# Evolution of alternate object pushing in a simulated embodied agent: Preliminary report

Răzvan V. Florian

Center for Cognitive and Neural Studies (Coneural)
str. Saturn 24, 400504 Cluj-Napoca, Romania
www.coneural.org
florian@coneural.org

August 14, 2004

**Abstract**

A push-release-seek reflex is evolved in a simulated simple agent interacting with objects. The agent is controlled by a spiking neural network. Succesful experiments involved a network of 125 neurons and spike-time-dependent synaptic plasticity. Plasticity seems to be actively used by the evolved network in memorizing some aspects of the environment that allow the performance of the task.

## 1 Introduction

Genuine, creative artificial intelligence can emerge only in embodied agents, capable of cognitive development and learning by interacting with their environment (Florian, 2003*a*). Before the start of the learning process, the agents need to have some innate (predefined) drives or reflexes that would induce the exploration of the environment. Otherwise, the agents would not do anything once emerged in their environment, and learning would not be possible.

In the experiments presented in this paper, some basic reflexes are evolved for a simple simulated agent, controlled by a spiking neural network, that is able to interact with the objects in its environment. These reflexes will be used in future research to bootstrap the ontogenetic cognitive development of the agent. By learning the structure of the environment, guided by value signals, the agent may be able, in the future, to perform tasks much more complex than the reflexes evolved here.

The preliminary evolutionary experiments also serve to provide working spiking embodied neural networks. Their functioning may be analyzed and the results may lead to new models of self-organization that can be implemented to ground the ontogenetic cognitive development of the agent. Also, these experiments serve to estimate the complexity of the behaviors and of the neural structures that can be evolved in reasonable computing time using current computers.

# 2 The agent, its environment and its task

## 2.1 The simulator

The agent and its environment are simulated using Thyrix, a simulator specifically designed for evolutionary and developmental experiments for embodied artificial intelligence research (Florian, 2003*c*). The simulator provides a two-dimensional environment with simplified, quasi-static (Aristotelian) mechanics, and supports collision detection and resolution between the objects in the environment.

## 2.2 The agent's morphology

The agent's morphology was chosen as the simplest one which would allow the agent to push the circular objects in its environment without the slipping of the objects on the surface of the agent. This slipping may appear, for example, if a circle pushes another circle, and the pushing force is not exactly oriented on the line connecting the centers of the two circles.

We wanted maximum simplicity both for economy (to allow evolution and development in less computing time) and for having few degrees of freedom, which may allow dynamical analysis and simpler statistical analysis of the behavior of the agent. However, we have tried to respect the principle of ecological balance (Pfeifer and Scheier, 1999, pp. 455–463) in the design of the agent's morphology and sensorimotor capabilities.

Thus, the agent is composed of two circles, connected by a variable length link. The link is "virtual", in the sense that it provides a force that keeps the two circles together, but it does not interact with other objects in the environment, i.e. external objects can pass through it without contact. With this morphology, the agent can easily push other circles in its environments, by keeping them between its two body circles, without the need of balancing them to prevent slipping.

## 2.3 The agent's effectors and sensors

The agent can apply forces to each of its two body circles. The forces originate from the center of the circles and are perpendicular to the link connecting them. They may be considered to originate from some virtual "rockets". Two effectors correspond to each of the two body circles, one

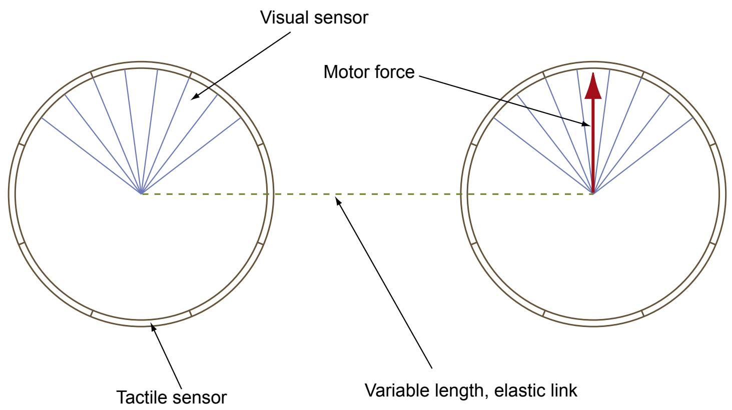


Figure 1: The agent's morphology.

commanding a forward-pushing force, and one commanding a backward-pushing force. There are thus four "rocket" effectors. The net motor force acting on one body circle is the sum of the forward and backward forces. These effectors allow thus the agent to move backward or forward, to rotate in place, and, in general, to move within its environment.

A fifth effector commands the length of the virtual link connecting the two body circles, between zero and a maximum length. If the actual length of the link is different from the commanded length, an elastic force (proportional with the difference between the desired and actual length) acts on the link, driving it to the desired length.

The agent has contact sensors equally distributed on the surface of its two body circles (8 contact sensors per circle, spanning a 45° angle each). The activation of the sensors is proportional to the sum of the magnitudes of the contact forces acting on the corresponding surface segment, up to a saturation value.

Each circle also has 7 visual sensors. Each sensor has a 15° view angle, originating from the center of the circle. Thus, each circle has a 105° viewing angle, centered around the "forward" direction. The activation of the sensors is proportional to the fraction of the viewing angle covered by external objects. The range of the sensors is infinite.

The agent also has proprioceptive sensors corresponding to the effectors. Each body circle has two velocity sensors, measuring the velocity in the forward and backward directions, respectively. The sensors saturate at a value corresponding to the effect of the maximum motor force that can be commanded by the effectors. The agent also has a proprioceptive sensor that measures the actual length of the link connecting the two body circles, that saturates at the maximum length that can be commanded by the link effector.

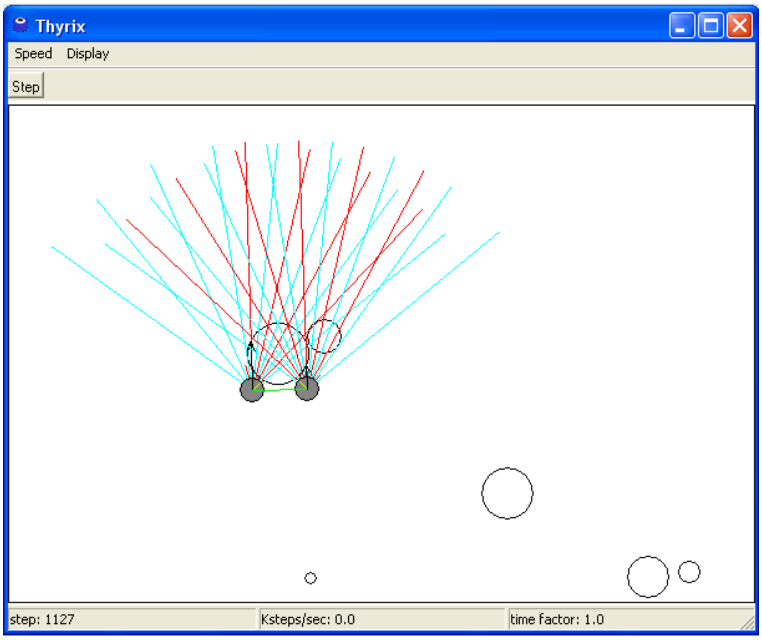


Figure 2: The agent used in the experiments, pushing a ball in the environment. The gray circles compose the body of the agents, the green segment represents the elastic link connecting them. The light blue rays emerging from them represent the viewing angles of the visual sensors, the red rays represent the activation of these sensors. The open circles are the balls.

Thus, the agent has a total of 5 effectors and 35 sensors (16 contact sensors, 14 visual sensors, and 5 proprioceptive ones). Each sensor or effector can have an activation between 0 and 1.

## 2.4 The environment

In the experiments presented in this paper, the environment consisted of one agent and 6 circles ("balls") that the agent can move around. The spatial extension of the environment was not limited. The balls have variable radiuses (between $0.06\,m$ and $0.26\,m$), comparable to the radius of the agent's body circles ($0.1\,m$).

During each trial, the agent and the balls were positioned randomly in the environment, without contact, in a rectangular perimeter of $6\,m$ by $4\,m$.

## 2.5 The task

The task of the agent was to move alternatively each of the balls in its environment, on a distance as long as possible, in limited time ($100\,s$ of simulated time). More specifically, the fitness of each agent was computed as the sum of the distances on which each ball was moved, but with a threshold of $d_t = 2\,m$ for each ball. Thus, the agent had to move all balls, instead of just detecting one ball and pushing it indefinitely. The sum of distances may thus range between 0 and $12\,m$.

This task was considered to evolve a push-release-seek reflex, that might be used in future experiments to bootstrap more complex behaviors, such as arranging the balls in a particular pattern, sorting the balls by size, or categorizing different kinds of objects.

If the agent would move in straight line at the maximum speed corresponding to the maximum forces it can produce, pushing the six balls for equal time, and neglecting the time needed for taking curves, seeking the balls, switching between balls, the distance that it may cover in the limited time is $55.945\,m$. Given the existence of distances between balls, the fact that the speed is lower when taking curves, that the agent has to release the balls when switching them, we may evaluate that the task is relatively difficult. It may require the coordination of the motor effectors for attaining high speeds, the evaluation of the distance or time spent pushing a certain ball, and eventually some memorization, either of objects size or positions, that would prevent the repeated pushing of the same balls.

To determine the fitness of a particular individual, we have averaged its performance on three trials, with random initial configurations of the balls.

## 3 The controller

### 3.1 The spiking neural network

The controller of the agent consisted of a recurrent spiking neural network. We have used this type of network as it seems to be the most suited for the control of embodied agents, among the classes of neural networks amenable to large scale computer simulation (Florian, 2003*b*). The controller has as input the activations of the agent's sensors, and as output the activations of the agent's effectors.

We have designed a fast, event-driven spiking neural simulator. The simulator was inspired by Mureşan's Neocortex (Mureşan and Ignat, 2004), however is simpler and faster, as we didn't need features like retinotopic maps, multiple types of neuron updating, and we used a single type of neuron.

The network consists of leaky integrate-and-fire neurons (Gerstner and Kistler, 2002, Chapter 4) with a resting potential of $-65\,mV$, a threshold potential of $-40\,mV$, a resistance of $10\,M\Omega$ and a decay constant of $10\,ms$.

The simulator uses discrete time with a resolution of $1\,ms$. At each time step, only the neurons that receive spikes are updated (hence the event driven nature of the updating of the network). If the updated neurons fire, their spikes are stored in a list. This spike list will be used during the next

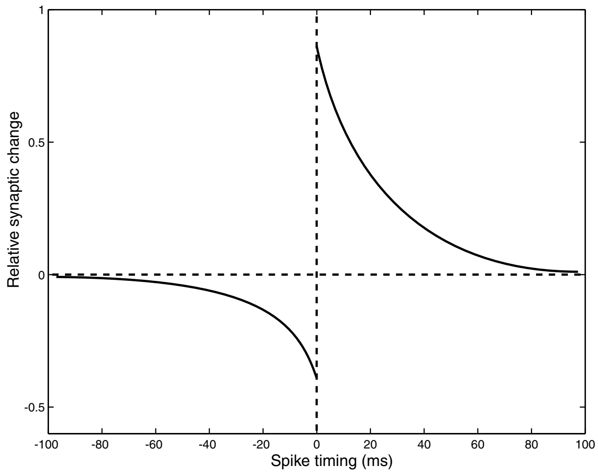


Figure 3: Typical example of spike-timing dependent plasticity. The relative change in synaptic efficacy $\Delta w/w$ is plotted against the delay $\Delta t$ between the postsynaptic and presynaptic spike. Synaptic potentiation ($\Delta w/w > 0$) appears when the postsynaptic spike follows (is caused by) the presynaptic spike ($\Delta t > 0$) (after Bi, 2002).

time step to update the affected postsynaptic neurons. Thus, the spikes propagate within an axonal delay of one time step.

## 3.2 Spike-time dependent plasticity

During some experiments, the neural network featured spike-time dependent plasticity (STDP). STDP is a phenomenon that was experimentally observed in biological neural systems (Bi, 2002). The changes $\Delta w$ of the synapse efficacies depend on the relative timing $\Delta t$ between the postsynaptic and presynaptic spikes (see Fig. 3). The synapse is strengthened if the postsynaptic spike occurs shortly after the presynaptic neuron fires, and is weakened if the sequence of spikes is reversed, thus enforcing causality. Notably, the direction of the change depends critically on the relative timing.

We have modeled STDP following the method of Song et al. (2000). The values of the parameters used were $A_+ = 0.005$, $A_- = 1.05 A_+$, and $\tau_+ = \tau_- = 50\,ms$.

Following Di Paolo (2002*b*), we have implemented directional damping for the synapse efficacies. The synapse efficacies, which are variable due to STDP are limited to the interval $[0, w_{max}]$, where $w_{max}$ can be either positive or negative, and is a genetically determined maximum (in absolute value) efficacy. We thus have $0 \leq w/w_{max} \leq 1$.

## 4 The agent-controller interface

In interfacing a spiking neural controller with an embodied agent, a conversion of the analog input and output signal to binary spikes has to be performed. Following Di Paolo (2002*b*), the analog values of the sensor activations were converted to a spike train using a Poisson process with a firing rate proportional to the activation. The maximum firing rate of the input neurons was set to $100\,Hz$.

The spikes of the motor neurons were converted to an analog value by a leaky integrator of time constant $\tau = 10\,ms$. The maximum value of the effector activation, 1, corresponds to a firing rate of the motor neuron of $100\,Hz$.

Each sensor of the agent, of activation $s$, $0 \leq s \leq 1$, drove two input spiking neurons, one being fed with activation $s$ and the other with activation $1-s$. Thus, both the activation of the sensor and its reciprocal was fed to the network, and there are 70 input neurons in the network. The reason of this duplication of the sensory signal in the spiking neural network is twofold. First, this allows the network to be active even in the absence of sensory input. For example, if the agent is in a position where nothing activates its sensors (there is no object in its visual range, no tactile contact etc.), there must be however some activity in the neural network, in order for the effectors to be activated and the agent to orientate to stimuli. Second, this mechanism implies that the total input of the network is approximately constant in time (the number of spikes that are fed to the network by the input). This might be important for future analysis of the network's activity in the framework of the theory of far-from-equilibrium systems.

## 5 The evolutionary algorithm

We have evolved controllers with both fixed and variable topology networks. The networks contain input neurons (driven by the sensors), hidden neurons, and motor neurons (that drive the agent's effectors).

The parameters determined by evolution were the values of the synaptic efficacies $w$ (in the non-plastic case), or the values of the maximum (in absolute value) synaptic efficacies $w_{max}$ (in the STDP case). In addition to these, in the case of variable topology networks, the topology was also determined by evolution.

### 5.1 Evolution of fixed topology networks

The fixed topology networks were fully connected (i.e., all neurons were connected to all neurons in the network, except input neurons, which had only efferent connections). In this case, there are 50 hidden neurons, in addition to the 70 input neurons and the 5 output neurons. The networks thus consists of 125 neurons and 6875 synapses. The genome directly encodes the synaptic efficacies ($w$ or $w_{max}$) of these synapses.

We have used a standard evolutionary algorithm, with a population of 80 individuals, truncation selection (the top 25 % individuals reproduce) and elitism. 10 % of the offspring result from mating with single cut crossover. Mutation is applied uniformly to all genes.

### 5.2 Evolution of variable topology networks

We have also evolved networks with variable topology. There are several advantages in evolving such networks. The number of hidden neurons does not have to be arbitrarily established before the evolution. The network is allowed to grow from a small initial number of neurons, up to the number needed to solve the task, through a process of complexification. Full connectivity is not needed: the connections between neurons are evolved, and thus the network might have a smaller number of connections than the maximum possible. Less neurons and less synapses lead to less computing time needed for the evaluation of these networks, and thus may result in faster evolution.

The method used for evolving growing networks was a modified version of the NeuroEvolution of Augmenting Topologies method (NEAT) (Stanley and Miikkulainen, 2002).

## 6 Results

We evolved first fixed topology networks. Networks with both static and plastic synapses evolved to solve the required task, with the performance of the best individuals reaching a plateau at about 11.3, very close to the maximum possible of 12 (see Fig. 4). Plastic networks evolved much faster, in terms of generations, than networks with static synapses. However, the simulation of plastic networks require a higher computational effort.

We have also tried to evolve variable topology networks. In this case, the best fitness reached plateaus around 4.5 and thus we could not evolve networks to solve the task. Further work is needed to investigate reasons for the failure and solutions for the growth of the networks.

## 7 Conclusion

We have succesfully evolved a fully connected spiking neural network of 125 neurons, that allows a simple agent to alternatively push the 6 balls in its environment. This push-and-seek reflex may be used in future experiments to bootstrap the development of more complex behaviors. Also, the dynamics of the network may be analyzed to uncover the mechanisms used by the

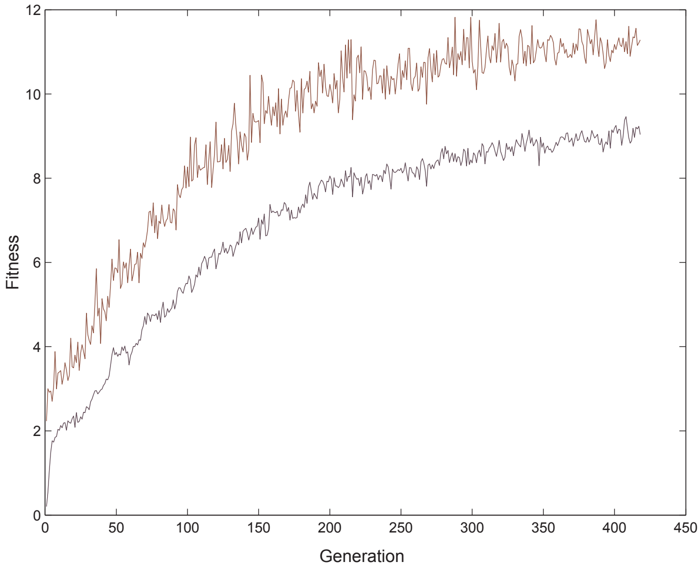


Figure 4: Evolution in the fixed topology, STDP case. The top curve represents the best fitness in a generation, and the bottom one represents the average fitness.

agent for the measurement of the distance it pushed a particular ball, or the memory mechanisms which permit the agent to choose balls not pushed yet.

The current experiment evolved a network much larger than the spiking neural networks evolved in previous studies for the control of embodied agent, which employed no more than 23 neurons (Floreano and Mattiussi, 2001; Floreano, Schoeni, Caprari and Blynel, 2002; Floreano, Zufferey and Mattiussi, 2002; French and Damper, 2002; Di Paolo, 2002*a*,*b*; Katada et al., 2003). This was possible because we used a fast agent-environment simulator especially designed for evolutionary and developmental experiments, and a fast event-driven neural network simulator.

Further research will analyze the evolved networks, will try to evolve variable topology networks able to solve the task, and will also try to evolve networks capable of solving more dificult tasks, such as sorting the balls by size.

# 8 Acknowledgements

This work was sponsored by Arxia. We thank to Raul Mureşan for providing the code of the Neocortex spiking neural network simulator.

## References

Bi, G.-Q. (2002), 'Spatiotemporal specificity of synaptic plasticity: cellular rules and mechanisms', *Biological Cybernetics* **87**, 319–332. Available from: `http://www.neurobio.pitt.edu/2_bi2002BCreview.pdf`.

Di Paolo, E. A. (2002*a*), 'Evolving spike-timing dependent plasticity for robot control', *EPSRC/BBSRC International Workshop: Biologically-inspired Robotics, The Legacy of W. Grey Walter, WGW'2002. HP Labs, Bristol, 14 - 16 August 2002* . Available from: `http://www.cogs.susx.ac.uk/users/ezequiel/dipaolo_wgw2002.ps`.

Di Paolo, E. A. (2002*b*), 'Spike timing dependent plasticity for evolved robots', *Adaptive Behavior* **10**, 243–263. Available from: `http://citeseer.nj.nec.com/598020.html`.

Floreano, D. and Mattiussi, C. (2001), Evolution of spiking neural controllers for autonomous vision-based robots, *in* T. Gomi, ed., 'Evolutionary Robotics IV', Springer-Verlag, Berlin. Available from: `http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/er2001b.pdf`.

Floreano, D., Schoeni, N., Caprari, G. and Blynel, J. (2002), Evolutionary bits'n'spikes, *in* R. K. Standish, M. A. Bedau and H. A. Abbass, eds, 'Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life', MIT Press, Boston, MA. Available from: `http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/alife02.pdf`.

Floreano, D., Zufferey, J. C. and Mattiussi, C. (2002), 'Evolving spiking neurons from wheels to wings', *Proceedings of the 3rd International Symposium on Human and Artificial Intelligence Systems, Fukui, Japan* . Available from: `http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/FloreanoEtAl_HARTS2002_color.pdf`.

Florian, R. V. (2003*a*), 'Autonomous artificial intelligent agents', *Technical Report Coneural-03-01* . Available from: `http://www.coneural.org/reports/Coneural-03-01.pdf`.

Florian, R. V. (2003*b*), 'Biologically inspired neural networks for the control of embodied agents', *Technical Report Coneural-03-03* . Available from: `http://www.coneural.org/reports/Coneural-03-03.pdf`.

Florian, R. V. (2003*c*), 'Thyrix: A simulator for articulated agents capable of manipulating objects', *Technical Report Coneural-03-02* . Available from: `http://www.coneural.org/reports/Coneural-03-02.pdf`.

French, R. L. B. and Damper, R. I. (2002), Evolution of a circuit of spiking neurons for phototaxis in a Braitenberg vehicle, *in* B. Hallam, D. Floreano,

J. Hallam, G. Hayes and J.-A. Meyer, eds, ‘From animals to animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior’, MIT Press, Cambridge, MA, pp. 335–344.

Gerstner, W. and Kistler, W. M. (2002), *Spiking neuron models*, Cambridge University Press, Cambridge, UK. Available from: http://diwww.epfl.ch/~gerstner/BUCH.html.

Katada, Y., Ohkura, K. and Ueda, K. (2003), ‘Artificial evolution of pulsed neural networks on the motion pattern classification system’, *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), July 16 - 20, 2003, Kobe, Japan* pp. 318–323. Available from: http://www.rci.scitec.kobe-u.ac.jp/~katada/Dpaper/Cira03_katada.pdf.

Mureşan, R. C. and Ignat, I. (2004), ‘The “Neocortex” neural simulator: A modern design’, *International Conference on Intelligent Engineering Systems, September 19-21, 2004, Cluj-Napoca, Romania* . Available from: http://www.raulmuresan.home.ro/Papers/Neocortex%20Muresan%202004.pdf.

Pfeifer, R. and Scheier, C. (1999), *Understanding intelligence*, MIT Press, Cambridge, MA.

Song, S., Miller, K. D. and Abbott, L. F. (2000), ‘Competitive hebbian learning through spike-timing-dependent synaptic plasticity’, *Nature Neuroscience* **3**, 919–926. Available from: http://people.brandeis.edu/~abbott/papers/STDP1.pdf.

Stanley, K. O. and Miikkulainen, R. (2002), ‘Evolving neural networks through augmenting topologies’, *Evolutionary Computation* **10**, 99–127. Available from: http://www.cs.utexas.edu/users/nn/downloads/papers/stanley.ec02.pdf.