Emergent robot behavior from the principle of homeokinesis

Ralf Der and Thomas Pantzer

Institut für Informatik,
Leipzig University, POB 920, D-04009 Leipzig

E-mail: `der@Informatik.uni-leipzig.de`

Abstract. The paper aims at demonstrating that basic behaviors can be acquired by an autonomous agent without the help by externally defined fitness functions or learning goals. For this purpose we introduce a new principle – homeokinesis – which is completely unspecific and yet induces specific, seemingly goal–oriented behaviors of an agent in a complex external world. The principle is based on the assumption that the agent is able of (i) learning an internal representation (self-model) of its current behavior and (ii) to adapt its behavior in such a way that the complexity gap between model and true behavior is minimized. Driven by the basic impetus of always moving and acting the agent in a noisy complex world develops smooth controlled behaviors as the minimal compromise between the high dynamical complexity of the world and the low complexity of the model. In these terms behavior may be called a byproduct (epiphenomenon) of the internal "desire" of the agent to be able of interpreting the world in simple terms.

1 Introduction

In the behavior based approach to robotics, there is a strong interest in exploiting the phenomenon of emergent behavior for the development of advanced robot controllers. One of the most clear cut ways in this direction is evolutionary robotics, cf. [15, 14, 10, 3, 9, 11, 2, 16, 12] which tries to breed behaviors oriented by conveniently defined fitness functions. Another way is reinforcement learning where the behaviors are driven into the desired directions by the rewards obtained from the environment. However despite some interesting achievements, there is also a lot of disappointment concerning the results. So far, all that has been evolved are rather basic behaviors like simple wall following. Moreover, most of the results obtained rest on computer simulations notable exceptions being several realizations achieved with real Khepera robots, cf. [5, 6] and [12] for an earlier review of the field. However, the way to a true *in vivo* evolutionary robotics of complex behaviors seems to be still very far.

Of course the main reason is the high complexity of the task so that reinforcement learning is slowly converging and evolution always has to run over many generations with many individuals. The solution to these difficulties seems to be the acceleration of those procedures. However our own practical experience with developping controllers for the Khepera robot have shown that the breeding – be it via artificial evolution or reinforcement learning – of behaviors itself faces a very basic problem not rooted in the specific kind of procedure by which it is realized.

We call this the design problem of the objective function. In artificial evolution this is the fitness function which must be designed by hand in view of the target behavior to be developed by the robot. However quite often the emerging behavior is radically different from expectation. In general, the design of the fitness function is seldom straightforward and often critical. Coevolution has recently been discussed as a way out, cf. [8, 7], but is not general and effective enough to cover the situations of interest for the present paper. On the other hand using reinforcement learning one faces the related problem that finding the right distribution of rewards is essential for success.

The common problem is that in order to breed behavior in a controlled way one has to formulate intermediate goals which have to be chosen carefully in order to drive evolution or learning in the desired direction. Certainly nature has not provided such luxury to evolution. Instead what we observe in nature are completely unspecific principles which produce domain-specific behaviors of a seemingly goal-oriented kind. The most famous principle in this respect probably is homeostasis, first introduced in physiology by Cannon as early as 1939 [1] in order to find a basic understanding of the goal-oriented reactions of the body in response to external perturbations. His observation was that the function of the complex interleaved circular chains of cause and effect is just to keep physiological quantities such as blood pressure at a constant level and that the reactions of the body so to say are a byproduct of this internal drive of the system to uphold its stationary state. This is a very general, unspecific principle not adapted to any specific process.

However, the principle has not found much applications in practice. In the present paper we will give some reasons for this situation and show that it is the focus on the stationary state of the system which restricts its applicability for the construction of realistic controllers. The principle of homeokinesis to be introduced is the dynamical *pendant* of homeostasis. Coarsely speaking it means that the internal goal of the agent is not the stabilization of a stationary state but of a definite internal kinetic regime. The latter is formulated via the **adaptive** self-model of the agent and we will show that the misfit between the model and the true behavior may be used as a permanently available learning signal which drives the emergence of behavior.

2 Self-model and kinetic regime

The most striking feature of both the homeostatic and the homeokinetic principle is the dominance of the internal perspective. Behavior is not bred by externally fixed goals but is a epiphenomenon of some internal drive of the agent. The internal world of the robot is built on its sensor values so that the above mentioned kinetic regime is to be formulated in sensor space. The controller of the robot may be viewed as a function mapping sensory inputs $x \in R^n$ to some outputs $y \in R^m$ governing the reaction of the robot to its specific environmental situation. The output determines the new situation (position) of the robot which on its hand corresponds to a new sensoric input. Thus, through the sensory-motor loop, each controller produces its specific trajectory $x(t)$ in sensor space which depends both on the environment and the starting position of the robot.

By way of example, let us assume the robot is instructed to keep a fixed velocity, the controller output determining the current turn speed w of the robot in physical space. Let us write the controller as (we assume that the controller does not refer to earlier sensor values as a PD controller would do. However the latter case can easily be integrated into the present concept)

$$w = g\left(\sum_{i=1}^{n} c_i x_i\right) \tag{1}$$

where $g() = \tanh()$ and the c_i are the parameters of the controller which are to be learned by the algorithm given below. The expression for the controller can be interpreted as a formal neuron. We assume $x_i \in [0,1] \quad \forall i$ without restriction of generality. Let us assume that the sensor values are read at regular intervals and that Δx is the change in sensor values in the current time step. Then the above mentioned dynamics in sensor space may be written as

$$\Delta x = F(x;c) \qquad x \in R^n, \quad c \in R^p \tag{2}$$

c being the parameter vector of the controller. The (stochastic) function F can be seen as the generator of a time evolution of x, i.e.

$$x(t+1) = x + F(x;c) \tag{3}$$

where $x = x(t)$. Eqs. (2), (3) may be interpreted as the update rule of a time discrete dynamical system, the structure of the function reflecting the complexity of the dynamics.

We may use the language of dynamical systems in order to classify different kinds of behavior. Obviously there are several behaviors corresponding to an attractor dynamics. The most prominent example is the case of wall following. If the controller supports a stable wall following behavior then after starting from any position in physical space the robot after some time will encounter a wall and follow it. This means that its sensors values are converging towards $x = x_{wall}$. In the language of dynamical systems we may say that x_{wall} is an attractor of the dynamics in sensor space. In the same way, trapping and following an object at a constant distance corresponds to an attractor dynamics in sensor space. Now we might be tempted to use this as a general tool for the domain invariant definition of "reasonable" behaviors in a given world. However this principle is not general enough. Consider a controller which is to guide a robot smoothly through a complicated maze with many doorways, corridors, and severals kinds of obstacles. This controller produces trajectories of a moderate (dynamical) complexity. What we mean by this is that they are somewhere between chaotic trajectories and trajectories produced by a kind of trivial controller like going straight ahead.

So we might formulate a domain-invariant principle for the learning of behavior in terms of a measure of dynamic complexity. However this is difficult to realize in practice in a domain-invariant way. The problem is that the level of complexity must be specified somehow. Our way is to define complexity via a self-model which is adaptive. Such a model might be realized by an adaptive predictor

$$\Delta x^P = \Phi(x;u) \tag{4}$$

predicting the change Δx in sensor values on the basis of the current values x, $u \in R^m$ being a set of parameters.

The function Φ defines a dynamics in the sense of eq. (3), i.e. the time discrete dynamical system

$$x^P(t+1) = x^P(t) + \Phi\left(x^P;u\right) \tag{5}$$

defines the time evolution of $\hat{x}$ which is the model behavior of our robot. Hence we may say that the function Φ defines the self-model of the agent. The function Φ is parameterized by the set of parameters $u \in R^m$. For u fixed the update rule (5) generates trajectories of a certain complexity which is depending on the value of u. The set of all trajectories for all (allowed) values of u may be seen as a complexity class which is defined by the *structure* of the function Φ. From the point of view of the principle of homeokinesis the structure of the function Φ defines the kinetic regime which the agent tries to reach and stabilize.

By way of example we consider the models given by

$$\Phi = 0 \tag{6}$$

and (writing Φ for each sensor explicitly)

$$\Phi_i = w \sum_j u_{ij} x_j \tag{7}$$

Eq. (6) underlies the hypothesis that there is no change in sensor values at all. The structure of the model so to say is the trivial or null structure. Eq. (7) states that every change in sensor values is proportional to the turn speed of the robot as given by the controller, the structure of the model being the bilinear form of the expression. Of course these are very simple models but we will see that they are sufficient for the emergence of various kinds of nontrivial behaviors.

In all cases of practical interest the complexity of the model behavior is inferior to that of the true behavior. Thus we speak of a dynamical complexity gap between model and reality. The idea now is to use the width of the complexity gap as learning signal for the adaptation of both the model and the controller. So the aim is to adapt behavior so that the complexity gap between true and model behavior is minimized.

3 The learning algorithm

This principle is simply realized by introducing an error function E which measures the distance between true and model behavior as

$$E = \frac{1}{2} \sum D_i^2 \tag{8}$$

where

$$D_i = \Delta x_i - \Delta x_i^P \tag{9}$$

and more explicitly

$$D_i = \Delta x_i - \Phi_i (x; u) \tag{10}$$

The observed change in sensor values Δx_i is a function of both the current sensor values x and the controller parameters c. Hence $E = E(x, c, u)$ so that the parameters c and u can bel learned by gradient descent. The update for u is

$$\Delta u_i = \varepsilon \sum_j (\Delta x_i - \Phi_i (x; u)) \frac{\partial}{\partial u_i} \Phi_j (x; u) \tag{11}$$

the partial derivatives being defined explicitly by the model. In order update c we need the derivatives

$$\frac{\partial}{\partial c_i} E = \frac{\partial E}{\partial w} \frac{\partial w}{\partial c_i} \tag{12}$$

The last derivative can be carried out explicitly using eq. (1), so that we are left with the derivative w.r.t. to w. This gradient information is gained by the following trick. We add a small perturbation $\phi(t)$ to the output w of the controller. This may be a noise with zero mean or a periodic oscillation. Assuming that the change of ϕ is small in the time interval between two sensor recordings, we may assume that the change in sensor values Δx and hence E is a function of the current value of ϕ. Writing E as $E = E(x,c,u) = E[x,w,u]$ the effect of the perturbation is

$$E[x, w+\gamma\phi, u] = E + \gamma\phi\frac{\partial E}{\partial w} + \ldots \tag{13}$$

where $E = E[x,w,u]$ and γ is a strength parameter. Now ϕ is to be chosen such that in the average over a longer time interval we have $\overline{\phi} << \overline{\phi^2}$. Then we may write

$$\overline{E\phi} \approx \gamma\overline{\phi^2}\frac{\partial E}{\partial w} \tag{14}$$

so that the gradient can be obtained by a time average over the product of $\phi(t)E$.

Collecting the results we have the following learning algorithm: We read the sensor values x_i at regular time intervals of length τ. The current turn speed of the robot is put to

$$w = g\left(\sum c_i x_i\right) + \gamma\phi(t) \tag{15}$$

where $x_i = x_i(t)$ is the value of the infrared sensor i at the current time step. Calling $x_i(t+\tau) - x_i(t) = \Delta x_i$ the update of the controller parameters is

$$\Delta c_i = -\varepsilon g' x_i \phi(t) E \tag{16}$$

where $E = \sum_j (\Delta x_j)^2$ in the case of the model given by eq. (6) which we have used mostly in the experiments below.

4 Experiments with physical Khepera robots

We have implemented the above algorithm on chip of a Khepera miniature robot and verified the emergence of behavior under different environmental conditions. The robot moves with a constant velocity, the controller defining the turn speed according to eq. (15). In the Figures given below we plot the values of the controller parameters c_i over time during the learning process. From this one may draw conclusions on the learning behavior of the robot.

In a typical experiment the parameters of the controller are initialized either randomly or with $c_i = 0 \quad \forall i$. The robot will crash usually after a short time. Then it must be removed from the wall by hand. After several such trials the robot very soon learns to either avoid the wall or to follow it. The latter behavior is stable as long as there are no dramatic changes in the environment. This could be corroborated in a long time experiment (In this experiment the controller was $w = g(\sum_i \theta_i (x_i - c_i) x_i)$ where θ_i is $+1$ for the sensors at the right and -1 at the left hand side.), cf. Fig. 1. It should be mentioned that during the long time experiment we did not change change the learning parameters, in particular we did not use any "cooling" of the learning rate ε. The remaining experiments presented in the Figures were carried out with the controller of eq. (15) with parameters initialized as $c_i = 0 \quad \forall i$ so that in the beginning the robot moves straight ahead ignoring its sensor values. The robot learns very rapidly to follow the wall, if one starts it a few times under a convenient angle to the wall.

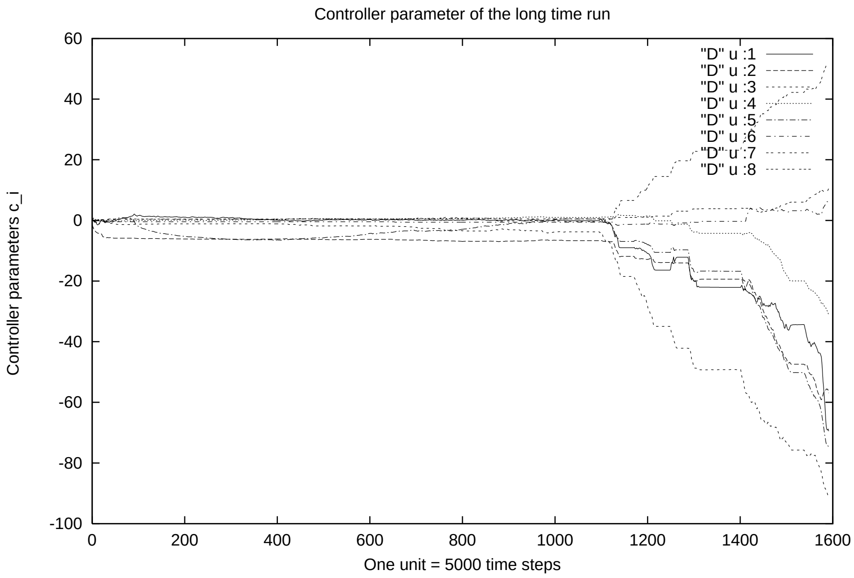

Figure 1. The controller parameters $c_i, \quad i = 1 \ldots 8$ for the long time (three days) run with about 8 km of robot path, the robot moving with a velocity of roughly $30mm/s$. The Figure displays the parameters c_i of the controller. After the initial period the robot was running for nearly two days in the maze in a stable wall following mode. During the wall following epoch learning essentially influences only the parameters connected with the sensors at the left hand side. Then due to some perturbation of the maze the robot was trapped and developed a rotating behavior. However the learning gets out of control in this situation. This is reflected by the divergence of the parameters.

It is interesting that when following the wall for some time the controller will change the behavior mode in the sense that it finds a new distance to the wall under which control is more stable. For smooth walls this means that it steers with two sensors, choosing the distance to the wall such that the sensor closest to the wall is at maximum value and the next one is at half maximum. This is a nontrivial effect and can be understood in terms of minimizing the influence of the noise as explained in [4].

A further point of general interest is that the robot changes its behavior depending on the environmental conditions. Typically one can learn it to balance a ball, cf. Fig. 3, and than put the learned robot under some angle close to the wall. Of course the robot will show the same reaction as with a ball, i.e. it turns until the wall is in front of it and then bangs against the wall. However after helping it severals times out of this crash situation it will learn to follow the wall afterwards.

It is also interesting that the controller may cope with the (simulated) damage of a sensor as demonstrated with the ball balancing behavior which the robot also learns if one of the sensors involved is damaged.

5 Conclusions

Our experiments with a real Khepera robot clearly demonstrated that the homeokinetic principle is a safe basis for the emergence of behavior with an autonomous robot. Depending on the

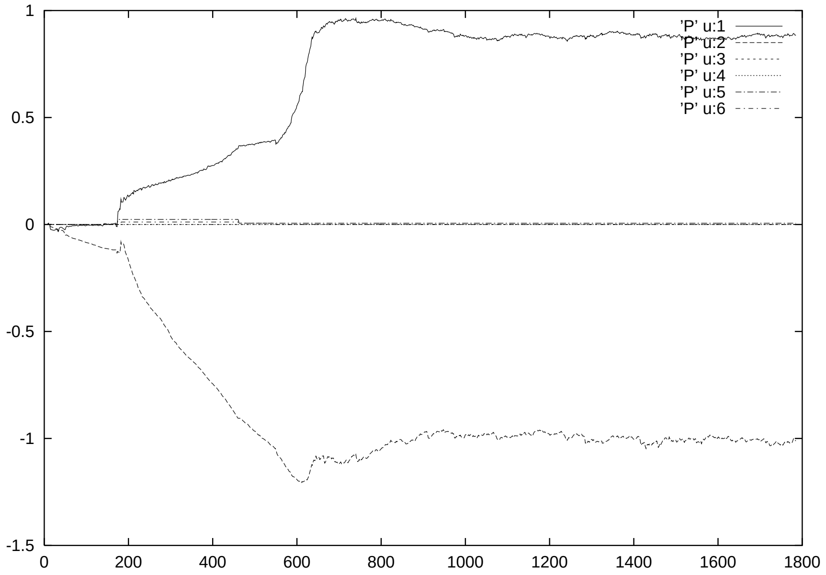

Figure 2. Wall following behavior. The robot sees the wall to its left. Plotted are the values of the controller parameters c_1 through c_6, one unit being about 2 cm. After the very short initial learning period the robot followed the inner wall of a circle of about 80 cm. After about 200 steps the robot was transmitted to the outer wall of a circle of about 60 cm. The robot rapidly adapts to the new situation and stays stable in the new behavior for about 400 steps. Initiated by an external perturbation starting by step 600 there is a transition into a more stable (closer) wall following modus. The latter behavior is completely stable.

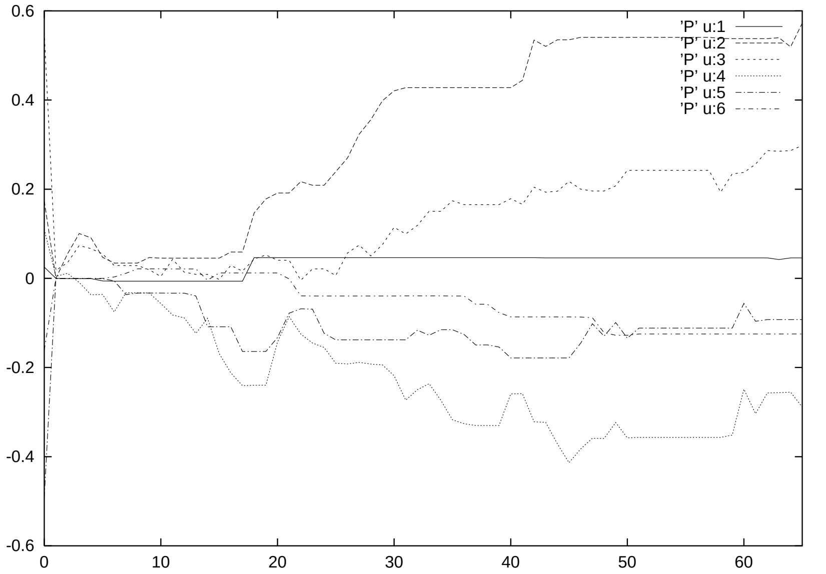

Figure 3. Ball balancing behavior. The robot moving with a constant velocity is presented a table tennis ball which is stabilized for some time by hand in front of it. After about 40 cm (one step covers about 2 cm) the robot is able of balancing the ball as long as its speed is sufficient to keep up with it. The plot displays the parameters c_1 through c_5 of the controller.

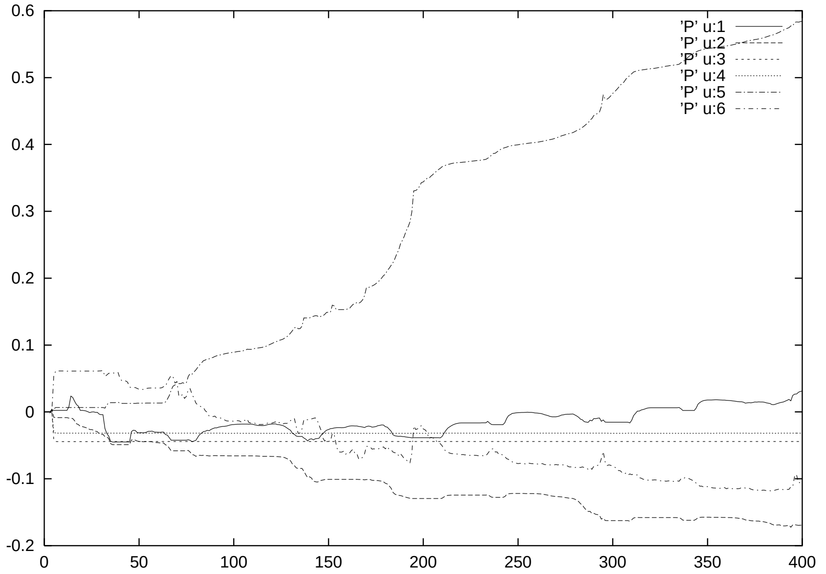

Figure 4. Corridor following: The robot moves in a circular corridor with diameter of about 80 cm. The width of the corridor varies from twice the robot diameter to a few millimeters wider than the robots diameter. The robot nearly immediately learns to follow the corridor. In the closer parts it keeps to the middle of the corridor while in the wider parts it keeps to the inner wall of the corridor. Parameter values (depicted are c_1 through c_6) are not symmetrical because of the different sensor characteristics.

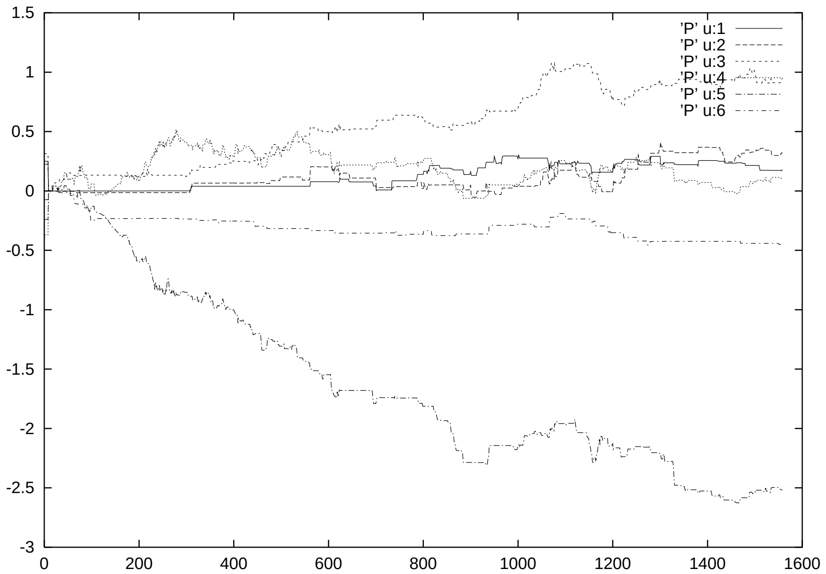

Figure 5. The right-hand front sensor was damaged nevertheless the robot learns a stable ball balancing behavior.

environmental conditions [1] we have observed a number of different behaviors with always the same adaptive controller and learning parameters. In particular we observed behaviors like

- Collision avoidance,
- Ball balancing,
- Navigation through corridors,
- Different modes of wall following,
- Navigation capabilities in chaotic mazes.

Moreover we observed the convergence towards best working conditions resulting from a nontrivial combination of nonlinearity and noise effects. Learning is found to be extremely fast and permanently alert while under stationary environmental conditions it is convergent and reproducible.

Our future work will be devoted to testing the current controller under new environmental conditions while including the camera and the ambient light sensors. We will also study emerging collective behaviors with several robots in a common environment each one being controlled by the adaptive homeokinetic controller. Moreover we also would like the robot to develop the "desire" to move which at the time being is given it from outside since we put the forward velocity to a prefixed value. This can be achieved by a predictor looking backward in time which leads to an amplification of the noise effects, cf. [4].

References

[1] W. B. Cannon. *The wisdom of the body*. Norton, New York, 1939.

[2] D. Cliff, P. Husbands, and I. Harvey. Analysis of evolved sensory-motor controllers. Technical Report Cognitive Science Research Paper CSRP264, School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, England, UK, 1992.

[3] D. Cliff, P. Husbands, and I. Harvey. Evolving visually giuded robots. In [13], pages 374–383, 1992.

[4] R. Der, U. Steinmetz, and F. Pasemann. Homeokinesis - a new principle to back up evolution with learning. In M. Mohammadian, editor, *Computational Intelligence for Modelling, Control, and Automation*, volume 55 of *Concurrent Systems Engineering Series*, pages 43 – 47. IOS Press, 1999. URL=http://www.informatik.uni-leipzig.de/ der/Veroeff/wienfin3.ps.

[5] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 26(3):396–407, 1996.

[1] The environmental conditions include the human interferences like choosing the starting position of the robot and the way of restarting it after helping it out of a crash situation.

[6] D. Floreano and F. Mondada. Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11:1461–1478, 1998.

[7] D. Floreano, F. Mondada, and S. Nolfi. Co-evolution and ontogenetic change in competing robots. *Robotics and Autonomous Systems*, 1999.

[8] D. Floreano and S. Nolfi. God save the red queen! competition in co-evolutionary robotics. In J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[9] I. Harvey. Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. Technical Report CSRP 222, University of Sussex, Brighton, UK, 1992.

[10] I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In [13], pages 364–373, 1992.

[11] P. Husbands, I. Harvey, and D. Cliff. Analysing recurrent dynamical networks evolved for robot control. Technical Report Cognitive Science Research Paper CSRP265, School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, England, UK, 1993.

[12] M. J. Mataric and D. Cliff. Challenges in evolving controllers for physical robots. *Journal of Robotics and Autonomous Systems*, 19(1):67–83, Oct. 1996.

[13] J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, editors. *From Animals to Animats 2: Proceedings of the 2nd International Conference on Simulation of Adaptive Bahavior*. MIT Press, 1992.

[14] S. Nolfi. Evolving nontrivial behaviours on real robots: A garbage collecting robot. Technical Report 96-04, Institute of Psychology, C.N.R. Rome, 1996.

[15] S. Nolfi and D. Parisi. Evolving non-trivial behaviors on real robots: An autonomous robot that picks up objects. *Lecture Notes in Computer Science*, 992:243, 1995.

[16] L. Steels. Emergent functionality in robotic agents through on-line evolution. In R. A. Brooks and P. Maes, editors, *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems $ArtificialLifeIV$*, pages 8–16, Cambridge, MA, USA, July 1994. MIT Press.