

Solving a Delayed Response Task with Spiking and McCulloch-Pitts Agents

Keren Saggie¹, Alon Keinan¹ and Eytan Ruppin^{1,2}

¹ School of Computer Sciences,
Tel-Aviv University, Tel-Aviv 69978, Israel,
{keren,keinan}@cns.tau.ac.il, ruppin@post.tau.ac.il

² School of Medicine,
Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract. This paper investigates the evolution of evolved autonomous agents that solve a memory-dependent delayed response task. Two types of neurocontrollers are evolved: networks of McCulloch-Pitts neurons, and spiky networks, evolving also the parameterization of the spiking dynamics. We show how the ability of a spiky neuron to accumulate voltage is utilized for the delayed response processing. We further confront new questions about the nature of “spikiness”, showing that the presence of spiking dynamics does not necessarily transcribe to actual spikiness in the network, and identify two distinct properties of spiking dynamics in embedded agents. Our main result is that in tasks possessing memory-dependent dynamics, neurocontrollers with spiking neurons can be less complex and easier to evolve than neurocontrollers employing McCulloch-Pitts neurons. Additionally the combined utilization of spiking dynamics with incremental evolution can lead to the successful evolution of response behavior over very long delay periods.

1 Introduction

The evolution of artificial neural networks of embedded autonomous agents constitutes a powerful tool for creating agents with complex behavioral abilities, and studying properties of the emergent networks dynamics. In this paper we investigate the evolution of Evolved Autonomous Agents (EAAs) that solve a delayed response task. Delayed response tasks are challenging since they are characterized by a significant delay between the stimulus and the corresponding appropriate response, which make them impossible to solve by a simple sensory-motor mapping. Such tasks are very common in real-life situations, and are thought to be solved by working memory. In the traditional delayed-response *match-to-sample* animal behavior task, an animal is presented with a cue, which determines the action that it should perform, but the animal’s action should be taken only after a delay of a few seconds. Here, an autonomous agent is navigating in an arena occupied with food and poison, and should consume as much food as possible, while avoiding poison. In order to eat, the agent has to remain still on a food item for exactly K steps, after which it can consume the food.

In order to be able to wait precisely K steps facing fixed, invariant inputs during the waiting time, the agent has to “remember” how many time steps it has already waited. Similarly to the match-to-sample task in animals, the cue (the presence of food) determines the corresponding action that should be performed after a substantial delay.

Two types of neurocontrollers are evolved: networks of McCulloch-Pitts (MP) neurons, and spiky networks of *Integrate-And-Fire* neurons. Models of spiking neurons have been extensively studied in the neuroscience literature. Spiky networks have a greater computational power than networks of sigmoidal and McCulloch-Pitts neurons [1], and are able to model the ability of biological neurons to convey information by the exact timing of an individual pulse, and not only by the frequency of the pulses [2, 3]. It is appealing to use spiky neural networks in EAAs studies since they are biologically more plausible: Biological neurons perform integration over their pre-synaptic inputs such that a neuron accumulates voltage over time, and fires if its voltage exceeds a threshold. After firing its voltage is returned to a resting voltage, having a refractory period that inhibits its firing. Spiking dynamics may be useful in solving delayed response tasks, since such tasks require memory. Recent studies that combine evolutionary computation with spiky neural networks have analyzed properties of the spiking dynamics in the evolved networks, for example, whether the spiking dynamics result in a time-dependent or a rate-dependent computation, and the effect of noise on the emerging network [4, 5].

This study is the first to address the questions of **whether an evolved network with spiking neurons is truly spiky, and how can we define and measure the spikiness level of each neuron**. In this work we try to answer these questions, by presenting two new fundamental ways by which we define and quantify the spikiness functional contribution, and the level of integration of inputs over time of a spiky neuron. Evolving MP and spiky neurocontrollers **enables us to compare and analyze the resulting network solutions** in terms of the difficulty of the evolution, the delayed response processing, and other network properties. The rest of this paper is organized as follows: Section 2 describes the network architectures and the evolutionary procedure. Section 3 analyzes the evolved neurocontrollers and their dynamics. In Sect. 4 we present and quantify two basic properties of spikiness in embedded agents, with which we analyze the evolved spiky agents. These results and their implications are discussed in Sect. 5.

2 The Model

2.1 The EAA Environment

The EAA environment is described in detail in [6]. The agents live in a discrete 2D grid “world” surrounded by walls. Poison items are scattered all around the world, while food items are scattered only in a “food zone” in one corner. The agent’s goal is to find and eat as many food items as possible during its life, while avoiding the poison items. The fitness of the agent is proportional to the

number of food items minus the number of poison items it consumes. The agent is equipped with a set of sensors, motors, and a fully recurrent neurocontroller of binary neurons.

Four sensors encode the presence of a wall, a resource (food or poison, without distinction between the two), or a vacancy in the cell the agent occupies and in the three cells directly in front of it. A fifth sensor is a “smell” sensor which can differentiate between food and poison underneath the agent, but gives a random reading if the agent is in an empty cell. The four motor neurons dictate movement forward (neuron 1), a turn left (neuron 2) or right (neuron 3), and control the state of the mouth (open or closed, neuron 4).

In previous studies [6], eating occurs if the agent stands on a grid cell containing a resource for one step. Here, we have modified this task to include a delayed-response challenge: In order to eat, the agent has to stand on a grid-cell containing a resource for a precise number of steps K , without moving or turning, and then consume it, by closing its mouth on the last waiting step. Closing its mouth after standing on a food item for more or less than K steps does not increase its fitness. Hence, in essence, the agent has to learn to precisely count to K . The agent’s lifespan, defined by the number of sensorimotor steps available to it, is limited. Waiting steps are not counted as part of the lifespan in order to facilitate the evolution of the delayed-response task.

2.2 The Neurocontrollers

All neurocontrollers are fully-recurrent with self-connections, containing 10 binary neurons (out of which 4 are the motor neurons), such that the 5 sensors are connected to all network neurons. We compare between neurocontrollers with McCulloch-Pitts (MP) neurons, employed conventionally in most EAA studies, and ones with spiky *Integrate-And-Fire* neurons. In both types of networks, a neuron fires if its voltage exceeds a threshold. The spiking dynamics of an Integrate-And-Fire neuron i are defined by

$$V_i(t) = \lambda_i(V_i(t-1) - V_{rest}) + V_{rest} + \frac{1}{N} \sum_{j=1}^N A_j(t)W(j, i), \quad (1)$$

where $V_i(t)$ is the voltage of neuron i at time t , λ_i is a **memory factor** of neuron i (which stands for its membrane time-constant), $A_j(t)$ is the activation (firing) of neuron j at time t , $W(j, i)$ is the synaptic weight from neuron j to neuron i , N is the number of neurons including the input sensory neurons, and V_{rest} stands for the resting voltage (set to zero in all simulations). After firing, the voltage of a spiky neuron is reset to the resting voltage, with no refractory period.

The voltage of a spiky neuron results from an interplay between the history of its inputs and the current input field. The memory factor, which ranges between 0 and 1, determines the amount of integration over time that the neuron performs: The higher the memory factor, the more important is the neuron’s history ($V_i(t-1)$), compared with the current input field (the last summand in (1)). The limit

case of $\lambda_i=0$ corresponds to an MP neuron, in which only the current input field determines the voltage. The memory factor is different for each neuron, as different neurons may have different roles, each demanding a different amount of integration over time.

A *genetic algorithm* is used to evolve the synaptic weights $W(j, i)$ and, for spiky neurocontrollers, the memory factor parameters, both directly encoded in the genome as real valued numbers. Evolution is conducted over a population of 100 agents for 30000 generations, starting from random neurocontrollers, using a mutation rate of 0.02 and uniform point-crossover with rate of 0.35.

3 Delayed Response Processing

In this section we present the basic evolutionary results for both types of agents (Sect. 3.1), after which the successfully evolved spiky and MP networks are compared in terms of localization of processing (Sect. 3.2), and in the way they solve the counting task (Sect. 3.3). This section is concluded with the results of incremental evolutions of the task (Sect. 3.4).

3.1 Performance Evaluation

Successful agents that solve the delayed-response task were evolved with both MP and spiky networks. The evolution of the delayed-response task is fairly difficult, and many evolutionary runs converged without yielding successful agents. We measure the difficulty of each task as the fraction of successful runs (Fig. 1A). Evidently, the task is harder as the agent has to wait for a longer delay period. More important, successful spiky neurocontrollers evolve more easily than MP networks.

3.2 Delayed Response Processing Localization

It is interesting to compare between the spiky and the non-spiky networks, in terms of localization of processing. The localization score [7] is a measure of how distributed is the agent’s task in the network. The lower this measure is, the more the processing is evenly spread in the network across many neurons. The localization score is obtained from lesioning experiments [7], where numerous multiple lesions are afflicted upon a network, and the corresponding performance score of the agent is recorded. Based on these data, the causal importance (contribution) of each element to the behavioral task is assessed by the Multi-lesion Shapley value Analysis (MSA) [8]. By lesioning the network’s neurons we compute the general neuronal contributions, from which the localization score is calculated.

We have shown that the difficulty of evolving the counting-task increases with the delayed response period (Sect. 3.1). Is there a correlation between the difficulty of evolving a network that solves a task, and the localization level of the resulting network? Figure 1B plots the average localization score as a function of the delayed response period. Since the tasks differ only by the duration of this

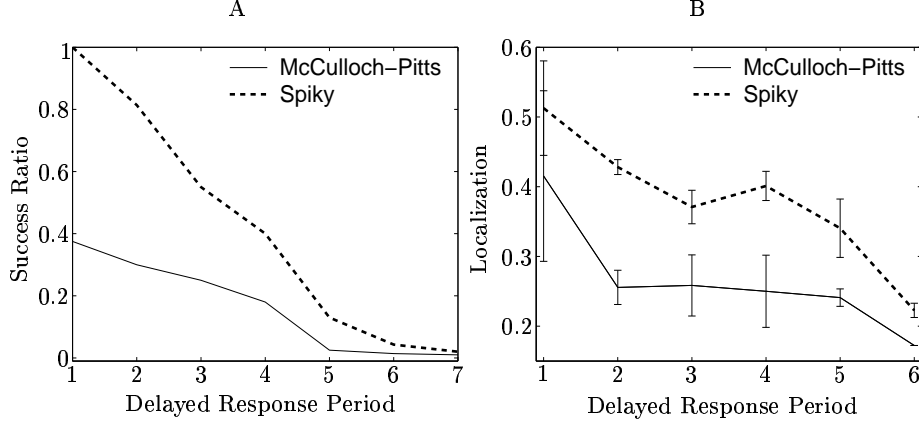


Fig. 1. (A). Task difficulty vs. delayed response period: Success ratio for a task is the fraction of evolutionary runs, out of 30, in which the best agent achieved a behavioral fitness above a pre-defined threshold. (B). Localization score (mean and standard deviation across several successfully evolved agents) vs. delayed response period

counting period, the localization differences can be attributed to the counting process. In both types of agents, the network’s localization decreases with the length of the delayed response period, and MP agents are less localized than the spiky ones. Interestingly, the correlation coefficient between the average localization score and the evolutionary success fraction is high: 0.87 for the spiky networks, and 0.79 for the MP ones.

3.3 Analysis of the Delayed Response Processing

We now turn to investigate the actual counting process in the successful neurocontrollers, studied by a receptive-field analysis, during which the activation patterns of an agent’s neurons are analyzed in view of its behavior. For a specific agent, the firing sequence during the delayed response period usually repeats itself precisely in all counting incidents. Figure 2A presents the activation pattern sequence during the delayed response period of agent MP5, an MP agent with a delay period of 5 time-steps. Since the agent does not move during the delayed response period, the sensory inputs are constant, inhibiting the first 3 motor neurons. On each time-step, a different sub-group of neurons is active, such that at the last waiting step the agent closes its mouth and eats, and a step after the forward neuron is reactivated, and the agent moves. In general, since the sensory inputs stay exactly the same throughout the counting period, for an MP agent to count to K , the network has to pass through K different activation states. Thus an MP neurocontroller with N neurons can theoretically count to 2^N .

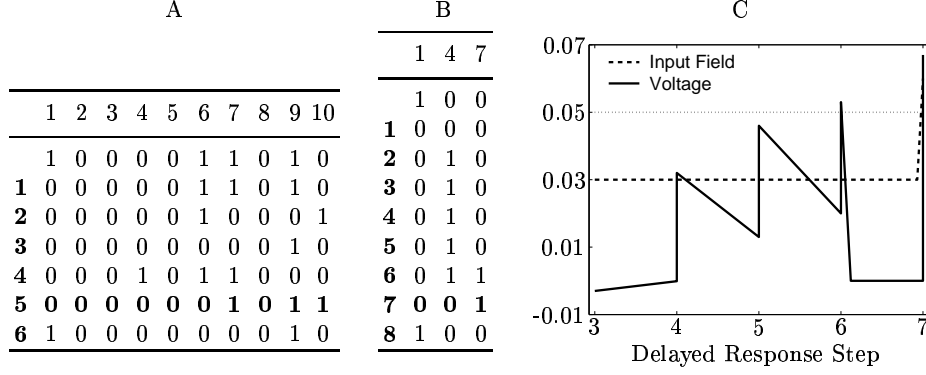


Fig. 2. Counting processing. (A). The activation pattern of the neurons of agent MP5 throughout the delayed response period. An activation value of 1 states that the neuron is firing. Each column represents a neuron, when the leftmost neurons are the forward, left, right and open-mouth motor neurons, respectively. Each row represents a different time-step, in an ascending order. Eating occurs at the fifth time-step (marked in bold) when the agent closes its mouth, which is open in the previous step. After eating the forward motor is activated and the agent moves. (B). The activation pattern of neurons 1, 4 and 7 of agent S7 during the delayed response period (all other neurons do not fire throughout this period). Columns and rows as in A, eating occurs at the seventh time-step. (C). The input field and the voltage of neuron 7 of agent S7 during time steps 3 to 7 of the delayed response period. The memory factor of neuron 7 is 0.43, the threshold above which the neuron fires is 0.05 (dotted line). The voltage decays between input ticks, and is recalculated with each new input field. This decay is exponential (1), the linear decay portrayed is only for illustration purposes. After firing at the sixth time step the voltage is returned to V_{rest} , and the neuron fires again at the seventh step without integration

Figure 2B shows the firing pattern of the counting neurons of S7, a spiky agent with a delay period of 7 waiting time-steps. The food inputs activate neuron 4, which has a positive synapse to neuron 7. Neuron 7, a spiky neuron with a memory factor of 0.43, gradually accumulates voltage during steps 3 to 5, till it passes the threshold and fires at the sixth time step. This firing inhibits the forth neuron on the seventh step, closing the agent's mouth to consume the food. After eating, the forward neuron is activated by neuron 7, and the agent moves to another grid-cell. As shown in Fig. 2C, during steps 3 to 6 of the delayed response period the input field of neuron 7 is constant, bellow the threshold, hence the neuron's history is the one causing the increase in its voltage. In a spiky network, the same network activation pattern can be repeated several times during the counting process, since the state of a neuron consists also of its voltage. Thus, theoretically, under spiky dynamics, a network can count with

a single neuron, that accumulates voltage over $K - 1$ steps, and fires on the K th step. The evolved spiky networks do not possess such efficient counting, but usually one or more neurons use their spikiness to accumulate voltage and “count” for a few steps, and as a result, less neurons participate in the counting process compared with MP networks, explaining the higher localization levels of the spiky networks (Sect. 3.2).

3.4 Incremental Evolution

Since evolving agents with a large delay period is a difficult evolutionary task (Sect. 3.1), we use the incremental evolution technique: Each evolutionary process starts with a population of 100 mutated copies of an already evolved agent with a delay period of K , and develops an agent with a delay period of $K + 1$. Incremental evolutions were easier and shorter than regular evolutions for both spiky and MP agents, and spiky incremental evolutions were easier and shorter than incremental MP ones (around 90% success ratio, 6000 generations needed in typical spiky incremental evolutions, and around 70% success ratio, 15000 generations needed in typical MP incremental evolutions). For MP networks, we have succeeded in developing agents that count to 15. Examining the activation patterns of the agents developed via this incremental evolution revealed that agents counting to 3 use two neurons in the counting process, agents that count to 4-5 do it with four neurons, agent that count to 6-7 use five neurons, and an additional neuron is used for counting to 8 and above, demonstrating that under MP dynamics more neurons are needed to count to a higher number.

For spiky networks, we have evolved agents that count up to 35! Out of these, the spiky networks with a delayed response period of 3 to 9 use two neurons for counting: one is spiky, and the other has a vanishing memory factor, and de facto behaves like an MP neuron. In the networks with a delayed response period of 10 and higher this MP-like neuron was transformed into a real spiky neuron with a memory factor of 0.38, such that the two spiky neurons count together. The difference between the agents is only the amount of time-steps that each spiky neuron “counts”, influenced mainly by the memory factors. Correspondingly the networks of the agents that count to 10 and higher have a constant localization value, irrespective of K . This result demonstrates that a spiky network can indeed employ a very efficient localized counting method (as described in Sect. 3.3). The evolutionary process does not always find such efficient solutions, since there is no evolutionary pressure towards this direction. Spiky agents with even larger delay periods can be easily constructed by manipulating the memory factors and the synaptic weights, as well as by continuing with the incremental evolutionary runs.

4 The Different Faces of “Spikiness”

4.1 Two Measurements of Spikiness

Are the evolved agents with spiking dynamics really spiky? First, having encoded the neuronal memory factors in the genome gives rise to the possibility that the

evolution will come out with non-spiky solutions. Second, **even if the memory factor is high (1), it does not ensure that the neuron indeed utilizes its “integration potential” in its firing.** For example, a neuron may receive a large excitatory input field in every time step and fire in a very high frequency, without performing any integration over its past input fields. That is, given its input field, its pattern of firing would be indistinguishable from an MP neuron. Third, **even if the spikiness is utilized for firing, it does not necessarily contribute to the agent’s performance.** Essentially, we aim to distinguish between the observation that a given neuron has been assigned *spiking dynamics* by evolution, i.e. obtained a non-vanishing memory factor, and the true level of its *spikiness*, i.e., the amount by which it really “utilizes” its spiking dynamics. In this section we present two methods for measuring the spikiness level of a neuron, based on two fundamentally different perspectives.

Relevant Spikiness (RS). The first measurement for spikiness answers the following question: **how much are the spiking dynamics of a neuron needed for good performance of the agent?** Intuitively, if while abolishing the spiking dynamics of a neuron, the agent’s performance deteriorates considerably, then its spiking dynamics contribute to the agent’s behavior. If, in contrast, the fitness of the agent is maintained, this neuron’s spiking dynamics are functionally insignificant.

To quantify this type of spikiness we define a new lesioning method: This λ -**lesioning** method *lesions only the memory factor of the neuron*, leaving the rest of its dynamics unaltered. Lesioning the memory factor is done by clamping it to zero, which turns the neuron into an MP one. Fitness scores are measured for all multi-lesion configurations with the λ -lesioning method, and the MSA is utilized to quantify **the contribution of the memory factor of each neuron to successful behavior.** .

Evident Spikiness (ES). The second index of spikiness measures **how much do the spiking dynamics of a neuron influence its firing:** If the firing pattern of a neuron stays the same regardless of whether it possesses spiking dynamics or not, then we can consider it as non-spiky. The ES index is calculated by comparing the firing of a spiky neuron to that of an MP neuron receiving an identical input field on each time step (last summand in (1)). The fraction of time steps in which there is a difference between the binary activations of the spiky neuron and a corresponding “benchmark” MP neuron, quantifies **the average percentage of lifetime steps in which the spiking dynamics “made a difference” in the firing of the neuron examined.** The RS and ES measures are normalized such that the sum over all neurons equals one.

4.2 Analysis of Spikiness

We examine the spikiness level of the neurocontrollers evolved with spiking neurons, focusing on two agents: S5 and S7, with a delay period of 5 and 7 waiting

time steps, respectively. For S5, Figure 3A compares the general contributions of the neurons (yielded by the MSA, Sect. 3.2) with the RS values, indicating how much the spikiness of each neuron contributes to successful behavior. Notably, neurons 1, 4, 9 and 10 contribute significantly to the agent’s behavior, as shown by their general MSA contributions, while the spikiness of only neurons 1 and 10 has a significant contribution, according to their RS values. Figure 3A also presents the ES values. Clearly, the two methods for measuring “spikiness” yield different results: Neuron 5 receives a very high ES score, and a near-zero RS score. A more pronounced difference is apparent in Fig. 3B, which shows both measures for agent S7, whose delayed response processing was analyzed in Sect. 3.3. In this case, the seventh neuron gets the highest RS value, but receives a pretty low ES score. Neurons 5 and 8 are the most spiky ones according to the ES measure, but receive low RS values.

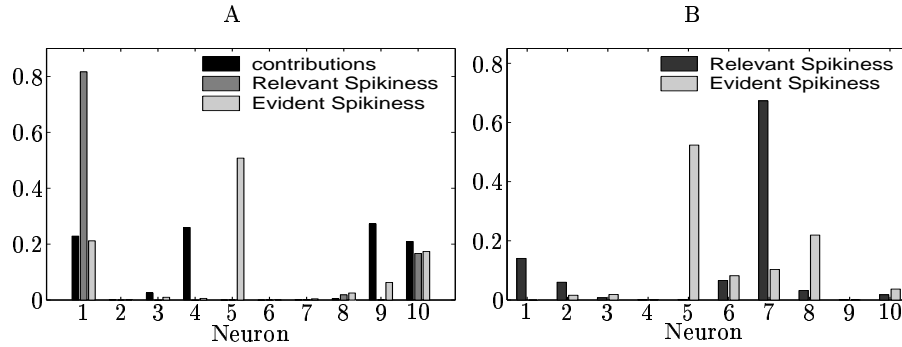


Fig. 3. *Comparison between spikiness measurements.* (A). RS and ES scores for the neurons of agent S5, along with their general neuronal contributions. (B). RS and ES scores for agent S7

The difference between the results of the two spikiness measurements originates in their different nature: The ES method measures the role that spiking dynamics play in determining the firing of a neuron *irrespectively of that neuron’s functional and behavioral role*. The RS method is a functional measurement, which considers as spiky only neurons whose spikiness contributes to behavior. Usually such neurons also have high neuronal contributions, and are generally important to behavior. Additionally, these neurons must have a non-zero ES score, since influencing the firing pattern of the neuron is a prerequisite of having a behavioral contribution. In the case of agent S7, as shown in Sect. 3.3, neuron 7’s spikiness plays a pivotal role in the agent’s counting ability. When this neuron is λ -lesioned, the agent does not eat, explaining the very high RS value assigned to it. However, since the fraction of steps in which the spiking dynamics influence the activation of neuron 7 is only about 3% of its total life-time steps (the counting steps), this neuron receives a low ES score. In almost all

evolved spiky agents, most of the network’s neurons have low spikiness functional contributions (RS), and usually, neurons with high RS scores are involved in the counting process. The two spikiness measurements show that the evolved agents with spiking dynamics are truly spiky, both in using past history to determine the activation patterns of some neurons, and in utilizing the spiking dynamics for successful behavior.

5 Conclusions

In this work we have succeeded in evolving agents solving a non-trivial, memory-dependent delayed response task, without using any special structure for counting, or giving an external reinforcement to the agent while waiting. Although incremental evolutions were easier and shorter than regular evolutions, even by a regular evolutionary process we evolved an agent that stands still on a food item for precisely 7 time-steps and then consumes it. By analyzing the counting process, we have shown how the ability of a spiky neuron to accumulate voltage is utilized for counting. We have further illustrated that in a delayed-response task, as may be the case for other memory-dependent tasks, networks of spiking neurons can be less complex and easier to evolve, compared with MP networks. These results demonstrate the importance of choosing a network architecture with characteristics and abilities that match the requirements of the given task.

Additionally, the study of spiky neural network in the context of EAAs brings forward basic questions regarding spiking dynamics that have not yet been raised. We have shown that the presence of spiking dynamics does not necessarily transcribe to actual spikiness in the network, and that the spikiness level can be defined and quantified in several functionally different ways. Specifically, the spikiness functional contribution (RS) and the spikiness evident influence (ES) each point to different neurons in the studied neurocontrollers.

References

1. Maass, W.: Networks of spiking neurons: the third generation of neural network models. *Neural Networks* **10**, 1656-1671 (1997)
2. Bugmann, G.: Biologically plausible neural computation. *Biosystems* **40**, 11–19 (1997)
3. Maass, W., Ruf, B.: On computation with pulses. *Information and Computation* **148**(2), 202–218 (1999)
4. Floreano, D., Mattiussi, C.: Evolution of spiking neural controllers for autonomous vision-based robots. *Evolutionary Robotics IV*, Berlin, Springer-Verlag (2001)
5. Paolo, E.A.D.: Spike-timing dependent plasticity for evolved robot control: neural noise, synchronization and robustness. To appear in *Adaptive Behavior* **10** (2003)
6. Aharonov-Barki, R., Beker, T., Ruppín, E.: Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation* **13** (2001) 691–716
7. Aharonov, R., Segev, L., Meilijson, I., Ruppín, E.: Localization of function via lesion analysis. *Neural Computation* **15** (2003)
8. Keinan, A., Hilgetag, C.C., Meilijson, I., E.Ruppín: Fair attribution of contribution: Shapley value analysis of neurocontrollers. preprint (2003)