-

Table 1: Proportion of runs with >0%/20%/40%/60%/80% of their final 1000 tournaments successful at level 1/2/3/4/5/6 of the 3D RC task.
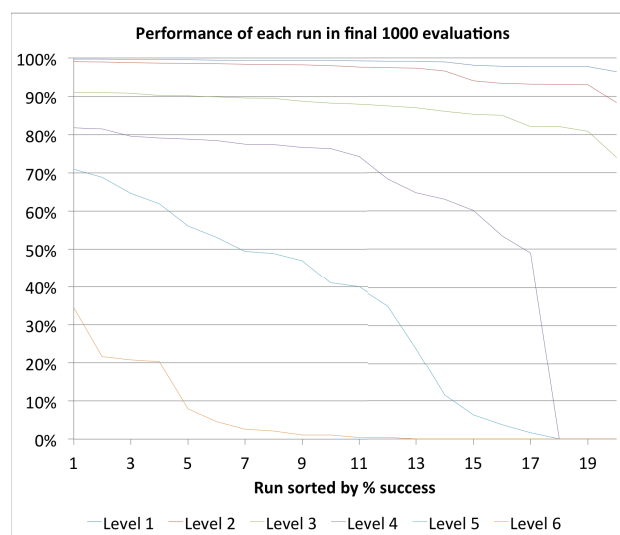

Figure 6: Success rates of all runs. The graph shows the performance of each 1000000-tournament run, evaluated from the final 1000 tournaments (3000 evaluations) of each run as the number of these evaluations that successfully completed each level of the 3D RC task. Runs are sorted in descending order for each level of the task.

Conclusions and Future Work

This work demonstrates that a standard evolutionary algorithm is sufficient to find parameters for a hybrid neural architecture comprised of loosely-coupled continuous-time and discrete-time neurons to produce reactive and deliberative behaviour in 3D, rigid-body virtual creatures requiring motion control.

By covering the range of task complexity over evolutionary time, species experience an evolutionary pressure

(no loss of gradient) whilst still being able to consolidate progress already made. This incremental approach allows species to first develop a locomotive behaviour, and then to use and adapt this ability to explore the space of solutions to the bridge-building river-crossing task.

This work has also shown that a hybrid approach to neuro-controller design that includes a generalised oscillatory component (in this case, an evolved network of leaky integrators) is sufficient to produce agents that exhibit task-dependent behaviours including locomotion, turning and avoidance. The architecture is also able to optimise the strategy for long-term deliberative planning in the 3D RC world at the same time.

The integration of a deliberative decision network and a mechanism to generate reactive behaviour in 3D virtual creatures, via a shunting landscape model, was successful and shows promise for future, more complex work in this area. The limitations of the model are due to the simplicity of the decomposition of the world into the agents' phenomenal space–there is no reason this relationship could not be integrated.

In order to generalise the applicability of this work to a broad range of tasks, it will be necessary to remove the problem-specific aspects of the neural architecture's design. A first step could be to make the distinction between the DN, SM and PN less explicit. Ultimately a single neural type and architecture, with genetically specified parameters, would be the most general design.

Other possibilities for increasing the coherence in the sensorimotor loop include finer-grained distinctions in the environment, for example iota values for boundary conditions, and the addition of noise to smooth behavioural transitions.

Associated Content

A video showing an agent completing a full run of tests is available at: http://eprints.keele.ac.uk/rt4eprints/file/2093/

References

Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.

Borg, J. M., Channon, A., and Day, C. (2011). Discovering and maintaining behaviours inaccessible to incremental genetic evolution through transcription errors and cultural transmission. In *Advances in Artificial Life: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems (ECAL 2011)*, pages 101–108.

Channon, A. D. and Damper, R. I. (1998). Evolving novel behaviors via natural selection. In *Proceedings of Artificial Life VI*, pages 384–388.

Cully, A. and Mouret, J.-B. (2015). Evolving a behavioral repertoire for a walking robot. *Evolutionary Computation*.

Glasius, R., Komoda, A., and Gielen, S. C. (1995). Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8(1):125–133.

Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural networks*, 1(1):17–61.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Physiology*, 117(4):500–544.

Langton, C. G. (1989). *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Addison-Wesley Longman Publishing Co., Inc.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of Artificial Life XI*, pages 329–336.

Lessin, D., Fussell, D., and Miikkulainen, R. (2013). Open-ended behavioral complexity for evolved virtual creatures. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 335–342.

Luo, C., Gao, J., Li, X., Mo, H., and Jiang, Q. (2014). Sensor-based autonomous robot navigation under unknown environments with grid map representation. In *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, pages 1–7.

Meng, M. and Yang, X. (1998). A neural network approach to real-time trajectory generation [mobile robots]. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1725–1730.

Miconi, T. and Channon, A. (2006). An improved system for artificial creatures evolution. In *Proceedings of Artificial Life X*, pages 255–261.

Pfeifer, R., Iida, F., and Lungarella, M. (2014). Cognition from the bottom up: on biological inspiration, body morphology, and soft materials. *Trends in Cognitive Sciences*, 18(8):404 – 413.

Reil, T. and Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168.

Robinson, E., Ellis, T., and Channon, A. (2007). Neuroevolution of agents capable of reactive and deliberative behaviours in novel and dynamic environments. In *Advances in Artificial Life: Proceedings of the Ninth European Conference on the Synthesis and Simulation of Living Systems (ECAL 2007)*, pages 345–354.

Rosch, E., Thompson, E., and Varela, F. J. (1991). *The embodied mind: Cognitive science and human experience*.

Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 15–22.

Stanton, A. and Channon, A. (2013). Heterogeneous complexification strategies robustly outperform homogeneous strategies for incremental evolution. In *Advances in Artificial Life: Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)*, pages 973–980.

Yaeger, L. (1993). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In *Proceedings of Artificial Life III*, pages 263–298.