Spiking Neural Controllers for Pushing Objects Around

Răzvan V. Florian[1,2]

[1] Center for Cognitive and Neural Studies (Coneural),
Str. Saturn nr. 24, 400504 Cluj-Napoca, Romania
florian@coneural.org
http://www.coneural.org/florian

[2] Babeş-Bolyai University, Institute for Interdisciplinary Experimental Research,
Str. T. Laurian nr. 42, 400271 Cluj-Napoca, Romania

Abstract. We evolve spiking neural networks that implement a seek-push-release drive for a simple simulated agent interacting with objects. The evolved agents display minimally-cognitive behavior, by switching as a function of context between the three sub-behaviors and by being able to discriminate relative object size. The neural controllers have either static synapses or synapses featuring spike-timing-dependent plasticity (STDP). Both types of networks are able to solve the task with similar efficacy, but networks with plastic synapses evolved faster. In the evolved networks, plasticity plays a minor role during the interaction with the environment and is used mostly to tune synapses when networks start to function.

1 Introduction

Genuine, creative artificial intelligence can emerge only in embodied agents, capable of cognitive development and learning by interacting with their environment [1]. Before the start of the learning process, the agents need to have some innate (predefined) drives or reflexes that can induce the exploration of the environment. Otherwise, the agents might not do anything once emerged in their environment, and learning would not be possible. In the experiments presented in this paper, we evolve a basic drive for a simple simulated agent that is able to interact with the objects in its environment. This drive could be used in future research to bootstrap the ontogenetic cognitive development of the agent.

The agent is controlled by a spiking neural network [2,3]. Among classes of neural network models amenable to large scale computer simulation, recurrent spiking neural networks are an attractive choice for implementing control systems for embodied artificial intelligent agents [4]. Spiking neural networks have more computational power per neuron than other types of neural networks [5,6,7]. Several studies [8,9,10,11] have shown that spiking neural networks achieve better performance for the control of embodied agents than continuous time recurrent neural networks or McCulloch-Pitts networks. More importantly, spiking neurons have a closer resemblance to real neurons than other neural models, which allows a bidirectional transfer of concepts and methodologies between neuroscience

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 570–581, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and artificial neural systems. Biological examples may suggest architectures and learning mechanisms for artificial models that would improve their performance. In the reverse direction, theories developed during the study of embodied artificial neural networks may lead to new concepts and explanations regarding the activity of real neural networks [12].

Evolved spiking neural networks have been used in the last few years for the control of simulated or real robots, but more rarely than other types of neural networks [8,13,14,15,16,17,18,9,19,20,21,22,23,24,10,11,25,26,27]. Among previous evolutionary studies, only one explored the properties of a plastic spiking neural network [18,9,19]. Very few studies used spiking neural controllers for embodied agents that were not evolved, but were taught using other learning methods [28,29,30,31,32].

This paper presents experiments where we evolved spiking neural networks with static as well as with plastic synapses. These networks are one of the largest spiking neural networks evolved to date. The evolved controllers display interesting minimally-cognitive capabilities, being able to discriminate relative object size.

2 The Agent, Its Environment and Its Task

2.1 The Simulator

The agent and its environment were simulated using Thyrix, an open source simulator specifically designed for evolutionary and developmental experiments for embodied artificial intelligence research [33]. 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 that allows the agent to push the circular objects in its environment without slipping of objects on the surface of the agent. Slipping may appear, for example, if a circle pushes another circle, and the pushing force is not positioned exactly on the line connecting the centers of the two circles.

We wanted maximum simplicity both for economy (in order to need less computing time for evolution) and for having few degrees of freedom, which allows a simpler analysis of the behavior of the agent. However, we have tried to respect the principle of ecological balance [34] 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 environment, by keeping them

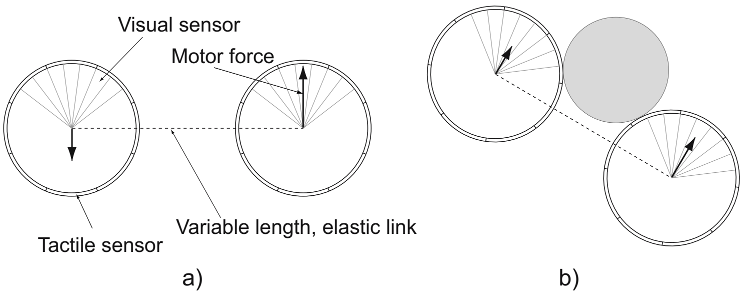

Fig. 1. a) The agent's morphology. b) The agent pushing a ball.

between its two body-circles, without the need of balancing them to prevent slipping. The agent was named Spherus and the code that defines it is available in the open source Thyrix simulator (`http://www.thyrix.com`).

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. Two effectors correspond to each of the two body-circles, one commanding a forward-pushing force, and one commanding a backward-pushing force. 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, centered around the "forward" direction. Each sensor has a 15° view angle, originating from the center of the circle. The activation of the sensors is proportional to the fraction of the view angle covered by external objects. The range of the visual 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.

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 (varying linearly between $r_1 = 0.06$ m and $r_2 = 0.26$ m), comparable in size 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 $6\,d_t = 12$ m.

This task was considered to implement a seek-push-release drive, 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 moves in straight line at the maximum speed corresponding to the maximum forces it can produce, pushing the six balls for equal time, and if we neglect 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 can see that the task is relatively difficult. From the perspective of an external observer, 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 the memorization of either objects' sizes or positions, that prevents 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. The controller had as input the activations of the agent's sensors, and as output the activations of the agent's effectors. The network was implemented by a fast, event-driven spiking neural simulator, inspired by Neocortex [35].

The network consisted of leaky integrate-and-fire neurons [3] with a resting and reset potential of -65 mV, a threshold potential of -40 mV, a resistance of

10 MΩ and a decay time constant of 10 ms. The network was fully connected: all neurons were connected to all neurons in the network, except input neurons, which had only efferent connections; there were no self-connections. There were 50 hidden neurons, in addition to the 70 input neurons and the 5 output neurons. The network was thus composed of 125 neurons and 6875 synapses.

The simulator used discrete time with a resolution of 1 ms. At each time step, only the neurons that received spikes were updated (hence the event driven nature of the updating of the network). If the updated neurons fired, their spikes were stored in a list. This spike list was used during the next time step to update the affected postsynaptic neurons. Thus, the spikes propagated within an axonal delay of one time step.

3.2 Spike-Timing-Dependent Plasticity

During some of the experiments, the neural network featured spike-timing-dependent plasticity (STDP). STDP is a phenomenon that was experimentally observed in biological neural systems [36,37,38,39]. The changes of the synapse efficacies depend on the relative timing between the postsynaptic and presynaptic spikes. 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 [40]. The values of the parameters used were $A_+ = 0.005$, $A_- = 1.05\ A_+$, and $\tau_+ = \tau_- = 50$ ms. Following [9], we implemented directional damping for the synapse efficacies. The synapse efficacies w, which were variable due to STDP, were limited to the interval $[0, w_{max}]$, where w_{max} could be either positive or negative, and was a genetically determined maximum (in absolute value) efficacy.

4 The Agent-Controller Interface

In interfacing a spiking neural controller with an embodied agent, a conversion of the analog input to binary spikes and then of spikes to an analog output has to be performed. Following [9], 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, corresponded 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. 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

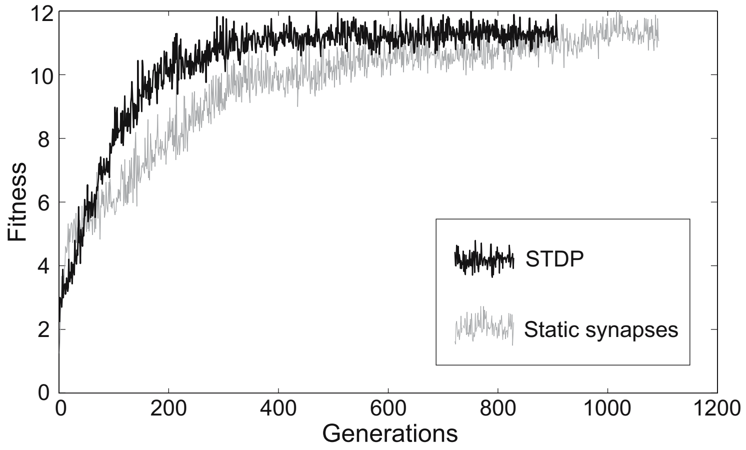

Fig. 2. Best fitness of the networks from a population over generations

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 simplifies the selection of the network's parameters and the analysis of the network's behavior.

5 The Evolutionary Algorithm

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). The genome directly encoded these values for the 6875 synapses. We used a standard evolutionary algorithm, with a population of 80 individuals, truncation selection (the top 25 % individuals reproduced) and elitism. 10 % of the offspring resulted from mating with single cut crossover. Mutation was applied uniformly to all genes.

6 Results

Networks with both static and plastic synapses evolved to solve the required task, with the fitness of the best individuals reaching a plateau at about 11.3, very close to the maximum possible of 12 (see Fig. 2). Plastic networks evolved faster, in terms of generations, than networks with static synapses. However, the simulation of plastic networks required a higher computational effort. Only one evolution has been performed for each case (STDP, and respectively static synapses), because of the required computing time (a few weeks on a standard PC).

7 Behavioral Analysis

The agents controlled by the evolved networks seek the closest ball, push it for a while, then release it and seek another ball. They use visual information for seeking the balls: when no ball is pushed, and a ball enters the visual field, the agents go towards it. If there is no ball in the visual field, the agents rotate in circles in order to visually scan the environment. When they push a ball, they keep the link that unite their two body-circles extended, in order to have most of their visual field not occupied by the pushed ball. However, the link is not extended to maximum, for not letting the ball pass trough it. The agents also move circularly when pushing a ball, in order to seek other balls in the environment. When another ball enters the visual field, they go towards it while still pushing the first ball, and release it only when they are close to the new ball. Release is performed by extending the link. Again, they use visual information in order to seek new balls while pushing one. For example, if a single ball is placed in the environment, they keep pushing it in circles indefinitely, without releasing it. The behavior of the evolved agents is thus relatively complex, from an external observer perspective, requiring the composition of three sub-behaviors: seek, push, release. Although the agents may come more than once to push a particular object, their strategy leads them to alternatively push, in most cases, all objects in their environment. A movie displaying the behavior of an evolved agent is available online at `http://coneural.org/reports/object_pushing/object_pushing.avi`.

The balls that the agents push have the same density, and thus larger balls are heavier. Since the environment obeys a quasistatic physics, velocity is inversely proportional to mass, for a given force. In order to optimize their behavior for solving the task, the agents have to push the balls as hard as possible, and, for constant (maximum) forces, they have to push larger (heavier) balls longer periods of time than the smaller ones, to move them on similar distances during a limited time interval. It is interesting that this behavior — pushing for longer periods the larger balls — actually emerges during evolution. On average, the balls from the set encountered during evolution are pushed on the same distances. This is illustrated in Fig. 3 a). We tried to uncover the mechanisms that determine this behavior by subjecting the evolved agents to several “psychological” experiments, where the reality they were accustomed to (through evolution) was modified. More precisely, we modified systematically the radiuses and/or the densities of the balls. It can be seen (Fig. 3) that the average distance on which a ball is pushed does not depend exclusively on its characteristics, such as radius or mass, but on the characteristics of the whole set of balls. The average distances do not depend exclusively on the geometry of the environment, but also on the interaction of the agent with the balls. There is no particular parameter on the basis of which the agent estimates for how long it should push a ball, but the constance of the average distances emerges as a property of the complex dynamical system constituted by the agent, its environment, and the neural network, a property which is found and selected by the evolution. However, an external observer could argue that the evolved

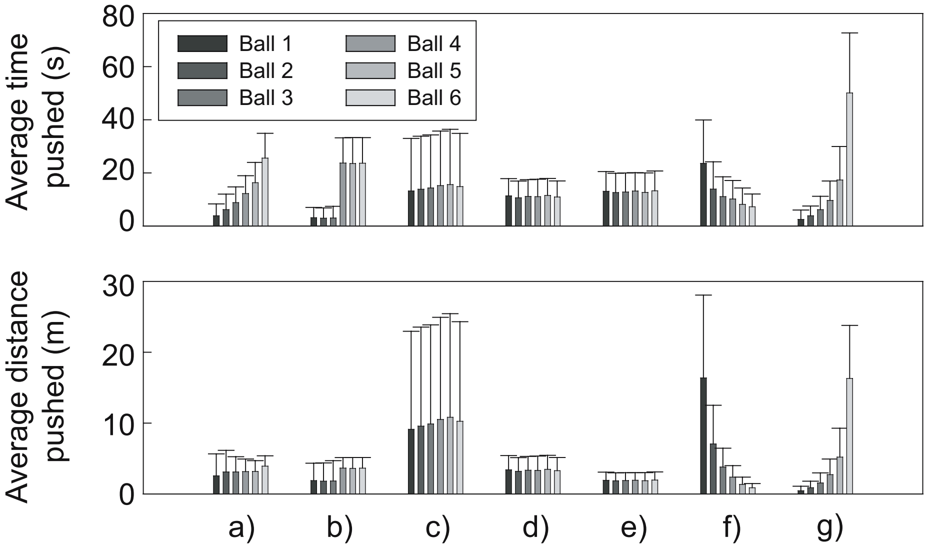

Fig. 3. Average (over 500 trials) and standard deviation of the average time and of the average distance a ball is pushed during a trial. a) The balls are like during the evolution, having radii varying linearly between r_1 and r_2. b) First three balls have radius r_1 and the other three have radius r_2. c)-e) The balls have equal radii: c) r_1; d) $r_a = (r_1 + r_2)/2$; e) r_2. f) The balls have equal radii r_a but variable masses (corresponding to radii varying linearly between r_1 and r_2, at default density). g) The balls have equal masses (corresponding to a radius r_a at default density) but radii varying linearly between r_1 and r_2. The relatively high variability of the displayed quantities is due by the variability of initial positions of the balls in the environment.

agents discriminate relative object size, by pushing larger objects for longer periods.

8 The Role of Plasticity

The evolved networks featuring STDP had a performance similar to the one of networks with static synapses, but evolved slightly faster. The improvement in evolution speed observed in plastic networks could be explained by a number of factors, including random exploration (smoothing) of the fitness landscape in the surroundings of individuals [41]. It is interesting to investigate whether plasticity also has an active role in determining the networks' performance. We have thus freezed the plasticity in networks evolved with STDP, either completely or after 1 s of activity (i.e., the first 1% of the duration of a trial). The results are presented in Fig. 4: freezing completely the plasticity leads to an important (80%) loss of performance, while freezing it after a short time that allows plasticity to act reduces performance only with about 33%. This means that most of the role of STDP in our evolved networks consists in tuning synapses to quasi-

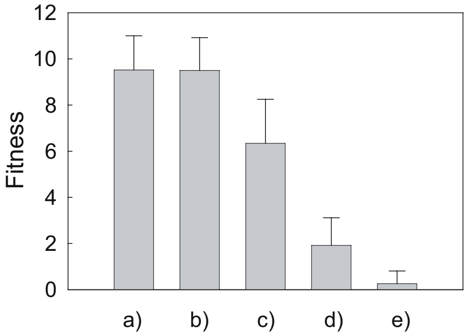

Fig. 4. Average fitness (over 500 trials) and standard deviation: a) Best network evolved with static synapses. b) Best network evolved with STDP. c) Best network evolved with STDP; plasticity is freezed after 1 s from the beginning of each trial. d) Best network evolved with STDP; plasticity is completely freezed. e) Random networks, not subjected to evolution.

stable, adaptive strengths, rather than contributing actively to the network's dynamics.

9 Conclusion

We have successfully evolved fully connected spiking neural networks consisting of 125 neurons and 6875 synapses, that allow a simple agent to alternatively seek, push and then release the 6 balls in its environment. This is one of the largest spiking neural network successfully evolved for the control of an embodied agent reported in the literature. 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. The evolved agents display interesting, minimally-cognitive behavior [42,43], by switching as a function of context between the three sub-behaviors (seek, push, release) and by being able to discriminate relative object size. We evolved networks with either static synapses or synapses featuring STDP. Plasticity proved to play a minor role in the dynamics of the evolved networks, contributing mostly to speeding up the evolutionary process and to tuning the synapses at the beginning of the networks' activity.

The evolved drive could be used in future experiments to bootstrap the development of more complex behaviors, by using, for example, reinforcement learning [32] to shape further the agent's behavior.

Acknowledgements

This work was partly supported by Arxia SRL and by a grant of the Romanian Government (MEdC-ANCS). We thank to Raul Mureşan for providing the code of the Neocortex spiking neural network simulator and for useful feedback.

References

1. Florian, R.V.: Autonomous artificial intelligent agents. Technical Report Coneural-03-01, Center for Cognitive and Neural Studies, Cluj, Romania (2003)
2. Maas, W., Bishop, C.M., eds.: Pulsed neural networks. MIT Press, Cambridge, MA (1999)
3. Gerstner, W., Kistler, W.M.: Spiking neuron models. Cambridge University Press, Cambridge, UK (2002)
4. Florian, R.V.: Biologically inspired neural networks for the control of embodied agents. Technical Report Coneural-03-03, Center for Cognitive and Neural Studies, Cluj, Romania (2003)
5. DasGupta, B., Schnitger, G.: Analog versus discrete neural networks. Neural Computation **8** (1996) 805–818
6. Maass, W., Schnitger, G., Sontag, E.D.: A comparison of the computational power of sigmoid and boolean threshold circuits. In Roychowdhury, V.P., Siu, K., Orlitsky, A., eds.: Theoretical Advances in Neural Computation and Learning. Kluwer Academic Publishers (1994) 127–151
7. Maas, W.: Networks of spiking neurons: the third generation of neural network models. Neural Networks **10** (1997) 1659–1671
8. Floreano, D., Mattiussi, C.: Evolution of spiking neural controllers for autonomous vision-based robots. In Gomi, T., ed.: Evolutionary Robotics IV. Springer-Verlag, Berlin (2001)
9. Di Paolo, E.A.: Spike timing dependent plasticity for evolved robots. Adaptive Behavior **10** (2002) 243–263
10. Saggie, K., Keinan, A., Ruppin, E.: Solving a delayed response task with spiking and McCulloch-Pitts agents. In: Advances in Artificial Life: 7th European Conference, ECAL 2003 Dortmund, Germany, September 14-17, 2003. Volume 2801 of Lecture Notes in Computer Science. Springer, Berlin / Heidelberg (2003)
11. Saggie-Wexler, K., Keinan, A., Ruppin, E.: Neural processing of counting in evolved spiking and mcculloch-pitts agents. Artificial Life **12**(1) (2005) 1–16
12. Ruppin, E.: Evolutionary embodied agents: A neuroscience perspective. Nature Reviews Neuroscience **3** (2002) 132–142
13. Floreano, D., Schoeni, N., Caprari, G., Blynel, J.: Evolutionary bits'n'spikes. In Standish, R.K., Bedau, M.A., Abbass, H.A., eds.: Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life. MIT Press, Boston, MA (2002)
14. Floreano, D., Zufferey, J.C., Mattiussi, C.: Evolving spiking neurons from wheels to wings. Proceedings of the 3rd International Symposium on Human and Artificial Intelligence Systems, Fukui, Japan (2002)
15. French, R.L.B., Damper, R.I.: Evolving a nervous system of spiking neurons for a behaving robot. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2001), San Francisco, CA (2001) 1099–1106
16. French, R.L.B., Damper, R.I.: Evolution of a circuit of spiking neurons for phototaxis in a Braitenberg vehicle. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., Meyer, J.A., eds.: From animals to animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge, MA (2002) 335–344
17. Damper, R.I., French, R.L.B.: Evolving spiking neuron controllers for phototaxis and phonotaxis. In Raidl, G., ed.: Applications of Evolutionary Computation, EvoWorkshops 2003. Volume 2611 of Lecture Notes in Computer Science. Springer, Berlin (2003) 616–625

18. Di Paolo, E.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 (2002)
19. Di Paolo, E.A.: Evolving spike-timing dependent plasticity for single-trial learning in robots. Philosophical Transactions of the Royal Society A **361** (2003) 2299–2319
20. Roggen, D., Hofmann, S., Thoma, Y., Floreano, D.: Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot. In: 2003 NASA/DoD Conference on Evolvable Hardware (EH'03). (2003) 199
21. Van Leeuwen, M., Vreeken, J., Koopman, A.: Evolving vision-based navigation on wheeled robots. Institute for Information and Computing Sciences, Utrecht University (2003)
22. Katada, Y., Ohkura, K., Ueda, K.: 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 (2003) 318–323
23. Katada, Y., Ohkura, K., Ueda, K.: An approach to evolutionary robotics using a genetic algorithm with a variable mutation rate strategy. In: Proceedings of The 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII). (2004) 952–961
24. Soula, H., Beslon, G., J.Favrel: Evolving spiking neural nets to control an animat. In: Proceedings of International Conference of Artificial Neural Networks and Genetic Algorithm 2003 - Roanne, France. (2003)
25. Hagras, H., Pounds-Cornish, A., Colley, M., Callaghan, V., Clarke, G.: Evolving spiking neural network controllers for autonomous robots. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, USA. (2004)
26. Federici, D.: A regenerating spiking neural network. Neural Networks **18**(5–6) (2005) 746–754
27. Federici, D.: Evolving developing spiking neural networks. In: Proceedings of CEC 2005 - IEEE Congress on Evolutionary Computation. (2005)
28. Damper, R., Scutt, T.: Biologically-motivated neural learning in situated systems. In: Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98). (1998)
29. Damper, R., French, R.L.B., Scutt, T.: Arbib: An autonomous robot based on inspirations from biology. Robotics and Autonomous Systems **31**(4) (2000) 247–274
30. Soula, H., Alwan, A., Beslon, G.: Obstacle avoidance learning in a spiking neural network. In: Last Minute Results of Simulation of Adaptive Behavior, Los Angeles, CA (2004)
31. Soula, H., Alwan, A., Beslon, G.: Learning at the edge of chaos: Temporal coupling of spiking neuron controller of autonomous robotic. In: Proceedings of AAAI Spring Symposia on Developmental Robotics, Stanford, CA (2005)
32. Florian, R.V.: A reinforcement learning algorithm for spiking neural networks. In Zaharie, D., Petcu, D., Negru, V., Jebelean, T., Ciobanu, G., Cicortaş, A., Abraham, A., Paprzycki, M., eds.: Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2005), IEEE Computer Society (2005) 299–306
33. Florian, R.V.: Thyrix: A simulator for articulated agents capable of manipulating objects. Technical Report Coneural-03-02, Center for Cognitive and Neural Studies, Cluj, Romania (2003)

34. Pfeifer, R., Scheier, C.: Understanding intelligence. MIT Press, Cambridge, MA (1999)
35. Mureşan, R.C., Ignat, I.: The "Neocortex" neural simulator: A modern design. International Conference on Intelligent Engineering Systems, September 19-21, 2004, Cluj-Napoca, Romania (2004)
36. Markram, H., Lübke, J., Frotscher, M., Sakmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science **275**(5297) (1997) 213–215
37. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of Neuroscience **18**(24) (1998) 10464–10472
38. Bi, G.Q.: Spatiotemporal specificity of synaptic plasticity: cellular rules and mechanisms. Biological Cybernetics **87** (2002) 319–332
39. Dan, Y., Poo, M.M.: Spike timing-dependent plasticity of neural circuits. Neuron **44** (2004) 23–30
40. Song, S., Miller, K.D., Abbott, L.F.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neuroscience **3** (2000) 919–926
41. Turney, P.: Myths and legends of the Baldwin effect. Proceedings of the Workshop on Evolutionary Computing and Machine Learning at the 13th International Conference on Machine Learning (ICML-96), Bari, Italy (1996) 135–142
42. Beer, R.: Toward the evolution of dynamical neural networks for minimally cognitive behavior. In Maes, P., Mataric, M., Meyer, J., Pollack, J., Wilson, S., eds.: From animals to animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior. Volume 421–429. MIT Press, Cambridge, MA (1996)
43. Slocum, A.C., Downey, D.C., Beer, R.D.: Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In Meyer, J.A., Berthoz, A., Floreano, D., Roitblat, H.L., Wilson, S.W., eds.: From animals to animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge, MA (2000) 430–439