

Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots

Dario Floreano and Claudio Mattiussi

Evolutionary & Adaptive Systems, Institute of Robotics
Swiss Federal Institute of Technology, CH-1015 Lausanne (EPFL)
Dario.Floreano@epfl.ch, Claudio.Mattiussi@epfl.ch

Abstract. We describe a set of preliminary experiments to evolve spiking neural controllers for a vision-based mobile robot. All the evolutionary experiments are carried out on physical robots without human intervention. After discussing how to implement and interface these neurons with a physical robot, we show that evolution finds relatively quickly functional spiking controllers capable of navigating in irregularly textured environments without hitting obstacles using a very simple genetic encoding and fitness function. Neuroethological analysis of the network activity let us understand the functioning of evolved controllers and tell the relative importance of single neurons independently of their observed firing rate. Finally, a number of systematic lesion experiments indicate that evolved spiking controllers are very robust to synaptic strength decay that typically occurs in hardware implementations of spiking circuits.

1 Spiking Neural Circuits

The great majority of biological neurons communicate by sending pulses along the axons to other neurons. A pulse is a small current charge that occurs when the voltage potential across the membrane of a neuron exceeds its threshold. The pulse is also known as “spike” to indicate its short and transient nature. After emitting a spike, a neuron needs some time to reestablish its electrochemical equilibrium and therefore cannot immediately emit a new spike, no matter how strong its excitatory input is. A typical neuron in the cortex “fires” approximately 10 spikes per second during resting conditions and can emit up to 300 spikes per second in operating conditions. Other neurons can fire more frequently (for example 500 spikes per seconds) clustered in short periods of time (“bursting neurons”).

In the field of artificial neural networks we find two different classes of models that differ with respect to the interpretation of the role of spikes. Connectionist models of neural networks [19], by far the most widespread models, assume that what matters in the communication among neurons is the *firing rate* of a neuron. The firing rate is the average quantity of spikes emitted by the neuron over a relatively long time window (for example, over 100 ms). This quantity is represented by the activation level of the neuron. For example, a neuron characterized by a sigmoid activation function, such as the logistic function $f(x) = 1/1 + \exp(-x)$,

that gives an output of 0.5 would be equivalent to a spiking neuron that emits approximately half its maximum rate of spikes (150 out of 300 spikes per second, e.g.). Models of pulsed neural networks [14] instead assume that the *firing time*, that is the precise time of emission of a single spike, can transmit important information for the post-synaptic neuron [23]. Therefore, these models use more complex activation functions that simulate the emission and reception of spikes on a very fine timescale.

Spiking circuits have at least two properties that make them interesting candidates for adaptive control of autonomous behavioral robots:

- The intrinsic time-dependent dynamics of neuron activation could detect and exploit more easily (e.g., with simpler circuits or with higher reliability) temporal patterns of sensory-motor events than connectionist neural networks.
- Since the physics of circuits of sub-threshold transistors (i.e., characterized by gate-to-source voltage differences below their threshold voltage) implemented with analog Very Large Scale Integration technology [15] match the properties of spiking neurons, it possible to implement large networks of spiking neurons in tiny and low-power chips [9].

Designing circuits of spiking neurons with a given functionality is still a challenging task and the most successful results obtained so far have focused on the first stages of sensory processing and on simple motor control. For example, Indiveri et al. [8] have developed neuromorphic vision circuits that emulate the interconnections among the neurons in the early layers of an artificial retina in order to extract motion information and a simple form of attentive selection of visual stimuli. These vision circuits have been interfaced with a Koala robot and their output has been used to drive the wheels of the robot in order to follow lines [10]. In another line of work, Lewis et al. have developed an analog VLSI circuit with four spiking neurons capable of controlling a robotic leg and adapting the motor commands using sensory feedback [13]. This neuromorphic circuit consumes less than 1 microwatt of power and takes less than 0.4 square millimeters of chip area.

Despite these interesting implementations, there are not yet methods for developing complex spiking circuits that could display minimally-cognitive functions or learn behavioral abilities through autonomous interaction with the environment. Furthermore, the potentially complex dynamics that a spiking circuit with feedback loops can display allows several alternative functioning modalities. For example, a spiking circuit could use inhibitory neurons to modulate the overall excitation of the network and/or selectively interact with other neurons to inhibit specific actions. Also, the same circuit could use firing rate instead of firing time as the preferred functioning modality (this concept will be discussed in more detail later on in this paper).

Artificial evolution is therefore an interesting method to discover spiking circuits that autonomously develop desired behavioral abilities for robots without imposing constraints on their architecture and functioning modality. In this paper, we describe some initial explorations in the evolution of spiking circuits for

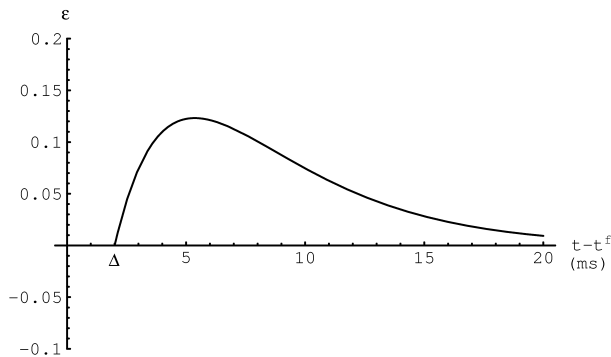


Fig. 1. The function describing the contribution ϵ of a spike from a presynaptic neuron emitted at time t^f . The contribution of the spike begins after some delay Δ (2 ms) due to the traveling time of the spike and eventually decreases its effect as time t flows from the firing time t^f . The synapse time constant τ_s is set to 10 ms and the membrane time constant τ_m is set to 4 ms.

a task of vision-based navigation using a Khepera robot with a linear CMOS vision system. We will then analyze the evolved spiking circuit and discuss the most important lessons learned from these experiments. Finally, we will describe some ideas for future developments in this emerging area of research.

2 The Spike Response Model

The state of a spiking neuron is described by the voltage difference across its membrane, also known as membrane potential v . Incoming spikes can increase or decrease the membrane potential. The neuron emits a spike when the total amount of excitation induced by incoming excitatory and inhibitory spikes exceeds its firing threshold θ . After firing, the membrane potential of the neuron resets its state to a low negative voltage during which it cannot emit a new spike, and gradually returns to its resting potential. This recharging period is called the *refractory period*.

There are several models of spiking neurons that account for these properties with various degrees of detail. In the experiments described in this paper, we have chosen the *Spike Response Model* developed by Gerstner [5]. It has been shown that several other models of spiking neurons, such as the class of Integrate-and-Fire neurons (where the membrane potential of the neuron is immediately reset to its resting value after a spike), represent special cases of the Spike Response Model [6].

In this model, the effect ϵ of an incoming spike on the neuron membrane is a function of the difference $s = t - t^f$ between current time t and the time when the spike was emitted t^f . The properties of the function are determined by a) the delay Δ between the generation of a spike at the pre-synaptic neuron

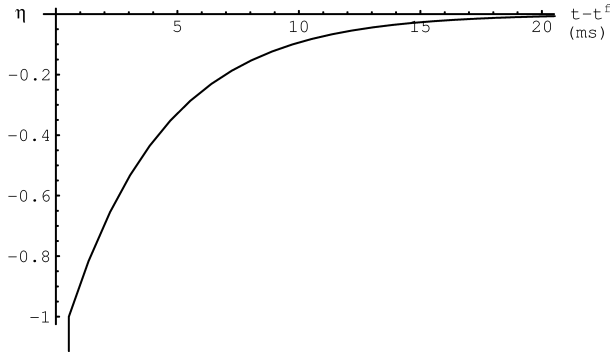


Fig. 2. The function describing the refractory period of the neuron after emission of a spike. Notice that at time 0, the membrane potential is set to a large negative value (larger than 1.0 shown on the y-axis) to prevent another spike. The refractory period is a function of time from last spike and its shape is given by the membrane time constant τ_m (here set to 4).

and the time of arrival at the synapse, *b*) a synaptic time constant τ_s , and *c*) a membrane time constant τ_m . The idea is that a spike emitted by a pre-synaptic neuron takes some time to travel along the axon and, once it has reached the synapse, its contribution on the membrane potential is higher as soon as it arrives, but gradually fades as time goes. A possible function $\epsilon(s)$ describing this behavior, shown in figure 1, is [6]

$$\epsilon(s) = \begin{cases} \exp[-(s - \Delta)/\tau_m](1 - \exp[-(s - \Delta)/\tau_s]) & : s \geq \Delta \\ 0 & : s < \Delta \end{cases} \quad (1)$$

In what follows, we will discretize time in small steps of 1 ms each and consider the effects of $\epsilon(s)$ only within a time window of 20 ms. At each time step t , the synaptic contribution to the membrane potential is the sum of all the spikes arriving at the synapse over a 20 ms time window weighted by the corresponding value of the function $\epsilon(s)$ at each time step. Therefore, if at time step t_1 the pre-synaptic neuron has emitted three spikes, respectively at $s = 4, s = 7, s = 15$, the total contribution E_{t_1} computed using equation 1 is

$$E_{t_1} = \epsilon(4) + \epsilon(7) + \epsilon(15) = 0.109945 + 0.112731 + 0.028207 = 0.250883 \quad (2)$$

At the next time step t_2 , assuming that there are no new spikes, we simply shift all the previous firing times of each spike by one position (respectively at $s = 5, s = 8, s = 16$) and obtain a new (lower) value $E_{t_2} = 0.2458538$.

Assuming that at time t the neuron membrane is at resting potential v_r , a spike is emitted if $E_t \geq \theta$, that is if the total contribution of the synapse is larger than the threshold. Once the neuron has emitted a spike, its membrane potential is set to a very low value to prevent an immediate second spike and then

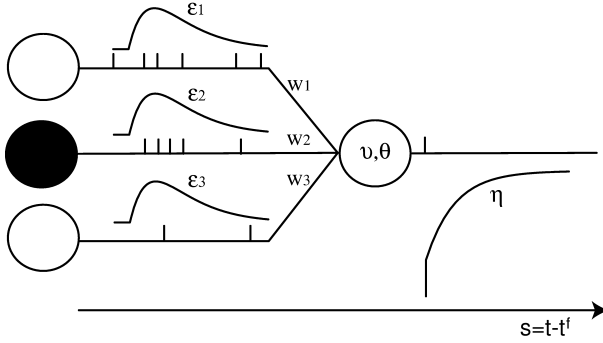


Fig. 3. A network of 4 neurons (white circle= excitatory, black circle=inhibitory) captured at a given time t with the functions ϵ and η superimposed. The postsynaptic neuron has just emitted a spike because the membrane potential v has exceeded its threshold θ .

it gradually recovers to its resting potential. The speed of recovery depends on the membrane time constant τ_m (that we have already seen in the computation of the synaptic contribution above). A possible function $\eta(s)$, shown in figure 2, describing the refractory function [6] is

$$\eta(s) = -\exp[-s/\tau_m] \quad (3)$$

We can now put together the equations describing synaptic contributions and the refractory period to describe the dynamics of a neuron that has several synaptic connections from excitatory and inhibitory neurons (figure 3). Each synaptic connection has a weight w_j whose sign, in this example, is given by the pre-synaptic neuron (positive if the neuron is excitatory, negative if the neuron is inhibitory). The membrane potential of a neuron i at time t is given by

$$v_i(t) = \sum_j w_j^t \sum_f \epsilon_j(s_j) + \sum_f \eta_i(s_i) \quad (4)$$

where $s_n = t - t_n^f$ is the difference between the time t and the time of firing t^f of neuron n . If the membrane potential $v_i(t)$ is equal or larger than the neuron threshold θ_i , the neuron emits a spike and η_i takes a very low value that prevents an immediate new spike. After that, η_i is computed according to equation 3.

It should be noticed that each synapse may have a different synaptic time constant τ_s and a different time delay Δ , which would affect the shape of its function $\epsilon(s)$, in addition of course to a different weight w . Similarly, each neuron may have a different membrane time constant τ_m and a different threshold θ , which would affect the contribution of its incoming synapses and its own spiking time.

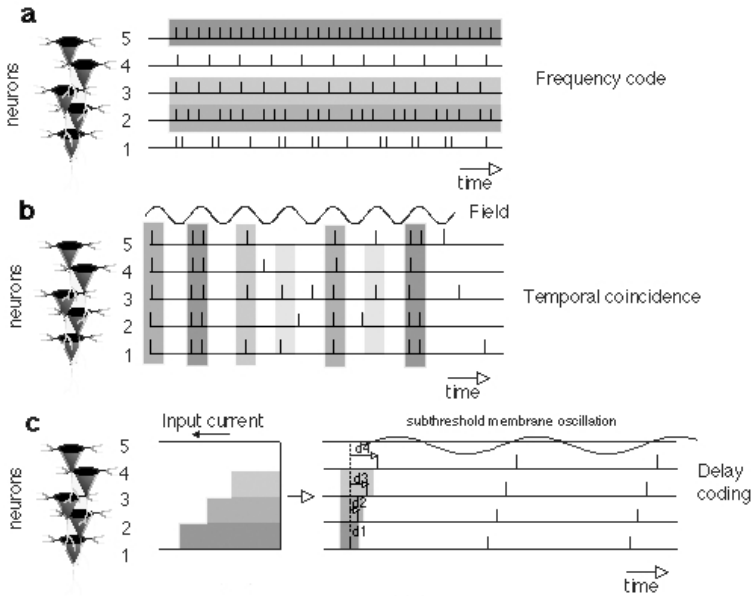


Fig. 4. Some models for encoding sensory information in spiking neurons (from [17]). Fictive spike trains recorded from five imaginary neurons. The different stimulus intensities (represented by gray scale) are converted to different spike sequences. (a) In the frequency code hypothesis [20] neurons generate different frequency of spike trains as a response to different stimulus intensities. (b) In the temporal coincidence hypothesis [22] spike occurrences are modulated by local field oscillation (γ). Tighter coincidence of spikes recorded from different neurons represent higher stimulus intensity. (c) In the delay coding hypothesis [7] the input current is converted to the spike delay. Neuron 1 which was stimulated stronger reached the threshold earlier and initiated a spike sooner than neurons stimulated less. Different delays of the spikes (d_2 - d_4) represent relative intensities of the different stimulus.

3 Interfacing Spiking Neurons with a Robot

Interfacing a neural network of sigmoid neurons to a robot is relatively straightforward. At regular intervals (100 ms, e.g.), the values read from the sensors of the robot to set the activation values of the corresponding input units. Similarly, the activation values of the output units are read at regular intervals to set the control parameters of the robot actuators, such as the speeds of the wheels.

In a spiking neural network, a single spike is a binary event that can encode only the *presence* or absence of a stimulus. Figure 4 shows three ways to map the *intensity* of sensory information into spiking neurons. A classic method (a) consists of mapping the stimulus intensity to the firing rate of the neuron. This method is based on the hypothesis that a neuron increases its firing rate to indicate stronger stimulation. For example, an often-cited result [2] shows that

the firing rate of a stretch receptor in the frog is a monotonically increasing function of the strength of the stimulation and saturates near the maximum firing rate of the neuron.¹ Another method (b) consists of encoding the sensory stimulation across several neurons and mapping the intensity of the stimulation into the number of neurons that spike at the same time. This method is based on the hypothesis that the brain represents meaningful information by synchronizing spiking activities across several neurons [21] and has been supported by measurements in the visual and temporal cortex of monkeys [1,22]. A recently suggested method (c) consists of encoding the strength of the stimulation in the firing delay of the neuron. The underlying hypothesis is that neurons that receive stronger stimulation fire earlier than neurons receiving weaker stimulation and has been supported by measurements in the olfactory neurons [7].

In the experiments reported in this paper, we have used a stochastic version of the firing rate method. In other words, the intensity of the stimulation is represented by the probability that the neuron emits a spike in a time interval. When repetitively measured over relatively long periods of time with respect to the time interval for the same stimulation intensity, the observed firing rate is proportional to the strength of the stimulation.

The transformation of spikes into motor commands presents similar issues. A simple model assumes that muscle stretching is a monotonically increasing function of the firing rate of one or more motor neurons over a short time window (from 20 to 60 ms) [11]. In these experiments, we have taken the firing rate of the motor neurons measured over 20 ms as speed commands to the wheels of the robot.

Another issue is the synchronization between the temporal dynamics of the spiking network and those of the robotic hardware. Biological neurons operate on the millisecond time scale and electronic spiking neurons can operate even faster, but current robotics technology may operate on a slower time scale. For example, in a Khepera robot equipped with a linear camera the visual information can be accessed every 25 ms at optimal lighting conditions, but in order to allow for lower illumination it is safer to access it every 50 or 100 ms. Furthermore, since in our experiments the network of spiking neurons runs on the workstation, we must allow for the time taken by the sensory signals to be transmitted through the serial connection and the time taken to send the motor commands to the wheels of the robot.

One possible solution is to set the elementary unit of time of the spiking neurons to the maximum time interval necessary to read the sensors and set the speeds of the wheels. Another solution is to let the neurons operate at their “natural” time scale (for example, with an update rate of 1 ms) while accessing the robotic interface at regular longer intervals. In this case, the neurons can

¹ More detailed studies indicate that this function is best described by a power function of the general form $R = KS^n + C$ where R is the observed firing rate, K is a constant of proportionality, and C is a constant given by the spontaneous firing rate of a neuron in the absence of stimulation (in our experiments, the spontaneous activation is 0) [16].

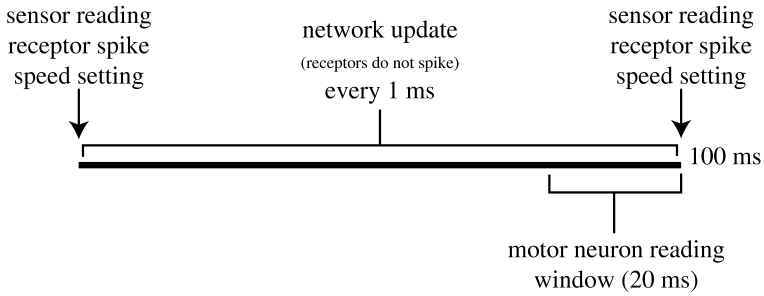


Fig. 5. Time management diagram (see text).

change their states faster than the sensors and actuators of the robot and may be left free to develop internal temporal dynamics.

In these experiments, we have chosen the second option (figure 5). Every 100 ms the sensors of the robot are pre-processed, scaled in the range $[0, 1]$, and used to set the probability of emitting a single spike at that precise time step, while the wheel speeds are set at the end of the 100 ms interval using the firing rate of the motor neurons measured during the previous 20 ms. During the 100 ms interval until the next sensory-motor access, all the neurons in the network are updated every 1 ms (except for the sensory neurons). In the meanwhile the robot moves using constant speed values (the real speed is given by a PID controller). We have included time functions in the code that make sure that the neural network is updated exactly once every millisecond. These functions allow us to keep the system synchronized when other processes are running in the background, when we activate new routines during analysis, and when we use different workstations.²

4 Evolution of Vision-Based Navigation

The experiments described in this paper are a preliminary exploration into the evolution of spiking controllers and a comparison with evolution of connectionist sigmoidal neurons using the same genetic representation and experimental settings.

We have attempted to evolve vision-based controllers for a navigation task in a rectangular arena with textured walls (figure 6). The walls are filled with black and white vertical stripes. The width and spacing of the stripes are random (within the interval $[0.5, 5]$ cm). Given the visual angle of the robot camera and the size of the arena, some areas display higher stripe frequency than other areas.

² If we had not such functions, a Pentium III at 700 MHz running Linux would update the networks used in our experiments between 1000 and 1200 times per 100 ms, depending on the number of active background processes.

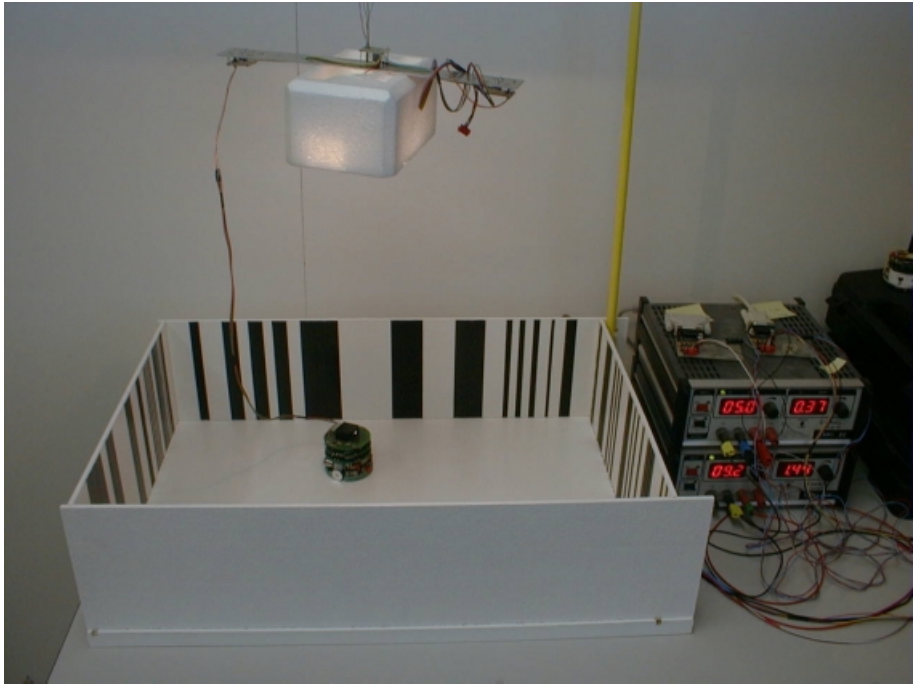


Fig. 6. A Khepera robot equipped with a linear camera is positioned in an arena with black and white vertical stripes of random size painted on the walls at irregular intervals. The arena is lit from above in order to let the evolutionary experiments continue at night. The robot is connected to a workstation through rotating contacts that provide serial data transmission and power supply. The spiking networks and genetic operators run on the workstation. The robot communicates with the workstation every 100 ms.

We have used a Khepera robot equipped with a linear camera module (figure 7). The vision system is composed of a linear array of 64 photoreceptors (left hole) spanning a visual angle of 36 deg and of a light sensor (right hole) used to adjust the sensitivity of the receptors to the global illumination level. Each photoreceptor returns a value between 0 (black) and 255 (white). Given the spacing of the stripes on the wall, we read only 16 photoreceptors equally spaced on the array. These values are convolved with a Laplace filter spanning three adjacent photoreceptors in order to detect contrast. Finally, the convolved image is rectified by taking the absolute values and scaling them in the range $[0, 1]$. The resulting 16 values represent the probabilities of emitting a single spike for each corresponding neural receptor at that precise instant only. The camera values are read every 100 ms (see also figure 5).

In these experiments we use a network of predefined size where the sign of the synaptic weight is given by the sign (excitatory or inhibitory) of the presynaptic neuron. In other words, all the connection coming from an excitatory

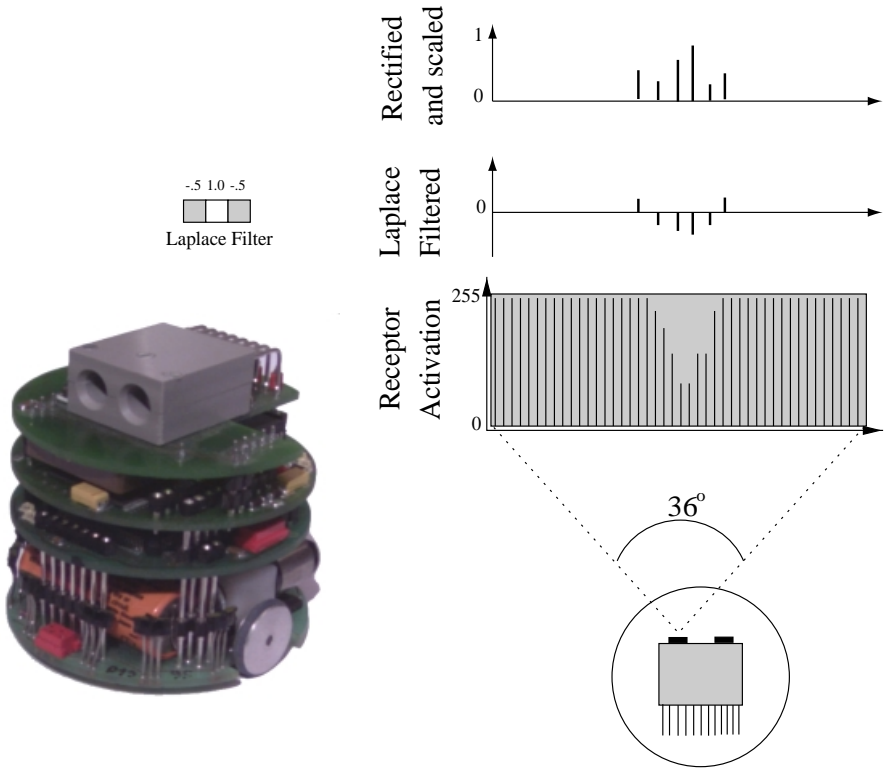


Fig. 7. The Khepera robot equipped with a linear vision system composed of 64 photoreceptors. Only 16 photoreceptors are read every 100 ms and filtered through a Laplace filter in order to detect areas of contrast. The filtered values are transformed into positive values and scaled in the range [0, 1]. These values represent the probability of emitting a spike for each corresponding neural receptor.

(inhibitory) neuron are positive (negative). Sensory receptors are always excitatory. We genetically encode and evolve only the signs of the neurons and the pattern of connectivity among neurons and between neurons and sensory receptors. The network consists of 10 fully-connected neurons, each connected to all sensory receptors (figure 8). There are 18 sensory receptors: 16 transmit vision signals and 2 transmit the error between the desired speeds of the wheels and the speed measured using on-board optical encoders.³

Four neurons are used to set the speeds of the wheels in push-pull mode. Two neurons are assigned to each wheel, one neuron setting the amount of forward speed and the other setting the amount of backward speed. The actual speed of the wheel is the algebraic sum of the two speeds. This value is mapped into a

³ The error is transformed in positive values in the range [0, 1] and used as a probability to emit a spike.

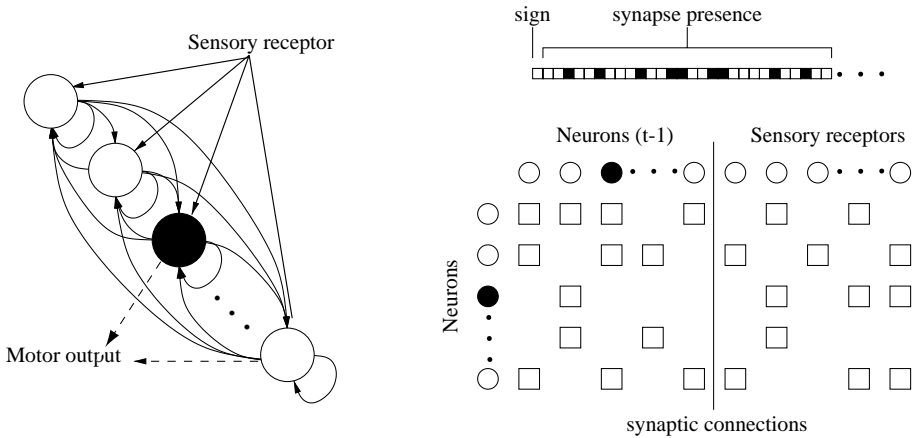


Fig. 8. Architecture of the neural network used in the experiments (only a few neurons and connections are shown) and genetic representation of one neuron. The network has 10 neurons that can be fully connected among each other. In addition, each neuron is connected to 18 spiking receptors (16 for vision and 2 for the error between speed commands and actual speed of the two wheels). Four neurons are used to set the speeds of the two wheels in push-pull mode. *Left:* A conventional representation showing the network architecture. *Right:* The same network unfolded in time (neurons as circles, synaptic connections as squares). The neurons on the column receive signals from connected neurons and receptors shown on the top row. The first part of the row includes the same neurons at the previous time step to show the connections among neurons. Sensory receptors do not have interconnections. The signs of the neurons (white = excitatory, black = inhibitory) and their connectivity is encoded in the genetic string and evolved.

maximum speed of 80 mm/s. However, since the neurons can fire at maximum once every 2 ms (because of the very low negative value taken by the neuron after emitting a spike), in practice the maximum speed is 40 mm/s.

A binary genetic string encodes only the sign of each neuron and the presence of a synaptic connection. The string is composed of n blocks, one for each of the n neurons in the network. The first bit of the block encodes the sign of the neuron and the remaining bits encode the presence/absence of a connection from the n neurons and from the s receptors. Therefore, the total length l of the string is $l = n(1 + n + s)$. In our experiments $l = 10(1 + 10 + 18) = 290$. The other parameters of the neurons and synapses (see section 2) are set as follows: $\theta = 0.1$, $\tau_m = 4$, $\tau_s = 10$, $\Delta = 2$, and $w = 1$ for all neurons and all synapses in the network. The shape of the synaptic and refractory functions for these parameters are shown in figures 1 and 2, respectively. Given these values, only the most recent 20 spikes arriving at each synapse are taken into account for computing the total synaptic contribution according to equation 1. In addition, some noise is added to the refractory period by multiplying the

value returned by the refractory function (equation 3) at each time step by a uniformly random value in the range $[0, 1]$. Preliminary experiments showed that without this added noise, the networks go very quickly into locked oscillations for a very large number of connectivity patterns and parameters.

Each individual of the population is decoded and tested on the robot *two times* for 40 seconds each (400 sensory-motor steps). The fitness function Φ is the sum of the speeds of the two wheels v_{left} and v_{right} measured at every time step t (100 ms) only if both wheels rotate in the forward direction averaged over all T time steps available ($T = 800$)

$$\Phi = \frac{1}{T} \sum_t^T (v_{left}^t + v_{right}^t) \quad (5)$$

If v_{left} or v_{right} are less than 0 (backward rotation), $\Phi^t = 0$. This fitness function selects individuals for the ability to go as straight as possible while avoiding the walls because it takes a few seconds to travel across the arena and if the robot is stuck against a wall, the wheels can hardly rotate due to the friction on the floor. The fitness function does not use the active infrared sensors available on the robot to judge the distance from the walls because the response profile of these sensors varies depending on the reflection properties of the walls (black stripes reflect approximately 40% less infrared light than white stripes) and on the spectrum component of ambient illumination.

A population of 60 individuals is evolved using rank-based truncated selection, one-point crossover, bit mutation, and elitism [18]. After ranking the individuals according to their measured fitness values, the top 15 individuals produce 4 copies each to create a new population of the same size and are randomly paired for crossover. One-point crossover is applied to each pair with probability 0.1 and each individual is then mutated by switching the value of a bit with probability 0.05 per bit (that is, on average 14.5 bits are mutated for each individual). Finally, a randomly selected individual is substituted by the original copy of the best individual of the previous generation (elitism).

We have run two sets of experiments, each generation taking 80 minutes on the physical robot. Each set of experiments consists of several evolutionary runs starting from a different random initialization of the genetic string. In the first set, we have evolved spiking controllers using the parameters described in this section. The graph on the left of figure 9 shows the average fitness values measured across six runs for 30 generations of evolutionary spiking networks. Since the initial populations are randomly created, the networks display on average 50% random connectivity. This value does not change significantly along generations. The best and average fitness values gradually increase and reach a plateau around the 30th generation.

In a second set of experiments, we have evolved connectionist sigmoid networks. In this case the activation of a neuron $v(i)$ is a value between 0 and 1 computed using the sigmoid function $v(i) = 1/1 + e^{(-A)}$, where $A = \sum_j w_{ij}v_j$. The receptors take on the values that were used as probabilities of emitting a spike for the spiking controllers. The network is updated only one time every 100

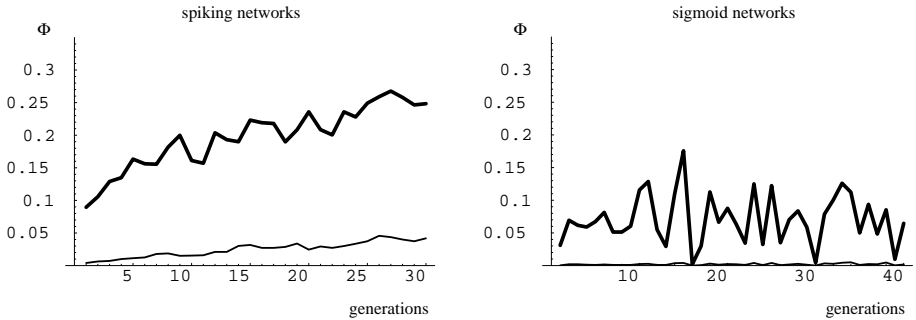


Fig. 9. Fitness values obtained on the physical robot Khepera (best fitness = thick line; average fitness = thin line). Each data point is the average of several evolutionary runs with different random initializations. *Left:* Evolution of spiking networks (average over six runs). *Right:* Evolution of connectionist sigmoid networks where 10 additional generations per run were allowed to check for signs of improvement (average over three runs).

ms.⁴ All the other parameters and genetic encoding were identical to those used for the spiking controllers (except for the fact that there is no threshold, synaptic function, and refractory function). The graph on the right of figure 9 shows the average fitness values measured across three runs of connectionist sigmoid networks. Despite allowing for an extra ten generations, none of the evolutionary runs could improve the fitness values along generations. The occasional higher values are given by individuals that perform wide circles independently of the sensory input until they meet a wall where they remain stuck.

The left side of figure 10 shows the architecture and the path of the best spiking controller of the 30th generation during 40 seconds. The right side of the figure shows the corresponding neural activity sampled every 100 ms. The robot moves along a looping trajectory, whose curvature depends on the visual input, without ever remaining stuck against a wall. The behavior does not change when the ambient illumination is increased or decreased since the visual input receives information about contrast, not about the grey levels returned by the photoreceptors.

If the robot is positioned against a wall (at any location), it slowly starts to rotate to the right until it gets away from it. Similarly, if a piece of white paper or one hand is positioned closed enough to its facing direction (2 to 3 cm), it rotates on the spot until it can get away.

⁴ We have also performed experiments where the sigmoid networks are updated 100 times between two sensory-motor intervals. These experiments generated the same results observed with a single update.

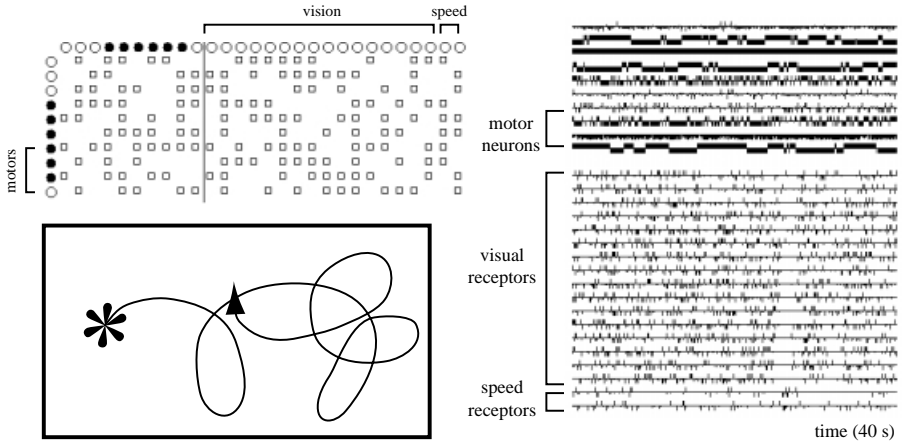


Fig. 10. *Top Left:* Architecture of the best spiking controller after 30 generations. Black circles = inhibitory neurons, white circles = excitatory neurons. *Bottom Left:* Typical trajectory of displayed by this spiking controller. The asterisk indicates the starting point. The curvature of the trajectory depends on the pattern of stripes seen by the robot. *Right:* Corresponding neural activity shown every 100 ms. Each line represents the resting potential of a neuron membrane. Dashes above the line indicate a spike, whereas signs below the line indicate that the neuron is inhibited.

5 Analysis of an Evolved Spiking Controller

In this section we analyze the best spiking controller evolved after 30 generations that has been described above. Since the neural receptors are driven solely by sensory information, do not have interconnections, and can emit a spike only every 100 ms (all spikes are shown on the right plot of figure 10), we focus our analysis on the remaining ten neurons, whose dynamics are updated every 1 ms.

While the robot was freely moving in the environment for 40 seconds, we recorded all the spikes emitted by each neuron. Table 1 shows that several neurons fire at almost maximum rate, which is 500 spikes per second, that is a spike every second millisecond because of the effects of the refractory period. Therefore, these neurons display a consistent self-sustained activity in the Inter Stimulus Interval (100 ms).

Table 1. Average number of spikes per second of the best spiking controller shown in figure 10 measured during 40s of autonomous navigation.

	neuron number									
	1	2	3	4	5	6	7	8	9	10
spikes/s	9	445	453	450	330	40	129	363	0	452

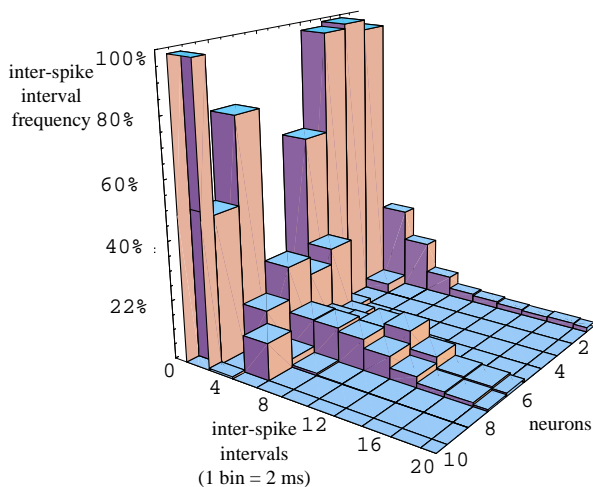


Fig. 11. Inter-spike interval frequencies measured during 10 seconds. The time intervals between adjacent spikes of a neuron are counted, grouped in incremental bins of 2 ms (2, 4, 6, 8, ..., 20), and normalized by the total number of spikes emitted by the neuron.

The firing rates of neurons 7–10 measured over 20 ms are used to set the speeds of the wheels. The firing rate of neurons 7 and 8 respectively set the backward and forward speed components of the right wheel, whereas the firing rates of neurons 9 and 10 set the backward and forward speed components of the left wheel. The spiking rates of these motor neurons suggests that the neural network controls the turning angle of the robot by changing the rotation speed of the right wheel (controlled by neurons 7 and 8) while the left wheel is kept at constant forward speed (by neuron 10).

The relative frequency of time intervals between spikes of each neuron provides further information on the the internal dynamics of the network (figure 11). This indicator is obtained by measuring the lengths of the intervals between two adjacent spikes over a 10 seconds period and grouping the occurrences of these intervals in bins of 2 ms (2 ms, 4 ms, 6 ms, 8 ms, etc.). These values are then divided by the total number of spikes fired by each neuron over 10 seconds in order to obtain a relative frequency. For example, a value of 0.2 in the third interval bin, means that that 20% of the spikes emitted by the neuron occur at intervals of 6ms. As expected from the firing rates shown in table 1, the data of figure 11 show that most neurons always fire at short intervals (2 and 4 ms). For example, this is the case of neuron 10 whose regular firing every 2 ms sustains constant forward speed of the left wheel. Some neurons fire also at longer intervals, but these intervals are rarely longer than 10 ms. Therefore, the dynamics of the evolved network settle into a stable state within approximately 10 ms after receiving sensory stimulation. To check this hypothesis, we have slowed down

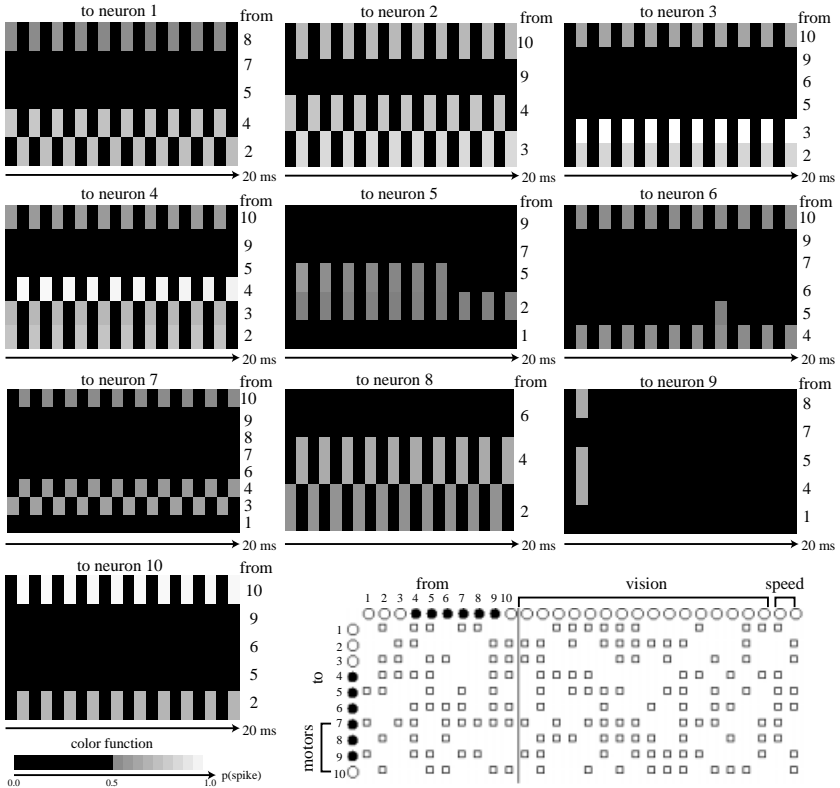


Fig. 12. Temporal Spike Correlograms. For each spike emitted by a neuron during a 10 s window, we count the occurrences of spikes emitted at within the preceding 20 ms (+2 ms to account for synaptic delay) by its pre-synaptic neurons, group them in 20 bins of 1 ms each, and normalize them by the total number of spikes emitted by the post-synaptic neuron.

the number of network updates between sensory updates from 100 to 50 and did not observe any difference in the behavior of the robot.

Since each neuron receives, on average, signals from five other neurons (without counting the connections from the sensory receptors), it is worth asking whether all the pre-synaptic neurons play the same role in causing a spike. To answer this question, we have developed the Temporal Spike Correlograms (figure 12). The correlations between the firing time of a neuron and the firing times of its pre-synaptic neurons within the 20 ms window preceding a spike⁵ provide an indication of the most important synaptic channels. These measures, shown in figure 12, are obtained as follows for each neuron. When the neuron emits

⁵ Remember that this is the time window of synaptic integration used in these experiments; see also figure 1.

a spike, we record the presence of a spike coming from each of its pre-synaptic neurons within a time window of 20 ms displaced by 2 ms in order to account for the synaptic delay (see equation 1 and figure 1). We repeat these measures over a period of 10 seconds arbitrarily chosen while the robot moves in the environment and count how many times at each millisecond there has been a spike from each connected pre-synaptic neuron. These counts are then divided by the total number of spikes emitted by the post-synaptic neuron. We take these values as the probability that pre-synaptic neuron j emits a spike t ms earlier than post-synaptic neuron i . If a neuron is connected to, say, 5 presynaptic neurons, we will have 5 series of 20 probabilities each. Figure 12 shows these probabilities as gray levels where white represents the highest probability and black represents probabilities lower or equal than 0.5.

These probability distributions tell us a number of things. To start with, not all synaptic channels are equally important. For example, when neuron 3 fires there is a probability lower or equal than 0.5 that neurons 5, 6, and 9 at any instant within the 20 ms window of synaptic sensitivity. On the other hand, we are almost certain of finding a spike every second millisecond coming from neurons 2, 3 (self-connection), and, to a lesser extent, from neuron 10. Considering that the threshold of the neurons is set to 0.1, a single spike emitted 6 ms earlier (including the time delay not shown in figure 12) by an excitatory neuron could cause a post-synaptic spike. In practice, things are more complex because we should also consider the interplay of excitation and inhibition. However, to a first approximation we can speculate that synaptic channels shown in black do not play an important role in controlling the behavior of the neuron.

If we weight the pattern of connectivity among neurons by the importance of the connections, we get an idea of which neurons are not playing an important role in the network. Neurons 1, 5, 6, and 9⁶ either do not make synaptic connections to other neurons or, when they do, their contribution does not play an important role in the firing of other neurons (as indicated by the black rows corresponding to these neurons). In the next section, we will explore these hypotheses by performing lesion studies.

Among the remaining six neurons, three (3, 4, and 10) have self-connections with high spike correlation. Notice that a self-connection does not necessarily imply high spiking correlation because of the effects of inhibitory channels and/or of long inter spike intervals (this is the case of neuron 5 and 6, for example). In the case of neurons 3 and 10 these self-connections are sufficient to generate a regular spiking activity independently of other inputs because there are not “important” inhibitory connections and the membrane threshold of 0.1 can be easily exceeded by their self-generated train of spikes over the 20 ms window (see also figure 1). Instead, the firing rates of neurons 2, 7, and 8 are determined by a delicate balance of inhibition and excitation coming from other neurons (and from neural receptors).

Another question is whether the evolved controller exploits spatio-temporal spike correlations from pre-synaptic neurons, as those postulated by the theories

⁶ Neuron 9 spiked only 3 times during the 10 seconds of observation.

of cortical activity based on coincidence detection [1,22,12]. If this is the case, for a given neuron we should observe a precise pattern of spikes across different pre-synaptic neurons that occurs regularly. From the distribution of inter spike intervals (figure 11), we know that this pattern must occur within the 15 ms window preceding the spike of the neuron. Although the plots of figure 12 clearly show regular occurrences of pre-synaptic spikes, there is not enough evidence that post-synaptic firing is caused by the coincidence of these spikes across different neurons. We can certainly rule it out for neurons 3 and 10 whose activity is mainly self-generated. We can also almost certainly rule it out for other neurons that fire mostly every second millisecond (2 and 4) because this time is insufficient to detect spatio-temporal patterns. Finally, we can rule it out for neuron 9 that almost never spikes. For the remaining five neurons, the analysis performed above is not sufficient to tell whether these patterns exist and play a role because their firing condition is the result of complex interactions between inhibitory and excitatory spikes. The tests described in the next subsection will help us to answer this question.

5.1 Lesion Studies

In the previous section we have speculated that neurons 1, 5, and 6 do not play a major role in the network on the basis of their “low contribution” to the firing of other neurons in the network. To check that hypothesis, we have systematically lesioned one neuron at a time and tested the lesioned controller in the environment three times for a duration of 80 seconds each. A lesion is done by silencing completely the output of the neuron. When neurons 1, 5, *or* 6 were lesioned the robot reported an average fitness of 0.22, practically identical to the fitness measured before the lesion (0.23), and no behavioral change was observed. We then lesioned all neurons 1, 5, *and* 6 and performed the same test. In this case the average reported fitness was 0.19, only slightly less than that observed before the multiple lesion and the behavior was almost the same. However, this time the trajectory was more straight when far from the walls and more jerky near the walls. This is probably due to the fact that the combined spiking rates of these neurons provide non-specific signal boost to the internal activity of the network. When they are lesioned all together, the robot behavior is slightly more dependent on the signals sent by visual receptors than on the internal activity.

Instead, when neuron 2 is lesioned, the robot rotates on the spot without influence of the visual input. Conversely, when neuron 3 is lesioned, the robot moves on a more or less straight trajectory (depending on the visual pattern) until the nearest wall. In both cases, the average fitness in the three tests is close to 0.0. Borrowing the method of double dissociation from cognitive neuropsychology [4], we can conclude that neuron 2 is responsible for moving straight on the basis of visual information, whereas neuron 3 generates a turn when the robot is gets too close to the walls. The role of neuron 2 is further supported by the pattern of connectivity from the visual receptors. It is the only neuron that has a full connection from the six vision receptors exactly in the middle

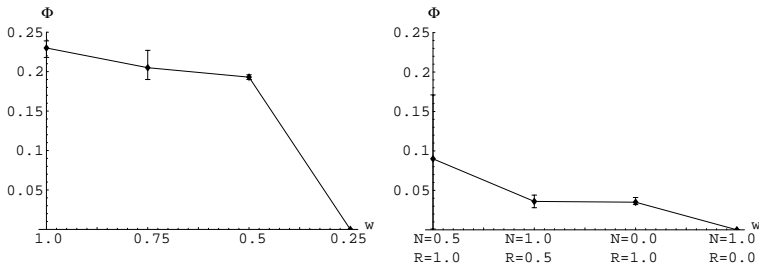


Fig. 13. Performance tests with synaptic decay. Data points represent average values (error bars indicate lowest and highest values) over three runs of 80 seconds each starting from a different location. The fitness value below which the robot is no longer capable of avoiding walls is approximately 0.15. *Left:* Synaptic strength is uniformly set to lower values w for all connections in the network. The first data point represents the performance of the evolved controller with intact synaptic strength ($w = 1$). *Right:* Synaptic strength is separately decreased for the synapses coming from the neurons (N) and for the synapses coming from the receptors (R).

of the visual field. Finally, when neuron 4 is lesioned the robot displays almost the same behavior, but cannot avoid the walls if it gets closer than 3 or 4 cm to them. Since neuron 4 is inhibitory and affects significantly both neuron 2 and motor neurons 7 and 8 (that control the right wheel to steer the robot), in normal conditions its role may be that of detecting a near collision and send stronger inhibition to neuron 2, which would result in a sharper turn.

5.2 Synaptic Decay

Another way to lesion the network consists of changing the strengths of the synaptic connections. These tests are of particular interest for hardware implementation of spiking neurons because the storage of weights for adaptive synapses is a major problem. Synaptic strengths are often stored and adapted by means of coupled capacitors whose voltages tend to decay relatively quickly. The same problem is found in analog VLSI implementations of adaptive weights. For example, Lewis et al. [13] report that their aVLSI neuromorphic Central Pattern Generator can store the values of synaptic weights for a few seconds with a decay rate of 0.1 V/s.

In order to assess the robustness of the evolved spiking controller with decaying synaptic strengths, we have performed three tests with weaker synaptic weights, at 0.75, 0.5, and 0.25 respectively. Each test consists of setting all the synaptic strengths (from the neurons and from the receptors) to a fixed value and measuring the fitness of the robot across three trials of 80 ms each starting at a different location in the environment. The average fitness values are plotted in the left graph of figure 13. The first data point shows the fitness of the evolved controller with its original synaptic strengths ($w = 1.0$). To some

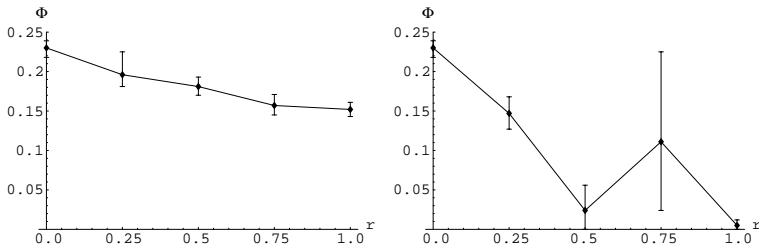


Fig. 14. Performance tests with noisy synaptic decay. Data points represent average values (error bars indicate lowest and highest values) over three runs of 80 seconds each starting from a different location. The fitness value below which the robot is no longer capable of avoiding walls is approximately 0.15. *Left:* A random value from the interval $0, r$ is subtracted from the original synaptic strength ($w = 1$) every millisecond. The random values are generated anew every millisecond and are different for every synapse. *Right:* Same as for the left graph, but the random values are subtracted only at the beginning of the test and the resulting weights are maintained constant for the whole duration of the test.

approximation, we can say that fitness values above 1.5 correspond to behaviors that retain the wall avoidance strategies and, to some extent, the trajectories of the original evolved controller. These tests indicate that the spiking controller can withstand uniform weight decay up to 50% of its original values. The lower fitness values are caused by slower movement or slightly jerkier movements.

In another series of tests, we have decreased the strengths separately for the synaptic connections among the neurons (N) and for the synaptic connections from the receptors (R). Four tests have been performed with the following combinations of strengths: $N = 0.5, R = 1.0$, $N = 1.0, R = 0.5$, $N = 0.0, R = 1.0$, and $N = 1.0, R = 0.0$. In none of these conditions the spiking controller could manage to navigate around the environment (graph on the left of figure 13). When only the connections among the neurons are lowered to 0.5 the robot goes straight but cannot avoid walls and when they are set to 0 it turns on itself. When only the connections from the sensors are lowered to 0.5 the robot turns on itself (but in the opposite direction than that used during normal behavior) and, of course, when they are set to 0.0 it does not move. These tests, combined with those on uniform synaptic decay, indicate that the evolved controller can withstand severe decay only if it is uniformly applied to all synapses in the network.

In another series of tests, we have studied the effects of noisy synaptic variations that may be caused, for example, by external radiation in analog VLSI microchips. In the first set of tests we change the strengths of each synapses every millisecond by subtracting a uniformly random number taken from the interval $[0, r]$ from the original strength ($w = 1$). The random numbers are generated separately for each synapse at each millisecond. Four tests have been performed with $r = 0.25$, $r = 0.5$, $r = 0.75$, and $r = 1.0$, respectively. Each test consists of three experiments of 80 seconds each, as above. The average fitness

data shown on the left graph of figure 14 (the first data point corresponds to a control condition without noise) indicate that the performance of the evolved controller degrades very slowly, but always preserves the same navigation abilities and overall trajectories. The small degradation is caused by the appearance of slightly jerky movements for $r = 0.25$ and lower speed for $r = 0.25$ and $r = 0.75$. For $r = 1.0$, we begin to observe straighter trajectories while still preserving the ability to avoid walls.

These results definitely rule out the possibility that the evolved controller may exploit spatio-temporal spike correlations because, if this were the case, the severe noise conditions of these tests would disrupt the precise contributions of individual spikes across the synaptic channels and would result in bad performance.

Since the synaptic noise is applied anew every millisecond, the average value over relatively long intervals approximates $r/2$. Therefore, if the network dynamics are based on firing rate, instead of firing time, the effects of this type of noise should be equivalent to those obtained by subtracting from each synapse a fixed value equal to $r/2$. This is indeed what we observe when we compare the results of these tests with those obtained with uniform synaptic decay (graph on the left of figure 13).

In a second set of tests with noisy synapse, we subtract from each synaptic strength a uniformly random number taken from an interval $[0, r]$ at the beginning of the test and maintain the resulting values for the whole duration of the test. Using the same procedure described above, we perform four tests with $r = 0.25$, $r = 0.5$, $r = 0.75$, and $r = 1.0$, respectively. The results shown in the graph on the right of figure 14 indicate that the controller is badly affected by non-uniform synaptic decrements. With the exception of $r = 0.25$, where the robot is still capable of maintain the basic navigation abilities, all the remaining conditions produce robots that mostly rotate on place. This last set of data fits the observations obtained with non-uniform synaptic decay across synaptic groups (neurons and receptors).

To summarize the results on synaptic lesions, we can conclude that the evolved controller can withstand strong synaptic decay, even if this is generated by a random process, as long as the decay is uniformly applied across the whole network.

6 Discussion

Despite the complex dynamics of spiking neurons and the higher number of free parameters with respect to sigmoid neurons, the results reported in this paper indicate that evolution can easily discover functional networks by searching the space of connectivity. Although the values of the neuron parameters have been set according to data reported in the literature [6] without attempting to optimize them for this specific implementation, we cannot exclude that different values would make the network harder or easier to evolve. As we already mentioned, the presence of noise in the refractory period is the only factor that we found

crucial for the evolvability of functional networks of spiking neurons. Since this type of noise affects the recovery time of the neuron, it reduces the probability that the networks fall into a state of locked oscillations that cannot be disrupted by sensory inputs.

The experimental data also show that it is harder to find functional networks of sigmoid neurons evolved under the same constraints. This does not mean that it is impossible to evolve networks of sigmoid neurons for this task, but that such networks probably require the variation and precise definition of other parameters, such as the strengths of the connections (in these experiments, they were all set to 1). It may also be the case that networks of sigmoid neurons with synaptic time delays and continuous dynamics [3] could be evolved under the same conditions used for spiking neurons. Even if that was the case, we think that spiking neurons represent a more powerful substrate for real-time mapping of sensory information into motor actions because these networks can potentially function both firing rate and firing time modes. In firing time mode, the time of arrival of a spike (or of a precise pattern of spikes distributed across several synapses) would be sufficient to trigger an appropriate response all the way up to the motor commands allowing the robot to respond in a few milliseconds to complex sensory patterns.

We should then ask why we did not find evidence that the neurons of our evolved spiking controllers responded in firing time mode (although we cannot exclude that some of their ancestors did). There could be several reasons for the evolutionary choice of firing rate instead of firing time. One of them is the relatively low membrane threshold $\theta = 0.1$ used in these experiments. Considering the values chosen for the other parameters, here a post-synaptic spike can be triggered by a single spike emitted by an excitatory pre-synaptic neuron 5 ms earlier (including the synaptic time delay). Therefore, it is unlikely that the neurons require multiple spikes across different synaptic channels in order to become active. A solution could be to either set higher threshold values, use smaller synaptic weights, or change the shape of the synaptic integrator (equation 1). Of course, these parameters could also be genetically encoded and evolved along with the pattern of connectivity.

Another reason is that the experimental settings are such to bias the system towards firing rate mode. Here the network is allowed 100 iterations between sensory-motor commands, which is sufficient to send several spikes along a connection and thus encode information in firing rate mode. Furthermore, the speeds of the robot wheels are set using the firing rate measured over a window of 20 ms. It would be interesting to evolve spiking controllers for robots where both sensory and motor dynamics match the dynamics of the spiking network. At the sensory level, these features could be provided by neuromorphic vision systems [9] whereas, at the motor level, one could use the spikes to directly drive the motors.

We also think that the way in which the sensory information is encoded into spiking neurons may affect the evolved functioning modality. In section 3 we have discussed some models of sensory encoding for spiking neurons. Whereas

sensory information encoded with the frequency model or the temporal coincidence model may be “understood” using a firing rate mode by simply summing up the quantity of incoming spikes across different channels over some time window, the delay coding model may force post-synaptic neurons to pay attention to the time of arrival of spikes in order to disambiguate the nature of the sensory stimulation.

Despite the relatively general form of the model of spiking neurons used in these experiments, one may not need all this level of detail when it comes to evolve networks of spiking neurons to be implemented in physical circuits. In that case, a simpler Integrate-and-Fire model which is relatively easier to realize with analog VLSI technology may provide similar functionalities.

7 Conclusion

The experiments described in this article are a preliminary exploration into the evolvability and properties of spiking neurons as control systems for autonomous robots. We have shown that artificial evolution can quickly discover small networks of spiking neurons to perform a non-trivial vision-based navigation task. The comparison with networks of sigmoid neurons indicate that, all the factors being equal, the intrinsic dynamics of spiking neurons provide more degrees of freedom that can be exploited by evolution to generate viable controllers.

The analysis tools described in this paper have allowed us to understand the roles played by individual neurons in the evolved network and make predictions about the influence of their activity on the behavior independently of their observed firing rate. These predictions have been confirmed with lesion studies on single neurons and groups of neurons.

The evolved network displays remarkable robustness in face of constant or irregular synaptic decay as long as this happens more or less uniformly across the entire network. Since this assumption is quite reasonable in the case of hardware implementation with adaptive or evolvable synaptic weights, we believe that it represents an interesting method for the evolution of analog VLSI spiking circuits characterized by tiny size and extremely small energetic consumption. This technology could be used for micro autonomous robots or for flying robots where payload and energetic autonomy are a major constraint.

Acknowledgements. This work was supported by the Swiss National Science Foundation, grant no. 620-58049.

References

1. M. Abeles. *Corticonics*. Cambridge University Press, Cambridge, 1991.
2. E. D. Adrian. *The basis of sensation: The action of the sense organs*. W. W. Norton, New York, 1928.
3. R. D. Beer and J. C. Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1:91-122, 1992.

4. A. W. Ellis and A. W. Young. *Human Cognitive Neuropsychology*. Lawrence Erlbaum Associates, London, 1988.
5. W. Gerstner. Associative memory in a network of biological neurons. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 84–90. Morgan Kaufmann, San Mateo, CA, 1991.
6. W. Gerstner, J. L. van Hemmen, and J. D. Cowan. What matters in neuronal locking? *Neural Computation*, 8:1653–1676, 1996.
7. J. J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376:33–36, 1995.
8. G. Indiveri. A Neuromorphic VLSI device for implementing 2D selective attention systems. *IEEE Transactions on Neural Networks*, page in press, 2001.
9. G. Indiveri and R. Douglas. ROBOTIC VISION: Neuromorphic Vision Sensors. *Science*, 288:1189–1190, 2000.
10. G. Indiveri and P. Verschure. Autonomous vehicle guidance using analog VLSI neuromorphic sensors. In W. Gerstner, A. Germond, M. Hasler, and J-D. Nicoud, editors, *Proceedings of the 7th International Conference on Neural Networks*, pages 811–816. Springer Verlag, Berlin, 1997.
11. E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. 4th edition. McGraw-Hill Professional Publishing, New York, 2000.
12. P. Koenig, A. K. Engel, and W. Singer. Integrator or coincidence detector? the role of cortical neuron revisited. *Trends in Neuroscience*, 19:130–137, 1996.
13. M. A. Lewis, R. Etienne-Cummings, A. H. Cohen, and M. Hartmann. Toward biomorphic control using custom aVLSI CPG chips. In *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE Press, 2000.
14. W. Maas and C. M. Bishop, editors. *Pulsed Neural Networks*. MIT Press, Cambridge, MA, 1999.
15. C. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA, 1991.
16. V. B. Mountcastle, G. F. Poggio, and G. Werner. The relation of thalamic cell response to peripheral stimuli varied over an intensive continuum. *Journal of Neurophysiology*, 26:807–834, 1963.
17. Z. Nadasdy. *Spatio-Temporal Patterns in the Extracellular Recording of Hippocampal Pyramidal Cells: From Single Spikes to Spike Sequences*. PhD thesis, Rutgers University, Newark, NJ, 1998.
available at <http://osiris.rutgers.edu/Buzsaki/Posters/Nadasdy/thesis.html>.
18. S. Nolfi and D. Floreano. *Evolutionary Robotics: Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.
19. D. E. Rumelhart, J. McClelland, and PDP Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MIT Press-Bradford Books, Cambridge, MA, 1986.
20. C. S. Sherrington, editor. *Integrative Action of the Nervous System*. Yale University Press, New Haven, CT, 1906.
21. W. Singer. Search for coherence: a basic principle of cortical self-organization. *Concepts in Neuroscience*, 1:1–26, 1990.
22. W. Singer and C. M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, 18:555–586, 1995.
23. A. Villa. Empirical evidence about temporal structure in multi-unit recordings. In R. Miller, editor, *Time and the Brain*. Harwood Academic Publishers, Reading, UK, 2000.