Sensor Evolution In A Homeokinetic System

Kai Labusch and Daniel Polani

Institut für Neuro- und Bioinformatik
Universität Lübeck
D-23569 Lübeck, Germany

Abstract

We study simulated Braitenberg agents controlled by a homeokinetic dynamic which evolve the ability to discriminate between two different types of objects. The free parameters of the homeokinetic control are varied by an evolutionary strategy. Two mirrored scenarios are used to show adaptation. Using a simple test scenario, we are able to evaluate whether agents are able to extract relevant information from their sensor values using only the temporal dynamics of the observed objects.

1 Introduction

In the last years it has become increasingly clear that one key to the understanding of embodied intelligence is the emergence of intelligent behavior due to the coevolution of sensors, actuators and control in an agent (Cariani 1992; Dautenhahn et al. 2001). An important part is taken by the evolution of morphological aspects (Sims 1994; Kawai and Hara 1998; Lipson and Pollack 2000). However, the real power of embodied evolution lies in the development of new ways to tap environmental information via evolution of suitable sensors. As opposed to evolving powerful morphological properties, attaining powerful sensors through evolution is not expensive for the individuals from a cost perspective and can be of significant usefulness for the survival of an individual. Therefore, the evolution of sensors is a crucial element in the process of embodied evolution and the creation of complex behaviors connected with it. Most of the studies consider functionally evolving sensors.

In this paper, though, we wish to explore the possibilities of the evolution of a "metasensorics". In other words, we are interested in the evolution not of the sensor itself (i.e. the "physical" interface between an agent and its environment), but of the way the information flow coming through a predefined sensoric interface can be interpreted by the agent. This information can then be seen as coming from a nonphysical metasensor that acts as a preprocessing filter interface between the physical sensor and the remaining agent control. The scenario will take place in the virtual agent environment XRaptor[1] (Bruns et al. 2001).

To contravene the possible influence of a specifically designed agent architecture, a general principle, the homeokinetic control mechanism from (Der 2000) is used to control agent behavior. It does not model explicitly the distinction of different types of sensoric input and is therefore useful in studying how a meta-sensoric approach is able to implicitly gain goal-relevant information from unspecific sensor values.

[1] As we operate in a virtual world, the use of the word "physical" sensor means a virtual physical sensor.

2 Scenario

2.1 Overview

We use two evolution scenarios. In both of them there exist two types of objects, immobile fruits and moving agents. Agents have an internal energy level, whose falling below a certain threshold will cause agents to die. In every time step, agents lose energy, thus they require interaction with other objects to survive. If an agent dies, it is replaced by reproduction and mutation of the remaining population. If an agent eats a fruit, a new fruit is created at a random location. In the first scenario eating fruits is beneficial and colliding with other agents is harmful. In the second scenario eating fruits will cause energy loss and colliding with other agents will result in an energy gain. The agents are controlled by a homeokinetic dynamics (Der et al. 1999; Der 2001). The free parameters of the dynamics are varied by an evolutionary strategy. To evaluate the behavior of the agents generated by the evolution, there is a third scenario consisting of eight specifically designed situations. Here we use the attention focus of the agents to measure the adaptation process.

2.2 Agent architecture

2.2.1 Actuators and Sensors

We use the XRaptor software as the simulation framework (Bruns et al. 2001). Simulated agents similar to Braitenberg's vehicle 3c (Braitenberg 1984) move around in a virtual, two-dimensional world consisting of other agents and fruits. The agents have two motors left and right from their orientation axis that can be controlled independently. The motor values can be set in the range of $[-1, 1]$. Setting both motors to same positive value will cause the agent to move forward ahead while setting both motors to the same negative value will cause the agent to move backward ahead. Setting the motors to different values will cause the agent to turn appropriately (Liese et al. 2001). The sensorics provides a list of positions of other agents or fruits up to a certain distance from the agent but not the type of these objects. The object positions are given relatively to the agent position and direction.

2.2.2 Agent Control

The update of the agents during a time step consists of three phases and is an extension of the homeokinesis mechanism developed in (Der et al. 1999). First the sensor readings are analyzed, second the actuators are adjusted, and third world and control model are trained. A detailed description of the mathematics of this process can be found in App. A.

The first task is accomplished by a simple world model which tries to predict the change of sensor values in the current time step based on the change of sensor values in the previous time step. From the change of sensor values in the previous time step

$$x = s^{t-1} - s^{t-2}$$

one obtains the prediction for the change of sensor values for the current time step

$$x^p = Wx$$

by the (linear) world model W. In the current time step the sensors supply a set of object positions

$$s_i^t = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2,\ i = 1 \ldots n$$

which induce a set of sensor value changes

$$x_i^t = s_i^t - s^{t-1}$$

For each observed object i the actual changes in sensor readings are compared to the predicted change x^p. The objects are then matched to that readings. That object i whose sensor change x_i^t fits best the prediction of the world model becomes the new *focus of attention* which will be used to train world and control model in the third phase of the update. Before that, the respective object position s_i^t is used to adjust the actuators of the agent by the control values

$$c^t = \begin{pmatrix} m_l \\ m_r \end{pmatrix} = C\left(s^t\right) + \gamma\varphi^t \ .$$

C is a (linear) mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$, φ^t is a time-dependent perturbation function whose magnitude is controlled by scaling factor γ. The function φ^t introduces slight variations into the control dynamics to probe the local dynamics of the environment (Der et al. 1999, and also Sec. A). If the actual prediction error does not exceed the expected prediction error which is calculated from the memory of the agent, the third phase is entered. The world model W^t is trained by gradient descent to minimize the prediction error in the next time step (Sec. A.3) The control model is subjected to a training with the same goal, though this procedure turns out to be more involved (Sec. A.4). For a full treatment of the calculation, see App. A

2.3 Evolution

The free parameters of the homeokinetic dynamic (see Sec. 2.3) are varied by an evolutionary strategy. The objective function is the ability to survive in the given environment which is measured by the amount of life energy an agent has. Because eating fruits gives an energy gain while colliding with another agent incurs a heavy penalty (in most cases, death), a necessary by-product of evolution is the ability to distinguish between both fruits and agents. To exclude the possibility that the model favors one type of object, we also set up a second, mirrored scenario where fruits are poisoned, while having contact to other agents is beneficial.

In both scenarios agents are randomly relocated after a certain period of time to ensure that the behavior evolved is robust. Otherwise, the selection pressure diminishes with time; for, in that case, an increasing number of noncompetitive agents finds itself in favorable positions by chance. For instance, in the second scenario, agents with a weaker behavior may find themselves in clusters of other agents, and thus be relieved from the need to actively seek collision partners. Though relocation is not perfectly realistic, it acts as a simple model for regularly changing environments.

We do not evolve discrete generations but perform a steady state evolution. A single agent which dies for lack of energy is replaced by recombination and mutation of the remaining population. Agents having a high energy level are more likely chosen for recombination.

Due to switch of attention focus or random replacement of the agents, the sensor readings regularly experience discontinuities. These discontinuities contradict the continuity assumptions that underlie the homeokinetic prediction model. Thus, they occasionally cause the divergence of the homeokinetic control parameters. Such a divergence can also occur if an agent is born with the wrong learning rates for world or control model. Such an agent is considered a “freak of circumstances” (or “of nature”, respectively) and killed, whereupon a new agent is created in the manner described previously.

The genome of an agent used in the evolution consists of the following real-valued entries:

α_W	learning rate of the world model
α_C	learning rate of the control model
γ	weight of the perturbation function
f_l	left perturbation frequency for φ
f_r	right perturbation frequency for φ
n	number of viewed objects in each time step
T	maximum relation between expected and actual prediction error (Sec. A)

The entries of the genome are recombined using crossover exchanging the entire value. Each entry of the genome mutates with a Gaussian distribution having a different standard deviation according to the strategy parameters $s = \begin{pmatrix} s_{\alpha_W} & s_{\alpha_C} & s_{\gamma} & s_{f_l} & s_{f_r} & s_n & s_T \end{pmatrix}^T$ These strategy parameters underlie intermediate recombination. The evolutionary strategy varies the genome of the agents as well as their strategy parameters which mutate with the global standard deviation s_{global} (Bäck et al. 1991).

2.4 Evaluation scenario

We used a special scenario to evaluate the behavior of the agents. Agents are selected for this scenario based on their age. In the evaluation scenario, agents are confronted with eight different situations. Beside the test candidate there are two other objects in each situation. On the one hand the test partner, a specifically designed agent moving around in circles, on the other hand a single fruit. The circular movement of the test partner is a simplification of the spiralling agent movement patterns often found during evolution.

Each agent tested in the evaluation scenario starts with the same position and orientation. The test partner and the fruit are positioned in the same distance to the test candidate, from the agent’s point of view in opposite directions. In this scenario, no bonuses or penalties are given. For each situation there is a mirror situation. Therefore, agents cannot distinguish fruits from agents by their position. To evaluate the information processing of the tested agents, their focus of attention is logged, i.e. the number of time steps is counted where the fruit (or the test partner, respectively) is in the agent’s focus of attention.

3 Results

As described above, we use two contrary scenarios to achieve opposite goals, namely beneficial fruits and harmful agents versus harmful fruits and beneficial agents. For each of these we performed 40 independent runs using a single parameter set for each scenario type, changing only the random seed. Each evolution experiment runs for 250000 time steps and generates between 15000 and 30000 agents. A single evolution run typically takes a couple of hours on a Pentium machine with 700 MHz. The XRaptor simulation requires in the order of magnitude of around 10–20 MByte.

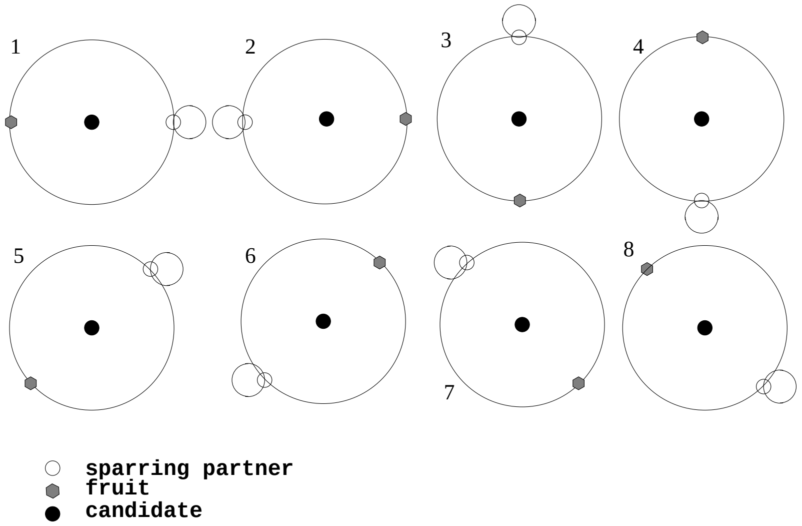

Figure 1: The eight evaluation scenarios. The evaluation candidate agent (black circle) is placed in the middle. A fruit and a "sparring partner" agent performing a fixed circle are placed at the same distance in opposite directions w.r.t. to the candidate.

	beneficial fruit scenario	poisoned fruit scenario
number of agents	55	34
number of fruits	34	120
energy loss per time step	80	80
collision energy	550000	17000
nutritive value of the fruits	34000	-700000

In our present analysis, we concentrate on a specific aspect of the agents' behavior, namely their attention focus. Future studies will include further aspects. We analyzed the behavior of two different sets of agents for each scenario. First we selected randomly 0.1% out of the entire population of agents for each scenario. Second we selected the best 0.1% of the entire population of agents according to the agents lifespan for each scenario.

These agents were exposed to the evaluation scenario. For a single agent run we calculate the ratio between the number of time steps the agent focused its attention on the fruit and the number of time steps the agent focused its attention on its test partner. We say the agent focuses its attention on a certain object if it selects the sensor values of this object to train its world and control model.

First we consider the randomly chosen agents. The first histogram below shows the distribution of the attention ratio of agents evolved to survive in an environment containing harmful agents and beneficial fruits. The second histogram below shows the distribution of the attention ratio of agents evolved to survive in an environment containing beneficial agents and harmful fruits.

Considering these histograms one can see that the peak indicating high attention to other agents grows as well as the peak indicating a high attention to fruits if the agents are forced by selective pressure to collide with other agents. The indifferent middle peak of the histogram decreases slightly.

Second we consider the best agents. The two histograms below show the distribution of the attention ratio of the best 0.1% of the population of each type of scenario. The selection criterion is the

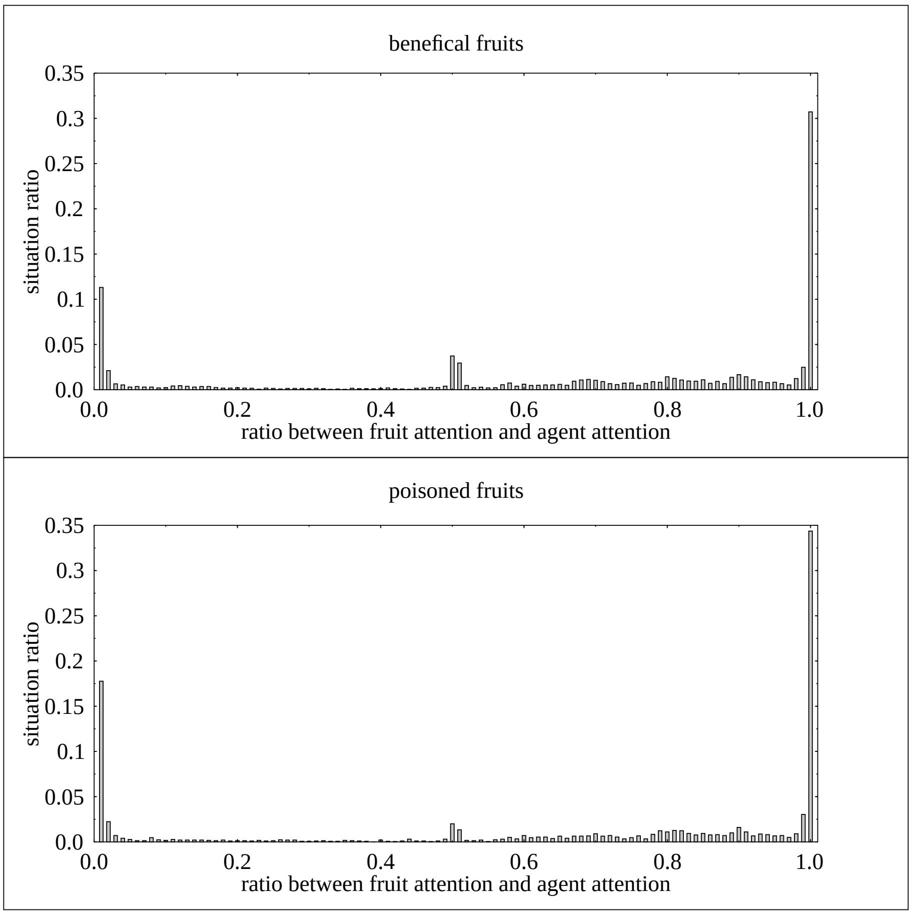

Figure 2: Attention ratio histograms for randomly selected agents from different evaluation runs. For each evaluation run the time ratio of the run is determined during which the agent attention focus is on the fruit (the rest of the run the attention focus can be assumed to be on the “sparring partner”). For a selection of agents created during the evolution runs (see text) these attention time ratios are then determined in all evaluation scenarios. This results in a distribution of attention time ratios which is plotted as histogram.

lifespan of the agents. Considering agents evolved to gain energy in an environment containing beneficial agents there are conspicuous attention peaks in the middle of the histogram. These do not occur in the histogram of the best agents evolved to gain energy in an environment containing harmful agents. Another conspicuous difference is the decrease of the peak indicating high attention to fruits. This obvious differences in the histogram indicate a more complex behavior which involves both objects in the environment the fruit and the test partner. The cause of the regular distances between the smaller peaks in the middle region has not been determined yet. This might be a result of the circular motion of the test partner. We will analyze this more precisely in future work.

The results show, that the agents evolve different behaviors if they are bred from different environments. The agents are forced to learn to treat immobile fruits and other, moving agents differently though neither the underlying model makes assumptions about these different types of objects nor the available sensors indicate what kind of object is scanned. An important aspect is that the agents were just evolved to survive in the given environment and not primarily to recognize a

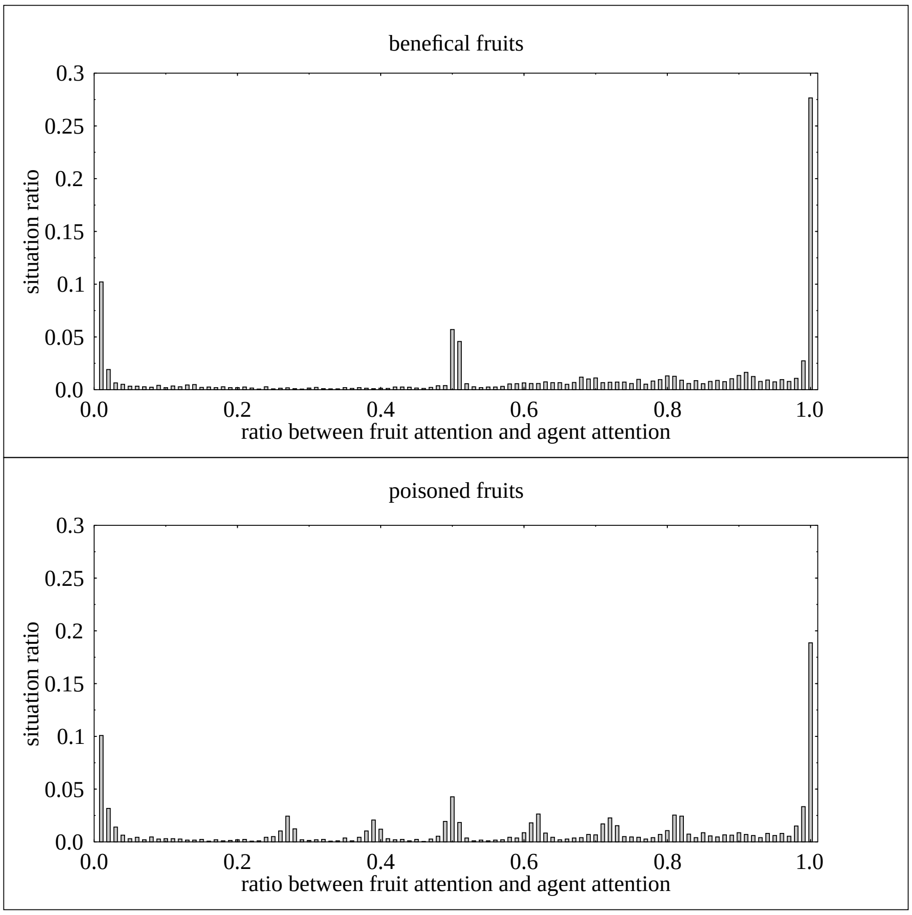

Figure 3: Attention probability histograms for best agents of two different run scenarios.

certain object type.

The findings demonstrate that agents develop the ability to interpret the sensors in a different way to acquire information relevant to survive. Hence meta-sensor evolution takes place as a byproduct of the adaptation process of the agents to their environment.

A The Extended Homeokinesis Model

We extend the controller architecture introduced in (Der et al. 1999) to more dimensions and add an attention mechanism as well as other mechanisms to cover special problems of the given environment.

A.1 Sensors

Let $s_i^t = \begin{pmatrix} x \\ y \end{pmatrix}$, $i = 1, \ldots, n$ be the sensor values for the next n objects in the agent's environment. The sensor readings are processed for a single object only, the *focus of attention*. We use those that fit best the prediction by the world model. This set of sensor values is denoted s^t in time step t. Let further $x = s^{t-1} - s^{t-2}$ be the actual change of the sensor values in the previous time step and $x_i^t = s_i^t - s^{t-1}$ be the actual change of the sensor values in the current time step for a given object i. Set $x^p = Wx$ to be the predicted change for the current time step according to the *world model* (weight matrix) W. We choose $D = x_i^t - x^p$ and get $E = D^T D = \left\|x_i^t - x^p\right\|^2$ as the prediction error by the world model for time step t. The homeokinetic principle now attempts not only to predict the future sensor input as accurately as possible, but also to choose actions which make the future sensor input as predictable as possible. This corresponds to choosing not only the prediction mapping W but also the control mapping C (see below) to minimize E.

In our model, we furthermore compare this prediction error to the expected error. Let $\overline{E} = \frac{\sum_{i=1}^{S} E^{t-i}}{S}$ be the expected error for the current time step. S denotes the number of time steps in the agent's memory. For $\overline{E} > \frac{E}{T}$ (T the maximum rate between actual and expected prediction error) learning takes place neither for the world model nor for the controller. This mechanism reduces discontinuities during sensor operation (see also Sec. 2.3). A parameter n controls the number of objects taken into account. Both n and T are entries of the agent genome.

A.2 Actuators

Agents have two motors, one on each side of an agent's orientation axis. Let $c^t = \begin{pmatrix} m_l \\ m_r \end{pmatrix}$ be the control vector of the current time step, which is computed by $c^t = \widehat{c} + \gamma\varphi^t$. $\widehat{c}$ is given by$\widehat{c} = Cs^t$ where C is the (linear) control mapping and φ^t a periodic perturbation function added to the controller output to probe the characteristics of the environment, scaled by γ.

A.3 World model

The world model predicts the change x^p of the sensor values for the next time step. As mentioned above, we use the linear mapping W for prediction according to $Wx = x^p$. The weights are adjusted by gradient descent. As above, with $D = x - x^p$ we want to minimize the prediction error $E = D^T D$. The corresponding gradient descent rule

$$W^{t+1} = W^t + \alpha_W \left[\frac{\partial E}{\partial w_{ij}}\right] \tag{1}$$

leads us to $W^{t+1} = W^t + \alpha_W \left[\left(x_l^t - Wx\right) x^T\right]$ (with l the focus of attention) as the learning rule for the world model. The learning rate α_W is an entry of the genome.

A.4 Control model

The control model computes the best action for the next time step. The best action minimizes the prediction error. We use a linear feed-forward network for this task as well $C(s) = Cs$ with C the weight matrix, and s the current sensor values. The weights are learned by gradient descent.

Because we want to minimize E,

$$C^{t+1} = C^t + \alpha_C \left[\frac{\partial E}{\partial c_{ij}}\right] \tag{2}$$

is the corresponding gradient descent. In this case we cannot obtain the learning rule directly because the function to obtain the gradient of involves the environment. Therefore one takes a closer look at $\frac{\partial E}{\partial c_{ij}}$.

Decomposing this into

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial}{\partial c_{ij}} \left(D^T D \right) = \left(\frac{\partial}{\partial c_{ij}} D^T \right) D + D^T \left(\frac{\partial}{\partial c_{ij}} D \right) \tag{3}$$

we get

$$\frac{\partial E}{\partial c_{ij}} = 2 \sum_k \left(D_k \frac{\partial}{\partial c_{ij}} D_k \right) \tag{4}$$

$\frac{\partial}{\partial c_{ij}} D_k$ can be written as

$$\frac{\partial}{\partial c_{ij}} D_k = x_k - [Wx]_k = G\left(s^{t-1}, C\left(s^{t-1}\right)\right)_k - s_k^t - [Wx]_k \tag{5}$$

where s^{t-1} is the knowledge of the agent about the state i.e. sensor input of the environment in the previous time step and $C\left(s^{t-1}\right)$ its action based on that knowledge. By G, the global world function, we summarize all other external factors having influence on the state of the environment leading to s^t, the knowledge of the agent about the current state of the environment. Now we can split up the derivative as follows

$$\frac{\partial}{\partial c_{ij}} D_k = \left. \frac{\partial}{\partial \widehat{c}} D\left(x, \widehat{c}, W\right)_k \right|_{\widehat{c}=C(s)} \frac{\partial}{\partial c_{ij}} C\left(s\right) \tag{6}$$

The latter part of the derivative can be given directly:

$$\frac{\partial}{\partial c_{ij}} C\left(s\right) = s_j \widehat{e_i} \tag{7}$$

where $\widehat{e_i}$ is the i-th canonical unit vector $(0, \ldots, \underset{i}{1}, \ldots, 0)$. To obtain the first part we perturb $\widehat{c}$ by a function φ where $\|\overline{\varphi}\| \approx 0$ and $\lambda_{min}\left[\overline{\varphi\varphi^T}\right] \gg \|\overline{\varphi}\|$ λ_{min} is the smallest eigenvalue of the matrix $\overline{\varphi\varphi^T}$ it corresponds to the selection of the scalar perturbation function in (Der et al. 1999). The choice (13) holds in the generic case this condition. Considering the Taylor series w.r.t. φ

$$D\left(x, \widehat{c} + \gamma\varphi, W\right)_k = D\left(x, \widehat{c}, W\right)_k + \frac{\partial}{\partial \widehat{c}} D\left(x, \widehat{c}, W\right)_k \gamma\varphi + o(\gamma\varphi) \tag{8}$$

we get

$$D\left(x, \widehat{c} + \gamma\varphi, W\right)_k \varphi^T = D\left(x, \widehat{c}, W\right)_k \varphi^T + \frac{\partial}{\partial \widehat{c}} D\left(x, \widehat{c}, W\right)_k \gamma\varphi\varphi^T + o(\gamma\varphi)\varphi^T \tag{9}$$

Averaging over time

$$\overline{D\left(x, \widehat{c} + \gamma\varphi, W\right)_k \varphi^T} = \underbrace{\overline{D\left(x, \widehat{c}, W\right)_k \varphi^T}}_{\approx 0} + \frac{\partial}{\partial \widehat{c}} D\left(x, \widehat{c}, W\right)_k \overline{\gamma\varphi\varphi^T} + o(\overline{\gamma\varphi\varphi^T}) \tag{10}$$

we can use the properties of the error function and obtain

$$\overline{D\left(x, \widehat{c}+\gamma\varphi, W\right)_k \varphi^T} \left[\overline{\gamma\varphi\varphi^T}\right]^{-1} \approx \frac{\partial}{\partial\widehat{c}} D\left(x, \widehat{c}, W\right)_k \tag{11}$$

Finally we get

$$C_{ij}^{t+1} = C_{ij}^{t} + \alpha_C \sum_{k=1}^{s} \left[D_k \left(\overline{D\left(x, \widehat{c}+\gamma\varphi, W\right)_k \varphi^T} \left[\overline{\gamma\varphi\varphi^T}\right]^{-1} s_j \widehat{e_i} \right) \right] \tag{12}$$

as the learning rule for the controller. Here we used

$$\varphi\left(t\right) = \gamma \left(\begin{array}{c} \cos\left(f_l t\right) \\ \cos\left(f_r t\right) \end{array} \right) \tag{13}$$

as perturbation function. The learning rate α_C, the perturbation factor γ, and the frequencies f_l, f_r are entries of the genome.

References

Bäck, T., Hoffmeister, F., and Schwefel, H.-P., (1991). A Survey of Evolution Strategies. In Belew, R. K., and Booker, L. B., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, 2–9. San Diego: Morgan Kaufmann.

Braitenberg, V., (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge: MIT Press.

Bruns, G., Polani, D., and Uthmann, T., (2001). Eine virtuelle kontinuierliche Welt als Testbett für KI-Modelle. *Künstliche Intelligenz*, 1:60–62.

Cariani, P., (1992). Some epistemological implications of devices which construct their own sensors and effectors. In Varela, F. J., and Bourgine, P., editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, 484–493.

Dautenhahn, K., Polani, D., and Uthmann, T., (2001). Special Issue on Sensor Evolution. *Artificial Life Journal*, 7(2).

Der, R., (2000). Selforganized robot behavior from the principle of homeokinesis. In Groß, H.-M., Debes, K., and Böhme, H.-J., editors, *Proc. Workhop SOAVE '2000 (Selbstorganisation von adaptivem Verhalten*, vol. 643 of *Fortschritt-Berichte VDI, Reihe 10*, 39–46. Ilmenau: VDI Verlag.

Der, R., (2001). Self-organized acqusition of situated behavior. *Theory Biosci.*, 120:1–9.

Der, R., Steinmetz, U., and Pasemann, F., (1999). Homeokinesis – A new principle to back up evolution with learning. In Mohammadian, M., editor, *Computational Intelligence for Modelling, Control, and Automation*, vol. 55 of *Concurrent Systems Engineering Series*, 43–47. IOS Press.

Kawai, N., and Hara, N., (1998). Formation of morphology and morpho-function in a linear-cluster robotic system. In Pfeifer, R., Blumberg, B., Meyer, J.-A., and Wilson, S. W., editors, *From Animals to Animats. Proc. of the 5th Int. Conference on the Simulation of Adaptive Behavior, SAB'98*, 459–464.

Liese, A., Polani, D., and Uthmann, T., (2001). Study of the Simulated Evolution of the Spectral Sensitivity of Visual Agent Receptors. *Artificial Life (Special Issue on Sensor Evolution)*, 7(2).

Lipson, H., and Pollack, J., (2000). Evolution of machines. In *Proceedings of 6th International Conference on Articial Intelligence in Design, Worcester MA, USA.*

Sims, K., (1994). Evolving 3D Morphology and Behavior by Competition. In Brooks, R., and Maes, P., editors, *Proc. Artificial Life IV*.