

# Emergence of Memory-Driven Command Neurons in Evolved Artificial Agents

Ranit Aharonov-Barki and Tuvik Beker  
Center for Computational Neuroscience  
The Hebrew University, Jerusalem, Israel  
*ranit@cns.tau.ac.il, tuvik@cns.tau.ac.il*

Eytan Ruppin  
Dept. of Computer Science and Dept. of Physiology  
School of Mathematical Sciences and School of Medicine  
Tel-Aviv University, Tel-Aviv 69978, Israel  
*ruppin@math.tau.ac.il*

[In **Neural Computation** 13(3), 2001]

## Abstract

Using evolutionary simulations we develop autonomous agents controlled by artificial neural networks (ANNs). In simple life-like tasks of foraging and navigation, high performance levels are attained by agents equipped with fully-recurrent ANN controllers. In a set of experiments sharing the same behavioural task but differing in the sensory input available to the agents, we find a common structure of a *command neuron* switching the dynamics of the network between radically different behavioural modes. When sensory position information is available the command neuron reflects a map of the environment, acting as a *location-dependent cell* sensitive to the location and orientation of the agent. When such information is unavailable the command neuron's activity is based on a spontaneously evolving *short-term memory* mechanism, which underlies its apparent place-sensitive activity. A two-parameter stochastic model for this memory mechanism is proposed. We show that the parameter values emerging via the evolutionary simulations are near optimal; evolution takes advantage of seemingly harmful features of the environment to maximize the agent's foraging efficiency. The accessibility of evolved ANNs for a detailed inspection, together with the resemblance of some of the results to known findings from neurobiology places evolved ANNs as an excellent candidate model for the study of structure and function relationship in complex nervous systems.

## 1 Introduction

In recent years a novel paradigm emerged in the study of ANNs. This paradigm uses genetic algorithms [Mitchell, 1996] and evolutionary computation [Fogel, 1995] to develop

ANNs. Work in this field can be divided to the development of *isolated* ANNs, evolving to maximize a certain target function on one hand [Kitano, 1990; Harrald and Kamstra, 1997], and the development of *embedded* ANNs, serving as the control mechanism for an autonomous agent, on the other hand [Gallagher and Beer, 1999; Gomez and Miikkulainen, 1997; Jakobi, 1998; Scheier, Pfeifer and Kuniyoshi, 1998; Kodjabachian and Meyer, 1998]. In the latter case the agents perform certain behavioural tasks, and their performance level in these tasks serves as the basis for evolutionary selection. This new paradigm of *Evolved ANNs (EANNs)* is clearly very interesting from the applicative point of view, opening new horizons to the development of robotic control mechanisms. However, its relevance to our understanding of biological neural systems has not yet gained wide recognition.

**There are two fundamental questions concerning EANNs we address in this paper: 1. Can we identify and analyze neurons and network structures emerging in EANNs which play an important information processing role in controlling autonomous agents? And if so, then 2. Do structures resembling findings from biology indeed occur in EANNs?** Using evolutionary simulations we developed autonomous agents controlled by ANNs, and conducted a thorough analysis of the evolved neuro-controllers. Previous works on evolved ANNs (e.g. [Cangelosi, Parisi and Nolfi, 1994; Beer, 1997]) analyzed the emerging neural structures and showed interesting structure to function relationship. These works typically used very small networks (3-5 neurons) or networks with pre-determined feed-forward structure. We extend this line of work by studying networks with tens of neurons and over 100 synapses and unconstrained recurrent architecture. Our results strongly suggest that EANNs are relatively tractable models to analyze, manifesting biological-like characteristics.

The EANN-controlled autonomous agents we develop use unconstrained network connectivity patterns to perform simple life-like behavioural tasks. Under these conditions, we show that networks maintaining steady activation levels can evolve, and moreover – serve to control agents that perform at a remarkably high level compared with algorithmic benchmarks. Non-trivial network structures evolve in these agents. We analyze these structures and demonstrate the existence of neurons whose functional repertoire strongly

resembles that of “command neurons” known from biological networks [Combes, Meyrand and Simmers, 1999; Teyke, Weiss and Kupfermann, 1990; Nagahama, Weiss and Kupfermann, 1994]. The command neuron’s activity is driven either by positional information or by a short-term memory mechanism, depending on the specific sensory information available to the agent.

The emergence of location-dependent cells in EANNs has previously been demonstrated by [Floreano and Mondada, 1996] who studied homing navigation of a real robot. They found a neuron in the controlling EANN that exhibits location- and orientation-dependent activity. [Jakobi, 1998] has described a simple memory-based behaviour in a small, bilaterally symmetrical EANN with 10 neurons and about 20 synapses. In this paper we revisit both of these issues, showing how they *emerge concomitantly* in agents with various sensory capabilities performing a task requiring simple navigation and foraging skills. We conduct a thorough analysis of the evolved networks and the computations they perform. We study the emergence of location-dependent cells also under conditions in which the agents are deprived of *any* positional cues, and find that the apparent place-dependent activity is actually the result of an emerging memory mechanism culminating in a single neuron. This demonstrates that different computations can result in the same phenomenon of a “place cell”. We show that the emerging place cell takes the role of a command neuron which modulates a complex switch in network dynamics between two distinct operational modes, that in turn manifest themselves in two different behavioural modes of the agent.

The rest of the paper is organized as follows: Section 2 gives an overview of the model. A more elaborate description can be found in Appendix 1. Section 3 describes the performance levels attained by evolved agents. In section 4 we demonstrate the emergence of command neurons modulating the behaviour, and define the two basic modes of behaviour. Sections 5 and 6 investigate the properties of the command neurons under two different sensory scenarios. We show that when sensory information is scarce a memory mechanism is evolved and helps to control the behaviour of the agent. Section 7 investigates in detail the network structure in a particular successful agent. Finally, section 8 discusses the results. Appendix 1 gives a self-contained, detailed description of the simulation model.

Appendix 2 describes the various behavioural measures we used to quantify the agents’ two basic behavioural modes and appendix 3 gives the algorithm we wrote as a benchmark for one of the behavioural tasks.

## 2 The Model

The basic environment consists of a grid arena surrounded by walls. In this arena two kinds of resources are scattered. “Poison” is randomly scattered all over the arena. Consuming this resource decreases the fitness of an agent. “Food”, the consumption of which increases fitness, is randomly scattered in a restricted “food zone” in the south-western corner of the arena. The agents’ behavioural task resembles other models found in the literature (e.g. [Nolfi, Elman and Parisi, 1994; Miglino, Nolfi and Parisi, 1996] and is seemingly very simple - to eat as much of the food while avoiding the poison. The complexity of the task in this work stems from the limited and low-level sensory information the agents have about their environment. The agents are equipped with a set of sensors, motors, and a *fully-recurrent* ANN controller. It is this neurocontroller that is coded in the genome and evolved; the sensors and motors are given and constant. The models we explore differ in the sensors the agents are equipped with, the size of the network (15-50 neurons), and whether or not the food zone border is marked.

The agents in all the models explored were equipped with a basic sensor we termed *somatosensor*, consisting of five probes. Four probes sense the grid cell the agent is located in and the three grid cells ahead of it (see Figure 1). These probes can sense the difference between an empty cell, a cell containing a resource (either poison or food – with no distinction between those two cases), an arena boundary and food zone boundary. The fifth probe can be thought of as a *smell probe*, which can discriminate between food and poison just under the agent, but which gives a random identification if the agent is not standing on a resource. This requires sensory integration in order to identify the presence of food or poison. In addition, in some models the agents are equipped with a *position sensor*, effectively giving the absolute coordinates of the agent in the arena. Note that the somatosensor alone gives only purely local information to the agent, making navigation in the environment a

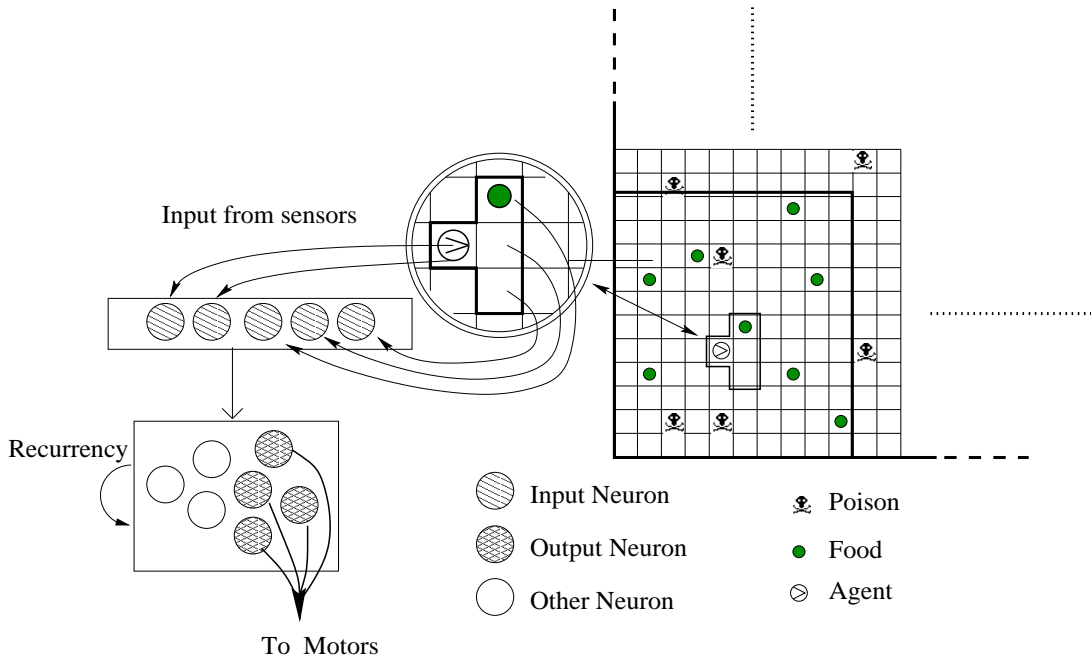


Figure 1: *An outline of the grid arena (southwest corner) and the agent's controlling network.* The agent is marked by a small arrow on the grid, whose direction indicates its orientation. The curved lines indicate where in the arena each of the sensory inputs comes from. Output neurons and interneurons are all fully connected to each other.

challenging task. Moreover, even in models where the agents are equipped with a position sensor, they lack any information about orientation, so navigation remains a difficult task. The motor system allows the agent to go forward, turn 90 degrees in each direction, and attempt to eat, a costly procedure since it requires a step with no movement.

In each generation a population of 100 agents is evaluated. The life cycle of an agent (an *epoch*) lasts 150 time steps, each step consisting of one sensory reading, network updating, and one motor action. At the end of its life-cycle each agent receives a fitness score calculated as the total amount of food it has consumed minus the total amount of poison it has eaten and normalized by the number of food items available to give a maximal value of 1. Simulations last a pre-defined number of generations, ranging between 10000 and 30000. Figure 2 shows a typical evolutionary run. The initial population consists of agents equipped with random neuro-controllers. In a typical simulation run, the average fitness of agents in the initial population is around -0.05. As evolution proceeds better controllers emerge and both the best and average fitness in the population increase until a plateau is reached. In the next section we assess the performance of the evolved agents at these final generations by

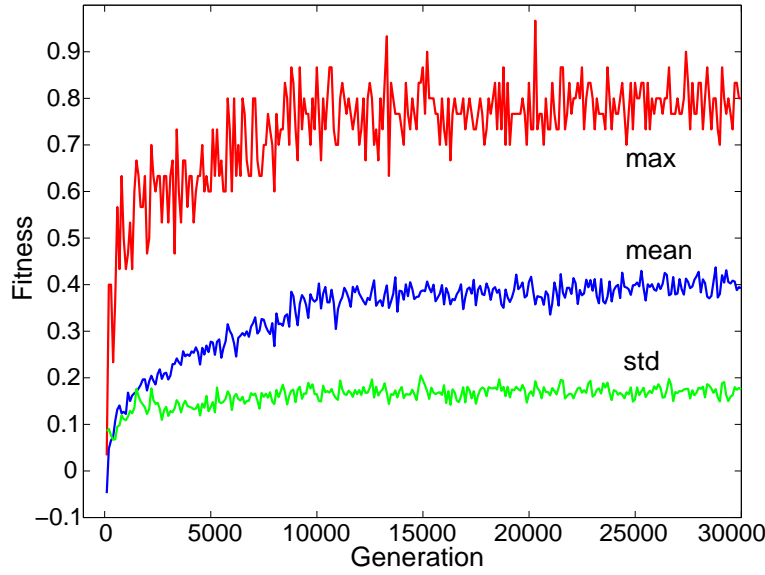


Figure 2: *A typical evolutionary run:* maximum, mean and standard deviation of fitness in a population plotted over 30000 generations of an evolutionary run. Values are plotted every 100 generations. Note that the fitness here is evaluated over a single epoch for each agent and the mean is the average of the fitness in the population (100 agents). Due to the big variety of possible environments the measured fitness is quite noisy. Throughout the paper, we assess the accurate fitness of an agent by averaging over 5000 epochs.

comparing them to benchmarks. For more details of the model and evolutionary dynamics see Appendix 1.

### 3 Agents’ Performance

In the framework described above ANN-controlled agents were evolved. We studied two types of agents: An SP-type agent possessing both a somatosensor and a position sensor, and an S-type agent possessing a somatosensor only (i.e. no positional cues available). These two types of agents were evolved in one of two possible environments. One in which the food zone borders were marked, and one in which they were not. These conditions define the four different models studied (see Figure 3).

In order to evaluate the performance of the evolved agents, we used two benchmarks. First, we compared their performance to the best *memoryless* algorithm designed by us to perform the same task. The basic idea behind all the designed algorithms we used was the same, and can be sketched as follows: “When reaching a resource – eat it if and only if it’s a food item. Unless you ‘believe’ you are inside the food zone, try to navigate into it, and

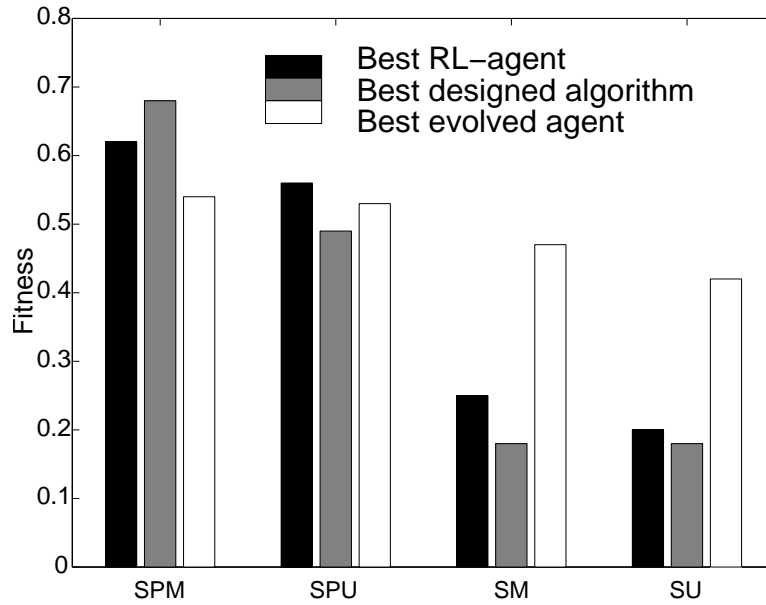


Figure 3: *Comparison between benchmark performance and that achieved by evolved agents on the four models studied.* S = somatosensor, SP = somato+position sensor. M = marked food zone, U = unmarked food zone. Fitness is averaged over 5000 epochs – the standard deviation is less than 0.5% of the fitness ( $< \pm 0.003$ ).

ignore all resources on your sides. If you 'believe' you are inside the food zone, switch to an efficient grazing mode, and try not to leave the food zone...". This basic idea translates into different instructions, according to the available sensory input. Appendix 3 describes one of these algorithms in more detail. As a second performance benchmark we solved the same tasks using Reinforcement Learning (RL) techniques, using an  $\epsilon$ -greedy SARSA algorithm [Sutton, 1996; Sutton and Barto, 1998] and training for up to ten million iterations.

Figure 3 summarizes the performance levels achieved by the above benchmarks, and those attained by the best evolved agents, for each of the four models studied. The fitness was averaged over 5000 epochs to achieve an accurate measure. (All fitness measures throughout the paper are averages of 5000 epochs, resulting in a vanishing standard deviation (less than 0.003)). As is evident, the evolved agents do well compared with the benchmarks in all possible scenarios. Since both benchmarks use no memory, a much higher score achieved by an agent indicated that it is employing some kind of memory. Indeed, as shall be seen below, the superior performance of evolved agents in models devoid of a position sensor is due to the development of memory under these scenarios.

## 4 Exploration vs. Grazing: The Emergence of Command Neurons

As a first step in the analysis, we investigated the emergent behaviour. We found that for both types of agents the most successful strategy relied upon a switch between two behavioural modes – *exploration* and *grazing*. The exploration mode consists of moving in straight lines, ignoring resources in the sensory field which are not directly under or in front of the agent, and turning at walls. When food zone borders are marked agents operating in exploration mode cross them. The grazing mode, on the other hand, consists of turning to resources to the right or left in order to examine them, turning both at walls and at food zone border markings, and maintaining the agent’s location on the grid in a relatively small, restricted region. In both modes when stepping on a food item eating occurs. Exploration mode is mostly observed when the agent is out of the food zone, allowing it to explore the environment and find the food zone. Inside the food zone, however, the agents almost always display grazing behaviour, which results in efficient consumption of food.

To characterize the agent’s mode in a quantitative manner, we defined behavioural measures (see Appendix 2). Using these measures and systematically clamping the activity of neurons in the network we identified a common structure in successful agents whereby the mode switch was always mediated by a central “*command neuron*”. The activity mode of this command neuron determines the agent’s behavioural mode. Clamping the activity of the command neuron, constantly maintaining an active or a quiescent state, reliably produces exploration or grazing modes respectively, regardless of the actual location of the agent in the environment.

As will be seen, although the functional role of the command neuron is nearly identical in all models studied, the computational basis underlying its activity is quite different. The dynamical properties of the command neuron depend on the type of sensors mounted on the agent and not on whether food zone markings exist. Hence, from this point on, we shall concentrate on the S vs. SP-type agents, without making the distinction between the marked (M) and unmarked (U) cases in each.



## 5 Location-driven Command Neurons: The SP model.

Examining the networks of successful agents equipped with a position sensor reveals an important common feature: Certain interneurons have a position-sensitive response. One of these neurons typically fires outside the food zone and remains quiescent inside the food zone and in its close vicinity. Figure 4 depicts the mean activity of such a neuron as a function of the agent’s location. The center sub-figure corresponds to the mean activity of the position-sensitive neuron in each grid cell, averaged over 5000 epochs. As is evident, its activity corresponds to the location of the food zone – active outside and inactive inside. Clamping this “*position-sensitive cell*” and measuring the behavioural mode of the agent, reveals that it acts as a command neuron. When this command neuron is active the agent assumes an exploration behaviour, *regardless* of where it is in the environment, whereas clamping it to a silent state results in grazing behaviour. Thus, the place-dependent activity map of the command neuron serves as an accurate predictor of the agent’s behaviour; where the map values are low the agent will graze, whereas where they are high it will explore.

Such place-driven command neurons were found consistently in agents evolved with somato and position sensors. The computational basis for these neurons’ activity is the absolute position of the agent in the environment, and is not influenced by the existence or location of resources in the arena. Shifting the food zone to a new location (adjacent to the middle of the northern wall), produces no change in the activity map of the command neuron, and thus no change in the behaviour of the agent relative to its location (see Figure 5 [1b] vs. [1a]). This obviously leads to a near-zero performance level since the commanded behavioural mode is no longer adequate. Given the direct sensory information about the location in the environment, the existence of a “position-sensitive cell” in these agents is not very surprising. It is the emerging “command neuron” function of these neurons which is interesting.

Further investigation of the activity of place-sensitive command neurons revealed that they are also *orientation selective*. The surrounding sub-figures in Figure 4 depict the mean activities when the agent was facing a given orientation. As is evident, when facing east or

north, the area where the command neuron is inactive (corresponding to grazing mode) is 'smeared' to the east or north respectively, compared to the opposite orientations. The grazing mode is thus maintained further out of the food zone when the agent is facing outward, increasing the chance to turn back and return to the food zone after accidentally leaving it, and in turn increasing these agents' fitness scores compared with agents controlled by a non-orientation-selective programmed algorithm (see the pertaining performance comparison in Figure 3).

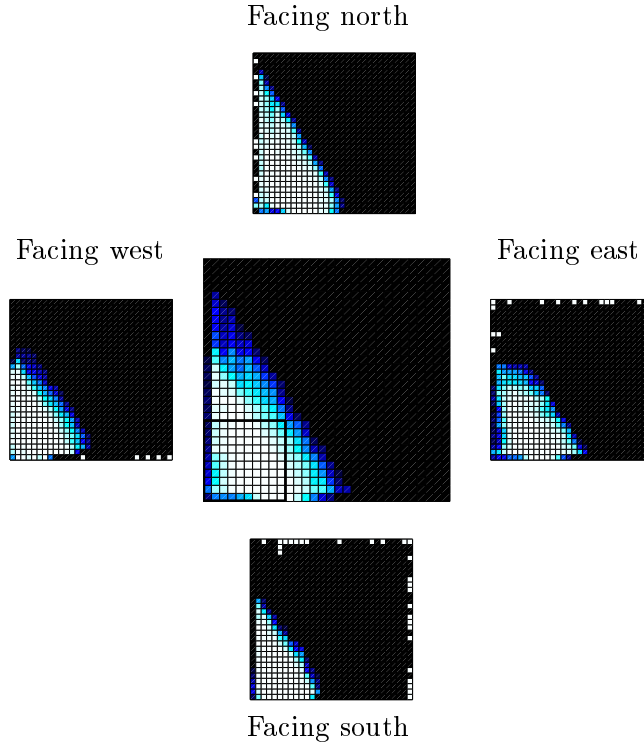


Figure 4: *Location- and Orientation-selectivity of command neuron activity in an agent with somato+position sensor.* The center sub-figure shows average command neuron activity over 5000 epochs. The peripheral sub-figures show average activity when the agent is facing a given orientation. Darker means higher average activity. White grid cells near walls correspond to locations the agent never visited with the given orientation. The thick line marks the border of the food zone (not seen by the agent in this scenario). The “smearing” of grazing activity towards north and east when facing these directions helps the agent to return to the food zone after accidentally leaving it.

## 6 Memory-driven Command Neurons: The S model.

### 6.1 Activity Maps of the Command Neuron

Due to the purely local sensory information and the lack of any positional cue in the S model, it is difficult to adopt a strategy that grazes efficiently inside the food zone and yet explores the environment quickly enough to reach the food zone within a reasonable time. The two atomic strategies – always moving in straight lines or always examining every resource in the sensory field – both yield near-zero performance.

Similar to the previous SP case we have identified in every network controlling a successful agent at least one neuron whose activity was place-dependent. Figure 5 [2a] depicts the activity map of such a neuron, demonstrating that it is quiescent when the agent is inside the food zone and increases its activity the further the agent is from the food zone. Again, this neuron takes the role of a command neuron, i.e. when it is active the agent manifests exploratory behaviour, whereas when it is quiescent the agent switches to grazing mode. In contrast to the SP-agents, shifting the food zone from its original location to a new location (Figure 5[2b]) causes no malfunction, as the selective activity pattern shifts accordingly.

Since the agents in the S-model received no explicit sensory cues regarding their location, the intriguing question is what constitutes the computational basis for the place-dependent activity of the command neuron. The observed behaviour led us to hypothesize that the successful agents utilize short-term memory, maintaining grazing mode for several time-steps after eating. This hypothesis was supported by the performance level of these agents, which actually surpassed both kinds of memory-less benchmarks for the same task (see Figure 3).

### 6.2 Command Neuron Activity Profiles

We now turn to study the dynamics of the S-type memory neuron in detail. To this end, we traced the average activity of the memory command neuron as a function of the elapsed time since the last eating episode (Figure 7, solid line, with 250 poison items). As is evident, the activity of the command neuron undergoes sharp inhibition immediately after

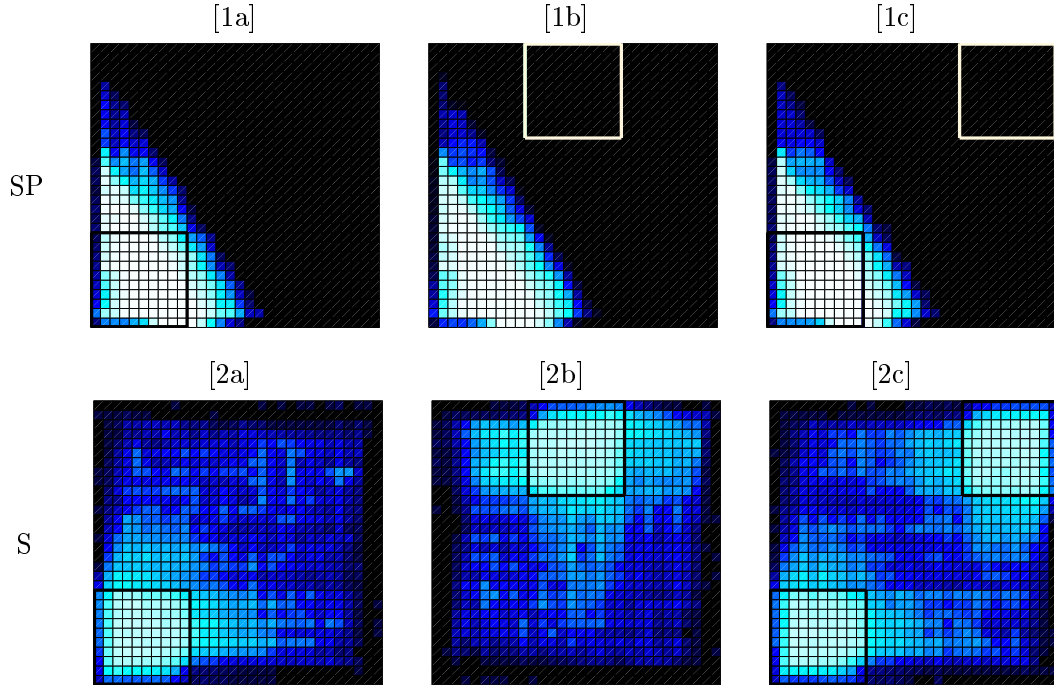


Figure 5: *Location-dependent activity maps of the command neurons.* Darker means higher average activity. Average is taken over 2000 epochs. **Somato+Pos Sensor (SP model):** [1a] Baseline condition, [1b] Shifted food zone [1c] Two food zones. **Somatosensor (S model):** [2a] Baseline condition, [2b] Shifted food zone [2c] Two food zones. In all cases agent populations were *evolved* in environments with one food zone at the south-western corner (baseline condition). Thick lines depict the food zone borders (the agents presented here were evolved in worlds with no food zone border markings, which were added here for clarity of exposition only).

eating, its probability of firing gradually increasing thereafter. Thus, the command neuron “remembers” the number of steps elapsed since the agent has last consumed a food item; its activity reflects *a stochastic short-term memory mechanism*.

Figure 6 presents a raster plot of the activity of a stochastic memory command neuron, with the small triangles indicating the times of feeding events. The baseline firing state of the neuron is fully active (i.e., commanding exploration mode)<sup>1</sup>, and it is inhibited following feeding. The post-eating quiescence periods vary in duration. Thus, the emerging dynamics of the neuron is sustained activity, interrupted by periods of inactivity triggered by the feeding events. This behaviourally translates into sustained exploration interrupted by grazing periods of varying durations which are triggered by feeding. From the agent’s

---

<sup>1</sup>At the beginning of each epoch the network is reset to an inactive state, resulting in the few steps of quiescence initiating each epoch.

perspective this is likely to constitute an optimal strategy: As long as the agent frequently encounters food it “knows” it is in a food zone and remains in grazing mode. However, if food is not encountered for a prolonged period, the agent switches to exploration mode to search back for the food zone. The robustness of this strategy can be demonstrated by distributing the food in two designated zones in the arena. Figure 5[2c] shows the resulting activity map, with the two zones clearly evident. This map correctly predicts the adequate behaviour of the agent. Indeed, S-type agents that evolved a memory mechanism in environments with one food zone but were evaluated in arenas with two zones, behaved adequately: Exploring the environment they reached one food zone, grazed there and after a while switched to exploration (since the food density decreased), reached the other food zone, and again switched to grazing mode. This is in sharp contrast to the case of the SP-type agents which rely on absolute location, whose behaviour was completely inadequate, grazing only in the south-west corner (Figure 5[1c]).

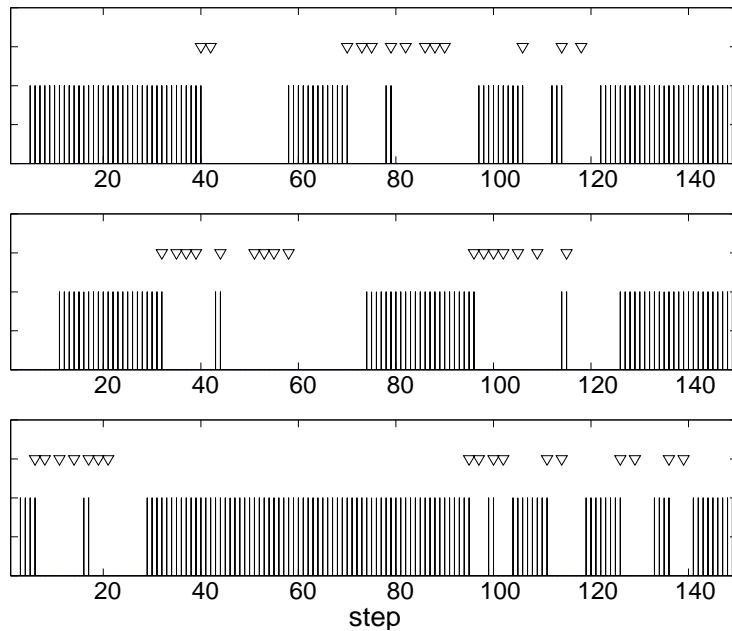


Figure 6: *Activity of a stochastic memory command neuron:* A raster plot of 3 different epochs. Vertical bars mark steps where the neuron fired. The small triangles above them mark feeding events.

The stochastic nature of such a memory mechanism which emerges in a binary neural network with deterministic dynamics can be accounted for by the random distribution of

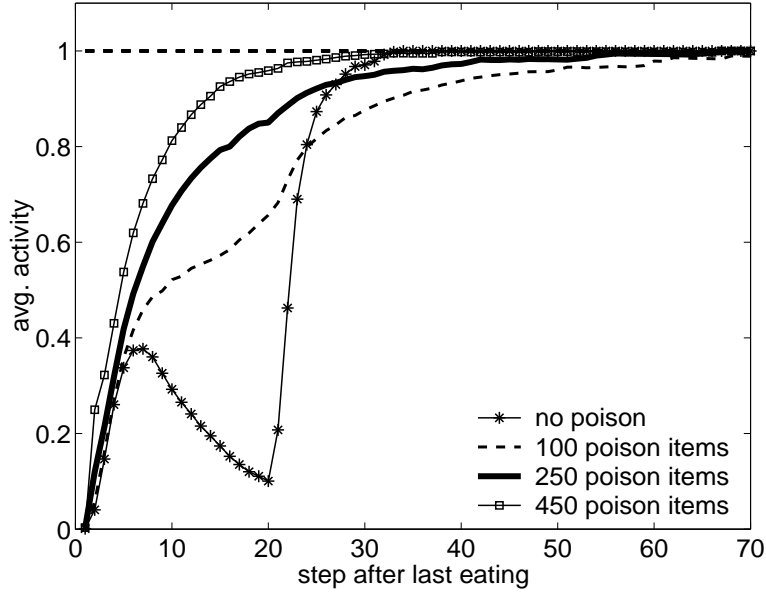


Figure 7: *Activity profiles of a stochastic memory command neuron:* The average activity level is portrayed as a function of the time elapsed since last eating episode. The agent was *evolved* with 250 poison items in the environment (solid line), but its activity is measured under various poison concentrations over 1500 distinct epochs.

resources in the arena, as well as the random component of input from the somatosensor. Indeed, we found that the activation of the memory neuron strongly depends on the poison distribution in the arena (Figure 7). For poison distributions deviating from the one the agents were evolved with, the activation as a function of elapsed time plot is perturbed. For higher poison concentration this merely changes the memory-span, whereas for lower concentrations it actually distorts the shape of the plot, eventually making it non-monotonous, and thus a poor correlate of the elapsed time. In the absent poison case this results in a 10% decrease in the performance of the agents. This is an interesting example of the way in which evolution harnesses harmful features of the environment: A high concentration of poison in the environment would seem like a burdening feature, the elimination of which should increase performance. However, the stochastic memory mechanism takes advantage of exactly this feature. As will be shown below, it finely tunes itself to near-optimal performance for the given “ecological niche” in which the agents evolved.

### 6.3 The Underlying Stochastic Memory

Examining the input field (post synaptic potential) of the stochastic memory neuron *excluding the input from itself* revealed that its firing threshold is around one standard deviation above the field’s mean. Given that the input field is noisy, a transition of the memory neuron from a quiescent state to firing will occur spontaneously. The synapse from the memory neuron to itself is strong enough to keep it above threshold once active, whereas an eating event induces a strong inhibition which shunts its activity. This corresponds well to the observed activity profiles described above.

To investigate the spontaneous return to an active state, we plotted the distribution of the post-eating quiescent period durations (see Figure 8[a]). After a refractory period that lasts one time-step, the length of the quiescence period (behaviourally commanding a grazing mode) can be roughly approximated to a first order by a continuous exponential distribution function. Thus, a first approximation for the underlying dynamics of the memory command neuron is a one-parameter geometric distribution model specifying the probability to resume firing at any quiescent step. This approximation assumes that the probability to resume firing is constant and does not depend on the sensory input in the step. However, further inspection reveals that the probability to resume firing does depend on the sensory input. Specifically, it is much higher near walls than in other places in the arena.

Therefore, to better assess the adaptation of the emerging memory mechanism to the behavioural task, we used a two-parameter stochastic model. According to this model, the normal state of the memory neuron is active, and it is shunted right after eating with probability one (whereas the probability to switch from activity to inactivity otherwise is zero). The probability to switch back from inactive to active state depends on two parameters: it is  $p$  if the agent senses a wall, and  $q$  otherwise <sup>2</sup>. This two parameter model was then used to determine the firing state of the command neuron at every step. Note that this is not an algorithmic benchmark. The model is “mounted” on an evolved S-type agent,

---

<sup>2</sup>It is straightforward to envisage a neural mechanism realizing such a two-parameter model, utilizing two different field distributions.

whose command neuron’s activity is determined by the values predicted by the model, while the rest of the network’s activity is updated naturally.

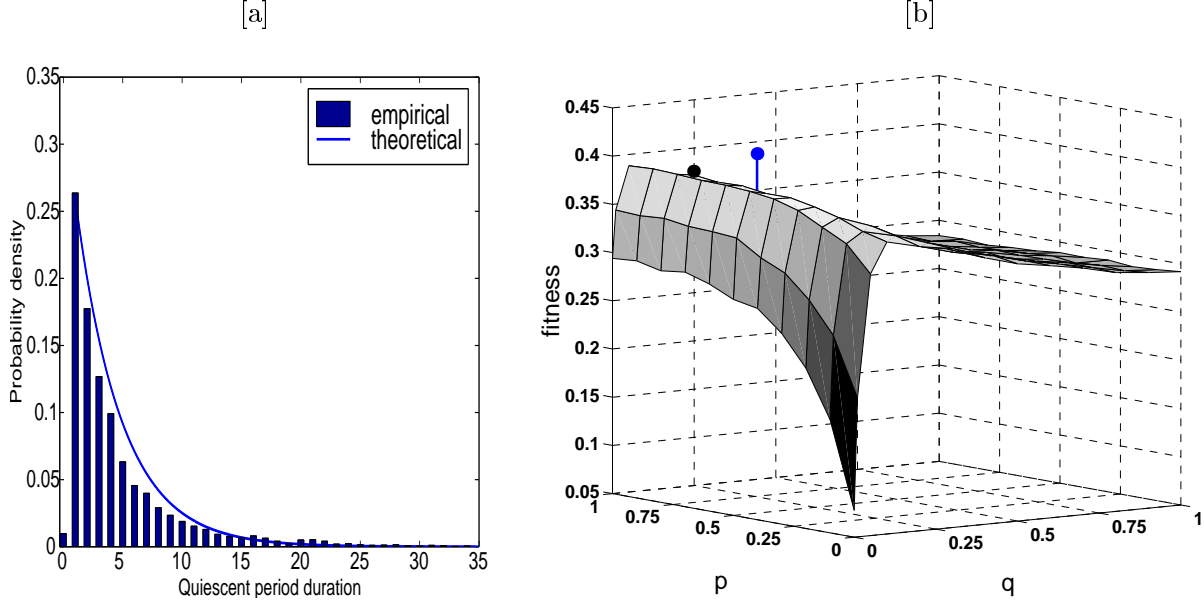


Figure 8: *Stochastic Behaviour of the Command Neuron* [a] The distribution of the command neuron’s quiescent period durations. The solid line corresponds to an exponential distribution with  $\lambda = 0.26$  (shifted by one time-step for the refractory period), corresponding to a mean memory maintenance of 4.85 time-steps compared with 4.74 observed experimentally. [b] A two-parameter stochastic model of the activity of the stochastic memory command neuron in the basic S model. The surface depicts the fitness attained for different values of  $p$  and  $q$ . The best fitness obtained using the model was 0.39 ( $p = 0.8$  and  $q = 0.1$  (lower circle)). The best evolved agent scored 0.42, with  $p = 0.51$  and  $q = 0.08$  (higher circle, the line pointing to the fitness obtained using the same parameters in the “mounted” model). The diagonal  $p = q$  corresponds to the first-order approximation, stating that the probability to resume firing is a constant.

The fitness values obtained by running agents whose command neuron’s activity was driven by this two-parameter stochastic model with  $p, q$  values ranging between 0 and 1 are shown in Figure 8[b]. The actual parameter values obtained by the evolutionary process lie on the high ridge of the fitness surface. The best evolved agents however still obtained a higher fitness (line connected circle, 0.42) than that of the two-parameter mounted model agents (unconnected circle, 0.39). This can be explained by the fact that the two-parameter model captures the essential dynamics of the memory-driven command neuron, yet neglects certain variations in the probability to switch between firing states, which enhance the



performance.

## 7 Neural Network Structure of An S-type Agent

Although successful neuro-controllers of different agents share the command neuron mechanism, they differ in their structure and in some aspects of their function. To demonstrate the evolving network’s structures we focus here on one of the evolved S-type agents which has a remarkably simple yet very successful network. Clamping the command neuron to its average activity has a marked effect on this agent’s behaviour (Figure 9), while clamping all the other interneurons to their average activity values results in no significant decrease in the agent’s fitness. The interneurons in this particular agent merely set the bias of other neurons, and the network can thus be reduced to the one depicted in Figure 10 (top).

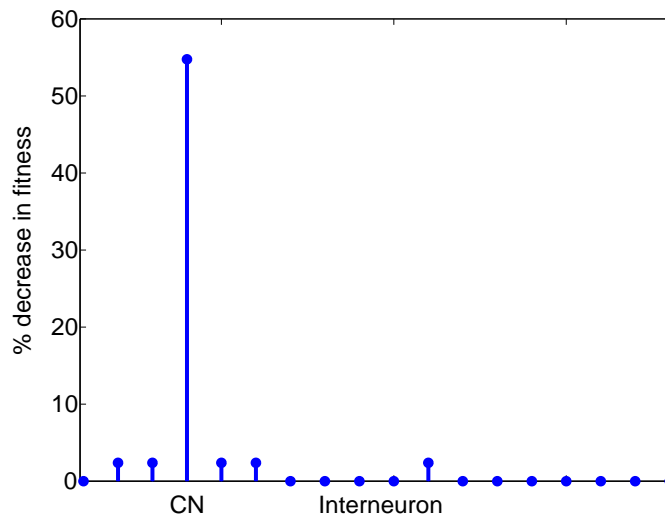


Figure 9: *Fitness decrease after selective clamping of interneurons*: Each interneuron was clamped in turn to its average activity value, and the fitness of the agent measured. The decrease in the fitness is plotted for each interneuron. Only one interneuron, the command neuron (CN), significantly effects the fitness when clamped alone. Fitness was averaged over 5000 epochs, standard deviation is less than 0.5% of the fitness.

The firing state of the command neuron, whose dynamics has been explained earlier, triggers a switch between two distinct input-output networks controlling exploration vs. grazing (Figure 10 (bottom)). These two basic sub-networks reside in the same network. They are modulated by the memory-based command neuron, which when active adds its set of weights to the network’s connectivity matrix. It should be noted that the ability to

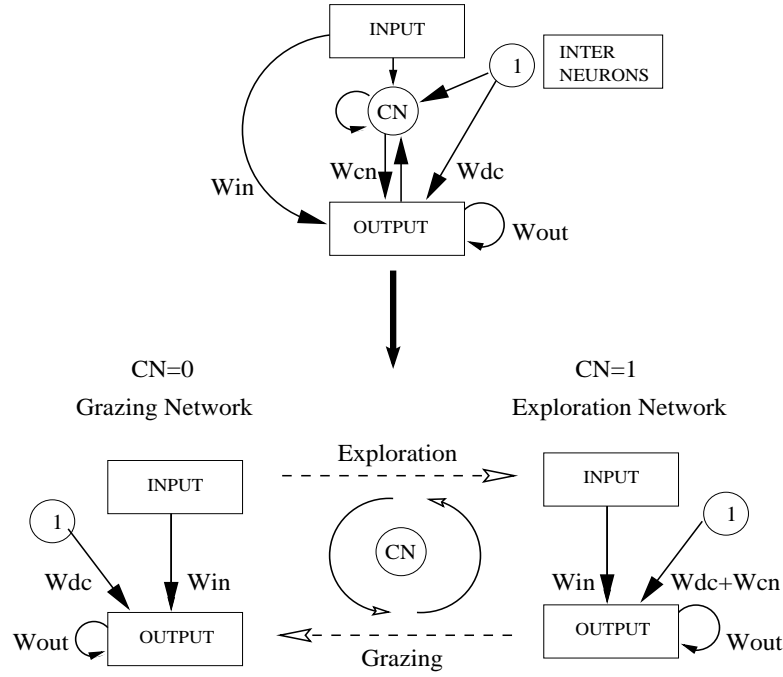


Figure 10: *An S-type neuro-controller*:  $W_{cn}$  is the weight vector from the command neuron (CN) to the output neurons.  $W_{in}$  is the weight matrix from the input to the output neurons.  $W_{out}$  are the recurrent connections within the output layer. The rest of the neurons can be reduced to a single neuron serving as a bias input to the command neuron and the neurons of the output layer. When in Grazing mode, the activity of the CN is 0, and the resulting network is the one depicted on the left. In exploration mode its activity is 1, and the equivalent network is the one on the right.

discriminate between food and poison remains intact with any manipulation of interneurons' activities. This basic ability relies on direct connections between the input and output layers, and indeed evolves at the first stages of the evolutionary process.

## 8 Discussion

This paper presents a novel attempt to perform an in-depth analysis of structure-to-function relations in evolved neuro-controllers for autonomous agents. To this end, we analyzed the control mechanisms evolving in autonomous agents performing a simple foraging task and governed by a recurrent ANN, without any predefined network architecture. In order to succeed in their behavioural task, the agents developed a mechanism for switching between two distinct types of behaviours – grazing and exploration. In all four experimental scenarios examined, the evolved agents managed to closely match the best memoryless algorithms for

the task, and in cases characterized by limited sensory input surpassed them by far. This was achieved with *completely unconstrained* network architectures.

We discussed in detail two types of evolved agents, differing in the sensory input available to them. In both cases, a similar mechanism has evolved, whereby a “*command neuron*” modulates the dynamics of the whole network and switches between grazing and exploration behaviours. In the case where the agents had no sensory position information a memory mechanism emerged, which then became the basis for the place-sensitivity of the command neuron. Using a two-parameter stochastic model for the memory mechanism we demonstrated that evolution fine-tuned this mechanism towards near-optimal parameters, using the inherent environmental noise and taking advantage of features of the environment that are a-priori harmful, such as the distribution of poison in the arena. An analysis of the neuro-controller of a simple S-type agent demonstrates that the command neuron essentially switches the dynamics between two basic input-output networks residing within the same network. Other networks controlling successful agents may be far more complex. It is the subject of further studies to fully understand all aspects of their structure and function, and will demand the utilization of more advanced analytical methods.

Several studies have previously dealt with the analysis of evolved neurocontrollers. [Cangelosi, Parisi and Nolfi, 1994] describe agents coping with a foraging task that demands navigation into areas with food and water respectively according to an externally-governed motivational state. They demonstrate the emergence of two distinct neural pathways for dealing with the two states. The architecture of the neurocontrollers they developed is constrained to be purely feed-forward, and their primary analysis method is receptive-field analysis. In a series of works, Beer et al. use dynamical systems approach to analyze continuous-time recurrent neurocontrollers. [Beer, 1997; Chiel, Beer and Gallagher, 1999; Beer, Chiel and Gallagher, 1999] deal with the evolution of small Pattern Generators (PGs) consisting of 3-5 neurons with recurrent connectivity. Similar to our analysis, the sensory input available during evolution is shown to affect the type of the evolving control mechanism. [Gallagher and Beer, 1999] evolves a 22-neuron symmetric feed-forward architecture for coping with a visually-guided walking task, and used selective lesioning to identify the

role of single neurons in the resulting controllers. The work presented in the current paper follows similar lines, extending previous work by using unconstrained architectures in the context of a non-trivial task requiring more than a purely reactive behaviour. The resulting structures are relevant to the understanding of an important biological neural archetype - that of a *Command Neuron*.

The idea of Command Neurons, i.e. single neurons whose activity commands a high level behavioural pattern, was first suggested some fifty years ago. Since then their existence was verified in a number of animal models, including crayfish [Edwards, Heitler and Krasne, 1999], Aplysia [Xin, Weiss and Kupfermann, 1996a; Xin, Weiss and Kupfermann, 1996b; Gamkrelidze, Laurienti and Blankenship, 1995; Nagahama, Weiss and Kupfermann, 1994; Teyke, Weiss and Kupfermann, 1990], Clione [Panchin, Arshavsky, Deliagina, Orlovsky, Popova and Selverston, 1996], crabs [Norris, Coleman and Nusbaum, 1994; DiCaprio, 1990] and lobsters [Combes, Meyrand and Simmers, 1999]. Command neuron activity has been shown to control a variety of motor repertoires. It has been demonstrated that in some cases behaviour is modulated by the command neurons on the basis of certain sensory stimuli [Xin, Weiss and Kupfermann, 1996b], and in particular by food arousal [Nagahama, Weiss and Kupfermann, 1994; Teyke, Weiss and Kupfermann, 1990]. Moreover, command neurons have been shown to induce different activity patterns in the same neural structures by modulating the activity of other neurons in a pattern-generating network [Combes, Meyrand and Simmers, 1999; DiCaprio, 1990]. One of the mechanisms by which command neurons act on other neurons is by modulating their excitability [DiCaprio, 1990], similar to the mechanism emerging in our simulations. The resemblance between these findings and the emerging properties of networks in the simulations we described is noteworthy. Nevertheless, biological reality is much richer. For example, chemical neuro-modulation plays an important role in the command neuron activity [Brisson and Simmers, 1998; Panchin, Arshavsky, Deliagina, Orlovsky, Popova and Selverston, 1996], while totally absent from our model. Another example is the finding of *Command Systems* comprising several command neurons that act in coordination to control a varied behavioural repertoire [Combes, Meyrand and Simmers, 1999; Edwards, Heitler and Krasne, 1999; Gamkrelidze, Laurienti

and Blankenship, 1995].

The current model, although simple, gives a computational insight into command neuron mechanisms. It shows for the first time a concrete model in which a single command neuron can switch the dynamics of a small neural network between two markedly different behavioural modes. This is achieved by dynamically setting the biases of other neurons in the network, thus effectively multiplexing two networks within the same set of neurons. We demonstrate that this structure does not have to be hand-crafted - it evolves spontaneously in a variety of scenarios, proving to be a robust computational mechanism. The characterization of the set of tasks that can be solved by such multiplexed networks still awaits further study. However, it is interesting to know that such non-trivial tasks indeed exist, that multiplexed networks solving them can be evolved and moreover that they can be successfully identified.

The results presented in this paper were obtained using the crudest form of genetic encoding – direct specification of all the synaptic weights in the genome. This bears a limiting effect on the scalability and speed of the evolutionary process. With the application of more sophisticated genetic encoding schemes, such as grammatical or ontogenic encodings [Kitano, 1990; Cangelosi, Parisi and Nolfi, 1994], and of more efficient selection procedures such as incremental evolution [Gomez and Miikkulainen, 1997], one may expect the evolution of larger recurrent EANNs, processing more complex sensory input to achieve more intelligent behaviours. Adding learning as an additional adaptive force can enhance the efficiency of the evolutionary process (see [Hinton and Nowlan, 1987; Ackley and Littman, 1991; Nolfi, Elman and Parisi, 1994]) and may be needed for the structural fine-tuning of large evolved networks.

The similarity to known neural mechanisms that was achieved even under the current basic and almost toy-like techniques leads us to believe that once better scalability is achieved, EANNs may provide an excellent vehicle to study the fundamental problem of structure and function relation in nervous systems. The accessibility of such EANN models to thorough analysis should make them important means of investigation in the tool-chest of computational neuroscientists.

**Acknowledgments:** We acknowledge the valuable contributions and suggestions made by Henri Atlan, Hanoach Gutfreund, Isaac Meilijson and Oran Singer. Lior Segev helped us in benchmarking the evolved agents. This research has been supported by the FIRST grant of the Israeli Academy of Science.

## References

- Ackley, D. and Littman, M. (1991). Interactions Between Learning and Evolution. In C. Langton, C. Taylor, J. Farmer and S. Rasmussen (Eds.), *Artificial Life II*, Volume 10 of *SFI Studies in the Sciences of Complexity*, pp. 487–509. Addison-Wesley.
- Beer, R. D. (1997). The dynamics of adaptive Behavior: A research program. *Robotics and Autonomous Systems*, **20**, 257–289.
- Beer, R. D., Chiel, H. J. and Gallagher, J. C. (1999). Evolution and Analysis of Model CPGs for Walking: II. General Principles and Individual Variability. *Journal of Computational Neuroscience*, **7**, 119–147.
- Brisson, M. T. and Simmers, J. (1998, March). Neuromodulatory inputs maintain expression of a lobster motor pattern generating network in a modulation-dependent state: evidence from long-term decentralization in vitro. *Journal of Neuroscience*, **18**(6), 2212–2225.
- Cangelosi, A., Parisi, D. and Nolfi, S. (1994). Cell division and migration in a 'genotype' for neural networks. *Network*, (5), 497–515.
- Chiel, H. J., Beer, R. D. and Gallagher, J. C. (1999). Evolution and Analysis of Model CPGs for Walking: I. Dynamical Modules. *Journal of Computational Neuroscience*, **7**, 99–118.
- Combes, D., Meyrand, P. and Simmers, J. (1999, May). Motor pattern specification by dual descending pathways to a lobster rhythm-generating network. *Journal of Neuroscience*, **19**(9), 3610–3619.

- DiCaprio, R. A. (1990). An interneurone mediating motor programme switching in the ventilatory system of the crab. *Journal of Experimental Biology*, **154**, 517–535.
- Edwards, D. H., Heitler, W. J. and Krasne, F. B. (1999, April). Fifty years of command neuron: the neurobiology of escape behavior in the crayfish. *Trends in Neurosciences*, **22**(4), 153–161.
- Floreano, D. and Mondada, F. (1996, June). Evolution of Homing Navigation in a real Mobile Robot. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, **26**(3), 396–407.
- Fogel, D. B. (1995). *Evolutionary Computation - Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press.
- Gallagher, J. C. and Beer, R. D. (1999). Evolution and analysis of dynamical neural networks for agents integrating vision, locomotion, and short-term memory. In W. B. et al (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pp. 1273–1280.
- Gamkrelidze, G., Laurienti, P. and Blankenship, J. (1995, October). Identification and characterization of cerebral ganglion neurons that induce swimming and modulate swim-related pedal ganglion neurons in *Aplysia brasiliana*. *Journal of Neurophysiology*, **74**(4), 1444–1462.
- Gomez, F. and Miikkulainen, R. (1997). Incremental Evolution of Complex General Behavior. *Adaptive Behavior*, **5**(3/4), 317–342.
- Harrauld, P. G. and Kamstra, M. (1997, April). Evolving Artificial Neural Networks to Combine Financial Forecasts. *IEEE Transactions on Evolutionary Computation*, **1**(1), 40–52.
- Hinton, G. E. and Nowlan, S. J. (1987). How Learning Can Guide Evolution. *Complex Systems*, **1**(1), 495–502.
- Jakobi, N. (1998). Evolutionary Robotics and the Radical Envelope-of-Noise Hypothesis. *Adaptive Behavior*, **6**(2), 325–368.

- Kitano, H. (1990). Designing Neural Networks Using Genetic Algorithms with Graph Generation System. *Complex Systems*, (4), 461–76.
- Kodjabachian, J. and Meyer, J.-A. (1998, September). Evolution and Development of Neural Controllers for Locomotion, Gradient-Following and Obstacle-Avoidance in Artificial Insects. *IEEE Transactions on Neural Networks*, **9**(5), 796–812.
- Miglino, O., Nolfi, S. and Parisi, D. (1996). Discontinuity in evolution: how different levels of organization imply pre-adaptation. In R. K. Belew and M. Mitchell (Eds.), *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison-Wesley.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: The MIT Press.
- Nagahama, T., Weiss, K. R. and Kupfermann, I. (1994, July). Body postural muscles active during food arousal in Aplysia are modulated by diverse neurons that receive monosynaptic excitation from the neuron CPR. *Journal of Neurophysiology*, **72**(1), 314–25.
- Nolfi, S., Elman, J. L. and Parisi, D. (1994). Learning and Evolution in Neural Networks. *Adaptive Behavior*, **3**(1), 5–28.
- Norris, B., Coleman, M. and Nusbaum, M. (1994, October). Recruitment of a projection neuron determines gastric mill motor pattern selection in the stomatogastric nervous system of the crab, *Cancer borealis*. *Journal of Neurophysiology*, **72**(4), 1451–1463.
- Panchin, Y., Arshavsky, Y., Deliagina, T., Orlovsky, G., Popova, L. and Selverston, A. (1996, May). Control of locomotion in the marine mollusc *Clione limacina*. XI. Effects of serotonin. *Experimental Brain Research*, **109**(2), 361–365.
- Scheier, C., Pfeifer, R. and Kuniyoshi, Y. (1998, October-November). Embedded Neural Networks: Exploiting Constraints. *Neural Networks*, (7-8), 1551–1569.
- Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In M. M. D.S. Touretzky and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, pp.



1038–1044. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press.

Teyke, T., Weiss, K. R. and Kupfermann, I. (1990, January). An Identified Neuron (CPR) Evokes Neuronal Responses Reflecting Food Arousal in Aplysia. *Science*, (247), 85–87.

Xin, Y., Weiss, K. and Kupfermann, I. (1996a, August). An identified interneuron contributes to aspects of six different behaviors in Aplysia. *Journal of Neuroscience*, **16**(16), 5266–5279.

Xin, Y., Weiss, K. and Kupfermann, I. (1996b, July). A pair of identified interneurons in Aplysia that are involved in multiple behaviors are necessary and sufficient for the arterial-shortening component of a local withdrawal reflex. *Journal of Neuroscience*, **16**(14), 4518–4528.

# Appendix 1: Detailed Model Description

## The Simulation System

A flexible simulation system was used to build a variety of evolutionary models incorporating autonomous agents acting in a life-like environment. The core system defines the basic notions of an agent, a simulation world in which several agents can coexist, and the simulation universe (in which several simulation worlds can coexist). It is implemented in C++ and runs on the Linux operating system. The specialized implementation in C++ allows for intensive optimization, resulting in running times of about 4 hours for 10,000 generations of 100 agents performing the task described above. Around the core system particular models are built. Each model defines the specific properties of simulation worlds and agents. At the level of the simulation world, this includes the geometry, appearance, and availability of different resources. At the level of the agent, it includes its sensory capabilities, its motor capabilities and the nature of the control mechanism mediating between the sensory input and motor output. Currently, the evolutionary process only affects the neural network control mechanism. A particular model also defines the specific genetic methods used in the simulation - whether it uses sexual or a-sexual breeding, the variational operators and the selection methods.

## The Environment and the Behavioural Task

The agents all operate in a grid arena of size 30x30 with two kinds of resources. One resource, defined as “poison” (i.e. causing a negative reward), is randomly scattered all over the arena. A second, “food” resource bringing positive reward is randomly scattered in a restricted food zone location in the environment. In most experiments, the food zone was of size 10x10 and was located at the south-west corner of the arena. In some of the models studied the boundaries of the food zone were marked, enabling the agents to sense them, while in other models this marker was absent. A life cycle of an agent (an *epoch*) lasts 150 time steps, in which one motor action takes place. At the beginning of an epoch the agent is introduced to the environment at a random location and orientation.

In each generation a population of 100 agents is evaluated. Each agent is evaluated in

it’s own environment, which is earlier initialized with 250 poison items and 30 food items. At the end of its life-cycle each agent receives a fitness score calculated as the total amount of food it has consumed minus the total amount of poison it has eaten divided by 30, the number of distributed food items. Thus the fitness ranges between -2.5 (eating the maximal number of poison items possible within the 150 steps of an epoch) and 1 (consuming all the distributed food items).

## The Controlling Network: Structure and Dynamics

Each agent is controlled by a neural network consisting of 15 to 50 neurons (the number was fixed within a given simulation run). Out of these,  $K_{in}$  neurons (5 or 7) are dedicated sensory neurons, whose values are clamped to the sensory input. Four neurons are designated as *output neurons* commanding the motor system. The network is composed of binary McCulloch-Pitts neurons which are fully connected, with the exception of the non-binary sensory neurons which have no input from other neurons. Network updating is synchronous. In every step a sensory reading occurs, network activity is then updated, and a motor action is taken according to the resulting activity in the designated output neurons.

## The Sensory System

Each agent is equipped with a basic sensor we termed *somatosensor*, consisting of five probes, to each of which a sensory neuron is associated. Four probes sense the grid cell the agent is located in and the three grid cells ahead of it (see Figure 1). These probes can sense the difference between an empty cell, a cell containing a resource (either poison or food – with no distinction between those two cases), an arena boundary and food zone boundary. The fifth probe can be thought of as a *smell probe*,<sup>3</sup> returning -1 or +1 if the agent is currently in a grid cell where there is poison or food respectively, and -1 or +1 randomly otherwise. Thus, the agent has to integrate the input from two sensors in order to identify the presence of food or poison. In addition, in some models the agents were equipped with a *position sensor*, consisting of two sensors, giving the agent’s absolute coordinates in the arena, where the origin is taken as the south-western corner. The agent has no information

---

<sup>3</sup>The term *somatosensor* is inaccurate due to the smell sensor, but we shall use it for brevity of notation.

about its orientation.

## The Motor System

The motor system of an agent consists of four motors, receiving binary commands from the four *output neurons*. The first motor induces forward movement when activated. Two other motors control right and left turns, inducing a 90 degrees turn in the respective direction when only one of them is activated, and maintaining the current orientation otherwise. The fourth motor controls eating, consuming whatever resource is available in the current location when activated. For eating to actually take place, however, there has to be no other movement (forward step and/or turn) in the same time step. This both enforces simple motor integration, and makes any attempt to eat a costly procedure.

## Evolutionary Dynamics

Each agent is equipped with a chromosome defining the structure of its N-neurons controlling EANN, consisting of  $N(N - K_{in})$  real numbers specifying the synaptic weights. At the end of a generation a phase of sexual reproduction takes place, in which chromosomes are crossed over and then mutated to obtain the agents of the next generation. There are 50 reproduction events each generation. In each of them two agents from the parents population are randomly selected with probability proportional to their fitness.

We used uniform point-crossover with probability 0.35, after which point mutations were randomly applied to two percent of the locations in the genome. These mutations changed the pertaining synaptic weights by a random value between -0.6 and +0.6. Simulations lasted a pre-defined number of generations, ranging between 10000 and 30000. In the last generation, every agent was evaluated during 5000 epochs, on a variety of initial conditions, to accurately measure its fitness.

## Appendix 2: Behavioural Indices of Exploration and Grazing

To quantify the behaviour of the agents we used the following indices:

1. *Time it takes the agent to reach the food zone* for the first time in an epoch. This index should be short for exploration mode and long for grazing mode.
2. *Time it takes the agent to return to the food zone* once it leaves it. This should be short for grazing mode and long for the exploration mode.
3. *Percentage of turns to resources*: Out of all steps in which there is a resource located on either side of the sensory field, we measure the percentage of steps in which the agent turned to inspect the resource. This should be high for grazing mode, and low for exploration mode.
4. *Percentage of turning steps out of all steps in the epoch*. Exploration mode consists of moving in straight lines, and thus this measure is low for exploration mode, and high for grazing mode.
5. *Extent of arena coverage*. In exploration mode, the main goal is to cover distances, but not to waste time on densely covering the explored areas. In grazing mode, on the other hand, the searched area should be small but it should be thoroughly covered. In order to measure this, we calculated two numbers. First, the percentage of arena cells visited (out of the total number of cells). Second, the area of the minimal rectangle encapsulating all the cells visited, divided by the total area of the arena. The index is the ratio of these two numbers. A high value signals grazing mode (i.e., high coverage), whereas a low one testifies to exploration mode.

Table 1 depicts the values of these indices in two successful agents, one of the S-type and one of the SP-type. Note the effects of clamping the command neuron: When it is clamped to an active state, in both agents the indices of exploration are high, while clamping to a silent state induces grazing behaviour. Also note that in the control state (no clamping) the behavioural mode is usually adequate, e.g finding the food zone under control conditions takes the same time as under a clamped-to-exploration mode, etc.

	S-type				
	Time to reach food zone	Return time to food zone	% turn to resource	% turn in general	Arena coverage
Control	55.7( $\pm 0.85$ )	24.2( $\pm 0.56$ )	22.5 ( $\pm 0.11$ )	17.3( $\pm 0.06$ )	0.22( $\pm 0.003$ )
Command neuron clamped to active state (exploration)	53.5( $\pm 0.84$ )	93.7( $\pm 0.39$ )	17.1( $\pm 0.08$ )	16.1( $\pm 0.06$ )	0.16( $\pm 0.0002$ )
Command neuron clamped to silent state (grazing)	113.8( $\pm 1.32$ )	15.2( $\pm 0.3$ )	30.1( $\pm 0.18$ )	20.7( $\pm 0.12$ )	0.53( $\pm 0.005$ )
	SP-type				
	Time to reach food zone	Return time to food zone	% turn to resource	% turn in general	Arena coverage
Control	45.4( $\pm 0.68$ )	13.5( $\pm 0.38$ )	23( $\pm 0.19$ )	16.3( $\pm 0.14$ )	0.3( $\pm 0.005$ )
Command neuron clamped to active state (exploration)	44( $\pm 0.66$ )	53.5( $\pm 0.36$ )	4.9( $\pm 0.05$ )	5.7( $\pm 0.03$ )	0.21( $\pm 0.002$ )
Command neuron clamped to silent state (grazing)	105.2( $\pm 1.37$ )	12.5( $\pm 0.3$ )	37.7( $\pm 0.3$ )	25.6( $\pm 0.28$ )	0.63( $\pm 0.009$ )

Table 1: *Behavioural mode indices*: For each of the agents we measured the five indices under three conditions: control (i.e. normal conditions with no clamping), the command neuron clamped to a constant active state, and to a constant silent state. Results are averages over 2000 epochs. The standard deviation of these averages is given in parenthesis.

## Appendix 3: Benchmark Algorithm for the SP-model

The following is the algorithm in pseudo-code.

```
if (stepping on food) eat
else
  if (in front of wall)
    if (x > y and ~infoodzone) turn right
    else turn left
  else if (in front of foodzone marking)
    if (infoodzone) turn left
    else move fwd
  else if (sense resource in front-left grid)
    if (infoodzone) move fwd and turn left
    else move fwd
  else if (sense resource in front-right grid)
    if (infoodzone) move fwd and turn right
    else move fwd
  else
    move fwd
```

Using the input from the position sensor the algorithm determines if the agent is in the food zone and takes the appropriate motor action. Loosening the “infoodzone” condition, i.e. acting in a grazing mode when the agent is within a certain range around the food zone, does not improve the performance of the algorithm.