Simulations of simulations in evolutionary robotics

Edgar Bermudez Contreras and Anil K. Seth

Department of Informatics, University of Sussex,
Falmer, Brighton, BN1 9QJ, UK.
Email: {e.j.bermudez-contreras, a.k.seth}@sussex.ac.uk

Abstract. In recent years simulation tools for agent-environment interactions have included increasingly complex and physically realistic conditions. These simulations pose challenges for researchers interested in evolutionary robotics because the computational expense of running multiple evaluations can be very high. Here, we address this issue by applying evolutionary techniques to a simplified simulation of a simulation itself. We show this approach to be successful when transferring controllers evolved for example visual tasks from a simplified simulation to a comparatively rich visual simulation.

1 Introduction

For more than a decade, evolutionary robotics (ER) has struggled with the challenge of producing controllers that function in real world environments. The approach of evolving in the real world itself is prohibitively time consuming in all but the simplest of cases [1],[3]. A popular alternative has been to evolve controllers in simulations, but simulations are often poor abstractions of the complexities of real world environments. This situation is changing. Recent years have witnessed enormous growth in the sophistication of simulation tools for modelling agent-environment interactions. Highly detailed physics-based simulations are now readily available 'off-the-shelf' which simulate not only complex morphologies but also rich streams of sensory input and motor output signals [5],[6]. While impressively realistic, these simulations can be highly computationally expensive and as a result can pose challenges similar to those posed by evolution in the real world.

This is not to say that evolution in a rich simulation is as problematic as evolving in the real world. Even a very rich simulation can likely be executed more rapidly (and with less chance of hardware failure) than a corresponding real world condition. If this is not true at present for a particular simulation, future increases in computational power will undoubtedly compensate. In addition, rich simulations offer the possibility of exploring detailed but non-physical agent-environment interactions, which may shed light on adaptive behavior by providing alternative comparison conditions to agent-environment interactions in real-world situations.

In this paper, we consider the construction of simplified simulations of simulations themselves, from the perspective of ER. We describe the construction of a simulation of a rich simulation of visually guided behavior and illustrate the value of this simplified simulation by evolving controllers to perform two visually guided tasks: object approaching and object discrimination. We show that controllers evolved in the simplified simulation transfer successfully into the rich visual simulation, despite there being significant differences in the structure of sensory input in the two cases. Our work therefore suggests the possibility of a "hierarchy" of simulations of progressively increasing complexity as a means of (i) evolving controllers for real world operation, and (ii) probing the dynamical structure of adaptive agent-environment interactions.

2 Methods

Simulated agents were evolved to perform two visually guided tasks. The first task, 'object approaching', required the agent to approach an object placed in the arena and be as close as possible to the object at the end of a fixed period of time. The second task, 'object discrimination', required the agents to discriminate between two different objects by approaching only one of them and remaining as close as possible to that object at the end of a period of time.

Two types of agents were simulated, one with a visual system using a simulated camera, called the "rich simulated agent" (RSA) and another using a simplified visual system called the "simple simulated agent" (SSA). As described below, the visual systems of both agents were tailored to each visual task. A genetic algorithm (GA) was used to evolve continuous-time recurrent neural network controllers (CTRNNs) for SSAs, and successful controllers were analyzed both as controllers for SSAs and as controllers for RSAs.

2.1 Rich simulated agent (RSA)

The RSA has a circular body with two wheels driven by two independent motors, and a camera on top of its body. The visual system of the RSA has a visual field which is a grey-scale region of the output of the simulated camera. This region is 512×32 pixels (see figure 1). The visual system has a blob detection mechanism and two types of sensors. The blob detection mechanism selects visual subregions of consistent pixel intensity with area in the range 10-50 pixels. Only one 'blob' is selected at any time. In cases where there is more than one blob in the visual field, the visual system selects the blob with largest area.

Two types of sensor respond to a selected blob. The first is a "location sensor" which responds to the proximity of the object to the edge of the visual field (the sub-region of the image captured from the camera). The RSA has left and right location sensors. These sensors are activated by the inverse of the distance (L or R) between the object and the corresponding edge of the visual field (see figure 1). The second sensor type, used only in the object discrimination task, are "colour sensors" that return the pixel intensity of the centroid of the selected

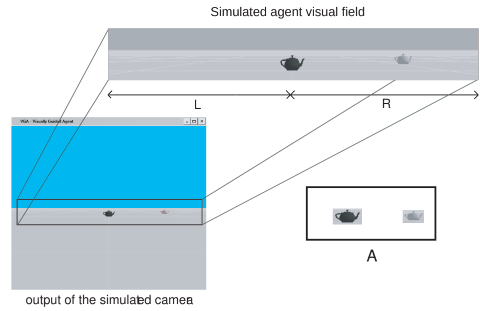

Fig. 1. Simulated visual system. The visual field of the agent is a region of 512×32 pixels. L is the distance from the object to the left edge of the visual field and R is the distance from the object to the right edge. The inset A in the figure shows the detected blobs (from a distance of 2.5 to the dark object and 3.0 units to the light object) containing the light and dark objects respectively. In this example, the dark object is the largest and so the sensor neurons will respond to this object.

blob. Because the rich visual simulation incorporates directional illumination and reflectance properties of the objects in the arena, the pixel intensity at any time is a complex function of intrinsic properties of the object detected and the reflectance of the object in the corresponding region of the visual field. Although the two colour sensors receive identical input (unlike the location sensors), they may still produce different outputs depending on intrinsic neuron properties (see below).

At the beginning of each evaluation, an RSA was randomly positioned within a region of 12x12 units in an unlimited arena. For object approaching experiments, a visual object (a dark-coloured kettle) was placed in a fixed position in the arena. For the object discrimination task, a light coloured kettle (target) and a dark coloured kettle (distracter) (see inset A in figure 1) were placed in the arena in positions (0, -4) and (0, 4), respectively. During evolution, each evaluation lasted for 200 time-steps; during analysis of evolved controllers, each evaluation lasted for 800 time-steps.

2.2 Simple simulated agent (SSA)

The SSA has a circular body with radius of 0.5 units and two wheels on both sides of the agent, driven by two independent motors. The simplified simulated visual system of this agent has a visual field that is restricted to a region of fixed width V. This region is also limited by two lines originating from the center of the agent extending ±45 deg from the orientation of the agent (see figure 2). It is important to emphasize that this region is spatial, in the sense that it is defined in terms of a subregion of the arena, rather than, as is the case for the RSA, as a subregion of a visual image. This difference means that sensory signals for the two agents will have different dynamical structures. For example, it is possible that a visual object will move in and out of view for the SSA (because of the fixed width V of the visual field) while remaining constantly within view for the RSA. (One situation in which this may occur is as an agent spins.)

It is also important to notice that there is a spatial region near to the agent where the SSA is blind (the light gray region in figure 2). As we describe below, this blind region is important in the explanation of evolved behaviour of the SSA.

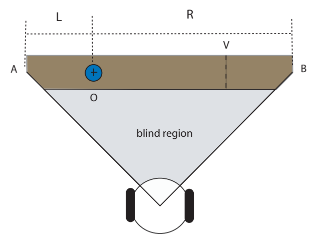

Fig. 2. Visual field of the SSA: the object (O) can only be sensed if it is within the dark region. This region is limited by two lines extending from the center of the agent with ± 45 deg from the orientation line and a width of V. L and R are the distances between the object and the left and right edges of the visual field, respectively.

As with the RSA, the SSA has two types of sensor which take input from the visual system. The location sensors of the SSA are activated by the inverse of the distance (L or R) between the object (if it is within the visual field) and the corresponding edge of the visual field. The colour sensors of the SSA return a similar value to the pixel intensity of the objects (40 and 130 for the dark and light kettles, respectively) used for the RSA. To deal with the variation in values of colour sensors for RSAs (resulting from changes in reflectance and in intrinsic properties of the selected blob), colour sensors for SSAs were modulated by a random value [-30, 30] (distributed uniformly).

2.3 Controller

The controllers for both types of agents were Continuous Time Recurrent Neural Networks (CTRNNs)[3], [2]. In a CTRNN, the state y of each neuron i changes in time according to the differential equation:

$$\tau_i \dot{y}_i = -y_i \sum_j w_{ij} \phi(y_j + \beta_j) + g_i \cdot I_i$$

where ϕ is the sigmoid activation function, τ is a time constant, β is a bias, and w_{ij} represent connection weights from neuron i to neuron j. The state of each neuron is therefore the integration of the weighted sum of all incoming connections (plus a gain modulated input $g_i \cdot I_i$ for input neurons).

For the object approaching task, the CTRNN consisted of eight neurons, specifically, two sensor neurons, four fully connected interneurons and two motor neurons. For the discrimination task, two more sensor neurons corresponding to the colour sensors were added (see figure 3). Parameter values for all neurons were initialised in the following ranges: $\tau \in [0.2, 2.0]$, $\beta \in [-10, 10]$, and connection weights $w_{ij} \in [-5, 5]$. In the object discrimination task, neurons 8 and 9 used $\tau \in [0.2, 10.2]$ and bias $\beta \in [-30, 30]$. All parameter values were shaped by the GA (see below).

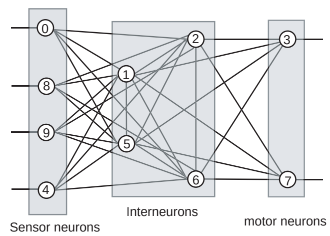

Fig. 3. Controller. Neurons: 0 and 4 are location sensors; 8 and 9 are colour sensors. Neurons 1,2, 5 and 6 are fully connected. Neuron 3 is the left motor neuron and neuron 7 is the right motor neuron. Note that the colour sensor neurons 8 and 9 were *not* used for the object approaching task.

2.4 Genetic algorithm

A distributed GA was used to evolve CTRNNs to perform the visually guided tasks. The genome of each individual was coded as a real vector of 32 elements for the object approaching controller and 39 elements for the object discrimination controller. For the 32 element vector, 4 elements were used to code the time constants of each neuron, 4 for the bias of each neuron, 2 for the sensor gains and 22 for the weights. Each element was coded as a real number in $[0, 1]$ and linearly scaled according to the parameters previously described in section 2.3. For the 10 neuron controller 7 elements were added, 2 for the bias of the two extra sensor neurons, 2 for the time constant, 1 for a sensor gain for these neurons and 2 for the weights. A population of 400 individuals was evolved with mutation probability of 80% for each genotype and 20% for mutation change for each vector element. There was also a 5% probability of crossover and an elitism probability of 80%.

The controllers were symmetrical (i.e., same parameters were used for each pair of sensor neurons, 0 and 4; 1 and 5; 2 and 6 and so on. See figure 3.), except for neurons 8 and 9 which had independent parameters.

Two fitness functions $F_1 = 1/d_f$ and $F_2 = 1/d_l - 1/d_d$ were used. F_1 was used for the object approaching task and F_2 for the object discrimination task. In F_1, d_f is the distance from the agent to the object at the end of the trial and in F_2, d_l and d_d are the final distances between the agent and the light and dark objects respectively. The fitness of each individual was calculated as the average across 5 independent trials (of 200 time-steps each).

3 Results

After several thousands of generations controllers were successfully evolved for both tasks. As mentioned previously, controllers were evolved using the SSA and were then tested in both types of agents, SSA and RSA.

3.1 Object approaching task

For this simple task successful controllers were found quickly (before 2000 generations). As we can see in figure 4B, the agents used an exploratory strategy, first spinning until the object was within the field of view and then approaching the object and rotating around or very close to it.

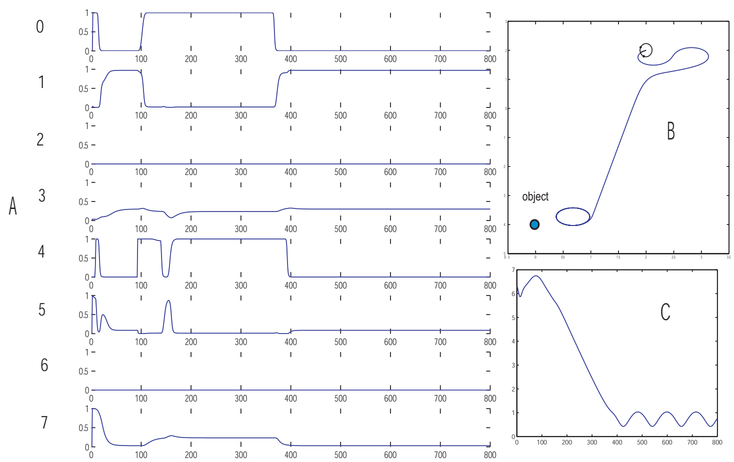

Fig. 4. Object approaching by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the distance between the agent and the object during the trial and [C] shows the distance between the agent and the object during the trial.

Successful controllers for SSAs were tested in agents using the rich visual system (RSAs). These evolved controllers also performed the object approaching task successfully (see figure 5B). The behaviour of the RSAs was similar to that observed for SSAs: rotate or explore until the object is within the visual field, approach the object and then rotate close to it. In the particular case shown in the figures, the circle described by the trajectory of the RSA at the end of the trial is bigger than that described by the trajectory of the SSA. This observation is highlighted by figures 4C and 5C, where the distance to the object is shown during the test trial.

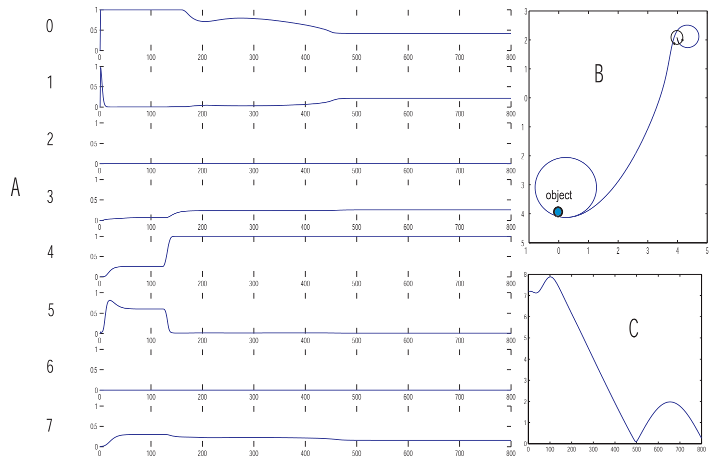

Fig. 5. Object approaching performed by an RSA using the evolved controller shown in figure 4. [A] shows the neural activity during the test trial of 800 time-steps. [B] shows the trajectory of the agent the trial and [C] shows the distance between the agent and the object during the trial.

3.2 Object discrimination task

In this case the task was to discriminate the objects using the pixel intensity information. Successful discrimination was reflected by approach to the target object (the light-coloured object). SSAs were successfully evolved to perform this task. Figure 6 shows a SSA performing the object discrimination task during a test trial.

As shown in figure 6, the dark (distracter) object is initially within the field of view but the agent nevertheless turns towards the target object and then approaches it. At the end of the trial the agent rotates in close proximity to the target object. The same controller transferred successfully to the RSA. Figure 7 shows an RSA performing object discrimination task with the evolved controller.

As in the first task, the behaviour of the RSA is similar to that of the SSA. The agent rotates until the object is within its visual field and then approaches it. In the trial shown in figure 7, the dark object is closer to the agent at the beginning of the trial, however, after a short time, the agent moves away from the dark object and subsequently approaches the target. Note that for the object discrimination task, both SSAs and RSAs stay very close to the target object (compare figures 6C and 7C).

It is important to emphasize that, for this task, certain aspects of simulation of the colour sensors were critical for the successful transfer of controllers. Specifically, evolutionary runs in which random variance in these sensor values was not incorporated (see section 2.2) showed considerably decreased performance when transfer to an RSA was attempted. During attempted transfer in these cases, variance in the RSA colour sensor values (due to the richness of

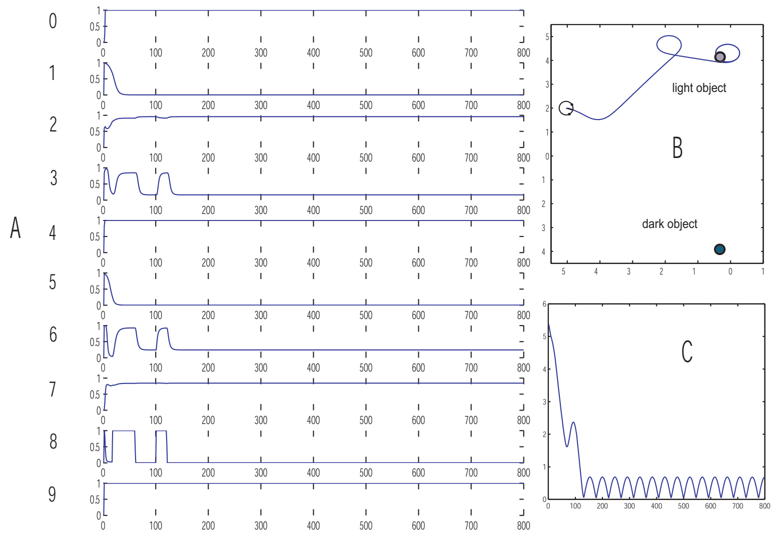

Fig. 6. Object discrimination performed by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the object during the test trial.

the visual simulation) resulted in these agents approaching both object types equally often.

4 Analysis

In general the strategies of both SSAs and RSAs can be described as follows. First, agents rotated until an object was within the field of view, then agents approached the object, and finally, agents rotated either close to or around the object, until the end of the trial.

In order to better understand the dynamics of evolved behaviours and the factors underlying successful transfer between simulations, we now examine evolved behaviors in terms of neural activity. For both agent types, the initial rotating behaviour can be attributed to the random initialisation of the CTRNN. This was shown by initialising the neurons uniformly, in which case SSAs and RSAs navigated in a straight line at an arbitrary heading (data not shown). The approach behaviour of both agent types can be attributed to sensor activation corresponding to an object perturbing the equilibrium point in neural dynamics corresponding to the spinning behaviour. This was shown by testing SSAs and RSAs without any object in the arena (data not shown).

For object approaching task, neurons 2 and 6 were always constantly saturated for both agent types and therefore can be discarded from the analysis (see figure 4A and 5A and figure 3), leaving only neurons 1 and 5 as modulators of motor neuron activity (see figure 3). For the object discrimination task, all

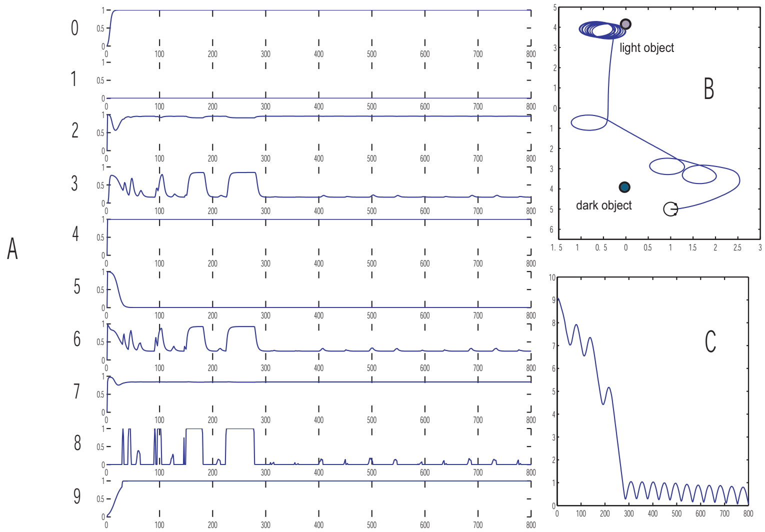

Fig. 7. Object discrimination performed by an RSA using the evolved controller shown in figure 6. [A] shows the neural activity during a test trial of 800 timesteps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the light object during the trial.

the sensor neurons are constantly saturated except for neuron 8 (again for both agent types). Since this type of neuron has a different weight for each connection, it is still able to modulate neuron 6 which in turn is responsible for regulating the motor neurons (see figure 6 and 7).

The final segment of successful agent behaviour involved rotating close to an object. This behavior was related to the initial rotating (described previously). Once the agents were sufficiently close to the object so that the object was within the "blind region" (see section 2.2), they reverted to spinning. In the object approaching task, when this happens, the agent could no longer sense any object and the situation was equivalent to the one where no object was present. For the object discrimination task, once the agent was spinning very close to the target object but was not able to sense it, the agent could still sense the dark (distracter) object (see neuron 8 in figure 7A, the small peaks correspond to the dark object and high peaks to the light object) but the activation of the sensor neuron was not high enough to trigger approaching behaviour. This situation is not shown in figure 6 because the agent is spinning too far away from the dark object to be able to detect it, however the same situation applies to both SSAs and RSAs.

In general, the behaviour of the evolved controllers shows that despite the differences in the dynamical structure of sensory signals between SSAs and RSAs, evolved controllers transferred successfully from one to the other. As the neural analysis shows, this transfer was possible because evolved agents relied on consistent features of sensory activity, and not on those aspects that varied between the agent types (see section 2.2).

5 Conclusions and future work

In this work, it was shown that evolved controllers for agents using a simplified visual system (SSAs) could be successfully transferred to agents using more complex visual information (RSAs). The behaviour of both agents (SSAs and RSAs) for object approaching and discrimination was fully explained by analysing the dynamics of their neural activity. In this way, it was shown that the complexity gap between SSAs and RSAs was crossed.

This demonstration is useful for evolutionary robotics in several ways. First, the development of increasingly complex simulations is blurring the distinction between simulation and reality, therefore an important future goal for ER will be to create adaptive controllers for agents in simulations, and not only as a bridge to real-world situations. On the other hand, a hierarchy can be envisaged in which controllers are initially evolved in simple simulations and then are incrementally refined in progressively more complex simulations until final deployment in a real world environment. Alternatively, rich simulations offer the possibility of exploring detailed agent-environment interactions which do not exist in real-world situations, thereby supplying potentially valuable comparison conditions for understanding mechanisms of adaptive behaviour.

Future work in this area could usefully consider the development of *minimal* simulations of rich simulations, in the sense described by Jakobi [4] . Minimal simulations incorporate extremely high levels of noise in specific loci in order to ensure that evolved controllers cannot rely on these aspects of agent-environment interaction. This method might extend the 'complexity gap' between simulations that can be feasibly traversed by evolutionary approaches.

References

1. D. Cliff, I. Harvey, and P. Husbands. Artificial evolution of visual control systems for robots. In M. Srinivisan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines*, pages 126–157. Oxford University Press, 1997.
2. Ken-Ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw.*, 6(6):801–806, 1993.
3. I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi. Evolutionary robotics: the sussex approach. *In Press*, 1996.
4. N. Jakobi. The minimal simulation approach to evolutionary robotics. In T. Gomi, editor, *Proceedings of Evolutionary Robotics - From Intelligent Robots to Artificial Life*. AAI Books, 1998.
5. N. Koening and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Proceedings of International Conference on Intelligent Robots and Systems*, volume 3, pages 2149–2154, 2004.
6. O Michel. Webotstm: Professional mobile robot simulation. *International Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.