Neuroevolution of Feedback Control for Object Manipulation by 3D Agents

Adam Stanton and Alastair Channon

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

Abstract

It has been shown that manipulation of objects by 3D virtual creatures can play an important role in the evolution of complex, embodied sensorimotor behaviours. In this work we examine the capacity of virtual creatures that use evolutionary and control architectures already shown to be capable of sensor-differential gradient-following locomotion (tropotaxis) to adapt to solve a physical problem involving the manipulation of 3D objects in their environments. Specifically, the creatures' task is to guide a physically-modelled cube through their environments in order to achieve maximum covered distance of the object. Agents were evolved in the manipulation environment from random initial genotypes and from genotypes previously optimised for performance in a different task. Performance was evaluated both before and after evolutionary adaptation. We show that the architecture achieves embodied feedback control in the block movement task. We observed some overlap between the earlier and later environments but also that success in the first environment does not preclude or entail success in the second. We found that species evolving from scratch do no better or worse than those optimised for a different environment, and that sensory feedback is necessary for correct approach and control behaviours in agents, although close control is less dependent on sensory input than distance approach.

Introduction

The evolution of virtual creatures in physically simulated three-dimensional worlds was first demonstrated in 1994 in the work of Karl Sims, who first evolved articulated agents to swim, walk, jump or follow a light source (Sims, 1994b) and then evolved such agents to compete to gain control of an object (Sims, 1994a). The diverse range of strategies and counter-strategies evolved through the latter task demonstrated both co-evolution's ability to generate increasingly complex behaviours and that object manipulation can play an important role in the evolution of sensorimotor intelligence (beyond mere locomotion and taxis) in simulation, as in nature.

The 3D River Crossing (3D RC) task, first presented in Stanton and Channon (2015), provides an ideal base from where the evolution of sensorimotor intelligence and related issues of physical embodiment can be explored. In that work we adapted the *shunting model* of Grossberg (1988) and

Yang and Meng (2000), used for the first time in an a-life context in Robinson et al. (2007), to build an evolutionary environment able to evolve control architectures of 3D virtual creatures that exhibit both reactive and deliberative behaviours. However, the problem-solving aspect of the 3D RC task in that work was abstracted from the physicality of the agent's morphology. Although each agent's joint motors were driven by some of the outputs from its neurocontroller, other neural outputs only notionally represented manipulation of physical objects in the agent's world.

An important extension of the earlier work into richer interactions is thus to introduce aspects of the deliberative problem to the agents' physical world, requiring an intricate manipulation of simulated objects to solve the challenge. In this work, we take a first step toward that goal by investigating whether the neural architecture outlined in that work can successfully constitute the control system for a simple manipulation task: displacement of a physically-modelled block in the agent's world, requiring feedback control, hereafter called the *block displacement* (BD) task.

Our general approach is to consider populations of agents in a new environment that provides the physical block challenge. The agents' neural control systems are sensitised to the location of the block by direct interaction with the shunting model, simplifying the adaptive problem. We investigate evolution on the BD task from both random (unevolved) populations and from populations of creatures previously evolved in the 3D RC environment. Hereafter we refer to random populations as *unevolved* populations and populations evolved only in the 3D RC environment as *naive* populations.

Hypotheses

The objective of this work is the evolution of agents able to successfully complete the BD task, as observed through 3D visualisation. In addition, we developed and tested the following hypotheses in order to further understand the interactions between the various components of the system and explore the limitations of the 3D RC architecture:

H1. The hybrid architecture is sufficient to achieve feedback

control that allows agents to successfully manipulate and guide an external object;

H2. There is some overlap between the earlier 3D RC task and the BD task due to the requirement for speedy and accurate movement in both environments;

H3. Species evolved in the 3D RC task show increased performance after evolution in the BD environment (i.e., it is possible to optimise this behaviour further), and that success in the 3D RC environment does not preclude success in the BD environment.

H4. Some 3D RC species are on evolutionary trajectories more suitable for the BD task than others.

The remainder of the paper presents an overview of the method used to generate the agents, and the results of the evolutionary and ablative experiments designed to test the above hypotheses. We then present conclusions and a discussion that relates the design of the base system to the observed results.

Methods

In this section, we describe how the overall objective of implementing a system capable of using an evolutionary algorithm to produce agents able to manipulate objects in a 3D, realistic physics world was achieved. The solution is split into three parts. The first part is the design of the evolutionary problem that the agent species must evolve to solve; the second part documents the abstractions made in the agent's morphology and control architecture that are under the control of the evolutionary algorithm and the third part describes the evolutionary algorithm itself. Finally we describe the data collection scheme we use to collect outputs from the experiments.

The Physical 3D RC Problem

The general problem used in this work is an adaptation of the 3D RC task described in earlier work (Stanton and Channon, 2015), following the same key ideas described and used to various ends in Robinson et al. (2007) and Borg and Channon (2011). The innovation in this work is the addition of a requirement for agents to physically manipulate objects in the environment; in our previous work only the body of the agent is physically simulated and all environmental interaction is through a two-dimensional, grid-world abstraction. In the original RC task, 2-dimensional agents are able to move between discrete cells in a 20x20 grid world containing hazards (*traps* and *water*) and resources (*stones* and *resource*. Stones can be carried by the agent and placed into water, enabling bridges to be built. Success in this environment is determined by agents' ability to avoid hazards and reach the single *resource* by learning an appropriate action policy given the current state; this includes capturing an element of deliberative planning in order to build bridges in worlds containing an otherwise impassable stretch of water.

This task was extended in our 2015 work to three dimensions (the 3D RC task), making the problem significantly harder. Agents are embodied in a four-legged fixed-morphology physical form that is simulated using a Newtonian rigid-body mechanics system, meaning that physical control (principally, the locomotive and orienting behaviours required for moving between grid cells) must be part of any solution. The agent's position in 3D is projected and quantised to the 2D RC world and any output from the control architecture translates directly into motor control in the 3D environment.

This work introduces the Physical 3D RC (P3D RC) task where the physical problem is extended beyond the agents' control of their bodies, to the wider environment. Solutions to the P3D RC task involve manipulation: in addition to the agents' bodies, a cube representing a *stone* in the world is also physically simulated. Any solution must use physical motor control to manipulate the cube into a configuration that allows the agent to access the resource objective.

As a step toward the P3D RC task, we first investigate simpler problems where agents must simply move blocks around in the world, without the requirement to solve the deliberative component of the RC challenge. This paper addresses the first of these challenges, where the problem is to move the environmental block as far as possible.

As in our earlier work and summarised here, agents have a symmetrical quadruped body plan comprising a torso (dimension $1.0 \times 1.0 \times 0.2$), four upper limbs and four lower limbs (dimensions $0.5 \times 0.2 \times 0.2$ each). Upper limbs are attached to the torso at each lower corner with a 2-axis constraint, limiting the range of motion relative to the torso. Knees connect upper libs to lower limbs, constraining their relative motion to a hinge. Four small sensors are also modelled in the physical environment as fixed appendages to the agent's torso; this is for convenience of updating sensor values based on their position and the sensors have no effect on the physical operation of the agent. The physical simulator used was Open Dynamics Engine (ODE) version 0.13.1, using a fixed timestep of 0.01s, friction pyramid approximation for contact response ($\mu = 10.0$) between agent and the ground plane, universal error reduction (ERP) of 0.2 and force-mixing (CFM) of 5×10^{-5}. In addition to the agents, a $1 \times 1 \times 1$ block is simulated at the centre of the environment ($\rho = 0.1$). On initialisation, agents are randomly positioned on a circle with radius 5 units from this point.

Agent Control

Given the above problem, a strategy to solve it necessarily requires a control architecture that receives sensor data from the environment and produces appropriate motor stimulation to guide the agent through the challenges of the world. We use a bespoke, hybrid neural network (HNN) to this end. The HNN comprises feed-forward networks for saliency calculation from sensor data, a locally-

connected, topologically-organised *shunting* neural network (Yang and Meng, 2000) for modelling the agent's world, a feed-forward bridge between this model and the motor control parts of the architecture and a series of recurrent leaky integrator networks in the style of Beer's Continuous-Time RNNs (Beer and Gallagher, 1992) that actually produce motor output from the control system. These components we label the *decision network* (DN), the *shunting model* (SM), the *physical network* (PN) and the *pattern generators* (PG). Together, these components are able to successfully solve the 3D RC task, as demonstrated in Stanton and Channon (2015).

Since the details of this hybrid architecture are elucidated in previous work, we present a only a summary of the architecture below, along with notes on aspects that have been modified for the present work. See figure 1 for a detailed exposition in graphical form.

In our 2015 work, the DN and SM follow the ideas presented in Robinson et al. (2007) closely. Together and properly configured, they provide a neural-like encoding of a fixed action policy relating current state (position, local objects and carrying state) to action (preferred movement direction, and a pick-up or put-down action). In the first part of the present work, the focus is on physical performance rather than the species' capacities to learn an appropriate state–action policy. As such, we hard-code ι-values (saliency values) for objects in the agents' worlds rather than learn appropriate weights in the DN.

The PN controls the agent's behaviour in the world. The state transition landscape produced by the agent's SM is sampled at four points physically located on the agent's body and these values are used as input to this network. Thus, information about desirable state transitions (in this model, directions to move) is available to the PN and can be used by agents to discriminate important features of the preferred state configuration relative to the agent's configuration. The agent's configuration can then be updated to climb the gradient in the state space.

Actual control of the agent's body to achieve this reconfiguration is mediated by the PG network. The network is an array of five three-neuron oscillator circuits, comprising simple leaky-integrator neurons governed by a set of coupled differential equations, modelled after those of Reil and Husbands (2002). The PG network receives input from the PN that perturbs the oscillating cycles which in turn affects the agent's behaviour in the world. The oscillator circuits are a given abstraction in the agents' design, generated by a pre-evolutionary phase that is documented in previous work and summarised in the next section.

Last, outputs from this network are used as target angles for the various joints in the agent's body; actual torques are applied according to a proportional-derivative (PD) equation based on the difference between the current and desired angles at the joint.
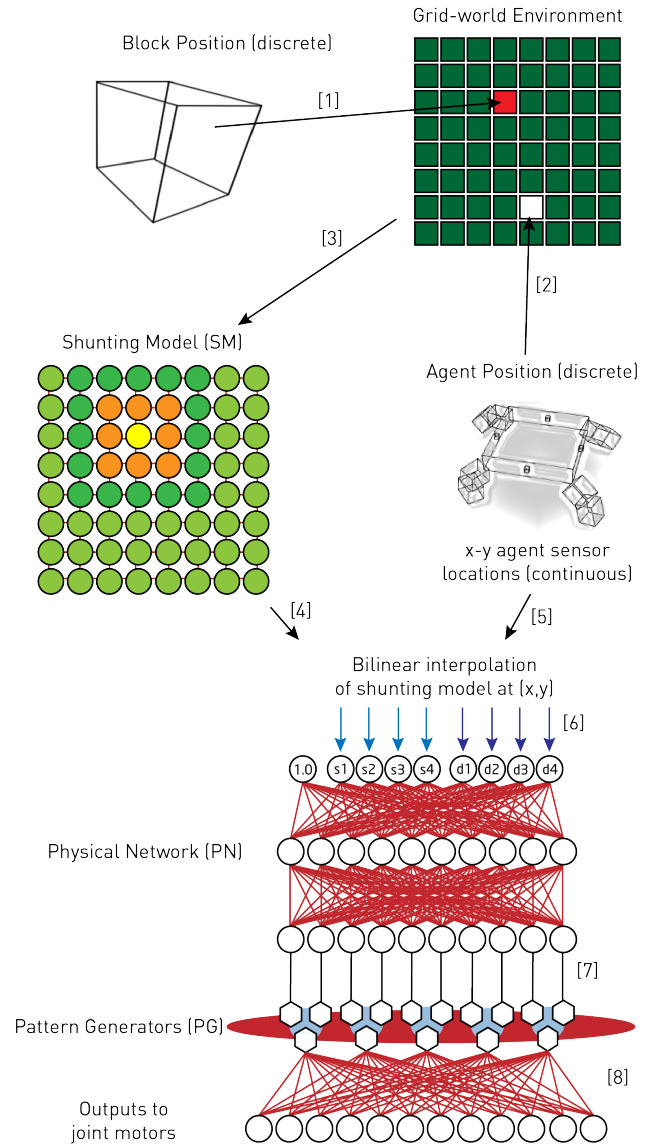

Figure 1: Neural architecture. The agent's 3D world, containing the agent and the block, is discretised into a 2D grid (1 and 2; cell-width is 1 unit in the physical model); grid locations are given ι-values where salient objects exist and this is used to generate the diffusive shunting model (3). Agents sample the landscape (4) at four different continuous positions given by their four sensors (5) by interpolation of values around the sensor location (6). These values pass through a feed-forward network and affect the dynamical trajectories of pattern generators (7) that ultimately output values to effectors via weighted links to joint motors (8). Links shown in red are subject to evolutionary optimisation, both in the pre-evolutionary phase and in the later block task. This includes the red region around the five preset pattern generators whose interneuron weights are also variable: within a single generator preset weights are adapted; across generators weights are initialised at zero but can also move from this value.

Evolutionary Algorithm

Pre-evolution As noted above, populations exploring the block task have been pre-evolved in other environments and also contain specific neural circuits that were produced in an additional, separate environment. These circuits were produced in isolation: three-neuron motifs were evaluated for their capacity to stably generate a 1Hz sinusoidal oscillation in the presence of an input signal and to be quiescent otherwise using an objective function based on the Fourier transform of their output over a 10-second window. The major pre-evolutionary phase involved the simulation of 20 species of agent in the original 3D RC environment. These species progressed through the documented incremental evolutionary phases of food collection, sprinting and hazard avoidance; the evolutionary process was halted before the deliberative part of the incremental challenge. (Specifically, agent populations were allowed 250k 3-individual tournaments; it was found that all 20 species had progressed to the deliberative component by this point.) The 20 species, all capable of tropotactic locomotion, were then installed in the block environment. All evaluation was carried out using a bespoke distributed evaluation system across approximately 200 CPU cores, achieving approximately 100 evaluations per minute.

Evolutionary Parameters In all cases the evolutionary algorithm is a three-individual tournament selection-based optimisation process, operating on a population of 150 genomes. Individuals' neuro-controllers are represented as an array of floating-point values. On reproduction, single-point crossover occurs between the two winning individuals in the tournament, and Gaussian mutation is applied to alleles of the resulting child genome with probability $1/l$, where $\mu = 0$ and $\sigma = 1$.

Objective Functions For the pre-evolution of oscillator circuits, the objective function was the number of non-1Hz frequencies in the frequency domain of a ten-second sample of the output neuron's signal in the input-high state, and the total number of frequencies in the input-low state. During the pre-evolution of gradient-ascending virtual creatures, the objective was as defined in (Stanton and Channon, 2015); agents of high fitness completed many of the incremental stages of the 3D RC task. For the evolved block-pushing task, the objective is to maximise the distance covered by the block in the discrete grid-world.

Data collection

To examine H1, we collected observations of agent behaviour, including extracting trajectory data from the highest-scoring individual from the BD task under various sensory ablation conditions. Deafferentation of control inputs was achieved by systematically disabling sensors, and the agent's progress in a controlled version of the BD task was recorded across a two minute time interval. For each ablation, we examine *approach* and *control*. In both cases the block is positioned at (20,20); for *approach* the agents start


Figure 2: Visualisation of a single agent in the block-displacement world. Agent is displaying a low, heavy gait suitable for block pushing.

far from the block at (5,5), and for *control* they start very close at (18,20). The trajectories followed by agents and block in the two scenarios illuminate the dependence of the gaits on sensory feedback. To examine H2, we used mean evolutionary performance data from the final 1000 tournaments of the 3D RC pre-evolution phase in comparison to the mean score of the same species in the BD task, evaluated for 2 minute and 10 minute periods (simulation time). To examine H3, we measure the naive BD score before evolution takes place of each individual in each population, in 10 randomly initialised trials. Each trial evaluates the individual for 10 minutes in the BD task. After the evolutionary phase, we repeat the process. We also collected evaluation data for each individual in each of 20 populations over 10 trials of 10 minutes each, after 100k tournament evolution when individuals begin with random genotypes. To examine H4, we use the evaluation fitnesses for naive and evolved species.

Results

H1: The hybrid architecture is sufficient to achieve feedback control that allows agents to successfully manipulate and guide an external object Visualisations of agent behaviour can be seen at `https://youtu.be/gZaUvXcdMK8`, and figure 2 provides a static view of an agent. The zoopraxiscopic figures (in the style of Eadweard Muybridge) show a time-series of snapshots that illustrate how agents approach the block from a distance (figure 3a), and manipulate the block in their world (figure 3b). The sensory ablation data are presented in figure 8. Figure 8a shows the planar trajectory followed by agents approaching the block from a distant point under various deafferentation conditions; figure 8b shows the response of agents to the same sensory culling in a closer, control scenario.

H2: There is some overlap between the 3D RC task and the BD task due to the requirement for speedy and accurate movement in both environments. A non-parametric correlation analysis was undertaken between the species' relative ranks for mean fitness during the final 1000 tournaments of the 250k-tournament 3D RC pre-evolutionary runs and the mean score on the BD task. Figure 4 presents this correlation graphically for both two minute and ten minute evaluation times. In the 10m trial we found a statis-
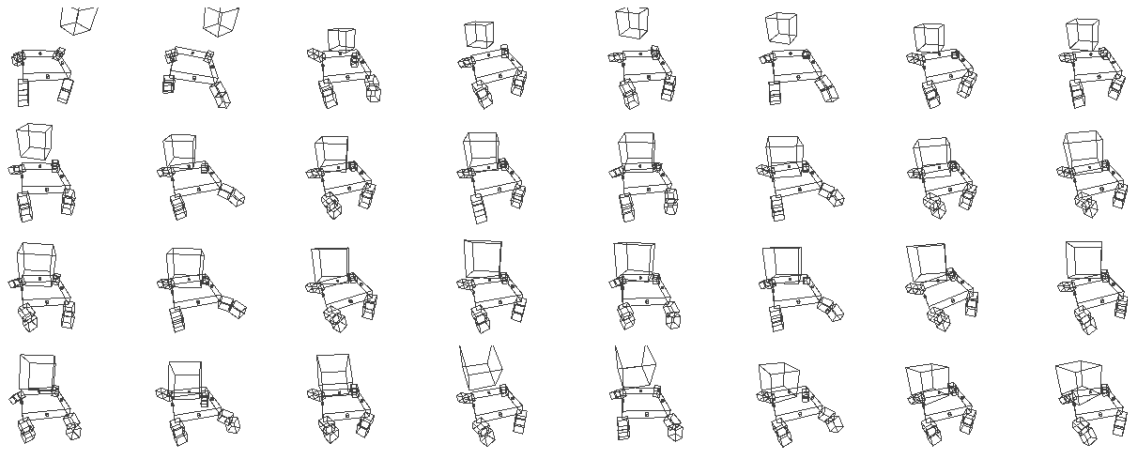
(a) Approach gait. The agent is moving toward the block from a distance. All limbs are contributing to the movement.
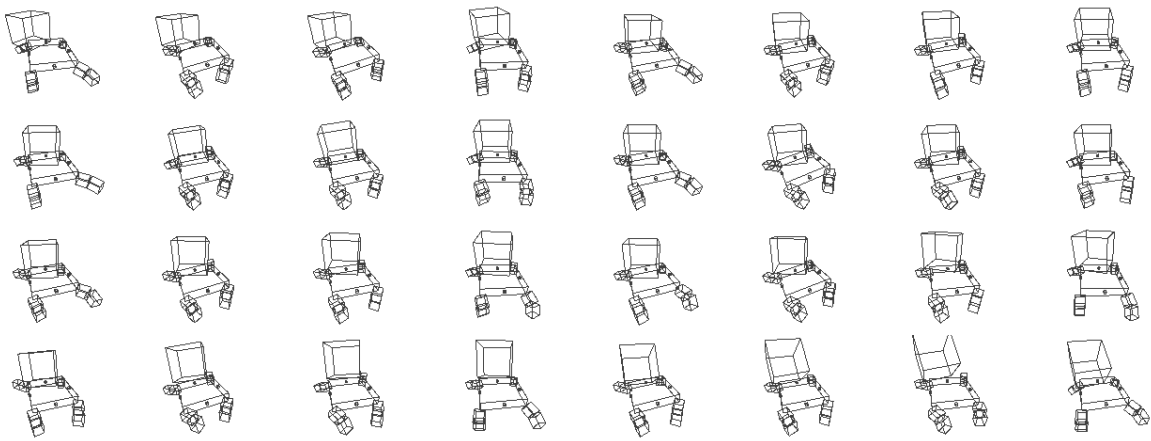
(b) Control gait. The agent is pushing forward with its 'back' limbs, maintaining the block between its forelimbs.

Figure 3: Zoopraxiscopic diagrams that show the gaits of the best evolved agent (run 11, individual 105). Presentation is in natural reading order, left-to-right, top-to-bottom. The viewpoint is fixed but tracks the agent as it moves through the world.

tically significant although weak correlation ($\rho = 0.38$; H_0 $p < 0.05$). The correlation between 3D RC and BD performance in the 2m BD trial is much stronger ($\rho = 0.51$; H_0 $p < 0.05$).

H3: Species evolved in the 3D RC task show increased performance after evolution in the BD environment. There is a clear improvement in all cases over the 25k tournament evolutionary run: the mean fitness over all naive populations was 37.29, compared to 124.16 in the evolved set (H_0 $p < 10^{10}$). Figure 5a shows progress of runs beginning from random genotypes over evolutionary time (100k tournaments in 1k tournament averages). Figure 5b shows the same view of populations beginning from naive genotypes, over 25k tournaments. Both treatments show a levelling off of fitness and there is no significant difference between the evaluation performance of the two starting conditions (figure 6).

H4: Some 3D RC species are on evolutionary trajectories more suitable for the BD task than others. We found a correlation between naive score and evolved scores across the 20 species ($\rho = 0.59$; H_0 $p < 0.01$) but no correlation between the naive scores and the magnitude of the change in fitness ($\rho = 0.26$; H_0 $p > 0.1$). Figure 7 demonstrates these relationships: δ-fitness is uncorrelated with naive fitness.

Conclusions and Discussion

We have shown that feedback motor control in evolved agents is possible with the given architecture, and that the architecture is flexible enough to support and adapt to a variety of evolutionary scenarios presented sequentially. This demonstrates that the platform has the potential to support environments that require even more sensorimotor control and is a reasonable starting point from where physical complexities can be added into the 3D RC task, eventually ap-
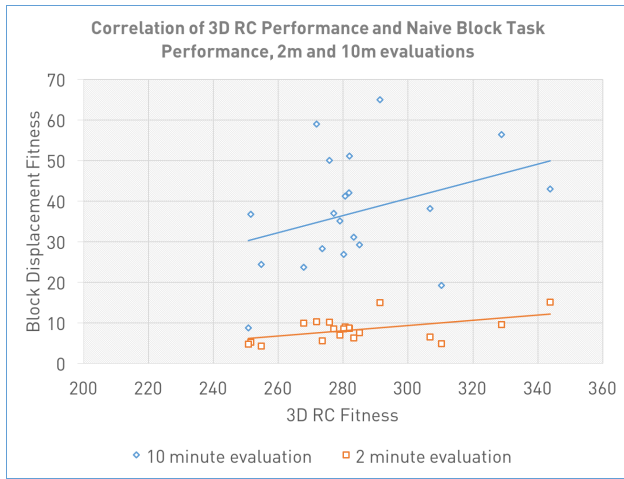
Figure 4: Across-species correlation comparing 3D RC performance and BD performance. Outcomes across the two tasks are more correlated when evaluation time is shorter ($\rho = 0.51$), indicating that movement speed is a factor in success in the block task and shared between the two problems. However, a strong gait is required to push the block and this is not selected for in the 3D RC task, hence the lesser correlation in the 10m task ($\rho = 0.38$).

(a) Fitness on the BD task (moving average over a 1000-tournament moving window) for evolution from a random (un-evolved) population.

(b) Fitness on the BD task (moving average over a 1000-tournament moving window) for evolution from a naive (evolved in 3D RC) population.

Figure 5: Progress of runs over evolutionary time; note that the x-axis differs due the different starting conditions and number of tournaments for each treatment.

proaching a full physical model of the problem. Observations of the agents' behaviour gained through 3D visualisation have revealed a rich variety of evolved strategies for solving the problem. Different classes of gait for approaching and manipulating the block appear due to the genetic heritage of species, and it is clear that low, heavy gaits work best for pushing the object in the BD task. From the deafferentation studies it can be seen that these gaits are not self-generating, blind gaits that simply aim the agent to the target location but are more complex aggregates of sensory data that depend on the agents position relative to the block in order to successfully achieve increased displacement.

When we consider the two evolutionary scenarios, 3D RC and BD, we found some overlap between the two problems. A strong correlation was observed between performance in the BD challenge before evolution in a two-minute evaluation, and performance at the end of the 3D RC task, indicating that some components of both challenges contribute similarly to relative agent fitness. This is likely to be the speed and directness of movement in the world which has a greater effect in a smaller evaluation period. As the evaluation period grows larger, this correlation decreases indicating that the block-pushing dimension of fitness in this scenario is not well captured in the 3D RC task and ultimately is the most important component. (It was also observed by measuring the time taken by agents to reach the block that most naive species sacrifice movement speed for block pushing capability during evolution, and this aspect should be investigated more thoroughly to determine whether this is an
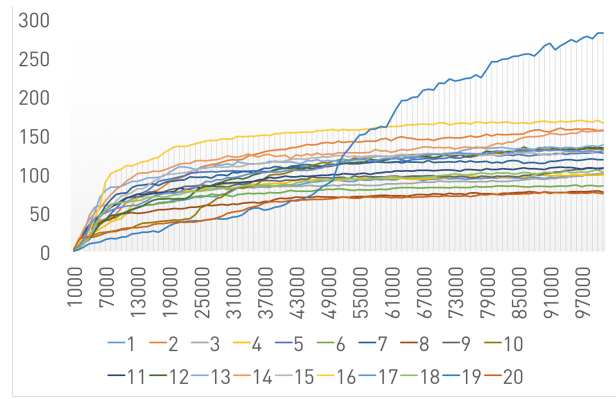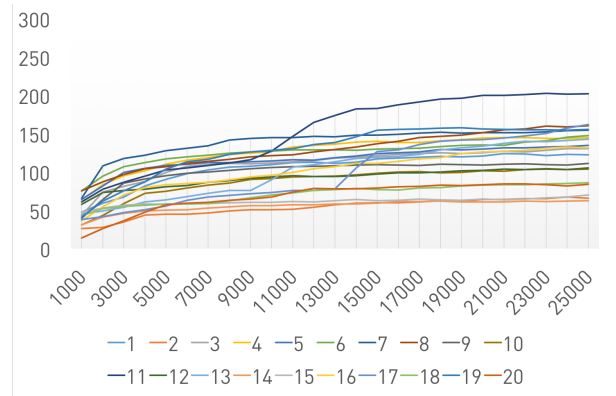
artefact or a consistent trend.) We showed that performance from either starting point (3D RC or unevolved genotypes) is comparable, demonstrating that an incremental approach incorporating both types of environment is possible in principle. We noted one extremely high-fitness run in the *random* category; upon visual inspection this species is a classic degenerate solution whose strategy is to rapidly vibrate the block to achieve high fitness. It is possible that more complex environments (such as 3D RC) prevent this kind of trivial solution by requiring a richer agent–environment interface. Our results comparing BD fitness before and after evolution demonstrate that whilst naive performance is an indicator of final performance, it is not an indicator of how much any particular species will improve. There is a risk that incrementally presenting new environments to only the most successful species could exclude good general solutions, a problem potentially mitigated by heterogeneous presentation of multiple environments.
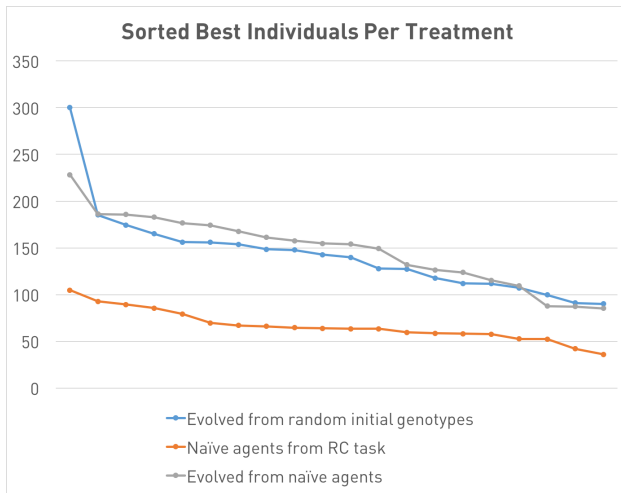
Figure 6: Comparison of the best individuals from the naive population, and from populations evolved from the random (unevolved) and naive-evolved populations.

Further work Ongoing work is toward the P3D RC task: a physically-embodied deliberative river crossing problem. The next step is to consider not just displacement but also positioning of the block using the shunting landscape. This is likely to demand significant revision of the underlying control architecture to incorporate reasoning about relative positioning. Additionally, the question of whether specific types of solutions in the 3D RC world have specific performance profiles in the BD world could be addressed by examining in detail whether some species always slow down and some always speed up. Additionally, it is possible that evolved morphology could significantly contribute to physical manipulation behaviours.
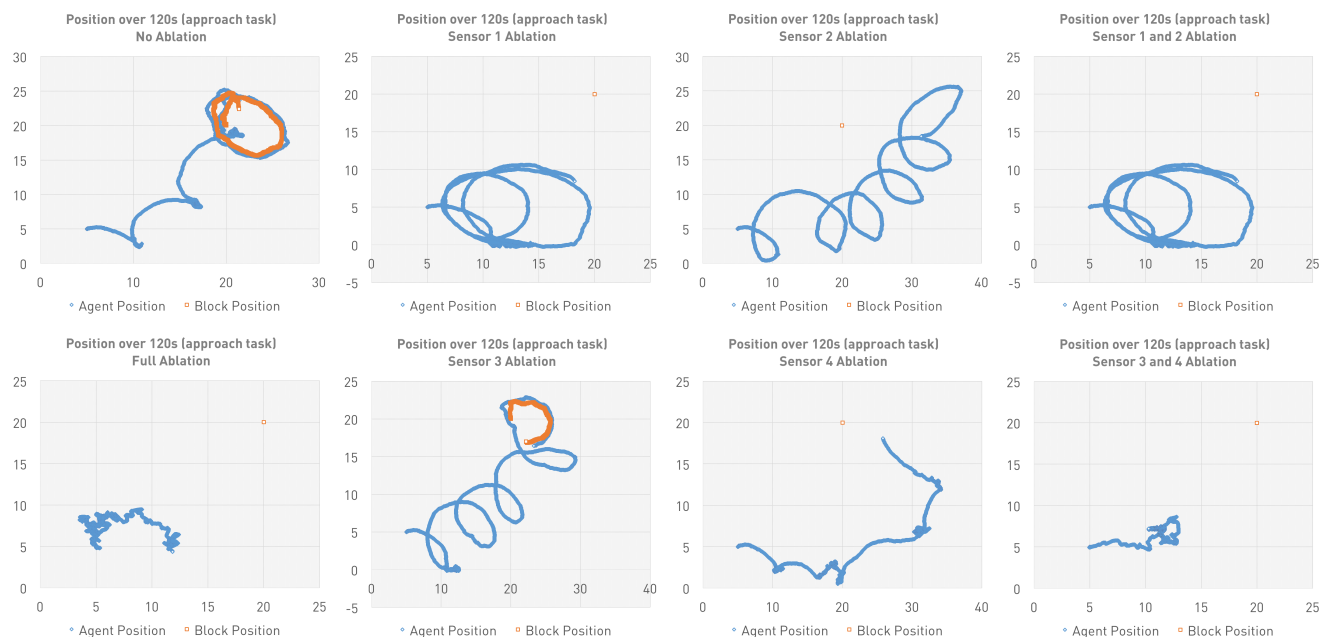
Acknowledgements

References

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

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

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

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*, pages 345–354. Springer.

Sims, K. (1994a). Evolving 3d morphology and behavior by competition. In *Artificial Life IV*, pages 28–39. MIT Press.

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

Stanton, A. and Channon, A. (2015). Incremental neuroevolution of reactive and deliberative 3d agents. In *Advances in Artificial Life, ECAL 2015: Proceedings of the Thirteenth European Conference on the Synthesis and Simulation of Living Systems*, pages 341–348. MIT Press.

Tange, O. (2011). Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

Yang, S. X. and Meng, M. (2000). An efficient neural network approach to dynamic robot motion planning. *Neural Networks*, 13(2):143 – 148.

Figure 7: Correlation of evolved fitness with naive fitness ($\rho = 0.59$), and delta fitness with naive fitness ($\rho = 0.26$).

(a) Approach task. The agent begins at (5,5) and attempts to reach the block. The unaltered agent's trajectory is shown in the top left; this agent tends to overshoot its target and then correct by rotating, as the two loops in the path record. All sensors have some effect on this behaviour although sensor 1 is by far the most pronounced difference in a single cut. In complete deafferentation (bottom right) the agent moves randomly.
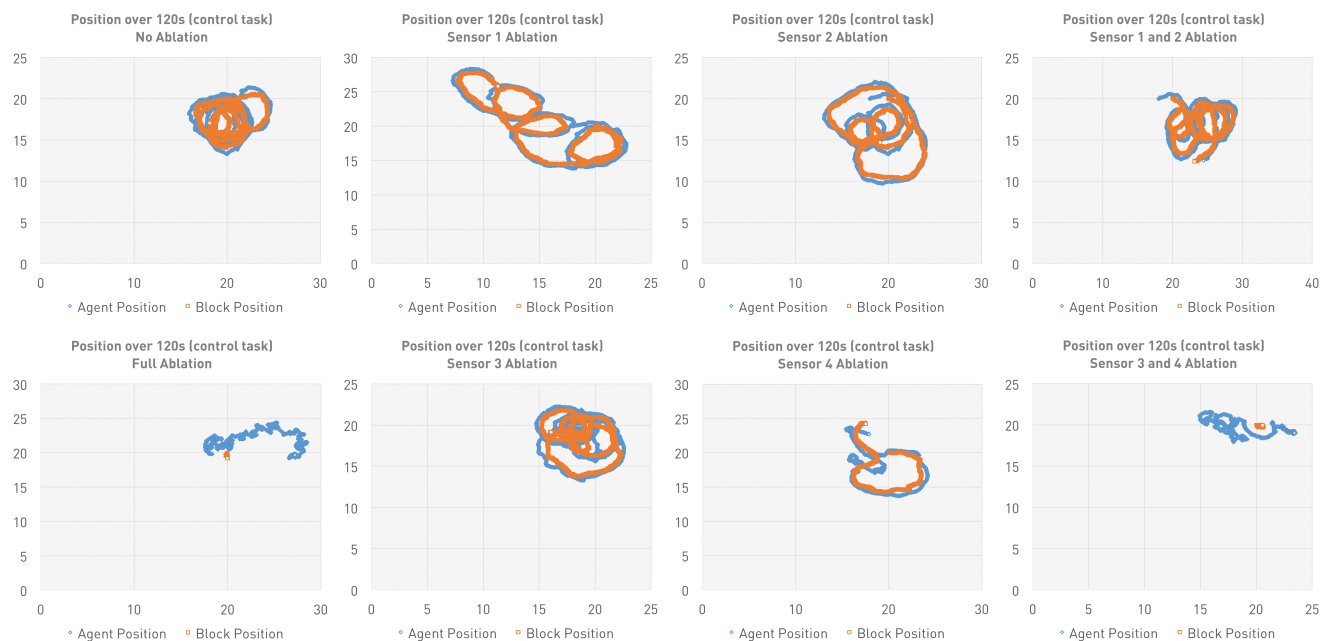
(b) Control task. The agent begins at (18,20), adjacent to the block. The unaltered agent pushes the block in a tight circle to maximise fitness (top left). Sensor ablations do not have a catastrophic effect as in the approach task; all single cuts still maintain block movement although the trajectory is less efficient, as does the dual cut of sensors 1 and 2. Only by cutting sensors 3 and 4 or complete deafferentation did we observe failure to displace the block at all.

Figure 8: Agent-block trajectories of best agent from best overall trained population under various sensor ablation treatments. The figure demonstrates how a combination of sensory inputs is necessary for reliable gait generation for distance approach and to control the block. In all cases the block initially rests at (20,20).