

---

# Evolutionary Robotics: Coping with Environmental Change

---

**Joseba Urzelai**

Microprocessors and Interfaces Lab  
Swiss Federal Institute of Technology  
Lausanne, Switzerland  
joseba.urzelai@epfl.ch  
+41 21 693 6696

**Dario Floreano**

Microprocessors and Interfaces Lab  
Swiss Federal Institute of Technology  
Lausanne, Switzerland  
dario.floreano@epfl.ch  
+41 21 693 5230

## Abstract

In this paper an evolutionary method consisting of encoding a set of local adaptation rules that synapses obey while a robot freely moves in the environment is compared to standard evolution of fixed-weight control networks. The results show that evolutionary adaptive controllers can adapt online without additional evolutionary training to strong environmental changes where instead the performance of evolutionary fixed-weight controllers is significantly degraded. Two cases are described: transfer of evolved controllers from simulated to real robots and across different robotic platforms that vary in size, shape, and sensor response profile. In both cases evolved adaptive controllers autonomously and quickly adjust synaptic weights to successfully accomplish the task in the new conditions.

## 1 INTRODUCTION

Evolutionary robotics is advocated as an automatic method to discover efficient controllers for robots that operate in real environments. The situated nature of the evolutionary approach is such that often evolved controllers find surprisingly simple –yet efficient– solutions that capitalize upon unexpected invariants of the interaction between the robot and its environment. For example, a robot evolved for the ability to discriminate between shapes can do so without resorting to expensive image processing techniques by simply checking the correlated activity of two receptors located in strategic positions on the retinal surface (Harvey, Husbands, & Cliff, 1994). Analogously, a robot evolved for finding a hidden location can display the performances similar to those obtained by rats trained under the same conditions without resorting to complex environmental representations by using

simple sensory-motor sequences that exploit geometric invariants of the environment (Lund & Miglino, 1998). The remarkable simplicity<sup>1</sup> and efficiency of these solutions is a clear advantage for fast and real-time operation required from autonomous robots, but it raises the issue of robustness when environmental conditions change. Environmental changes can be a problem also for other approaches (programming, learning, e.g.) to the extent in which the sources of change have not been considered during system design, but they are even more so for evolved systems because these often rely on environmental aspects that are often not predictable by an external observer.

Environmental changes can be induced by several factors such as modifications of the sensory appearance of objects (e.g., different light conditions), changes in sensor response, re-arrangement of environment configuration, transfer from simulated to physical robots, and transfer across different robotic platforms.

Some authors have suggested to improve the robustness of evolved systems by adding noise (Miglino, Lund, & Nolfi, 1996; Jakobi, 1997) and by evaluating fitness values in several different environments (Thompson, 1998). However, both techniques imply that one knows in advance what makes the evolved solution brittle in the face of future changes in order to choose a suitable type of noise and of environmental variability during evolutionary training. Another approach consists of combining evolution and learning “during life” of the individual (see (Belew & Mitchell, 1996) for a comprehensive review of the combination of evolution and learning). This strategy not only can improve the search properties of artificial evolution, but can also make the controller more robust to changes that occur faster than the evolutionary time scale (i.e., changes that occur during the life of an individual) (Nolfi & Floreano, 1999). This is typically achieved by evolving neural controllers that learn with an off-

---

<sup>1</sup>This does not imply that evolutionary approaches are restricted to forms of reactive intelligence; see for example (Floreano & Mondada, 1996a)

the-shelf algorithm, such as reinforcement learning or back-propagation, starting from synaptic weights specified on the genetic string of the individual (Ackley & Littman, 1992; Nolfi, Miglino, & Parisi, 1994). Only initial synaptic weights are evolved. A limitation of this approach is the “Baldwin effect” (see (Hinton & Nowlan, 1987) for an example of the Baldwin effect in a computational model), whereby the evolutionary costs associated with learning give a selective advantage to the genetic assimilation of learned properties and consequently reduce the plasticity of the system over time (Mayley, 1996).

In previous work (Floreano & Mondada, 1996b) we have suggested to *evolve the adaptive characteristics* of a controller instead of combining evolution with off-the-shelf algorithms. The method consists of encoding on the genotype a set of four local Hebb rules for each synapse, but *not the synaptic weights*, and let these synapse use these rules to adapt their weights online starting always from random values at the beginning of the life. Since the synaptic weights are not encoded on the genetic string, there cannot be genetic assimilation of abilities developed during life. In other words, these controller can rely less on genetically-inherited invariants and must develop on-the-fly the connection weights necessary to achieve the task.

When comparing evolution of genetically-determined weights with evolution of adaptive controllers on a simple navigation task, we have shown that the latter approach generates similarly-good performances in less generations (Floreano & Mondada, 1998) by taking advantage of the combined search methods. In another paper we have empirically demonstrated that our method of evolving adaptive controllers can solve a complex sequential task involving multiple behaviors whereas evolution of genetically-determined controllers fail to do so satisfactorily (Floreano & Urzelai, 1999). Finally, in a very recent paper (Urzelai & Floreano, 2000) we have shown that evolutionary adaptive controllers can adapt to environmental changes that involve new sensory characteristics and new spatial relationships of the environment.

In this paper, we describe a new set of experiments designed to further test the robustness of this approach to environmental changes. In particular, here we address two important types of change for robot controllers: the transfer of evolved controllers from a simulated robot to a physical robot (Khepera) and across different robots for the sequential task described earlier (Floreano & Urzelai, 1999). The results are compared to those obtained from evolution of genetically-determined weights and evolution of noisy synaptic weights (control condition). Evolutionary adaptive controllers not only report significantly better performances, but also display qualitatively different ways

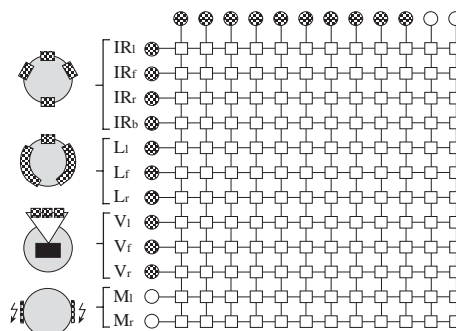


Figure 1: The neural controller is a fully-recurrent discrete-time neural network composed of 12 neurons giving a total of  $12 \times 12 = 144$  synapses (here represented as small squares of the unfolded network). 10 sensory neurons receive additional input from one corresponding pool of sensors positioned around the body of the robot shown on the left (l=left; r=right; f=front; b=back).  $\vec{IR}$ =Infrared Proximity sensors;  $\vec{L}$ =Ambient Light sensors;  $\vec{V}$ =vision photoreceptors. Two motor neurons  $\vec{M}$  do not receive sensory input; their activation sets the speed of the wheels ( $M_i > 0.5$  forward rotation;  $M_i < 0.5$  backward rotation)

of coping with the task at hand.

In the next sections we give an overview of the evolutionary method and then briefly describe the sequential task that is used as a context to measure adaptivity to environmental changes. We then present the results on the transfer from simulation to physical robots and across different robotic platforms. Finally, we discuss the future perspectives of this new evolutionary approach.

## 2 METHOD

The controller we have used in our experiments is a fully-recurrent discrete-time neural network (figure 1). It has access to three types of sensory information from the robot:

1. *Infrared light*: the active infrared sensors positioned around the robot (figure 2, a) measure the distance from objects. Their values are pooled into four pairs and the average reading of each pair is passed to a corresponding neuron.
2. *Ambient light*: the same sensors are used to measure ambient light too. These readings are pooled into three groups and the average values are passed to the corresponding three light neurons.
3. *Vision*: the vision module (figure 2, b) consists of an array of 64 photoreceptors covering a visual

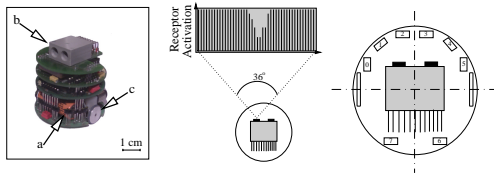


Figure 2: The Khepera robot used in the experiments. Infrared sensors (a) measure object proximity and light intensity. The linear vision module (b) is composed of 64 photoreceptors covering a visual field of 36° (center). The output of the controller generates the motor commands (c) for the robot. Right figure shows the sensory disposition of the Khepera robot.

field of 36° (figure 2, center). The visual field is divided up in three sectors and the average value of the photoreceptors (256 gray levels) within each sector is passed to the corresponding vision neuron.

Two motor neurons are used to set the rotation speed of the wheels (figure 2, c). Neurons are updated every 100 ms according to the following equation

$$y_i \leftarrow \sigma \left( \sum_{j=0}^N w_{ij} y_j \right) + I_i,$$

where  $y_i$  is the activation of the  $i$ th neuron,  $w_{ij}$  is the strength of the synapse between presynaptic neuron  $j$  and postsynaptic neuron  $i$ ,  $N$  is the number of neurons in the network,  $0 \leq I_i < 1$  is the corresponding external sensory input, and  $\sigma(x) = (1 + e^x)^{-1}$  is the sigmoidal activation function.  $I_i = 0$  for the motor neurons.

Each synaptic weight  $w_{ij}$  is randomly initialized at the beginning of the individual's life and can be updated after every sensory-motor cycle (100 ms),

$$w_{ij}^t = w_{ij}^{t-1} + \eta \Delta w_{ij},$$

where  $0.0 < \eta < 1.0$  is the learning rate and  $\Delta w_{ij}$  is one of the four modification rules specified in the genotype:<sup>2</sup>

1. *Plain Hebb rule*: strengthens the synapse proportionally to the correlated activity of the two neurons.
2. *Postsynaptic rule*: behaves as the plain Hebb rule, but in addition it weakens the synapse when the postsynaptic node is active but the presynaptic is not.
3. *Presynaptic rule*: weakening occurs when the presynaptic unit is active but the postsynaptic is not.

<sup>2</sup>These four rules co-exist within the same network.

Encoding	Bits for one synapse / node				
Genotype	1	2	3	4	5
A	sign	strength			
B	sign	Hebb rule		rate	
C	sign	strength		noise	

Table 1: Genetic encoding of synaptic parameters for Synapse Encoding and Node Encoding. In the latter case the sign encoded on the first bit is applied to all outgoing synapses whereas the properties encoded on the remaining four bits are applied to all incoming synapses. A: Genetically determined controllers; B: Adaptive synapse controllers; C: Noisy synapse controllers.

4. *Covariance rule*: strengthens the synapse whenever the difference between the activations of the two neurons is less than half their maximum activity, otherwise the synapse is weakened.

Synaptic strength is maintained within a range  $[0, 1]$  (notice that a synapse cannot change sign) by adding to the modification rules a self-limiting component inversely proportional to the synaptic strength itself (Floreano & Mondada, 1996b, 1998, for more details).

Two types of genetic (binary) encoding are considered (see table):

1. *Synapse Encoding*: also known as direct encoding (Yao, 1993), every synapse is individually coded on 5 bits, the first bit representing its sign and the remaining four bits its properties (either the weight strength or its adaptive rule).
2. *Node Encoding*: each node is characterized by 5 bits, the first bit representing its sign and the remaining four bits the properties of all its incoming synapses (consequently, all incoming synapses to a given node have the *same* properties).

In our first experiments (Floreano & Mondada, 1996b, 1998; Floreano & Nolfi, 1997) we always used Synapse Encoding and showed that evolution of adaptive synapses for obstacle avoidance and chasing tasks develops levels of performance faster, better, or more robust than those obtained by evolution of genetically-determined synapses.

Synapse Encoding allows a detailed definition of the controller, but for a fully connected network of  $N$  neurons the genetic length is proportional to  $N^2$ . Instead Node Encoding requires a much shorter genetic length (proportional to  $N$ ), but it allows only a rough definition of the controller. In recent work (Floreano & Urzelai, 1999) we showed that our evolutionary

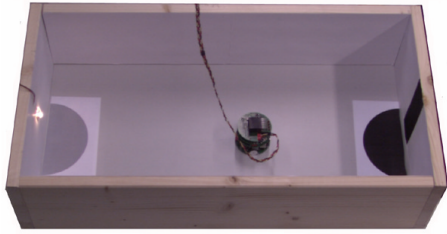


Figure 3: A mobile robot Khepera equipped with a vision module gains fitness by staying on the gray area only when the light is on. The light is normally off, but it can be switched on if the robot passes over the black area positioned on the other side of the arena. The robot can detect ambient light and the color of the wall, but not the color of the floor.

adaptive approach does not need a lengthy representation because the actual weights of the synapses are always shaped at run-time by the genetically specified rules. However, this is not possible in the traditional approaches where it is necessary to assign good initial weights to the controller. Therefore, the experiments reported in this paper compare evolution of genetically-determined networks using Synapse Encoding with evolution of adaptive networks using Node Encoding.

What is encoded on the remaining four bits depends on the evolutionary condition chosen, namely:

1. *Genetically-determined*: 4 bits encode the synaptic strength. This value is constant during “life”.
2. *Adaptive synapses*: 2 bits encode 4 adaptive rules and 2 bits the learning rate. Synaptic weights are always randomly initialized at the beginning of an individual’s life and then updated according to their own adaptation rule.
3. *Noisy synapses*: 2 bits encode the weight strength and 2 bits a noise range. The synaptic strength is genetically determined at birth, but a random value extracted from the noise range is freshly computed and added after each sensory motor cycle. This latter condition is used as a control condition to check whether the effects of Hebbian adaptation (condition above) are equivalent to random synaptic variability.

### 3 A SEQUENTIAL TASK

In this initial set of experiments, we have compared the performance of evolutionary adaptive controllers with respect to the evolution of synaptic weights and the evolution of noisy synapses in a sequential task that is complex enough to require non-trivial solutions.

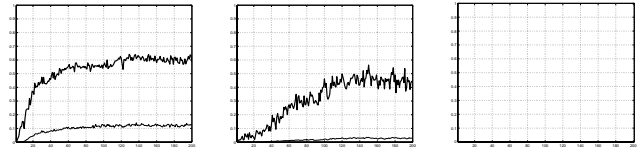


Figure 4: Comparison of adaptive synapses with Node Encoding (*left*) versus genetically-determined synapses with Synapse Encoding (*center*) and genetically-determined synapses with Node Encoding (*right*). Individuals evolved with genetically-determined synapses and Node Encoding (graph on the right) never managed to complete the task reliably in any of the ten replications. Thick line=best individual; thin line=population average; dashed line=genetic diversity. Each data point is an average over 10 replications with different random initializations. Population size is 100 and 20 best individuals reproduce by making 5 copies. Crossover probability is 0.2 and mutation probability is 0.05 (per bit).

A mobile robot Khepera equipped with a vision module is positioned in the rectangular environment shown in figure 3. A light bulb is attached on one side of the environment. This light is normally off, but it can be switched on when the robot passes over a black-painted area on the opposite side of the environment. A black stripe is painted on the wall over the light-switch area. Each individual of the population is tested on the same robot, one at a time, for 500 sensory motor cycles, each cycle lasting 100 ms. At the beginning of an individual’s life, the robot is positioned at a random position and orientation and the light is off.

The fitness function is given by the number of sensory motor cycles spent by the robot on the gray area beneath the light bulb *when the light is on* divided by the total number of cycles available (500). In order to maximize this fitness function, the robot should find the light-switch area, go there in order to switch the light on, and then move towards the light as soon as possible, and stand on the gray area<sup>3</sup>. Since this sequence of actions takes time (several sensory motor cycles), the fitness of a robot will never be 1.0. Also, a robot that cannot manage to complete the entire sequence will be scored with 0.0 fitness.

A light sensor placed under the robot is used to detect the color of the floor—white, gray, or black—and passed to a host computer in order to switch on the light bulb and compute fitness values. The output of this sensor is *not* given as input to the neural con-

<sup>3</sup>Notice that the fitness function does not explicitly reward this sequence of actions, but only the final outcome of the overall behavior chosen by the robot. Therefore, we call it a *behavior-based* fitness function.

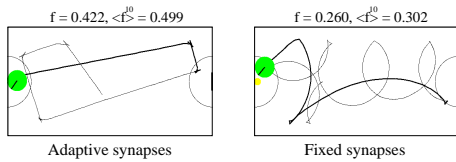


Figure 5: Behaviors of two best individuals (from last generation) with adaptive synapses and Node Encoding (*left*) and with genetically-determined synapses and Synapse Encoding (*right*). When the light is turned on, the trajectory line becomes thick. The corresponding fitness value is printed on the top of each box along with the average fitness of the same individual tested ten times from different positions and orientations.

troller. After 500 sensory motor cycles, the light is switched off and the robot is repositioned by applying random speeds to the wheels for 5 seconds.

The experiments have been carried out in simulations sampling sensor activation and adding 5% uniform noise to these values (Miglino et al., 1996).<sup>4</sup>

The fitness results reported in figure 4 show that individuals with adaptive synapses and Node Encoding (graph on the left) are much better than individuals with genetically-determined synapses and Synapse Encoding (graph in the center) in that:

1. Both the fitness of the best individuals and of the population report higher values (0.6 against 0.5). The performance difference measured on best individuals of the last generation is statistically significant ( $p < 0.05$  for a two-tailed t-test of the difference of the means).
2. They reach the best value obtained by genetically-determined individuals in less than half generations (40 against more than 100).

Individuals with genetically-determined synapses and Node Encoding (figure 4, right) never managed to complete the task in any of the ten replications.

Figure 5 shows the behaviors of two best individuals evolved with adaptive synapses and Node Encoding (left) and with genetically-determined weights and Synapse Encoding (right). In both cases individuals aim at the area with the light switch<sup>5</sup> and, once the light is turned on, they move towards the light and

<sup>4</sup>All experimental conditions have also been repeated on the physical robot and yielded similar results (Floreano & Urzelai, 1999).

<sup>5</sup>Their performance is badly affected if the vision input is disabled, indicating that they do not use random search to locate the switch (data not shown).

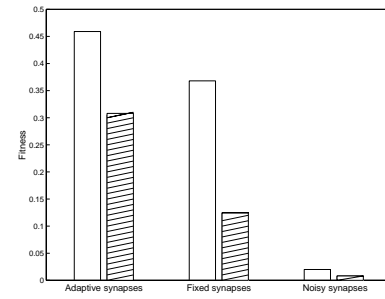


Figure 6: Comparison of adaptive synapses with Node Encoding (*left*) versus genetically-determined synapses with Synapse Encoding (*center*) and noisy synapses with Node Encoding (*right*) in simulated environments (white) and on a real Khepera robot (striped). Each fitness value is an average over 3 replications with different random initializations.

remain there. The better fitness of the adaptive controllers (given on the top of each box, see figure caption) is given by straight and faster trajectories showing a clear behavioral change between the first phase where they go towards the switching area and the second phase where they become attracted by the light. Instead, genetically-determined individuals display always the same looping trajectories around the environment with some attraction towards the stripe and the light. This minimalist behavior that depends on invariant geometrical relations of the environment gives them a chance to accomplish the task but with a lower performance.

The fitness curves and the behavior of robots evolved with Node Encoding and noisy synapses have been reported elsewhere (Floreano & Urzelai, 1999), and are not showed here due to space constraints. These results showed that the effects of Hebbian adaptation are *not* equivalent to random synaptic variability.

Additional tests have been carried out to assess the role of adaptation in the behavior of the individuals with adaptive synapses. For example, one might argue that what matters is the sign of the synapse and not its strength as long as it is non-zero, or that adaptive synapses may have the same effect of fixed synapses with strengths set to their average values<sup>6</sup>. The results reported by our control experiments and analyses clearly indicated that evolved adaptive networks modify their parameters in ways that are functionally related to the survival criterion (Floreano & Urzelai, 1999).

<sup>6</sup>This latter suggestion was made by Flotzinger (Flotzinger, 1996) who replicated our previous experiments on Adaptive Synapses with Synapse Encoding (Floreano & Mondada, 1996b)



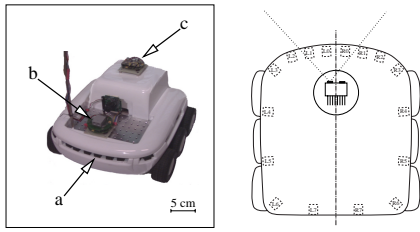


Figure 7: The Koala robot used in the experiments. Infrared sensors (a) measure object proximity and light intensity. The linear vision module (b) is the same as used in the experiments with the Khepera robot. The localization module (c) provides the position of the robot at every time step. Right figure shows the sensory layout of the Koala robot. Only 8 equally-spaced sensors are selected as input to the network.

## 4 FROM SIMULATIONS TO REAL ROBOTS

One way of measuring the adaptive abilities of evolved controllers is to transfer them from simulated to real robots. Since physical robots and environments often have characteristics different from simulations, solutions evolved in simulation typically fail when tested on real robots.

We have transferred on a physical Khepera robot the best individuals of the last generation for each of the 10 populations evolved in simulation (walls are covered with white paper, figure 3). Figure 6 shows that the performance of adaptive individuals is less affected by the transfer to the physical environment than genetically-determined individuals (the difference in performance loss is statistically significant,  $p < 0.05$  for a two-tailed t-test of the difference of the means). Individuals with noisy synapses are not affected by the transfer because their behavior is always random and not effective in both simulated and physical environments.

Some performance loss in adaptive individuals is caused by the fact that in some cases the robot performs looping trajectories around the fitness area without coming to rest on it. Instead, the two major reasons for failure of genetically-determined individuals are that they often get stuck on the walls and they cannot manage to move efficiently towards the light. These failures are due to differences between simulated and real sensors. Since the weights are fixed, genetically-determined individuals cannot accommodate these changes as adaptive individuals do.



Figure 8: A mobile robot Koala equipped with a vision module gains fitness by staying near the lamp (right side) only when the light is on. The light is normally off, but it can be switched on if the robot passes near the black stripe (left side) positioned on the other side of the arena. Position of the robot is controlled by an external positioning system and passed to the computer in order to control the light and to compute the fitness.

## 5 CROSSPLATFORM EVOLUTION

Cross-platform transfer is a very useful feature, but we are not aware of any control system that can be transferred across different robots without changes. One may develop (or evolve) control systems for a desktop sturdy robot like the Khepera and then download them to larger and consequently more fragile robots<sup>7</sup>. In previous work we have shown that this can be achieved by using incremental evolution of genetically-determined networks (Floreano & Mondada, 1998). However, even for a simple reactive navigation behavior it took additional 20 generations to re-adapt to the new robot.

Here we test the adaptive properties of the evolutionary adaptive strategy by transferring on a physical Koala robot (figure 7, left) the best individuals of the last generation evolved in each of the 10 simulations of the experiment presented in section 3. The Koala robot has six wheels driven by two motors (one on each side) and 16 infrared sensors (figure 7, right) with a different and stronger detection range. A mobile robot Koala equipped with a vision module is positioned in the rectangular environment shown in figure 8. As in the previous experiment with the Khepera robot, the Koala robot must find the light-switching area, go there in order to switch the light on, and then move towards the light as soon as possible and stay there in order to score fitness points.

An external positioning system emitting laser beams at predefined angles and frequencies is positioned on the top of the environment and the Koala robot is

<sup>7</sup>Obviously, the two robots must share some characteristics, such as type of sensors and actuators used, that allow a suitable interfacing of the control system.

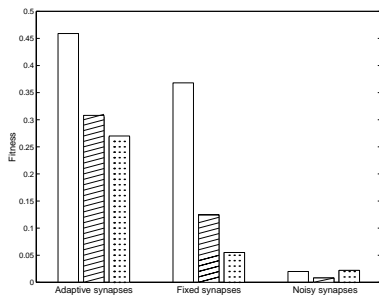


Figure 9: Comparison of adaptive synapses with Node Encoding (*left*) versus genetically-determined synapses with Synapse Encoding (*center*) and noisy synapses with Node Encoding (*right*) in simulated environments (white), on the Khepera robot (striped), and on the Koala robot (dotted). Each fitness value is an average over 3 replications with different random initializations.

equipped with an additional turret capable of detecting laser and computing in real-time the robot displacement. This information is used in order to control the light and to compute the fitness. Figure 9 shows that the performance of adaptive individuals is not affected by the transfer from the Khepera robot (striped bars) to the Koala robot (dotted bars), whereas genetically-determined individuals report a significant fitness loss ( $p < 0.05$  for a two-tailed t-test of the difference of mean performance loss measured on adaptive and on genetically-determined individuals). Individuals with noisy synapses are not affected by the transfer because their behavior is always random and not effective in both Khepera and Koala robots.

Individuals evolved in simulation for the Khepera robot display a satisfactory behavior when tested on the Koala robot. They correctly approach the light-switching area and they are clearly attracted by light (figure 10, left). As in the case of real Khepera robot, once arrived under the light the Koala robot moves around the fitness area while remaining close to it until the testing time is over.

On the other hand, genetically-determined individuals (right) perform spiralling trajectories around the environment and do not display any attraction by the black stripe or the light. They eventually manage to pass through the light-switching area, turn the light on, and occasionally score fitness points passing through the fitness area. In several cases, genetically-determined individuals get stuck on the walls of the environment (behaviors not shown). Individuals with noisy synapses score a low performance because their strategy is based in random navigation.

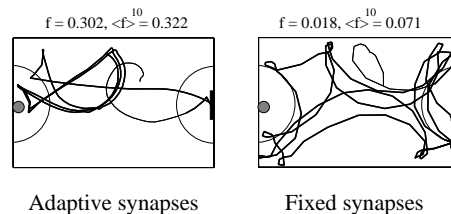


Figure 10: Behaviors of individuals with adaptive synapses (left) and genetically-determined synapses (right). Individuals belong to the last generation evolved in simulation for the Khepera robot.

## 6 CONCLUSIONS

We have shown through a set of systematic comparisons that evolution of adaptive synapses provides better adaptation capabilities than standard evolution of synaptic weights in the transfer from simulations to physical robots and across different robotic platforms. Evolutionary adaptive controllers can autonomously modify their synaptic weights and behavior online to the new environmental conditions without requiring additional evolution or ad-hoc manipulation of the evolutionary conditions.

We think that the evolutionary technique presented here represents a significant step forward towards making Evolutionary Robotics applicable to real-world applications of autonomous robotics. In scenarios like those—for example—of robots probing an asteroid surface or robots interacting with an handicapped person it is impossible to evolve the control system on the spot (not even incrementally). However, one might reproduce the working conditions in the laboratory to some degree of approximation and evolve the adaptive controller in there. The controller would then be transferred on the final robot and let free to adapt to actual working conditions in a few seconds.

We also think that our adaptive strategy will be useful for evolving more complex and powerful control architectures. In current methods there is a trade-off between the complexity the genotype/phenotype mapping and the evolvability of such systems which is partly due to the fact that the phenotype largely depends on genetic instructions. By evolving the adaptive characteristics along with other high-level parameters (position and type of nodes, e.g.) of the controller, one may obtain simpler genetic encodings and a higher tolerance to mutations. This would make the evolved controllers more viable, add neutrality to the genetic landscape, and ultimately improve evolvability.

## Acknowledgments

Joseba Urzelai is supported by grant nr. BF197.136-AK from the Basque government. Thanks to Patrick Saucy for his help on the crossplatform experiment.

## References

- Ackley, D. H., & Littman, M. L. (1992). Interactions between learning and evolution. In Langton, C., Farmer, J., Rasmussen, S., & Taylor, C. (Eds.), *Artificial Life II: Proceedings Volume of Santa Fe Conference*, Vol. XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, Redwood City, CA.
- Belew, R. K., & Mitchell, M. (Eds.). (1996). *Adaptive Individuals in Evolving Populations. Models and Algorithms*. Addison-Wesley, Redwood City, CA.
- Floreano, D., & Mondada, F. (1996a). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26, 396–407.
- Floreano, D., & Mondada, F. (1996b). Evolution of plastic neurocontrollers for situated agents. In Maes, P., Mataric, M., Meyer, J., Pollack, J., Roitblat, H., & Wilson, S. (Eds.), *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp. 402–410. MIT Press-Bradford Books, Cambridge, MA.
- Floreano, D., & Mondada, F. (1998). Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11, 1461–1478.
- Floreano, D., & Nolfi, S. (1997). Adaptive behavior in competing co-evolving species. In Husbands, P., & Harvey, I. (Eds.), *Proceedings of the 4th European Conference on Artificial Life* Cambridge, MA. MIT Press.
- Floreano, D., & Urzelai, J. (1999). Evolution of Neural Controllers with Adaptive Synapses and Compact Genetic Encoding. In Floreano, D., Nicoud, J., & Mondada, F. (Eds.), *Advances In Artificial Life: Proceedings of the 5th European Conference on Artificial Life (ECAL'99)*, pp. 183–194. Springer Verlag, Berlin.
- Flotzinger, D. (1996). Evolving plastic neural network controllers for autonomous robots. Msc dissertation 9580131, COGS, University of Sussex at Brighton.
- Harvey, I., Husbands, P., & Cliff, D. (1994). Seeing The Light: Artificial Evolution, Real Vision. In Cliff, D., Husbands, P., Meyer, J., & Wilson, S. W. (Eds.), *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*. MIT Press-Bradford Books, Cambridge, MA.
- Hinton, G. E., & Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1, 495–502.
- Jakobi, N. (1997). Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In Husbands, P., & Harvey, I. (Eds.), *Advances in Artificial Life: Proceedings of the 4th European Conference on Artificial Life*, pp. 348–357. MIT Press.
- Lund, H. H., & Miglino, O. (1998). Evolving and breeding robots. In Husbands, P., & Meyer, J.-A. (Eds.), *Proceedings of the First European Workshop on Evolutionary Robotics* Berlin. Springer Verlag.
- Mayley, G. (1996). Landscapes, Learning Costs and Genetic Assimilation. *Evolutionary Computation*, 4(3), 213–234.
- Miglino, O., Lund, H. H., & Nolfi, S. (1996). Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life*, 2, 417–434.
- Nolfi, S., & Floreano, D. (1999). Learning and evolution. *Autonomous Robots*, 7(1), forthcoming.
- Nolfi, S., Miglino, O., & Parisi, D. (1994). Phenotypic Plasticity in Evolving Neural Networks. In Nicoud, J.-D., & Gaussier, P. (Eds.), *Proceedings of the conference From Perception to Action*. IEEE Computer Press, Los Alamitos, CA.
- Thompson, A. (1998). On the automatic design of robust electronics through artificial evolution. In M. Sipper, D. M. . A. P.-U. (Ed.), *Proceedings of the 2nd International Conference on Evolvable Systems: From biology to hardware (ICES98)*, pp. 13–24. Springer-Verlag, Berlin.
- Urzelai, J., & Floreano, D. (2000). Evolutionary robots with fast adaptive behavior in new environments. In Fogarty, T. C., Miller, J., Thompson, A., & Thomson, P. (Eds.), *Third International Conference on Evolvable Systems: From Biology to Hardware (ICES2000)* Berlin. Springer-Verlag.
- Yao, X. (1993). A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4, 203–222.