Chapter 8
Robot Learning by Guided Self-Organization

Georg Martius, Ralf Der, and J. Michael Herrmann

8.1 Introduction

Self-organizing processes are not only crucial for the development of living beings, but can also spur new developments in robotics, e. g. to increase fault tolerance and enhance flexibility, provided that the prescribed goals can be realized at the same time. This combination of an externally specified objective and autonomous exploratory behavior is very interesting for practical applications of robot learning. In this chapter, we will present several forms of guided self-organization in robots based on homeokinesis.

Self-organization in the sense used in natural sciences means the spontaneous creation of patterns in space and/or time in systems consisting of many individual components. This involves the emergence, meaning the spontaneous creation, of structures or functions that are not directly explainable from the interactions between the constituents of the system. Examples are for instance spontaneous magnetization, convection patterns and reaction diffusion systems leading to the wonderful coloring of shells or animal coats. For robotic applications it is important to translate self-organization effects to a single robots considered as complex physical systems consisting of many constituents that are constraining each other in an intensive

Georg Martius · Ralf Der
Max Planck Institute for Mathematics in the Sciences,
Inselstraße 22, D-04103, Leipzig, Germany
e-mail: {martius,ralfder}@mis.mpg.de

J. Michael Herrmann
Bernstein Center for Computational Neuroscience,
Am Faßberg 17, 37077 Göttingen, Germany
Institute for Perception, Action and Behaviour, School of Informatics,
University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, Scotland, U.K.
e-mail: michael.herrmann@ed.ac.uk

M. Prokopenko (ed.), *Guided Self-Organization: Inception,*
Emergence, Complexity and Computation 9,
DOI: 10.1007/978-3-642-53734-9_8,

manner. This is what homeokinesis (Der 2001; Der and Liebscher 2002; Der and Martius 2012) or information theoretic approaches (Martius et al. 2013; Klyubin et al. 2005) to behavioral self-organization are after.

Homeokinesis or homeokinetic learning is based on a dynamical systems formulation of sensorimotor loops and introduces an objective function, called the time-loop-error. Intuitively it maximizes the sensitivity to sensor inputs while maintaining predictability with respect to an internal adaptive forward model. In practice homeokinetic control enables a robot to self-organize its behavior in a playful interaction with its environment and explores the suitable movement patterns for its particular embodiment. A short introduction to homeokinesis will be given in the following section. Then, we will face the question how goals can be introduced into a self-organizing system. Instead of imposing a goal we will aim at guiding the agents towards the desired behavior using as much of the intrinsic behavior as possible.

For the combination of self-organizing and external drives we coined (Martius et al. 2007) the term *guided self-organization* (GSO), which was before only rarely used e. g. in nano technology (Choi et al. 2005) or swarm robotics (Rodriguez 2007) and gained now a much larger scientific interest (Prokopenko 2009). Goal-oriented methods optimize for a specific task and require a prestructuring of the control problem in high-dimensional systems. Self-organization, on the other side, can generate coherent behavior and structure in the behavior space. Furthermore self-organizing systems show a great flexibility and high tolerance against failures and degrade gracefully rather than catastrophically (Prokopenko 2008, 2009). The perspective of GSO is to obtain a system which unites benefits of both. In the main sections of this chapter we discuss several approaches for guided self-organization with homeokinesis (GSOH). These methods span the range from incorporation of supervised learning signals to reward based methods and to teaching of structural relations.

8.2 Homeokinesis

Homeokinesis (Der 2001; Der and Liebscher 2002; Der and Martius 2012) is about establishing/stabilizing an internally defined dynamic regime of the sensorimotor dynamics and is thus conceptually similar to homeostasis (Cannon 1939; Wikipedia 2013), where a system has a internal set of states that are stabilized against external perturbation. So homeostasis is about keeping things fixed whereas homeokinesis is about keeping things moving. In effect homeokinesis produces a variety of behaviors in dependence on the interaction between control, internal dynamics and environment. Homeokinetic control arises from optimizing the sensorimotor coordination of an embodied agent to stay in a certain dynamical regime of sensitive but well controlled behavior. For that the movements are compared to the predictions by an internal adaptive model, and it works best with a controller and a model of similar complexity. The robot is thus controlled by a quasi-linear controller that receives sensor values and determines the motor values. If the coefficients of the controller are fixed then we have a purely reactive setup which can produce a particular

reactive behavior. If, however, the parameters change the robot can produce a variety of behaviors. If done appropriately, e. g. as proposed below, a sequence of behaviors is obtained that are all locally smooth and simple but globally rather complex. The approach consists of adapting the parameters to maximize prediction quality and simultaneously to maximize sensitivity to changes in the sensor values.

Formally, we denote the vector of sensor values at time t by $x_t \in \mathbb{R}^n$. The vector of motor values $y_t \in \mathbb{R}^m$ is generated by a controller function

$$y_t = K(x_t, C, h) = g(Cx_t + h) \ , \tag{8.1}$$

where $g(\cdot)$ is a componentwise sigmoidal function, e.g. a hyperbolic tangent. The matrix C contains the modifiable parameters of the controller and h is a vector of bias values. The predictive internal model M uses sensor values and motor values to predict the sensory inputs one time step ahead.

$$x_{t+1} = M(x_t, y_t; A, S, b) + \xi_{t+1} \ , \tag{8.2}$$

$$M(x_t, y_t; A, S, b) = Ay_t + Sx_t + b \ , \tag{8.3}$$

where ξ is the deviation of the actually observed sensor values from their predictions. The matrices A and S are adapted such as to represent the effect of the actions and the previous sensory values, respectively, onto the new sensor values. The vector b, similar to h above, serves as an offset. Inserting Eq. (8.1) into Eq. (8.2) yields

$$x_{t+1} = M(x_t, K(x_t, C, h); A, S, b) + \xi_{t+1} = \psi(x_t) + \xi_{t+1} \ , \tag{8.4}$$

which is a stochastic dynamical system describing the temporal evolution of the sensor values. Considering that all the information the robot obtains arrives through its sensors, the dynamics (8.4) describes the behavior of the robot completely. Note, however, in this interpretation, Eq. (8.4) assumes a Markov property with respect to a fixed time step which may not be realizable in real robots in general.

While the controller determines the behavior of the robot and changes its state in the environment, the internal predictive model learns any new arriving sensory inputs by an online adaptation of the parameters A, S, and b via gradient descent. As a consequence, the prediction error $\|\xi\|^2$ (8.4) tends to decrease.

If the parameters C and h of the controller are also adapted by the minimization of the prediction error then the robot dynamics is subject to stabilization. The resulting behavior reflects the complexity of the environment to some extent, but is typically relatively simple or may simply approach a resting state.

Activity in the sensorimotor loop can by achieved by the homeokinetic paradigm, namely by considering instead the reconstruction which is given by

$$v_t = x_t - \psi^{-1}(x_{t+1}) \tag{8.5}$$

between the previous sensory inputs x_t and their reconstructed values obtained by $\psi^{-1}(x_{t+1})$, where it is assumed that ψ is invertible. It can be interpreted as the amount by which the sensor values would have had to be changed in order to

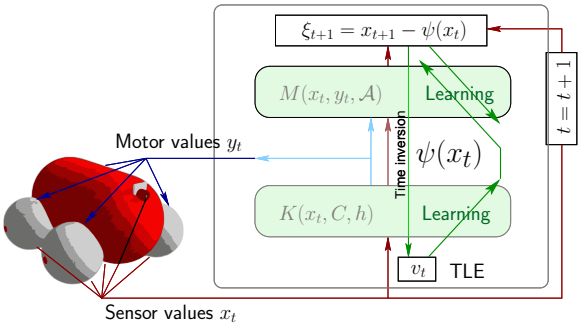


Fig. 8.1 The homeokinetic controller connected to a wheeled robot in a sensorimotor loop. The robot is equipped with wheel counters and a camera. The controller is represented by the function K and the predictor M, both together form the map ψ (Eq. (8.4)). The TLE is obtained by propagating ξ_{t+1} through the inverse of ψ.

preempt any prediction error. The objective function minimizing the reconstruction error v_t is called *time-loop error* (TLE) and it can be approximated using the linearization $v_t = L^{-1}\xi_{t+1}$:

$$E^{\text{TLE}} = \|v_t\|^2 = \xi_{t+1}^\top \left(L_t L_t^\top\right)^{-1} \xi_{t+1}\,, \tag{8.6}$$

where $(L_t)_{ij} = \frac{\partial \psi(x_t)_i}{\partial (x_t)_j}$ is the Jacobian matrix of ψ at time t. The entire framework is sketched in Fig. 8.1. Note that minimizing this error quantity increases the small eigenvalues of L, i. e. it tends to destabilize the system which is, however, confined by the nonlinearity $g(\cdot)$ (8.1). This eliminates the trivial fixed points (in sensor space) and enables spontaneous symmetry breaking.

The parameters of the controller (C,h) are adapted by a gradient descent on the TLE (8.6). This gives rise to the parameter dynamics

$$\Delta C = -\varepsilon_c \frac{\partial}{\partial C} E = \varepsilon_c \mu v^\top - \varepsilon' y x^\top\,, \tag{8.7}$$

$$\Delta h = -\varepsilon_c \frac{\partial}{\partial h} E = -\varepsilon' y\,, \tag{8.8}$$

where ε_c is a global learning rate and ε' is channel-dependent learning rate given by $\varepsilon'_i = 2\varepsilon_c \mu_i \zeta_i$, where $\mu = G' A^\top \left(L^\top\right)^{-1} v$, and $\zeta = Cv$ and G' is the diagonal matrix defined as $G'_{ij} = \delta_{ij} g'_i(Cx+h)$. The derivation of the learning rules can be found in Der and Martius (2012). In our parameterization the Jacobian matrix is given as

$$L = AG'C + S. \tag{8.9}$$

We will generally assume that there are more sensors than motors, which, for $S = 0$, implies that the Jacobian matrix L cannot be inverted such that a pseudo-inverse is being used instead in the above formulas. The parameters A, S and b (8.3)

are adapted online in order to minimize the prediction error $\|\xi\|^2$ (8.4). However, the minimization is ambiguous with respect to A and S because y is a function of x, see (8.1). In order to capture as much as possible of the relationship by the matrix A we introduce a bias by using partly the TLE for learning of the model:

$$\Delta A = \varepsilon_A \xi_{t+1} \left(y_t + \rho_A G' C v\right)^\top , \tag{8.10}$$

$$\Delta S = \varepsilon_S \xi_{t+1} x_t^\top , \quad \Delta b = \varepsilon_b \xi_{t+1} , \tag{8.11}$$

where the parameter $\rho_A = 0.1$ controls the bias. The learning rates ε_S and ε_b are chosen to be smaller than ε_A, but the exact parameter values are not critical; for $\rho_A = 0$ the original delta-rule is restored.

The learning rates are chosen to result in a fast dynamics for the weights. Assuming sensory noise, the TLE is never zero nor has a vanishing gradient such that the rule (8.7) produces an itinerant trajectory in the parameter space, i. e. the robot traverses a sequence of behaviors that are determined by the interaction with the environment. An intuitive idea of the resulting dynamics can be obtained for a robot with just two wheels each equipped with a proprioceptive velocity sensor (see for instance Fig. 8.16(a)). Initially the robot rests, but after a short while it starts to drive autonomously forward and backward or to turn. If the robot arrives at an obstacle, the wheels stop, thus causing a large error because of which the learning dynamics will quickly stop the motors and eventually drive in the free direction. Also high-dimensional systems such as snake- or chain-like robots, quadrupeds, hexapods and wheeled robots can be successfully controlled with the learning dynamics of Eqs. (8.7) to (8.11) (Der and Martius 2012).

8.2.1 Example of Emergent Behavior

To get a more clear idea of what homeokinetic control is about we will present two examples here: the spherical robot and the Cricket robot. The design of the spherical robot is inspired by the artist Julius Popp (2004). It has a ball shaped body and is equipped with three internal masses whose positions are controlled by motors, see Fig. 8.2(a). The motor values define the target positions of the masses along the axes which are realized by simulated servo motors. Collisions of these masses especially at the intersection point are ignored in the simulation.

If we put the spherical robot on level ground and connect the homeokinetic controller initially only small fluctuations due to the sensor noise occur. The learning dynamics increases the feedback strength steadily so that the controller is getting more and more sensitive to the sensor values. Once the critical level is exceeded fluctuations get amplified so that the symmetry of the system is spontaneously broken and the body starts to roll into a decided direction. This is the first moment when the sensor values show a defined response to the actions. The most simple of the natural modes of the robot is realized by rotating around one of the internal axes with the mass on that axis being used for steering and the other ones for shifting the

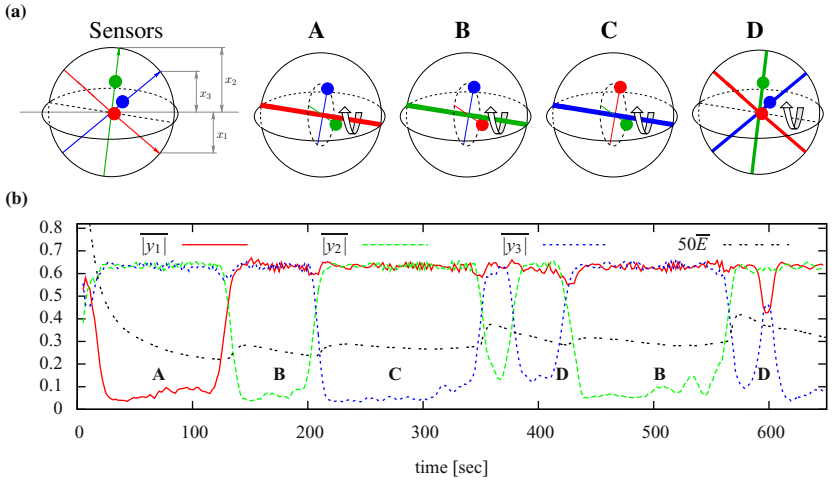


Fig. 8.2 The spherical robot exploring its behavioral capabilities. (a) Sensor setup and sketch of four typical behaviors (**A-D**), namely the rolling mode around the three internal axis (**A-C**) and around another axis (**D**). **(b)** Amplitudes of the motor value oscillations ($y_{1...3}$) and the time-loop error E (scaled for visibility) averaged over 10 and 30 sec, respectively. Corresponding behaviors are indicated with letters A-D.

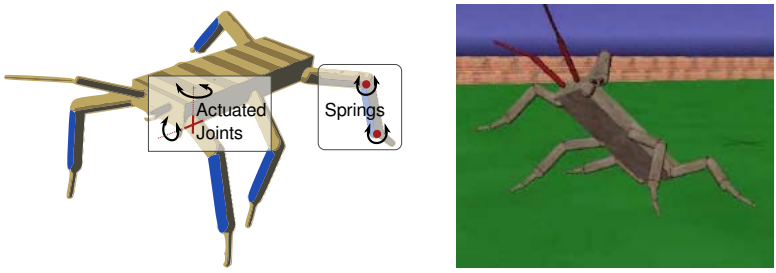
center of gravity. The experiments demonstrate, Fig. 8.2, that the controller picks up such a rolling mode and amplifies it very quickly. The explorative nature of the control algorithm is illustrated in the fact that different rolling modes emerge.

8.2.2 Behavior and Critical Dynamics in High-Dimensional Cricket Robot

In simplified systems the self-organization of the movement parameters of the robots can be studied analytically (Der and Martius 2012), which provides an intuition about the noise amplification and the emergence of behavior in such systems. It is beyond the scope of the present text to represent these results here. Instead we take a phenomenological look at a more realistic system, namely a cricket robot Fig. 8.3.

As before the robot would not move after initialization until the self-amplification of the sensor noise will eventually lead to an initial movement. Because all legs are connected to the trunk their movements are physically coupled, which is automatically extracted by the learning algorithm. The robot starts to sway and becomes more and more active until it starts to lift the feet from the ground. A range of jumping and wobbling motions is emerging that are coming and going.

Theoretically homeokinetic learning should bring the sensorimotor loop into a critical state also termed the edge of chaos. In this state small perturbation in the



(a) (b)

Fig. 8.3 Cricket robot with realistic leg sizes, ranges and mass distribution, cf. (Cruse et al. 2006). The robot has twelve active degrees of freedom (DoF) and 14 passive DoF (lower legs and antennae). **(a)** Schematic diagram of the robot and actuated joints. **(b)** Screen shot from the computer simulations using LPZROBOTS (Martius et al. 2012).

sensor values are neither damped nor amplified. An indication for this state can be obtained from the largest eigenvalue of the mapping from current sensor values to future sensor values, which should be 1. That this is indeed the case for the cricket robot shows Fig. 8.4, where the linearization of the map was used. In linear systems, eigenvalues of unity represent an on-going movement, however, the nonlinearity of the controller or of certain interactions with the environment, such as collisions between feet and ground, require a more powerful controller which leads to larger eigenvalues as shown in the figure. So homeokinetic learning works also in dynamically complex systems and leads to an exploration of the behavioral capabilities of the system under control.

8.3 Guided Self-Organization

The homeokinetic learning rule causes a robot to move actively and to react sensitively to its environment. The resulting behaviors are, however, waxing and waning and their time span and transitions are hard to predict. There are only a number of exceptional cases where a robot could directly make use of the above learning scheme. Assume for instance that the robot has a number of options or schemes to follow in specific situations, but when none of these are applicable then a generic search behavior is certainly helpful. Moreover, if the robot has received a prescribed plan it can still explore similar behaviors which may be more smooth or better with respect to an external reward.

In many robotic applications, however, a defined and goal-oriented behavior is desired. With traditional learning methods these may be hard to obtain, especially if the control space is high-dimensional. A promising route, reflecting some properties of biological learning, is to allow the robot to explore its basic behaviors in a playful

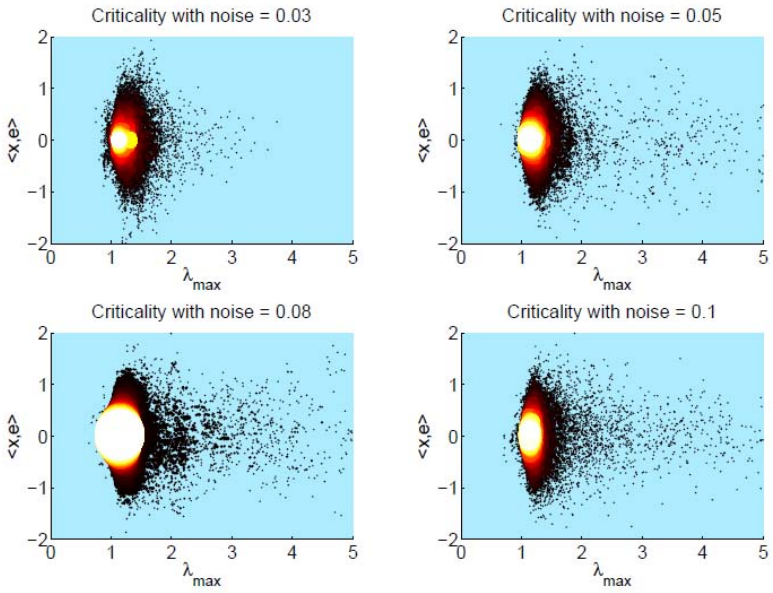


Fig. 8.4 Analysis of the Cricket robot (s. Fig. 8.3). In order to demonstrate the criticality in a complex robot we considered the main eigenvalue of the sensorimotor loop. In accordance with the analytical results for the one-dimensional case (Der and Martius 2012), we observe here eigenvalues with a mean values of approximately 1.2. The *y*-axes in the plots show the projection of the state of the robot onto the corresponding eigenvector which shows a symmetric distribution. Data was obtained in a single run for each of the plots with a different level of noise in each case. During the run the dominant eigenvector frequently changed its orientation. It is interesting that maximal flexibility is reached at an intermediate noise level as indicated by the heat map. White areas represent a high density of points; in low-density areas individual points are drawn in black.

and self-organized phase and internalize some of the intrinsic properties of the sensorimotor loops. Why should it be more effective? Self-organizing systems tend to scale well to higher dimensions and may exploit the constraints and properties of the embodiment. Also, self-organizing systems show a great flexibility and tolerance against failures and degrade gracefully rather than catastrophically (Prokopenko 2008, 2009). After this or even already during this self-exploratory phase, the robot receives information about the task it is expected to execute. This information can be imposed on the robot in an imperative way, but this is possible only if the exploratory properties cease to have an effect on the robot. Taking it further, we need a continuous balance between external and intrinsic learning: The robot continues to behave exploratory, but will preferentially choose those behavioral patterns that comply best with the external information. This is what *guided self-organization* (GSO) for robot control is about, which we introduced (Martius et al. 2007) for the combination of a desired goal with self-organizing behavior. The term has been used before in contexts such as nanotechnology (Choi et al. 2005), city

development (Butera 1998) or swarm robotics (Rodriguez 2007) representing essentially the same idea: exploiting the intrinsic complex dynamics to achieve a goal without much engineering effort or strong interference with the intrinsic dynamics of the system. An illustrative example from nature is again the shell patterns (and animal coat pigmentation). The self-organizing reaction-diffusion systems creating the patterns are guided by comparably simple chemical gradients leading to a specific (species typical) formation. It would have been much more difficult (in terms of e. g. coding length) for evolution to come up with a precise description of pattern in the genome. More importantly this GSO system act as a pattern factory. A new pattern only needs different gradient. However, to engineer a new desired pattern may be difficult, which is part of the challenge of guided self-organization.

By the way, the same general idea also underlies chaos control (Ott et al. 1990) and, more recently, self-motivated learning. GSO is different from active learning, reinforcement learning (Sutton and Barto 1998) and evolutionary learning (Nolfi and Floreano 2001) at least because the exploration is self-organized rather than following a defined scheme or being exhaustive.

To get an intuitive idea how guidance could look like we consider again the emergence of self-organized behavior. In terms of the theory of dynamical systems, the homeokinetically controlled behavior can be considered to consist of series of symmetry breaking events. E. g. a simple robot that is not moving initially, starts to choose to move either forward or backward. If the robot's hardware does not indicate a preference for either direction, the robot chooses a random orientation caused by a possibly tiny fluctuation at the critical moment when the breaking of the symmetry happened. Obviously, the same effect can be achieved if the robot is biased (namely, to move forwards rather than backwards) either by a hardware asymmetry or by any external information. It can be further expected that the external input that the robot receives does not need to be strong. In all cases the robot will continue to self-organize its behavior, but with the difference that the specific decision which was previously due to a noise effect, is now due to an external guidance.

More formally, we will distinguish a number of possibilities for guidance in dependence on the type of information the robot receives. The first one allows for the incorporation of supervised learning signals, e. g. specific nominal motor commands. To make this possible we study the integration of problem-specific error functions into the homeokinetic learning dynamics in the next section. Using a distal learning (Jordan and Rumelhart 1992) setup we also study the use of teaching signals in terms of sensor values and give an example of guidance by visual target stimuli. Interestingly we find a remarkable robustness to sensorimotor disruptions. The second mechanism is discussed in Sect. 8.6 and uses online reward signals to shape the emerging behaviors. The third mechanism for guiding the self-organization can be used to formulate relationships between motors, see Sect. 8.7. This will be proven to be an effective and simple way to introduce constraints into the system and facilitate the unsupervised development of specific behaviors.

8.4 Guidance by Mild Supervision

8.4.1 Integration of Problem-Specific Error Functions

The combination of self-organizing processes and additional constraints is not trivial and essentially an instance of the well-known dilemma that arises when both exploration and exploitation is desired at the same time. A problem-specific error function expressed the goal, i.e. a specification what is to be exploited in a given context, while the behavioral self-organization provides an efficient means for exploration. Whether or not the exploration indeed serves the goal in the long run, is a question of the balance between the two which we are going to discuss in this section. A particular goal can be specified in terms of a problem-specific error function (PSEF) that is minimal if the goal is met.

A suggestive way of combining the TLE and a PSEF could be a weighted sum of the two error functions. Performing gradient descent on this sum minimizes then this combination such that we could expect learning to both improve the active engagement with the environment as well as top approach the goal. It is, however, likely that either one of the learning tasks may improve on the cost of the other one.

The optimal balance between the exploration and exploitation depends not only on the specific problem but also on the course of learning and the current state of the system. The reason is that the size of the TLE varies often over several orders of magnitude, whereas the goal-specific terms will usually stay in a smaller range or will not covary with the TLE. Therefore, a fixed weighting in the combined error function cannot be expected to exist in non-trivial problems.

In order to achieve a goal-orienting effect without destroying the self-organization process, we have proposed to scale the gradient on the PSEF in order to be compatible with the TLE (Martius and Herrmann 2010, 2012). This approach was motivated by the natural gradient method (Amari 1998). This method is based on the fact that for an arbitrary Riemann metric of the parameter space the steepest direction is given by the transformed gradient, which is obtained by multiplying with the inverse of the metric. We use a metric which is defined by the matrix $JJ^\top$, where J is the Jacobi matrix of the sensorimotor loop, similar to Eq. (8.9) but now in motor space. We can think of this procedure as map of the error into the action space of the robot.

The PSEF is denoted by E^G and it must be non-trivially dependent on the controller parameters such that the gradient can be effective. So the main formula for guided self-organization with homeokinesis is the new update rule for the controller matrix C as

$$\frac{1}{\varepsilon_c}\Delta C = -(1-\gamma)\frac{\partial E^{\mathrm{TLE}}}{\partial C} - \gamma Q \frac{\partial E^G}{\partial C}\,, \tag{8.12}$$

where $\frac{\partial E^{\mathrm{TLE}}}{\partial C}$ is the homeokinetic learning rule (8.7), $1 \leq \gamma \leq 0$ is the guidance factor defining the weighting between goal following and self-organization, and

$Q = \left(JJ^\top\right)^{-1}$ defines the metric. The latter can also be expressed as $Q = A^\top \left(LL^\top\right)^{-1} A$, see Martius (2013). For $\gamma = 0$ there is no guidance and we obtain the unmodified dynamics, and for $\gamma = 1$ there is no homeokinetic adaptation but only guidance.

The entire update size is still controlled by the learning rate ε_c. For the update of the parameter h we apply an analogous procedure.

Below we will look at a few concrete examples of problem-specific error functions (PSEFs) that implement the guidance by *teaching signals*. In this way a supervised learning procedure is introduced which, however does not imprint its effect on the system but rather have the system explore the learning objective implied by the PSEF.

8.4.2 Direct Motor Teaching

In order use motor-teaching signals we define a PSEF, which penalizes the mismatch

$$\eta_t = y_t^G - y_t \tag{8.13}$$

between motor teaching values y_t^G and actual motor values y_t (output by the homeokinetic controller). Similarly to the prediction error for the forward model we find

$$E^G = {\eta_t}^\top \eta_t \ . \tag{8.14}$$

Using the gradient descent we get the additional update for the controller matrix C as

$$\frac{\partial E^G}{\partial C} = -G' \eta_t {x_t}^\top \ , \tag{8.15}$$

where G' is the diagonal matrix given by $G'_{ij} = \delta_{ij} g'_i(Cx + h)$. Similarly, for h we obtain $\frac{\partial E^G}{\partial h} = -G'\eta_t$. These additional terms are integrated into the final learning rule using Eq. (8.12). The guidance factor γ regulates the strength of the additional drive and has to be determined empirically. A small value of γ leads to a small influence of the teaching signal and results in a behavior that is mostly dominated by the original homeokinetic controller. For large values of γ the teaching signals are followed narrowly and few exploratory actions are performed, however, with the increasing danger to break down the self-organization.

8.4.2.1 Experiment

Using a two-wheeled robot, see Fig. 8.16(a), we will show that teaching signals can be used to specify a certain behavior and that the influence of the teaching can be conveniently adjusted using γ. For that let us consider two different motor teaching signals, which are subsequently applied. First the nominal motor values are given by a sine wave and then by a rectangular function with the same value for both motors, i. e.

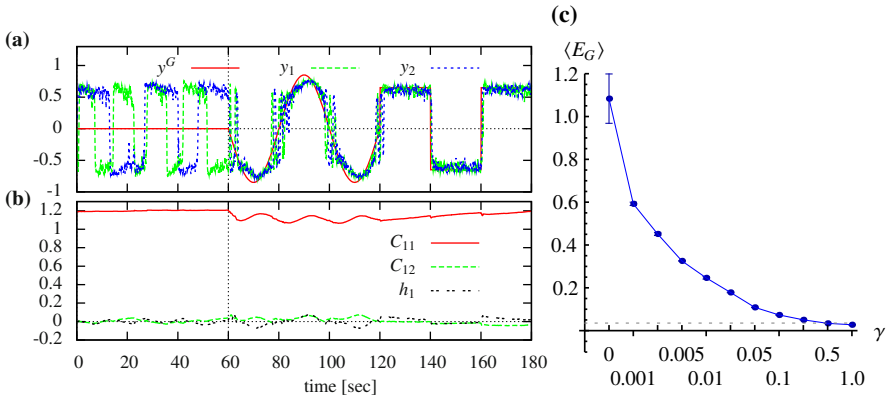


Fig. 8.5 Two-wheeled robot controlled with homeokinetic controller and direct motor teaching signals. **(a)** The teaching signals y^G (identical in both components) are followed partially by the motor values $y_{1,2}$ after teaching was switched on with $\gamma = 0.01$ at 60 sec. **(b)** Time evolution of the controller parameters affecting the first motor is shown to illustrate that only little changes are necessary, however, the adaptations do not vanish. **(c)** Average value of the PSEF E^G (for 5 experiments à 5 min) in dependence of γ (note the logarithmic scale). The noise level (*dotted gray line*) is reached at $\gamma = 1$. Parameters: $\varepsilon_c = \varepsilon_A = 0.1$, $\gamma = 0.01$ (a,b).

$$(y_t^G)_i = \begin{cases} 0.85 \cdot \sin(\omega t) & t < 75 \\ 0.65 \cdot \mathrm{sgn}(\sin(\omega t)) & \text{otherwise} , \end{cases} \tag{8.16}$$

with $i = 1,2$ and $\omega = 2\pi/50$. For the choice of the teaching signal we have to consider that the nominal motor values should not be too large because otherwise the controller is driven into the saturation region of the motor neurons. The fixed point of the sensor dynamics in the simplified world condition is at $y \approx \pm 0.65$. This is a good mean teaching signal size, which was also used in Eq. (8.16). As a rule of thumb we recommend confining the motor teaching values to the interval $[-0.85, 0.85]$.

In Fig. 8.5 the produced motor values and the parameter dynamics are displayed for different values of the guidance factor γ. For a low value of γ the desired behavior is only followed by trend, whereas for higher values, e. g. $\gamma = 0.01$, the robot mostly follows the given teaching value with occasional exploratory interruptions. These interruptions cause the robot, for example, to move in curved fashion instead of strictly driving in a straight line as the teaching signals dictate. The exploration around the teaching signals might be useful to find modes which are better predictable or more active. The long performance of a single low-dimensional behavior can lead to the inaccuracy of the adaptive forward model. Thus, the explorative actions can supply the forward model with necessary sensation-actions pairs for complete learning.

The experiment demonstrated that motor teaching signals can be used to achieve a specific behavior. This result is not very surprising, because the system is very

simple and the target behavior did not conflict with the homeokinetic principle (sensitive and predictable). However, it served as a proof of principle and showed that the balance between target behavior and remaining self-organized behavior can be adjusted using a single parameter.

8.4.3 Direct Sensor Teaching and Distal Learning

In this section we transfer the direct teaching paradigm from motor teaching signals (Sect. 8.4.2) to sensor teaching signals. This is a useful way of teaching because desired sensor values can be more easily obtained than motor values, for instance by moving the robot, or parts of the robot by hand. This kind of teaching is also commonly used when humans learn a new skill, e. g. think of a tennis trainer that teaches a new stroke by moving the arm and the racket of the learner and is a subset of imitation learning (Schaal et al. 2004). Thus, a series of nominal sensations can be acquired that can serve as teaching signals. Setups where the desired outputs are provided in a different domain than the actual controller outputs are called *distal learning* (Jordan and Rumelhart 1992; Stitt and Zheng 1994; Dongyong et al. 2000). Usually a forward model is learned that maps actions to sensations (or more generally to the space of the desired output signals). Then the mismatch between a desired and occurred sensation can be transformed to the required changes of action by inverting the forward model.

The distal learning error is the mismatch between desired sensations x_t^G and actual sensations x_t

$$\xi_t^G = x_t - x_t^G \,. \tag{8.17}$$

The mismatch η_t in motor space can be obtained via the forward model M (8.3) in linear order

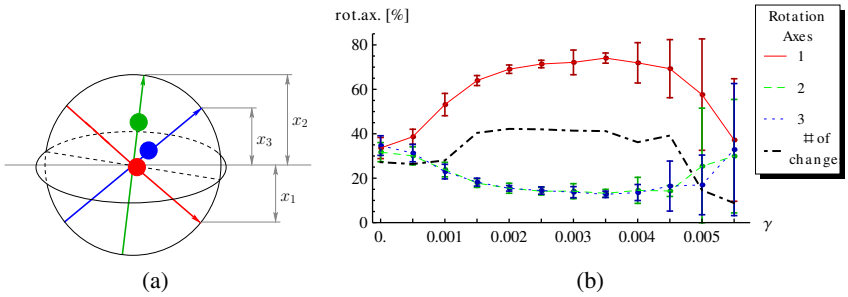
$$\eta_t = A^+ \xi_t^G \,, \tag{8.18}$$

where $A = \frac{\partial M(x,y)}{\partial y}$ and the A^+ denotes the pseudoinverse of A. Now the update formulas (8.15) for C and h from the direct motor teaching setup can be used based on the teaching error $E_G = \|\eta_t\|^2$.

8.4.3.1 Experiment

For the two-wheeled robot (Fig. 8.16(a)) the forward model is simply a multiple of the unit matrix. The spherical robot (Fig. 8.6(a)), however, has a non-trivial relation between sensor and motor values and is thus better suited for an illustrative experiment to show that a simple teaching signal in terms of sensor values can be effective in guiding the behavior.

A desired behavior for the spherical robot could be to rotate around the one of its internal axes. For the particular sensor setup we need to assure the the corresponding



(a) (b)

Fig. 8.6 The spherical robot in a homeokinetic plus distal learning setup. **(a)** Illustration of the robot with its sensor values. **(b)** Behavior with the distal learning signal, Eq. (8.19). The plot shows the percentage of rotation time around each of the internal axes and the number of times the behavior was changed for different values of the guidance factor γ (no teaching for $\gamma = 0$). The rotation around the red (first) axis is clearly preferred for non-zero γ. The mean and standard deviation are plotted for 20 runs each 60 min long, excluding the first 10 min (initial transient, no guidance). For too large values of the guidance factor the self-organization process is too much disturbed such that the robot gets trapped in a random behavior (*dash-dotted line*). Parameters: $\varepsilon_c = \varepsilon_A = 0.1$.

sensor returns consistently a low absolute value. This can be directly specified in the distal learning scheme, here for the first axis:

$$x_t^G = \begin{pmatrix} 0 \\ (x_t)_2 \\ (x_t)_3 \end{pmatrix} . \tag{8.19}$$

Now only the first component of the sensor value produces an error signal. The resulting behavior is characterized in Fig. 8.6.

The distal learning scheme requires a well trained forward model. Therefore pure self-organization was used during the first 10 min of the experiment ($\gamma = 0$). As a descriptive measure of the behavior, we used the index of the internal axis around which the highest rotational velocity was measured at each moment of time. Figure 8.6(b) displays for different values of the guidance factor and for each of the axes the percentage of time it was the major axis of rotation. Without teaching there is no preferred axis of rotation. With distal learning the robot shows a significant preference (up to 75 %) for a rotation around the first axis. For overly strong teaching, a large variance in the performance occurs. This is caused by a destructive influence of the teaching signal on the homeokinetic learning dynamics. Remember that the rolling modes can emerge due to the fine regulation of the sensorimotor loop to the working regime of the homeokinetic controller, which cannot be maintained for large values of γ.

The robot will not stay in the rotational mode about one axis. While the robot is in this rotational mode the teaching signal is negligible. However, the sensitization

property of homeokinetic learning increases the impact of the first sensor, such that the mode becomes eventually unstable again. Again this may be considered as an advantage since the temporary breaking out avoids a too narrow specialization of the internal model. Note, moreover, that the learning success in the current setting of controller and forward model could *not* be achieved by the distal learning alone, at least not with a constant learning signal.

To recapitulate, the direct teaching mechanism allows us to specify motor patterns that are more or less closely followed, depending on the strength of integrating the additional drives into the learning dynamics. In this section we considered sensor teaching signals that were transformed into motor teaching signals using the internal forward model. We have shown that the spherical robot with the homeokinetic controller can be guided to locomote mostly around one particular axis, by specifying a constant sensor teaching signal at one of the sensors Martius and Herrmann (2010).

8.5 Self-Organized Interaction with the Environment

Let us know consider a more involved application with direct sensor teaching using a camera sensor.

8.5.1 Integration of Vision into the Sensorimotor Loop

Vision adds a new level of complexity to any robotic system. In particular in most of the applications of the homeokinetic principle, mostly proprioceptive sensors have been used, which helps to generate a sensible control of the body, but may not be sufficient to produce a tight interaction with complex environments. In the following we will discuss the integration of visual information into the framework of self-organizing control, see also Martius (2013).

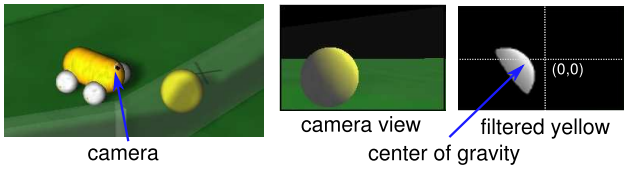


Fig. 8.7 Camera setup, image processing and sensor values

In the following we will describe experiments with a four-wheeled robot (Fig. 8.1). The robot is operated such that the two motors on one side of the robot receive the same target velocity. The two velocity sensors (x_l and x_r) return the average of the actual wheel velocities on one side.

A simplification can be reached by restricting the interacting of the robot to with objects of a certain color, yellow in our case. We start by calculating the center of mass (x_h, x_v) over all pixels of this color based on the assumption that only one

yellow object is visible. If this not the case the approach of the robot will help with the disambiguation. Next, we approximate the size (x_s) of the object by the sum of all yellow pixels (normalized to $[0,\sqrt{2}]$). This is prone to light and shadow effects and is again a crude approach but it will turn out to be sufficient for our purposes. In addition we also use the time derivatives of the quantities such that the vector of sensor values reads

$$x = (x_l, x_r, x_h, \dot{x}_h, x_v, \dot{x}_v, x_s, \dot{x}_s)^\top. \tag{8.20}$$

We are often adding a small amount of sensory noise to the simulated sensors which is not only more realistic, but also has the side effect that the TLE does not become zero. The vision sensors are, if any objects are visible at all, rather inaccurate and noisy, e.g. due to illumination, such that additional noise is not required here. Therefore, only the wheel velocities sensors x_l and x_r are subject to Gaussian noise with a small standard deviation.

Exteroceptive sensors in general and our vision sensors in particular may not be active for substantial periods during operation. For instance the position sensor (x_h, x_v) is essentially undefined if no object is in sight. Since the predictive model is to correlate actions with perceptions, the absence of any object nullifies the correlations such that a prediction becomes impossible. A simple solution is to prevent learning of the predictive model on invalid sensor values. We implement this by assuming an undefined sensor value to be zero and set the prediction error $(\xi_i)_t$ to zero as well, if $(x_i)_t = 0$ or $(x_i)_{t-1} = 0$ while it remains unchanged otherwise.

8.5.2 Guiding towards an Object

We will now define a guidance mechanism that drives the robot towards a visible object. In order to fixate the object in the center of the field of vision, the position sensors (x_h, x_v) should approach zero unless a specific target position (p_h, p_v) is given. If the robot should push objects, e. g. a ball, then the value of the size sensor (x_s) should be large. Alternatively if the robot should keep a certain distance, for instance when interacting with other robots, then a smaller value is required. We denote the desired size by s.

The linear predictive model can represent the relation between actions and position/size only in certain situations. In particular we deal here with stationary and moving objects that cause a different sensory response. A new mechanism could make use of the desired value for the derivatives, too. Fortunately, we can use proportional set-point control formula with damping: $\dot{x} = -\alpha(x - x^{\text{desired}}) - \beta\dot{x}$, where α is a rate and β is the damping constant. This differential equation has a fixed point at $x = x^{\text{desired}}$.

The sensor teaching vector x^G is thus given in components as

$$x_l^G = x_l, \qquad x_r^G = x_r, \tag{8.21}$$

$$x_h^G = p_h, \qquad \dot{x}_h^G = -\alpha(x_h - p_h) - \beta\dot{x}_h, \tag{8.22}$$

$$x_v^G = p_v, \qquad \dot{x}_v^G = -\alpha(x_v - p_v) - \beta\dot{x}_v, \tag{8.23}$$

$$x_s^G = s, \qquad \dot{x}_s^G = -\alpha\,(x_s - s) - \beta\dot{x}_s\,, \tag{8.24}$$

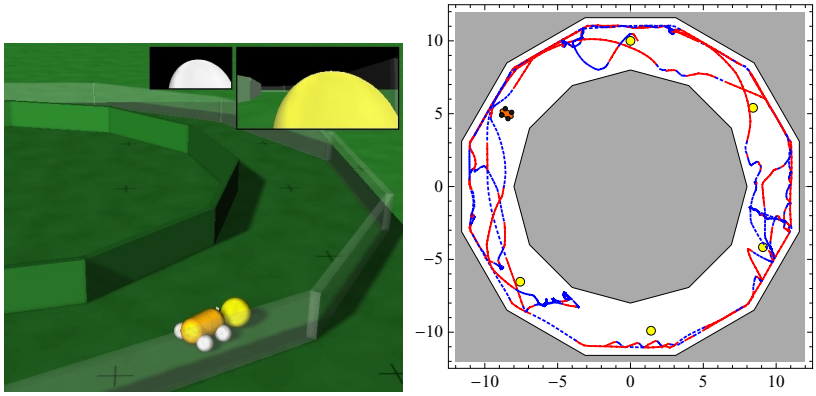
where $\beta = 0.1$ and $\alpha = 1$ here. Note, the wheel velocity sensors x_l and x_r produce no teaching signal. For the following experiments we use for the center position $p_h = p_v = 0$ and set the maximal size to $s = \sqrt{2}$.

8.5.3 Emergent Behaviors

The first experiment should test whether the guidance mechanism is able to influence the self-organized behavior to find and push balls. This involves the establishment of the required sensorimotor mappings from scratch in a changing environment (balls can move). All the experiments are performed in virtual reality in our robot simulator (Martius et al. 2012). The formal definition of the goal is specified by the target sensor state x^G Eqs. (8.21–8.24). We place the robot together with five balls into a circular corridor, as displayed in Fig. 8.8(a), such that the robot can possibly push a ball for a long distance without getting stopped by corners. Those parameters of the model (A) connecting to the vision sensors are initialized with zero, such that the guidance has no effect independently of the guidance factor. Recall that the forward model transforms the teaching signal to nominal changes in motor values Eq. (8.18), which will be zero if the model did not learn anything. Once the robot learns to move, the model starts to correlate actions with the visual sensors. In this way the guidance starts to actually influence the behavior, such that the robot sees a ball more often and the model can improve further. Eventually the robot starts to steer at a ball and pushes it along the arena. Note that the robot has a round front shape such that the ball easily drifts away to either side while pushed. From time to time the robot still performs exploratory actions such that the ball gets lost and a ball needs to be found again. A part of a trajectory of the guided robot is shown in Fig. 8.8(b).

Note that there can be more than one ball in the field of view at the same time. However, the sensors cannot distinguish different objects, since the visual sensor (x_h, x_v) provides a position between the objects and the size (x_s) sensor returns a sum of the sizes. Nevertheless, the robot copes with this situation without problems. The robot steers at a group of balls and decides rather spontaneously which one it will touch. The final choice depends on how well the different balls are visible, when they leave the field of view, and other perturbations.

In order to analyze quantitatively the behavior of the robot, we consider the average distance to the closest ball and the cumulative time a ball was in the sight of the robot. This gives a good measure on whether the guidance was followed and the robot is indeed approaching the balls. If the robot is also pushing the balls along the arena, then the traveling distance of the balls raises, which we display together with the other quantities in Fig. 8.9(a). Indeed, for intermediate values of the guidance factor the time a ball was in sight increases from 100 sec to 600 sec. The same holds for the average distance of the robot to the closest ball which decreases from 5 to a



(a) (b) $\gamma = 0.1$

Fig. 8.8 Ball playing scenario. The robot is placed in a circular corridor. **(a)** Screen shot from the simulation. The right inlet shows the camera image and the left displays the color filtered image; **(b)** Part of a sample trajectory of the robot (minutes 5–10) for $\gamma = 0.1$ colored in red (solid) if the robot is close (within two body length) to the ball and it was in sight, and in blue (dashed) otherwise. The yellow disks show the initial positions of the balls.

value of 2. The size of the robot is 1 and the ball has radius of 0.3, resulting in a minimum of 0.8. Why does not the average distance go much below 2? Firstly, the plots include the entire simulation time including the phase where the robot has to acquire basic knowledge about its body. Secondly, it can take a long time and driving distance to find a ball again when it is lost, for instance through an exploratory action. Due to the inner circular walls of the arena the balls are not visible everywhere and finally the distribution of distances is skewed, see below.

The traveling distance of the balls raises from nearly zero to more than 7500 units, which corresponds to about 100 rounds in the arena (in 30 min).

In Fig. 8.9(b) we show that the aspects of the behavior that are not particularly subject to the guidance, namely the covered area of the arena by the robot and its average velocity are not negatively effected by the guidance, at least for moderate guidance strengths. The area coverage and the velocity go up when the task is performed, because the robot drives much more straight and forward than without the guidance.

When the guidance is too strong self-organized adaptation and external pressures become out of balance and the performance drops. Especially visible is this effect at $\gamma = 1$ where no homeokinetic learning takes place (Eq. (8.12)) and the robot fails to move in a coordinated fashion, see Fig. 8.9(b).

Taking a closer look at the distance to the closest ball, we find that the mean is not such an appropriate measure in the guided situation since the distribution of distances is not Gaussian but rather skewed as shown in Fig. 8.10. Without guidance the distribution of distances is almost flat, whereas for weak and intermediate guidance strengths the distribution is skewed with a strong preference for short distances. For

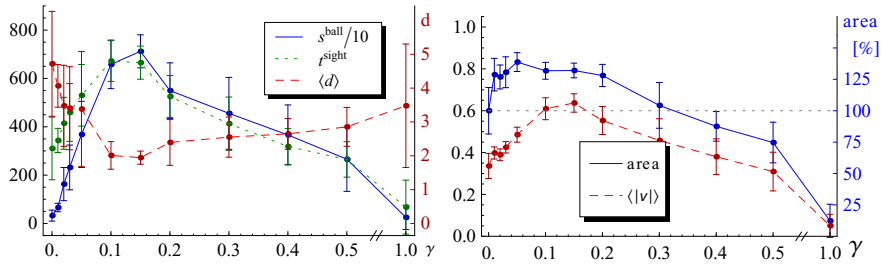


Fig. 8.9 Behavioral quantification of the ball playing scenario. Both panels show the mean and standard deviation of 10 simulations each 30 min long, in dependence of the guidance factor γ. **(a)** Traveling distance of the balls s^{ball} (scaled), cumulative time a ball was in sight t^{sight} (in sec), and average distance to the closest ball $\langle d \rangle$ (right axis, minimum 0.8). **(b)** Average absolute velocity of the robot (left axis) and area coverage (box counting method), given in percent of the case without guidance ($\gamma = 0$) (right axis).

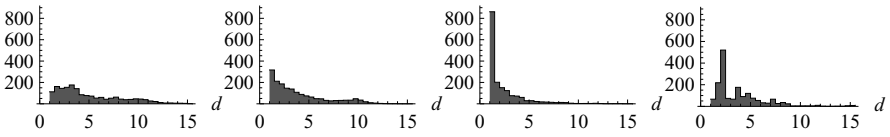


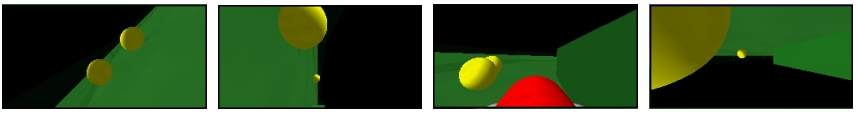
Fig. 8.10 Distribution of distance to the ball in the ball playing scenario. All panels show the histogram (in sec) of the distance d averaged over all simulations for one particular value of the guidance factor γ. **(a)** No guidance; **(b)** weak guidance; **(c)** intermediate guidance; **(d)** overly strong guidance (no self-organization).

overly strong guidance ($\gamma = 1$) the robot gets predominantly stuck at the walls because the sensorimotor coordination is pushed away from its sensitive regime, such that the histogram is rather arbitrary.

8.5.4 *Robustness against Structural Changes*

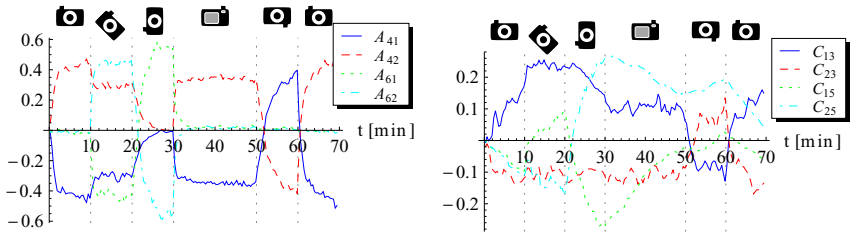
In fact we performed quite radical changes to the camera setup, namely to rotate and flip the camera abruptly, see Fig. 8.11. These changes have severe consequences for the sensorimotor dynamics, because some sensor values swap signs or change from being useless to becoming important and vice versa.

We use the same circular arena as in the previous section. In our simulated experiments the camera setup is initially normal and is changed every 10 minutes to the setups shown in Fig. 8.11. Only the backwards view is kept for 20 min. Finally the normal setup is used again, such that an experiment lasted 70 min in total. We conducted 10 experiments with $\gamma = 0.1$ and present the evolution of the relevant model and controller parameters in Fig. 8.12.



(a) Tilt 45° (b) Tilt −90° (c) Backward view (d) Upside down

Fig. 8.11 Radical changes to the visual perception. In addition to the normal setup of the camera (Fig. 8.8) it is rotated by 45° **(a)**, −90° **(b)**, and 180° **(d)** along the optical axis, and lifted and rotated by 180° **(c)** along the vertical axis yielding a backward view. Note the different perspective and the appearance of the robot's body in the camera view in **(c)**.



(a) Model – motion (b) Controller – position

(c) Controller – motion (d) Controller & Model – size

Fig. 8.12 Fast relearning: evolution of parameters for a changing camera. The camera is changed every 10 min, illustrated by the vertical lines. Its orientation on the body is shown by the icons. All values are mean values for 10 independent runs. Shown are elements of the model matrix (A) and controller matrix (C). The indexes refer to the sensor and motor value vectors, see Eq. (8.20). **(a)** Model parameters connecting left and right motor command with visual motion input ($\dot{x}_h$,$\dot{x}_v$). **(b)** Controller parameters connecting visual position (x_h,x_v) with left and right motor neuron. **(c)** Controller parameters connecting visual motion ($\dot{x}_h$,$\dot{x}_v$) with left and right motor neuron. **(d)** Controller and Model parameters connecting visual size (x_s,$\dot{x}_s$) and left wheel. The model parameters adapt very quickly to the new camera configurations. The controller utilizes both the position and the motion of the ball, however its adaptation is much slower compared to the model. Parameters: $\gamma = 0.1$.

Especially the model parameters relating motor values with the motion sensors, Fig. 8.12(a), evidently show that the correct correspondence is learned within a few minutes after each switching event. This, however, is only possible if the behavior of the robot is such that a ball remains frequently in the field of vision, which is very hard, if e.g., the positional sensation just swapped sign. In this situation the major strength of the homeokinetic controller shows its fruits, namely its continuous and embodiment related explorative and drive. The controller parameters show that the incorporation of the vision sensors is changed drastically for the different situations, but also that both motion and position information is used. The positional information is required to steer towards the ball and the motion sensor is used avoid overshooting. The parameters C change slower than the model parameters. Note that the behavior is also influenced by the parameters h (not shown). These change more rapidly and help to realize the teaching signals on a shorter timescale until the C matrix captures the correspondence with the sensor values.

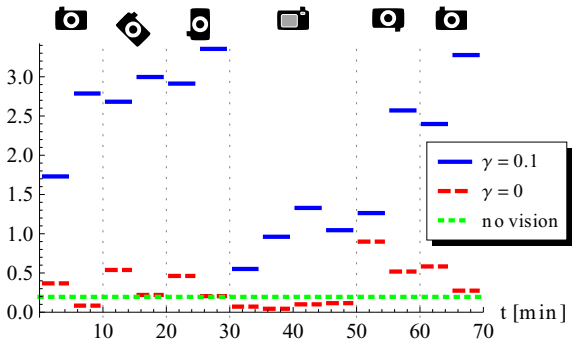


Fig. 8.13 Performance recovery for a changing camera configuration. Depicted is the summed average velocity of all balls within intervals of 5 min corresponding to the simulations in Fig. 8.12. For comparison the case without guidance ($\gamma = 0$) is displayed. The base line (green, dotted) represents the average ball movement of a blind robot.

How is the performance in the task after the structural changes? To answer this question we present in Fig. 8.13 the average ball velocities within 5 minute intervals summed over all balls. Note, that since the balls are subject to rolling friction a constant pushing is required. For comparison the values without guidance and without vision (chance level as a baseline) are displayed. The performance within the first 5 minutes is already far above the baseline and it is doubled from the first to the second 5 minute interval. After each structural disruption the performance drops a bit and is recovered in the second 5 min interval for each setting. Only the setting with camera pointing backward yields worse performance, which is due to the partial obstruction of the visual field by the body. Then the most drastic disruption occurs when the view is switched from backward to forward, but upside down. Here all visual sensor modalities change sign. Nevertheless the performance raises in the second 5

min interval to the performance of before. We can conclude that the performance is rapidly recovered even after severe changes in the sensor modalities.

At the beginning of an experiment the robot learns the behavior from scratch. When the camera is first turned by 45° comparably small adaptations occur, see interval 10-20 min in Fig. 8.12. For instance the sensors for vertical position and motion get slowly integrated, but the remaining structure stays the same and in fact the performance drops only slightly (Fig. 8.13). When the camera is turned to −90° a drastic change occurs. The meaning of the size sensor does not change, but the position and motion sensors require a completely different coupling, which is slowly established (interval 20-30 min). This may be called learning from scratch, but in fact it is worse, it is learning from a wrong configuration. When the switch occurs the controller acts to avoid the ball. To manage this challenge an exploration is required that focuses on the wrong aspects of the model, which is what happens in our approach, where the adaptation speed is actually increased if the prediction errors raise (see Eq. (8.6)). Since the controller does not explicitly know when a structural change occurs it is always adapting in a continuous manner. However, there is no long-term memory such that the controller cannot remember previously experienced configurations.

To summarize, the entire sensorimotor coordination to fulfill the task was learned by the robot within a few minutes. This involves the basic coordination to drive the robot and the integration of the vision sensors such that the balls are approached and balanced while pushed. The task to push the balls is not very complicated and can be achieved with a simple hand-crafted controller. However, to learn it from scratch in a short amount of time is hard. On top of that the orientation of the camera was abruptly changed such that a completely different sensorimotor coordination becomes necessary. We found that guided self-organizing with homeokinesis can cope with a wide range of configuration changes, even those where a complete change in the visual sensation occurs. To our knowledge there is no other system that offers this kind of robustness and the rapid on-line learning.

8.6 Reward-Driven Self-Organization

8.6.1 Reinforcement Learning and Guided Self-Organization

In many applications an explicit objective function is not available, instead a qualitative signal is given that can be interpreted as reward of punishment of a recent state or action of the robot. Reinforcement learning studies the generation of policies under such conditions typically relying on an exploration mechanism that discovers better solutions from present ones. While it is possible to apply a learning rule similar to homeokinesis to shape exploration in reinforcement learning (Smith and Herrmann 2012), we will consider here the usage of the reward signal for the guidance of the homeokinetic exploration.

For example the behavior of the simple robot with one-degree of freedom shows a systematic sweeping through the accessible frequencies of the sensor state reflected

by rolling modes with different velocities (Der and Martius 2012). In the case of the spherical robot with its three dimensional motor and sensor space we also observed a sweeping through a large set of possible behaviors. In a setup where the robot can move freely, it will exhibit different slow and fast rolling modes around different axes.

Before introducing the new mechanisms, let us recall that well predictable behaviors persist longer than others. Due to this effect the well predictable behaviors are also quickly found because badly predictable ones are left quickly. Translating this into the case of reward and punishment, we want that rewarded behaviors persist longer than punishment ones and that predictable ones are found quickly. Thus we have to modulate the learning speed according to the online reinforcement signal in a way that in rewarded situations the adaptation speed is reduced and in punished ones the speed in increased. At first glance it seems to be counterintuitive that we have to reduce learning speed in order to keep a behavior, but the self-organized search should be slowed down to find even better behaviors locally. Moreover, the controller is already able to produce the behavior at the time it is exhibited by the robot.

The real-valued reward signal $r(t)$ for each time t is supposed to act as a reward for positive values and as a punishment for negative values. It is incorporated into the error function in the following way

$$E^{\mathrm{r}} = (1 - \tanh(r(t)))E \,, \tag{8.25}$$

where E is the usual TLE (8.6) and $r(t)$ is expected to assume values mainly in the interval between -1 and $+1$. Larger amplitudes are squashed by the hyperbolic tangent such that differences tend to be ignored for high positive or negative rewards. The effect of the factor $(1 - \tanh(r(t)))$ is is the same as a rescaling of the learning rate which is increased for negative and decreased for positive rewards. Therefore, we can expect that rewarded behaviors persist longer and punished behaviors are left quicker. We will demonstrate the effect of the reward-based weighting in shaping the behaviors of the spherical robot (see Fig. 8.5a).

8.6.2 Modulation of Behavior in a Spherical Robot

8.6.2.1 Reinforcing Speed

In the following experiment we will use the spherical robot, see Fig. 8.6(a). One of the simplest possible desired behaviors of this robot is fast unidirectional rotation. A reward function for this goal can be constructed from the angular velocity of the robot. For small velocities the reward should be negative, thus causing a stronger change of behavior, whereas larger velocities should result in a positive reward. To achieve that, the reinforcement signal can be expressed as

$$r(t) = \frac{1}{3}\|v_t\| - 1 \,, \tag{8.26}$$

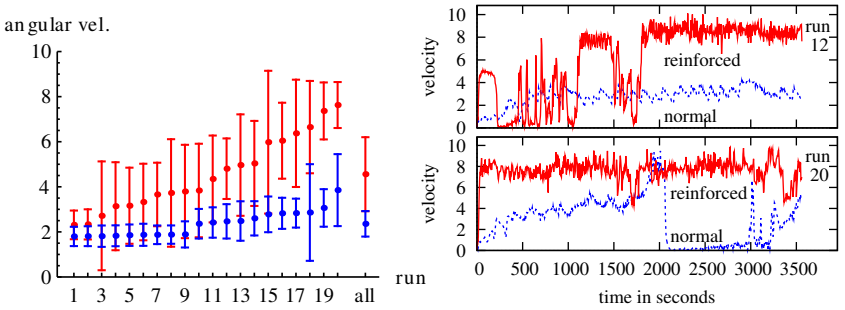


Fig. 8.14 Performance of the spherical robot rewarded for speed. **(a)** Mean and standard deviation of the velocity of the spherical robot for 20 runs each 60 min long with (**red**) and without (*blue*) speed reinforcement, sorted by velocity. The label '**all**' denotes the mean and std. deviation over the means of all runs, which is significantly ($p < 0.001$) higher for the reinforced runs. **(b)** Time course of the robot's velocity for run number 10 and 14, where *blue/dotted* shows the normal case and **red/solid** line shows the reinforced case.

where v_t is the velocity vector of the robot, see Fig. 8.6(a). In order to compare the results with the unguided case the reward is shifted, such that it is zero for the average velocity of normal runs. The scaling is done to keep the reward within the effective range.

We conducted 20 trials with the spherical robot with reinforcement and 20 trials without reinforcement, all with random initial conditions, each for 60 min in simulated real time on a flat surface without obstacles. The robot also experiences rolling friction, so that fast rolling really requires continuous motor activity. In Fig. 8.14 the mean velocity (measured at the center of the robot) for each simulation is plotted and the velocity trace of the robot for two reinforced and two normal runs are displayed as well. The simulations are sorted by performance and plotted pairwise for comparison. As desired, the mean velocities of the reinforced runs are larger than the ones of the normal runs. This is especially evident in the overall mean (mean of means marked by '**all**' in Fig. 8.14(a)), which is significantly different. The null hypothesis that the set of means of the reinforced runs and of the normal runs have an indistinguishable mean was rejected with $p < 0.001$ using the t-test. However, since straight and also fast rolling modes are easily predictable and active they are also exhibited without reinforcement for a long time. It is important to note that the fast rolling modes are also found again, after the robot was moving slower, see Fig. 8.14(b).

The guidance of the homeokinetic controller using a reward for fast motion has shown to increase the average speed of the robot significantly. Although there are also trials where no increased speed was found.

8.6.2.2 Reinforcing Spin

In a different setup we want the robot to follow curves and spin at the spot. We use the angular velocity ω_z around the z-axis of the world coordinates system, which is perpendicular to the ground plane, as depicted in Fig. 8.6(a). The reward function is now given by

$$r(t) = \frac{1}{3}\|\omega_z\| - 1\,. \tag{8.27}$$

Again the reward is scaled and shifted to be zero for normal runs and to be in an appropriate interval. Positive reward can be obtained by rolling in a curved fashion or by entering a pirouette mode. The latter can be compared to a pirouette done by figure-skaters—with some initial rotation the masses are moved towards the center, so that the robot spins fast in place. The robot also experiences rolling friction, so that fast pirouettes are not persistent.

Again, we conducted 20 trials with reinforcement and 20 trials without reinforcement, each for 60 min simulated real time on a flat surface without obstacles. In Fig. 8.15(a) the mean angular velocity ω_z for each simulation is plotted, again sorted by performance. The time evolution of the angular velocity for two reinforced and two normal runs are displayed in Fig. 8.15(b). In this scenario the differences between the normal runs and the reinforced runs are remarkable. Nearly all reinforced runs show a large mean angular velocity. The reason for this drastic difference is that these spinning modes are less predictable and therefore quickly abandoned in the non-reinforced setup. The traces show that the robot in a normal setup rarely performs spinning motion, whereas the reinforced robot performs, after some time of exploration, very fast spinning motions, which are persistent for several minutes.

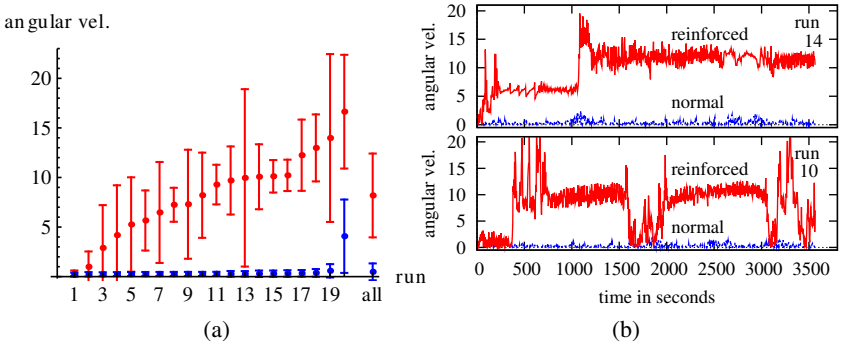


Fig. 8.15 Performance of the spherical robot rewarded for spin. **(a)** Mean and std. deviation of the angular velocity ω_z of the spherical robot for 20 runs each 60 min long with (**red**) and without (*blue*) spin reinforcement, sorted by angular velocity. The label '**all**' denotes the mean and std. deviation over the means of all runs. **(b)** Time course of the velocity for run number 12 and 20, where *blue/dotted* shows the normal case and **red/solid** line shows the reinforced case.

In this setup it can also be seen that the rewarded behaviors are found again after they were lost, see Fig. 8.15(b).

The mechanism to modulate the learning speed by a reward signal showed a strong effect on the behavior of the spherical robot. When controlled by the homeokinetic controller without guidance the robot rarely exhibits narrow curves or spinning behavior. In contrast the guided controller engaged the system into curved motion most of the time. One might wonder how it is possible that this technique is able to reach a behavior that is normally not exhibited. The reason is that when the robot is starting to follow a curve, then the learning rate of the controller goes down, although the forward model is still learning normally. In the unguided case the prediction error rises (because it is a new behavior) and thus the controller will quickly leave this behavior. This actually happens before a fast spinning is reached. In the rewarded case the forward model is able to capture the behavior before it is left (because of the slower drift), which in turn enables the control system to enter modes of more narrow curves.

8.7 Channeling Self-Organization

Periodic behaviors, such as observable in locomotion, are characterized by a particular spatio-temporal structure which can be described in terms of phase relations between the joints. Vice versa, by imposing certain phase relations a bias towards a specific behavior can be conveniently introduced into the dynamical system. For this purpose we will use again soft constraints that break symmetries in a particular way, reduce the effective dimension of the sensorimotor dynamics, and guide thus the self-organizational process towards a subspace of the original control problem. In biological systems similar constraints are known to be effective on a low level of neuronal circuitry, e. g. linking pairs of antagonistic muscles such that the activity of one muscle inhibits activity of the other via inter-neurons in the spinal cord (Pearson and Gordon 2000).

We will apply here an analogous regulation method which refers to motor values of one effector as teaching signals for another one, and will call this scheme *cross-motor teaching*. It will be used to prescribe which motor neuron receives a teaching signal from which other neuron. Note that despite the use of 'teaching signals' the algorithm is completely unsupervised, because the signals are generated internally. The self-organization progress preserves a high amount of symmetries of the physical system. As an example, consider a two-wheeled robot that drives forward and backward and rotates clockwise and counterclockwise equally often. The physical system (morphology of the body and interaction) is essentially symmetric with respect to forward-backward, left-right (lateral), and also straight-rotational behavior. To the contrary, if the robot lacks forward-backward symmetry and, more importantly, also straight-rotational symmetry because of friction and inertia. This is also reflected in behavior in that the robot is more driving straight than rotating.

8.7.1 From Spontaneous to Guided Symmetry Breaking

To achieve symmetry breaking in a predefined way, we will first consider pairwise relations as constraints for the broken-symmetric state. Later we will generalize this by using permutation relations. Let us, e.g. influence the controller to prefer a pairwise *in-phase* or *antiphase* relations in the motor patterns (Martius and Herrmann 2010). For a particular pair of motors (r,s), we place a bidirectional cross-motor connection from r to s, which means that the motor s receives its teaching signal from motor r and vice versa. In this way both motors are guided towards an in-phase activity. The (internal) teaching signal is

$$\left(y_t^G\right)_r = (y_t)_s \qquad \text{and} \qquad \left(y_t^G\right)_s = (y_t)_r \,, \tag{8.28}$$

which is used then in Eqs. (8.12–8.15).

Likewise, an antiphase teaching relation can be expressed by $\left(y_t^G\right)_r = -(y_t)_s$ and vice versa. In this simple setup the cross-motor connections have either a positive or negative sign. For those motors i that are not part of a connected pair we need to set $\left(y_t^G\right)_i = (y_t)_i$, in order to suppress the error signal, see Sect. 8.4.2.

8.7.1.1 Experiment

To illustrate the concept we will consider the above-mentioned two-wheeled robot, cf. Fig. 8.16c. The robot has two motors actuated according to y_1 and y_2 and is subject to the goal of straight driving. This can be obtained by an in-phase relation between both motors following Eq. (8.28), i. e.

$$\left(y_t^G\right)_1 = (y_t)_2 \qquad \text{and} \qquad \left(y_t^G\right)_2 = (y_t)_1 \,. \tag{8.29}$$

For experimental evaluation we placed the robot in an environment cluttered with obstacles.

We performed, for different values of the factor γ, five runs of 20 min length. In order to quantify the influence of the cross-motor teaching we recorded the trajectory, the linear velocity, and the angular velocity of the robot. We expect an increase in linear velocity because the robot is to move straight instead of turning. For the same reason the angular velocity should go down. In Fig. 8.16 a sample trajectory and the behavioral quantifications are plotted. Additionally, we plot the relative area coverage which is calculated from the trajectory using a box-counting method. It reflects how much area of the environment was covered by the robot with cross-motor teaching compared to the original homeokinetic controller. As expected, the robot shows a distinct decrease in mean turning velocity and a higher area coverage with increasing values of the guidance factor. Note that the robot is still performing turns and drives both backwards and forwards and does not get stuck at the walls, as seen in the trajectory in Fig. 8.16(c). The properties of the homeokinetic controller, such as sensitivity and exploration, remain.

We have seen that a pairwise cross-motor teaching can be used to guide the self-organizing control to drive mostly straight in the two-wheeled robot. The strength

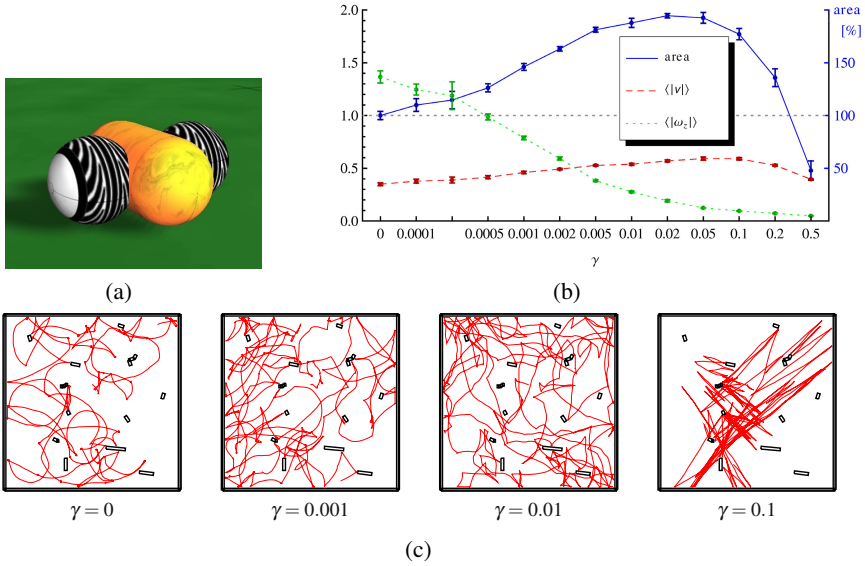


Fig. 8.16 Behavior of a two-wheeled robot (a) guided to move preferably straight. (b) Mean and standard deviation (of 5 runs each 20 min) of the area coverage, the average velocity $\langle|v|\rangle$, and the average turning velocity $\langle|\omega_z|\rangle$ for different values of the guidance factor γ. Area coverage (box counting method) is given relative to the the case without influence ($\gamma=0$: 100%) (right axis). The robot drives straighter and covers more area for increasing γ, until at large γ the teaching strictly dominates the behavior of the robot. **(c)** Example trajectories for different guidance factors. Parameters: $\varepsilon_c = \varepsilon_A = 0.01$.

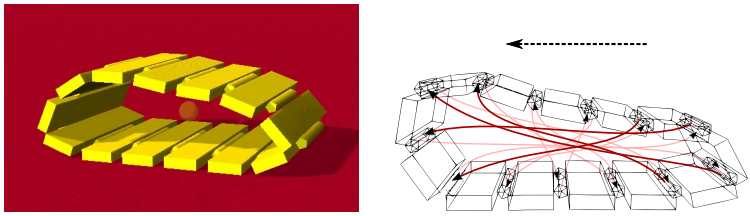
of this preference can be adjusted by the guidance factor. The algorithm is self-supervised and the only specific information that is given is the pair of motors to be synchronized.

8.7.2 *Multiple Motor Relations*

Now we want to consider a more general cross-motor connection setup where each motor has one incoming and one outgoing connection, such that there is still only one teaching signal per motor neuron (Martius and Herrmann 2011). The cross-motor connections can be described by a permutation π_m of the m motor neurons assigning each motor neuron a source of teaching input. The teaching signal is then given by (dropping the time index)

$$y_i^G = y_{\pi_m(i)} \qquad \text{for } i = 1,\ldots,m. \tag{8.30}$$

Additionally a sign function could be used defines whether the motors are supposed to be in-phase or antiphase, but we do not need it in the following. The pairwise



(a) (b)

Fig. 8.17 The armband robot. **(a)** Screen shots of the simulation. The transparent sphere in the center marks the center of mass of the robot. **(b)** Track-robot armband with cross-motor connections. The arrows indicate unidirectional cross-motor connections, where the head points to the receiving unit. All links are equal, but for visibility reasons only four links are drawn boldly. For this connection setup the robot preferably moves leftwards.

setup (Eq. (8.28)) is of course a special case of this notation. Note, that with a cyclic schema of connections also a group of motors can be synchronized.

8.7.3 Guiding to Directed Locomotion

In order to study a robot with a scalable complexity, we will consider the *armband* robot—a bracelet- or track-like structure. We will see that we can explicitly guide the robot to a directed and fast locomotion by organizing the initially decentralized control into a cooperative mode which can be considered as the emergence of a single controller for the entire robot.

This robot consists of a sequence of m flat segments placed in a ring-like configuration, where subsequent segments are connected by the m hinge joints. The resulting body has the appearance of a bracelet or chain, see Fig. 8.17(a). Each joint is driven by a servo motor and has a joint-angle sensor. The center positions of the joints are such that the robot is in a perfectly circular configuration (deviating by an angle of $2\pi/m$ from a straight positioning). The motor values and sensor values are represented as well as joint angle deviations, see Fig. 8.17(b). The joints are highly coupled through the ring configuration. Therefore, an independent movement of a single joint is not possible. Instead it has to be accompanied by a movement of the neighboring joints and of distant joints.

Since the robot is symmetric there is by construction no preferred direction of motion, meaning that the robot controlled by the homeokinetic controller will equally probable move forward or backward. The robot cannot turn or move sideways, but it can produce a variety of postures and locomotion patterns.

With the method of cross-motor teaching we can help to break different symmetries, such that the robot is more likely to perform a directed motion. One possibility is to connect motors on opposite sides of the robot with a bias in clockwise or counterclockwise direction. For that we define the permutation (used in Eq. (8.30)) as

$$\pi_m(i) = (i + k + \lfloor m/2 \rfloor) \mod m\,, \tag{8.31}$$

where $k \in \{-1, 0, 1\}$ and $\lfloor \cdot \rfloor$ denotes the truncation rounding (floor). We will only use positive connections, such that the sign function is not required. Thus, the teaching signals are (omitting the time index)

$$y_i^G = y_{(i+k+\lfloor m/2 \rfloor) \mod m} \qquad \text{for } i = 1, \ldots, m\,. \tag{8.32}$$

The choice of k depends on the desired direction of motion and on whether the number of joints m is even or odd. If m is even then $k = -1$ and $k = 1$ are used for both directions (forward or backward) and $k = 0$ represents a symmetric connection setup. In the latter case the robot will not prefer a direction of motion and the behavior is similar to the one without guidance. For an odd value of m, which is also used here, $k = 0$ and $k = 1$ need to be used for backward and forward motion.

In the following experiments the robot has $m = 13$ motors. The motor connections for $k = 1$ are illustrated in Fig. 8.17. Each motor connection is displayed by an arrow pointing to the receiving motor. Note that the connections are directed and a motor neuron is not teaching the motor neuron from which it is receiving teaching signals. For $k = 0$ all arrows are inverted, meaning that for each connection the sending and receiving motor neurons swap roles.

To evaluate the performance we conducted, for different values of the guidance factor γ, 5 trials each 30 min long. In a first setting the cross-motor connections were fixed ($k = 1$) for the entire duration of the experiment. Without guidance the robot moves equally to both directions but with comparably low velocity. This can be seen at the mean of the absolute velocity in Fig. 8.18(a). If the value of the guidance factor is chosen conveniently, we observed the formation of a locomotion behavior after a very short time and the robot moves in one direction with varying speed see Fig. 8.18(b) for three velocity traces. Note that this behavior requires all joints of the robot to be highly coordinated. We also observe a peak of high velocity after the first few minutes, which is followed by a dip before a more steady regime is attained. During this time the controller is going from a subcritical regime (at $t = 0$) to a slightly supercritical regime.

The locomotory behavior can also be seen in Fig. 8.19 for a low value of guidance factor ($\gamma_s = 0.001$) and in Fig. 8.20 for a medium value of guidance factor ($\gamma = 0.003$). The average velocity of the robot increased distinctively with rising guidance factors, see Fig. 8.18(a). However, for excessively large values of the guidance factor the velocity goes down again. This occurs for two reasons: First, the cross-motor teaching has a too strong influence on the working regime of the homeokinetic controller and second the actual motor pattern of the locomotion behavior does not perfectly obey the relations between the motor values as specified by Eq. (8.32). In order to satisfy the constraints imposed by Eq. (8.32) all motor values need to be equal, which is of course not the case in the locomotion behavior.

In a second setup we changed the cross-motor connections every 5 min, i. e. k was changed from 0 to 1 and back. A value of $k = 0$ should lead to a negative velocity and a value of $k = 1$ to a positive velocity.

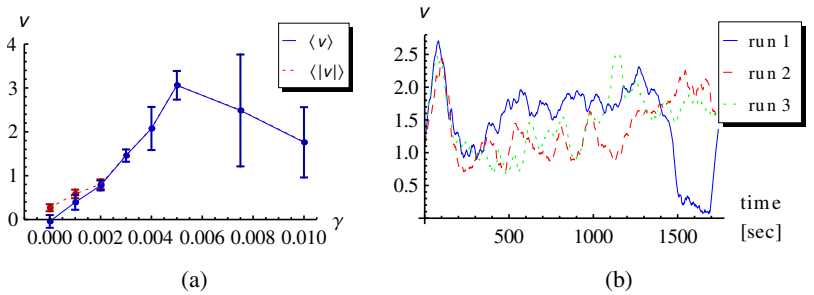


Fig. 8.18 Performance of the armband robot with constant cross-motor teaching. **(a)** Mean and standard deviation of the average velocity $\langle v \rangle$ and the average absolute velocity $\langle |v| \rangle$ of 5 runs for different value of the guidance factor γ. **(b)** Velocity of the robot $\overline{v}$ (averaged over 1 minute sliding window) for 3 runs at $\gamma = 0.003$. Parameters: $k = 1$, $\varepsilon_c = \varepsilon_A = 0.1$.

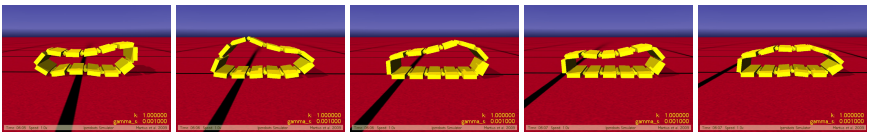


Fig. 8.19 The armband robot learns to locomote by weak guidance. Behavior of the robot with cross-motor teaching and weak guidance ($\gamma = 0.001$). A slow locomotive behavior with different velocities is exhibited. Explorative actions cause the posture of the robot to vary in the course of time.

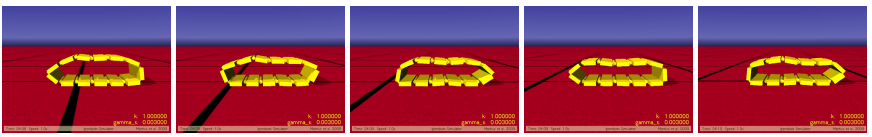


Fig. 8.20 The armband robot quickly learns to locomote. Behavior of the robot with cross-motor teaching and medium guidance ($\gamma = 0.003$). Comparable fast locomotive behavior emerges quickly and is persistent. Nevertheless the velocity varies. Only small exploratory actions are takes, such that the posture is mainly constant.

To study the dependence on the guidance factor and to measure the performance we use the average absolute velocity $\langle |v| \rangle$ and the correlation of the velocity with the configuration of the coupling $\rho(v,k)$, see Fig. 8.21(a). Without guidance ($\gamma = 0$) there is, as expected, no correlation with the supposed direction of locomotion. For a range of values of the guidance factor we find a high total locomotion speed with a strong correlation to the supposed direction of motion. Note that the size of the correlation depends on the length of the intervals of one connection setting. For long intervals the correlation will approach one. In Fig. 8.21(b) the velocity of the robot is plotted for different runs with the same value of the guidance factor that was used in the previous experiment ($\gamma = 0.003$). We observe that the robot changes the

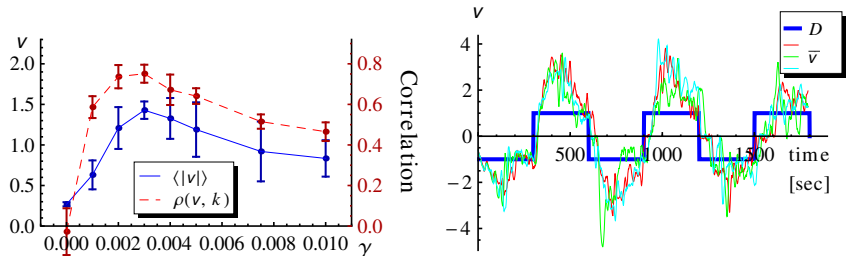


Fig. 8.21 Performance of the armband robot for variable cross-motor teaching. **(a)** Mean and standard deviation of the average absolute velocity $\langle|v|\rangle$ and the correlation $\rho(v,k)$ of the velocity with the configuration of the coupling for 5 runs with different values of the guidance factor γ. **(b)** Velocity (averages over 10 sec sliding windows) of the robot for 3 runs at $\gamma = 0.003$ and the target direction of motion $D = 2k - 1$ for better visibility. Parameters: $\varepsilon_c = \varepsilon_A = 0.1$.

direction of motion shortly after the configuration of the coupling was changed, see Fig. 8.21

8.7.4 Scaling Properties

The locomotion of the robot is essentially influenced by the number of cross-motor connections. To study this we use again the fixed connectivity. In a series of simulations a number $0 \leq l \leq m$ of equally spaced cross-motor connections (Fig. 8.17) are used. With increasing l the robot starts to locomote earlier. Full performance is reached already if 8 out of the 13 connections are used, see Fig. 8.22(a).

In order to study the scaling properties of the learning algorithm we varied the number of segments m of the robot and thus the dimensionality of the control problem. The results are astonishing, see Fig. 8.22(b): The behavior is learned with the same speed also for large number (40) of segments. There is no scaling problem here for the following reason. In the closed loop with an approximate feedback strength (self-regulated by the homeokinetic controller) the robot needs only very little influence to roll. The length of the robot can even help because other behavioral modes (e. g. wobbling) are damped increasingly due to gravitational forces. For the same reason, small robots are slower than medium ones. Large robots are again slower because the available forces at the joints become too weak.

The experiment illustrates that specific behaviors can be achieved in a high-dimensional robot by using cross-motor teaching. Cross-motor connections can break the symmetry between the two directions of motion such that a locomotion behavior is produced quickly. When the connections are switched during runtime, the behavior of the robot changes reliably.

The mechanism proposed here can also be transferred to sensor space using the direct sensor teaching (Sect. 8.4.3) instead of the motor teaching. One obtains a

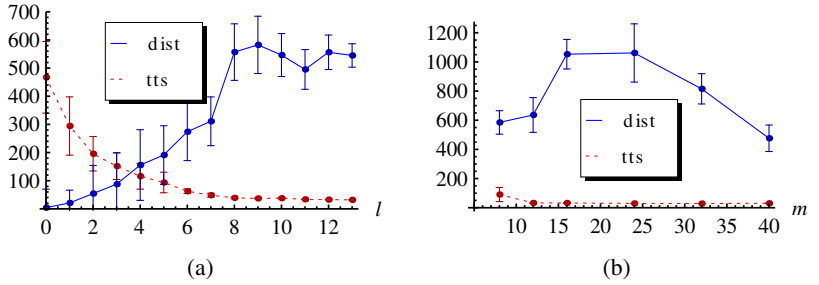


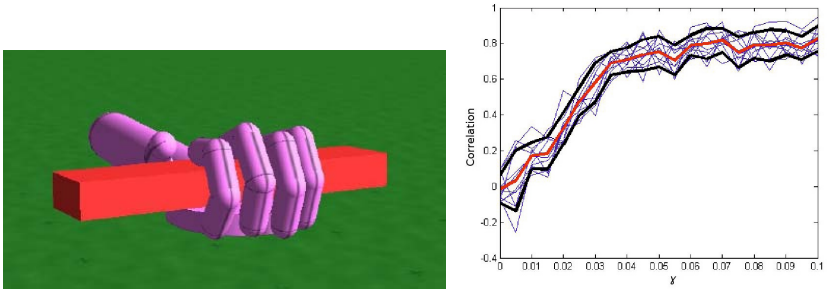
Fig. 8.22 Scaling of learning time and performance for different robot complexity. The plots show mean and standard deviation of the distance traveled by the robot ('dist' in units of 1 segment size) and of the time-to-start ('tts' in seconds) of 20 runs à 10 min ($\gamma = 0.003$). **(a)** Performance as a function of the number of cross-motor connections l (equally spaced around a robot with $m = 13$ joints). **(b)** Performance for different numbers of segments m (DoF) with full cross-motor connectivity ($l = m$).

cross-sensor teaching analogously to the definitions given above. This can become useful, for example if a certain behavior is demonstrated by a human operator by activity moving the robot. In the case of the armband robot, one can imagine pushing the robot along the ground forcing it into a locomotion pattern. Based on the observed sensor readings, the correlations between the sensor channels may be determined and used as a basis for the construction of a specific cross-sensor teaching setup. This highly interesting idea was, however, not yet implemented and remains for future work.

Starting from the guidance by teaching we introduced the concept of cross-motor teaching allowing for the specification of abstract relations between motor channels. There are no external teaching signals required, because the motor values are used mutually as teaching signals. The only specific information put into the system is the cross-motor relation. First we studied simple pairwise relations and shaped the behavior of the two-wheeled robot to drive mostly straight through the coupling between both motors. The couplings introduce soft constraints that guide the self-organization process to a subspace of the entire sensorimotor space and therewith the effective dimension of the search space for behaviors is reduced. This was demonstrated using the high-dimensional armband robot. With a simple cross-motor teaching the robot developed within a short time fast locomotion behaviors from scratch. The direction of motion was altered by a change in the connection setup. Remarkable is also the scaling property with respect to the dimensionality of the control problem.

8.7.5 Coordination of Finger Movements for Grasping

An interesting application of the above method is in neuroprosthetics, where often little information is available in complex control problems. We will consider the control of a prosthetic hand in a simulation. The simulated hand has six controllable



(a) (b)

Fig. 8.23 Guided self-organization with homeokinesis in a simulated hand prosthesis. **(a)** The prostheses has six controllable degrees of freedom. A teaching term based on the similarity of the finger angles keeps the fingers in near synchrony while they are moving independently when controlled by the basic homeokinetic rule. **(b)** For a guidance factor $\gamma \approx 0.03$ correlations between fingers are reached that are similar to observations in healthy humans (Santello and Soechting 2000). The red line shows the mean value over 10 trial and black lines the respective standard deviation.

DoF, two for the thumb and one for the other fingers, i.e. the fingers have a fixed coupling between their three degrees of mobility, see Fig 8.23(a). All fingers are equipped with position sensors of the joints and proximity sensors in the tip. If the fingers are controlled by the homeokinetic controller they develop independent movements because no physical coupling is present in this robotic model. Since in a natural environment the fingers interact mostly because of the manipulation of objects and because of physiological constraints, we can also try learning such correlations by a guidance principle. In this way we will arrive at a measure of the required interaction which then can be compared with observations in healthy humans.

In order to enforce movement synergies between the fingers, we implemented finger correlation by cross motor teaching between all fingers. Here we have multiple teaching signals for each motor neuron, where simply the arithmetic average is used.

The mean correlation of the homeokinetic controller without guidance ($\gamma = 0$) is 0, which means here the fingers move independently. At high values of γ the correlation approaches unity, which indicates that the fingers have lost independence which however would be needed in grasping applications. Considering the correlation value of 0.582 given for human fingers (Santello and Soechting 2000), the optimal value of γ would be around 0.03, see Fig. 8.23(b). Again a very weak guidance is sufficient to influence the behavior in the desired direction.

8.8 Discussion

In this chapter we have presented several mechanisms for guided self-organization of robot behavior based on homeokinetic control (GSOH). Homeokinesis bootstraps the exploration process of embodied systems and leads to self-organization of behavior. Various patterns of behavior emerge depending on the robotic hardware and its environment. With a general framework of problem specific error functions we set the foundation for guidance by teaching signals and guidance by cross-motor teaching. The balance between self-organized behavior and target behavior can be adjusted with a single parameter.

Interestingly, teaching signals can as well be provided in terms of desired sensor values. In this setting, for instance a spherical robot was taught to rotate around one particular axis solely by requesting a zero value of the sensor value corresponding to that axis. In a more elaborate example we show how the task of finding balls and pushing them around in an environment can be achieved by simply providing a desired visual sensor state. The entire sensorimotor coordination to fulfill this goal was learned by the robot within a few minutes. This involves the basic coordination to drive the robot and to integrate the vision sensors such that the balls are approached and balanced while pushed. To probe the robustness of the approach the orientation of the camera was abruptly changed such that a completely different sensorimotor coordination becomes necessary. We found that GSOH can cope with a wide range of configuration changes, even those where a complete change in the visual sensation occurs (signs of all visual sensors swapped).

The teaching mechanisms form the basis for a higher level guiding mechanism, namely cross-motor teaching. It allows to specify relations between motor channels to be in-phase or antiphase activity. This induces soft constraints and therewith reduces the effective dimensionality of the system. This was especially illustrative with the high-dimensional armband robot. A cross-motor teaching with only one connection per joint leads to a fast and coordinated locomotion behavior. Similar to the robustness in the vision experiments, we observe here a rapid and reliable change in the direction of locomotion by an altered connection setup. A particularly promising result is that the performance and speed of learning is almost independent of the dimensionality of the system, at least in the here considered cases of up to 40 DoF. The discrete cross-motor connections offers a good way for higher level control structures to direct the behavior of the robot.

We also presented a simple method to guide the self-organizing behavior using online reward signals, originally published in (Martius et al. 2007). In essence the original time-loop error is multiplied by a strength factor, obtained from the reward signal. The approach was applied to the spherical robot with two goals, fast motion and curved rolling, which was successfully achieved. Notably, the exploratory character of the paradigm still remains intact.

To compare the different methods of GSOH we can ask for the amount and type of information that is required about the behavior and the robotic system. For the direct teaching methods a rather detailed insight into the sensorimotor patterns of the desired behavior is required. In sensor space this is typically easier than in

motor space as demonstrated by the examples. For the cross-motor teaching a more high-level knowledge is sufficient, for instance about the symmetries of the body and of the desired motion. In both cases the designer needs to expect a specific behavior, e. g. a locomotion behavior with a certain gait. In the reward based method, on the other hand, the realization is not specified, e. g. the gait would be found autonomously. To be successful, however, the exploration needs to be structured enough to produce short segments of locomotion behavior to be picked and amplified. Here the newest methods for behavioral self-organization using information-theoretic quantities show promising results (Martius et al. 2013; Der and Martius 2013).

Let us briefly compare GSOH with other approaches to learning of autonomous robot behavior, namely evolutionary algorithms (EA) (Nolfi and Floreano 2001) and reinforcement learning (RL) (Sutton and Barto 1998). EA and RL can optimize the parameters of the controller (e. g. a neural network) and can in principle achieve the behaviors demonstrated here. There are many impressive results where systems of similar dynamical complexity have been successfully controlled, see for example (EA) Chemova and Veloso (2004); Bongard et al. (2006); Mazzapioda and Nolfi (2006); de Margerie et al. (2007); Ijspeert et al. (1999) and (RL) Peters and Schaal (2008). In high-dimensional systems, however, identical subcomponents are typically used or the problem is appropriately prestructured by hand. Additionally, long learning times are required (many generations with many individuals or repetitions) which is often prohibitively long for physical robots. Here we see the main strength of our system: The desired behaviors are found very fast even in high-dimensional and dynamically complex systems—we have very fast online-learning. Another difference is that the finally evolved or learned controllers are typically static, such that it only works in the conditions it was evolved/trained in. In contrast we demonstrated the robustness of GSOH to extreme sensor disruption, which is successful due to a continuous self-modeling and exploration.

Of course there is also a downside, namely that the here proposed approaches are rather limited in which behaviors can be achieved and how for instance the reward can be given. Also in GSOH the desired behaviors are only partially followed and no optimality guarantees can be given which is in contrast to RL that was proven to converge to the optimal solution under certain conditions (Sutton and Barto 1998). However, for practical applications these proves are of questionable value because a prohibitive amount of time is required.

To conclude, the GSOH methods offer a fast development of goal-oriented behaviors in high-dimensional continuous-domain robotic systems from scratch, which cannot be achieved with other learning control systems so far. However, the implementation of goals is comparably limited. The reward-based guidance allows any reward signals, but no time delays are tolerated and it is not guaranteed that the reward is maximized. The cross-motor teaching method is suitable to select a subset of behaviors, but cannot be generalized to all behaviors. A combination of both methods is also conceivable, namely using cross-motor teaching to be very effective in high-dimensional systems and additionally using rewards to give a more fine grain control over the behavior. Another line of future research would be the

proposed cross-sensor teaching that would allow for the specification of behaviors on the level of sensor relations. We also expect that superior results can be obtained when the here proposed methods are combined with the new algorithms for behavioral self-organization (Der and Martius 2013) as they produce more structure in the emerging behaviors.

Acknowledgment. We thank Raluca Scona and Guillaume de Chambrier for providing us with some of the figures. RD thanks Nihat Ay for the hospitality in his group. The project was supported by BCCN Göttingen grant #01GQ0432, BFNT Göttingen grant #01GQ0811 and DFG (SPP 1527).

References

Amari, S.: Natural gradients work efficiently in learning. Neural Computation 10 (1998)

Bongard, J.C., Zykov, V., Lipson, H.: Resilient machines through continuous self-modeling. Science 314, 1118–1121 (2006)

Butera, F.M.: Urban development as a guided self-organisation process. In: The City and Its Sciences, pp. 225–242. Springer (1998)

Cannon, W.B.: The wisdom of the body. Norton, New York (1939)

Chemova, S., Veloso, M.: An evolutionary approach to gait learning for four-legged robots. In: Proc. IEEE IROS 2004, vol. 3, pp. 2562–2567 (2004)

Choi, J., Wehrspohn, R.B., Gösele, U.: Mechanism of guided self-organization producing quasi-monodomain porous alumina. Electrochimica Acta 50(13), 2591–2595 (2005)

Cruse, H., Dürr, V., Schmitz, J., Schneider, A.: Control of hexapod walking in biological systems. In: Adaptive Motion of Animals and Machines, pp. 17–29. Springer (2006)

de Margerie, E., Mouret, J.-B., Doncieux, S., Meyer, J.-A.: Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV. Bioinspiration and Biomimetics 2, 65–82 (2007)

Der, R.: Self-organized acquisition of situated behaviors. Theory Biosci. 120, 179–187 (2001)

Der, R., Liebscher, R.: True autonomy from self-organized adaptivity. In: Proc. Workshop Biologically Inspired Robotics, Bristol (2002)

Der, R., Martius, G.: The Playful Machine - Theoretical Foundation and Practical Realization of Self-Organizing Robots. Springer (2012)

Der, R., Martius, G.: Behavior as broken symmetry in embodied self-organizing robots. In: Advances in Artificial Life, ECAL 2013 (accepted, 2013)

Dongyong, Y., Jingping, J., Yuzo, Y.: Distal supervised learning control and its application to CSTRsystems. In: SICE 2000. Proc. of the 39th SICE Annual Conference, pp. 209–214 (2000)

Ijspeert, A.J., Hallam, J., Willshaw, D.: Evolving Swimming Controllers for a Simulated Lamprey with Inspiration from Neurobiology. Adaptive Behavior 7(2), 151–172 (1999)

Jordan, M.I., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. Cognitive Science 16(3), 307–354 (1992)

Klyubin, A.S., Polani, D., Nehaniv, C.L.: Empowerment: a universal agent-centric measure of control. In: IEEE Congress on Evolutionary Computation, pp. 128–135. IEEE (2005)

Martius, G.: Robustness of guided self-organization against sensorimotor disruptions. Advances in Complex Systems 16(02n03), 1350001 (2013)

Martius, G., Der, R., Ay, N.: Information driven self-organization of complex robotic behaviors. PLoS ONE 8(5), e63400 (2013)

Martius, G., Herrmann, J.M.: Taming the beast: Guided self-organization of behavior in autonomous robots. In: Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J.-A., Mouret, J.-B. (eds.) SAB 2010. LNCS, vol. 6226, pp. 50–61. Springer, Heidelberg (2010)

Martius, G., Herrmann, J.M.: Tipping the scales: Guidance and intrinsically motivated behavior. In: Advances in Artificial Life, ECAL 2011, pp. 506–513. MIT Press (2011)

Martius, G., Herrmann, J.M.: Variants of guided self-organization for robot control. Theory in Biosci. 131(3), 129–137 (2012)

Martius, G., Herrmann, J.M., Der, R.: Guided self-organisation for autonomous robot development. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 766–775. Springer, Heidelberg (2007)

Martius, G., Hesse, F., Güttler, F., Der, R.: LpzRobots: A free and powerful robot simulator (2012), `http://robot.informatik.uni-leipzig.de/software`

Mazzapioda, M., Nolfi, S.: Synchronization and gait adaptation in evolving hexapod robots. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 113–125. Springer, Heidelberg (2006)

Nolfi, S., Floreano, D.: Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines. MIT Press, Cambridge (2000) (1st print) (2001) (2nd print)

Ott, E., Grebogi, C., Yorke, J.: Controlling chaos. Phys. Rev. Lett. 64, 1196–1199 (1990)

Pearson, K., Gordon, J.: Spinal reflexes. In: Kandel, E., Schwartz, J.H., Jessell, T.M. (eds.) Principles of Neural Science, 4th edn., pp. 713–736. McGraw-Hill, New York (2000)

Peters, J., Schaal, S.: Natural Actor-Critic. Neurocomputing 71(7-9), 1180–1190 (2008)

Popp, J.: Spherical robots (2004), `http://www.sphericalrobots.com`

Prokopenko, M.: Design vs self-organization. In: Prokopenko, M. (ed.) Advances in Applied Self-organizing Systems, pp. 3–17. Springer (2008)

Prokopenko, M.: Guided self-organization. HFSP Journal 3(5), 287–289 (2009)

Rodriguez, A.: Guided Self-Organizing Particle Systems for Basic Problem Solving. PhD thesis, University of Maryland (College Park, Md., USA) (2007)

Santello, M., Soechting, J.F.: Force synergies for multifingered grasping. Experimental Brain Research 133(4), 457–467 (2000)

Schaal, S., Ijspeert, A., Billard, A.: Computational approaches to motor learning by imitation, vol. 1431, pp. 199–218. Oxford University Press (2004)

Smith, S.C., Herrmann, J.M.: Homeokinetic reinforcement learning. In: Schwenker, F., Trentin, E. (eds.) PSL 2011. LNCS, vol. 7081, pp. 82–91. Springer, Heidelberg (2012)

Stitt, S., Zheng, Y.F.: Distal learning applied to biped robots. In: Proc. of the IEEE Intl. Conf. on Robotics and Automation, pp. 137–142. IEEE Computer Society (1994)

Sutton, R.S.: Reinforcement learning: Past, present and future. In: McKay, B., Yao, X., Newton, C.S., Kim, J.-H., Furuhashi, T. (eds.) SEAL 1998. LNCS (LNAI), vol. 1585, p. 195. Springer, Heidelberg (1999)

Wikipedia (2013). Homeostasis — wikipedia, the free encyclopedia (Online accessed July 23, 2013)