Guided Self-organisation for Autonomous Robot Development

Georg Martius[1,2,3], J. Michael Herrmann[1,2,3], and Ralf Der[4]

[1] Bernstein Center for Computational Neuroscience Göttingen,
[2] University of Göttingen, Institute for Nonlinear Dynamics,
[3] Max Plank Institute for Dynamics and Self-Organization,
Bunsenstrasse 10, D-37073 Göttingen, Germany
`{georg,michael}@nld.ds.mpg.de`
[4] University of Leipzig, Institute of Computer Science,
PF. 920 D-04009 Leipzig, Germany
`der@informatik.uni-leipzig.de`

Abstract. The paper presents a method to guide the self-organised development of behaviours of autonomous robots. In earlier publications we demonstrated how to use the homeokinesis principle and dynamical systems theory to obtain self-organised playful but goal-free behaviour. Now we extend this framework by reinforcement signals. We validate the mechanisms with two experiment with a spherical robot. The first experiment aims at fast motion, where the robot reaches on average about twice the speed of a not reinforcement robot. In the second experiment spinning motion is rewarded and we demonstrate that the robot successfully develops pirouettes and curved motion which only rarely occur among the natural behaviours of the robot.

Key words: autonomous robots, self-organised behaviour, reinforcement learning, developmental robotics, homeokinesis

1 Introduction

Self-organisation is a key phenomenon in many disciplines ranging from physics over chemistry to the life sciences and economy. It centres on the spontaneous creation of patterns in space, time or space-time in complex systems. The dynamical systems approach to robotics describes robotic behaviour as a spatio-temporal pattern which is formed in the complex interaction of the robot and its environment. Our interest is in developing a systematic approach to the behavioural self-organisation of such systems.

Self-organisation needs a general paradigm which has to be domain invariant. An exemplary paradigm of such generality is homeostasis meant in the early days of cybernetics to be a basis of self-organisation. There are a few attempts to introduce homeostatic mechanisms in robotics, cf. [1, 2]. However, while obviously helpful in stabilising systems the principle of homeostasis seems of limited use for the construction of behaviour systems.

One of the authors proposed some time ago *homeokinesis* as a dynamical counterpart to homeostasis, see [3, 4]. The idea is that in a behaving system the components like neurons, sensors, motors or muscles have to cooperate their activities in a common kinetic state. As with homeostasis this paradigm is not constructive, because it does not tell how to reach the pertinent state. In particular it gives no answer to the basic question why the robot should do anything at all. One solution is the so called time loop error (TLE) see [5, 6] and Sec. 3.2 below. There, the drive for activity has been rooted into the principle itself, and the creation of activity and the adaptation to the environment are combined into one single quantity. The development of the robot is driven by the minimisation of the TLE, which is entirely defined in internal terms of the robot.

Applications with both real [7] and simulated robots have shown many interesting and unexpected behaviours ranging from coiling, hurling and jumping modes in snake like artifacts, over stable rolling modes of spherical robots [8] to dogs climbing over walls and the like, see our video page [9]. What we observe in these experiments are behaviours with high sensorimotor coordination, emerging in a "playful" exploration of the bodily affordances. However, so far all the emerging behaviours are contingent, depending on the concrete body and environmental conditions. Moreover, emerging behaviours are in general transient which may be viewed as the sequential creation and destruction of behavioural primitives.

In the present paper we report a first result of guiding self-organisation into the direction of desired behaviours. In the specific case we consider a spherical robot which earlier has been demonstrated to develop different rolling modes of various velocities and modalities [10]. Our aim now is to tell the robot to move fast or to spin and let self-organisation do the rest. This goal is reached by modulating the TLE with a conveniently defined reward signal as defined below. This simple principle is shown to work in a surprisingly effective way and the presented results may indicate a general approach to influence self-organisation by general reinforcement signals.

There is a close relationship to the attempts of guiding autonomous learning by internal reinforcement signals [11] and to task independent learning [12, 13]. The difference is that these approaches work best with discrete state-action spaces of not too large dimensions. Our approach on the other hand was demonstrated to work in real time and continuous space with robots of up to 25 active degrees of freedom [9].

The paper is structured as follows: In the next section we will explain the used robots, then our general principle for self-organisation is formulated. After that a short example for exploration is given followed by the main section about the guiding experiments and finally we close with a discussion and an outlook.

2   Robots

For the experiments we use both a simulated spherical robot called "Sphere" and a cylindrical robot called "Barrel", see Fig. 1. The Sphere was inspired orig-

inally by Julius Popp [14]. We constructed the Barrel because it is easier to analyse and shows clear effects. We used the ODE library (open dynamic engine
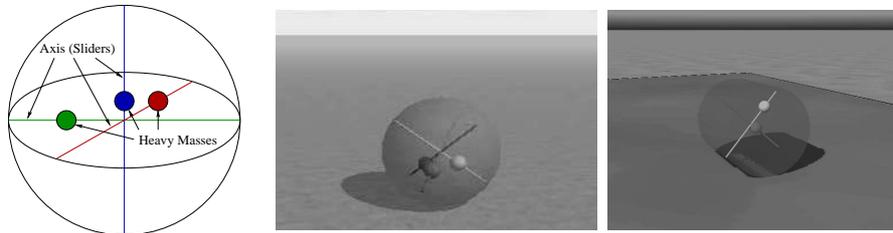

Fig. 1. Simulated spherical robot "Sphere" and cylindrical robot "Barrel". *Left:* Sketch of a the Sphere with three internal sliders. The Barrel has only two sliders; *Center:* Picture of the Sphere on the ground; *Right:* Picture of the Barrel on the ground.

[15]) embedded in our simulation framework [16] for the computer simulations. The robots are driven by shifting the centre of mass which is realised by shifting internal masses by servo motors, situated on the orthogonal axes (three in the Sphere an two in the Barrel). The motor values are the target positions of each of the masses on its axis, symmetric around the centre ranging to half of the radius. Collisions of these masses are ignored. The servo motors move the masses by applying forces to them, which are calculated by a PID controller. This provides more reliable control of the mass positions and stabilises them against perturbations and centrifugal forces.

The Sphere is equipped with three proprioceptive sensors, which measure the projections of the axes vectors on the z-axis of the world coordinate system, i.e. the z-component of each axis vector. The Barrel only has two such sensors.

Both Sphere and Barrel are physical objects with a complicated mapping of motor to sensor values. In fact, shifting of a mass position will have quite different consequences due to inertia. The task of the controller is to close the sensorimotor loop so that a rolling motion of the robot is achieved. This would be usually done by constructing the controller conveniently. In our case the rolling motion will emerges from our general principle given below.

3 A General Approach to Self-organisation

We will give here a short review of the general homeokinesis approach. Central to our approach is the internal perspective, i.e. everything is based on the stream of the sensor values represented by $x_t \in \mathbb{R}^n$ where $x_t = (s_{t1}, \ldots, s_{tn})$ are the n sensor values at time $t = 0, 1, 2, \ldots$. The controller is given by a function $K : \mathbb{R}^n \to \mathbb{R}^m$ mapping sensor values $x \in \mathbb{R}^n$ to motor values $y \in \mathbb{R}^m$

$$y_t = K\left(x_t\right) \ . \tag{1}$$

In the example we have $y_t = (y_1^t, y_2^t, y_3^t) \top$, y_i^t being the servo target positions of the internal masses on the axes. Our controller is adaptive, i.e. it depends on a set of parameters $C \in \mathbb{R}^c$. In the cases considered here the controller is realised by a one layer neural network defined by the pseudo-linear expression

$$K_i(x) = g(z_i) \qquad\qquad g(z) = \tanh(z) \qquad (2)$$
$$z_i = \sum_j C_{ij} x_j + h_i \qquad (3)$$

again all variables at time t. This seems to be overly trivial concerning the set of behaviours which are observed in the experiments. Please note however, that in our case the behaviours are generated essentially also by an interplay of neuronal and synaptic dynamics (Eq. 11) so that our robots are not simple reactive agents.

3.1 World Model and Sensorimotor Dynamics

The robot has a minimum ability of cognition. This is realised by a world model $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ mapping the actions y and previous sensor values x_t, x_{t-1} of the robot on the new sensor values x_{t+1}, i.e.

$$x_{t+1} = F(x_t, x_{t-1}, y_t) + \xi_t \qquad (4)$$

where ξ_t denotes the model error. We make the following ansatz for the world model F,

$$x_{t+1} = Ay_t + S(x_t - x_{t-1}) + b + \xi_t \qquad (5)$$

where A is a $n \times m$ matrix, S is a $n \times n$ matrix, b, ξ are column vectors. This model is in contrast to earlier work enhanced by the S-term. The model is trained by gradient descent on the error $E_F = \xi^\top \xi$ as

$$\Delta A_{t+1} = \varepsilon_M\, \xi_t y_t^\top, \quad \Delta S_{t+1} = \varepsilon_M\, \xi_t (x_t - x_{t+1})^\top, \quad \Delta b_{t+1} = \varepsilon_M\, \xi_t\,. \qquad (6)$$

where ε_M is the learning rate chosen conveniently. Again, the model seems to be oversimplified. However, model learning is very fast so that the model parameters change rapidly in time and different world situations are modelled by relearning. Moreover, the model only has to represent the coarse response of the world to the actions y of the robot. Behaviour is organised such that this reaction is more or less predictable. Hence, the world model is sufficient to provide a qualitative measure of these response properties.

With these notions we may write the dynamics of the sensorimotor loop in closed form, where ψ denotes the internal model of the sensorimotor loop

$$x_{t+1} = \psi(x_t, x_{t-1}) + \xi_t \qquad (7)$$
$$\psi(x_t, x_{t-1}) = AK(x_t) + S(x_t - x_{t-1}) + b \qquad (8)$$

using Eq. 1 $y_t = K(x_t)$.

3.2 Realising Self-organisation

As known from physics, self-organisation results from the compromise between a driving force which amplifies fluctuations and a regulating force which tries to stabilise the system. In our paradigm the destabilisation is achieved by increasing the sensitivity of the sensor response induced by the taken actions. Since the controls (motor values) are based on the current sensor values, increasing the sensitivity in this sense means amplifying small changes in sensor values over time which drives the robot towards a chaotic regime.

The counteracting force is obtained from the requirement that the consequences of the taken actions are still predictable. This should keep the robot in "harmony" with the physics of its body and the environment. It has been shown in earlier work that these two objectives can be combined in the time loop error namely finding the input $\hat{x}_t$ which is mapped by ψ to the true new sensor values x_{t+1}, i.e. $\|x_{t+1} - \psi(\hat{x}_t, x_{t-1})\|$ is minimal. We define:

$$E = v^\top v \tag{9}$$

where $v = \hat{x}_t - x_t$. Using Taylor expansion we get from Eq.7

$$\xi_t = L v_t$$

where ξ_t is the model error as introduced above and $L = \partial\psi/\partial x_t$ is the Jacobi matrix of the sensorimotor dynamics. If L^{-1}exists we can write

$$E = \xi^\top Q^{-1}\xi \tag{10}$$

with the positive semidefinite matrix $Q = LL^\top$.

Using gradient descent the parameter dynamics is

$$\Delta c_t = -\varepsilon\frac{\partial E_t}{\partial c_t} \ , \qquad \Delta h_t = -\varepsilon\frac{\partial E_t}{\partial h_t} \ . \tag{11}$$

More detail and explicit expressions for the parameter dynamics can be found in previous publications [7, 5]. Note that the parameter dynamics Eq. 11 is updated in each time step so that the parameters in practical applications may change on the behavioural time scale if the update rate ε is chosen conveniently. This means that the parameter dynamics is constitutive for the behaviour of the robot.

4 Self-organised Sweeping Through Behaviour Space

Let us consider first the case of the Barrel for the demonstration of the exploratory character of the system. The Barrel is a physical object with strong inertia effects so that it is not possible for instance to drive it with a pattern generator emitting a fixed frequency, where the Barrel will normally execute a rather erratic behaviour. However, if connected to our controller with both the C and A matrix in a "tabula rasa" condition (equal to the unit matrix), the parameter dynamics described above will after a short time excite a rolling mode

with the velocity systematically increasing up to a maximum value, after this the velocity decreases to zero and increases again with inverted sign.

In Fig. 2 one can see a part of the state and parameter dynamics of the system for one such cycle. Note, that the velocity of the robot can be directly read from the oscillations of the sensor value x_1, high frequency corresponding to high velocities. The direction however depends on the phase relation between x_1 and x_2 (not shown). We can analyse the controller matrix C during the course of
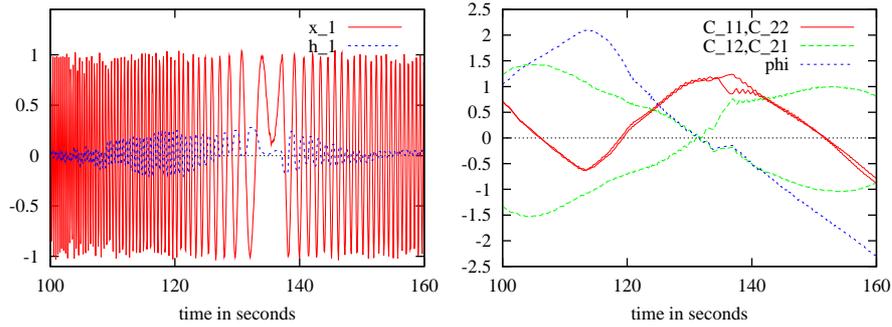

Fig. 2. Dynamics of controller with the Barrel in the time interval 100 to 160 seconds. The region covers the period where the robot actively slows down and then inverts its velocity and then rolls backwards with increasing speed. *Left:* one sensor value $x_1(t)$ and one bias term $h_1(t)$; *Right:* elements of controller matrix C and rotation angle ϕ.

time. It is obvious from the right plot of Fig. 2 that despite the unit initialisation C develops into a matrix with scaled $SO(2)$ structure. That means basically that C is a scaled rotation matrix:

$$C = u \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}$$

In the experiment the controller matrix C runs through the entire range of rotation angles ϕ and hence through the accessible velocities of the robot. The described behaviour sweeping repeats more or less periodically. It is important to note that the sweeping effect is a consequence of the interplay between the state dynamics and the learning dynamics of the threshold values h_i see [17] for details.

5 Guiding Self-organisation

In the previous section we showed that the controller will explore the action space and in particular the frequency space. In the case of the Sphere with three dimensional motor and sensor space we observe also frequency sweeping behaviour, however the situation is more complex since the robot can change

the axis of rotation and so on and so forth. However, in a normal setup, where the Sphere can move freely, it will exhibit different slow and fast rolling modes. Behaviours which are well predictable will persist longer than others, but due to the exploratory character of the controller all modes are transient in nature.

In order to shape the behaviour of the robot, we define a reinforcement signal given $r(t) \in \mathbb{R}$, which can be negative for punishment and positive for reward. In order to incorporate the reinforcement signal we can modify the error function with the following formula.

$$E_r = (1 - \tanh(r(t)))E \qquad (12)$$

where E is the error defined in Eq. 9.

The effect is based on the fact that E is small if both the prediction error ξ is small (smooth, stable behaviour) and the dynamics is instable (due to L in the denominator, see Eq. 10). The latter effect is what makes the system explorative so that emerging behaviours are transient. The life time of a transient depends also on the strength of ξ so that transient behaviours which can be well modelled have a longer life time. The prefactor in the error function (Eq. 12) regulates the life time of transients as well since it essentially multiplies the learning rate of the parameter dynamics. Behaviours with small or even negative reinforcement are left rapidly, whereas large positive reinforcement tends to increase the life times. The life time of behaviours is maximal if they both are rewarded and can be well modelled. In the following sections we will demonstrate two different nominal behaviours, fast motion and spinning.

5.1 Speed Reinforcement

As one possible goal one could want the robot to move fast. In this case the reinforcement signal is:

$$r(t) = \frac{1}{3}\|\boldsymbol{v}_t\| - 1$$

where $\boldsymbol{v}_t$ is the velocity vector of the robot in world coordinates. In order to avoid saturation of the tanh function in Eq. 12 the reward is scaled and shifted. For the average velocity of the normal runs the reward is about zero. For small velocities the reward is negative and causes a stronger change of behaviour, whereas larger velocities give a positive reward and due to small changes in the behaviour the robot stays longer in this regimes.

We conducted 20 experiments with reinforcement and 20 experiments without reinforcement all with random initial conditions, each 60 minutes in simulated real-time on a flat surface without obstacles. The robot also experiences rolling friction, so that fast rolling really requires constant acceleration. In Fig. 3 the mean velocity for each simulation is plotted and the velocity trace of the robot for two reinforced and two normal runs are plotted. One can see, especially at the overall mean, that the mean velocities for the reinforced runs are significantly larger than the ones of the normal runs. However, since straight and also fast rolling modes are easy predictable they are also exhibited in the normal
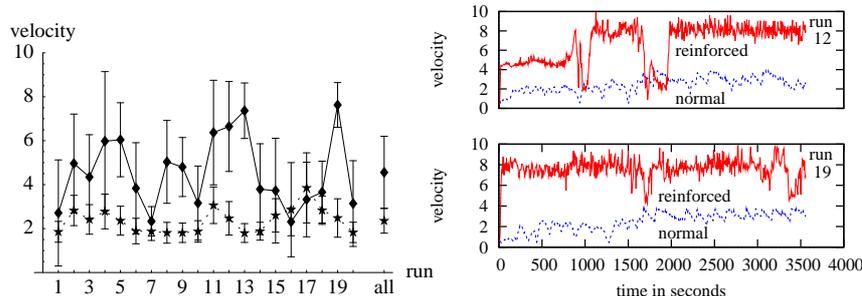
Fig. 3. *Left:* Mean and std. deviation of the velocity of the Sphere for 20 runs each 60 minutes long with (*diamonds/solid line*) and without (*stars/dotted line*) speed reinforcement; *all* denotes the mean and std. deviation over the means of all runs; *Right:* Time course of the velocity during 2 runs, i.e. 4 independent simulations (*upper:* run 12, *lower* run 19).

runs. The traces illustrate that the robot with reinforcement reaches quicker a faster motion behaviour and also stays longer in these behaviours.

5.2 Spin Reinforcement

In a different setup we want the robot to drive curves and spin around the z-axis of the world coordinate system. The reinforcement function looks as follows:

$$r(t) = \frac{1}{3}\|\omega_z\| - 1$$

where ω_z is the angular velocity of the robot around the z-axis (in world coordinates). Again the reward is scaled 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 centre, so that the robot spins fast at the place. The robot also experiences rolling friction, so that fast pirouettes are not persistent. We conducted again 20 experiments with reinforcement, each 60 minutes simulated real-time on a flat surface without obstacles. In Fig. 4 the mean angular velocity ω_z for each simulation is plotted and the angular velocities of the robot in two reinforced and two normal runs are displayed. In this scenario the difference between the normal runs and the reinforced runs are tremendous. Nearly all reinforced runs show a very large mean angular velocity. The reason for this drastic difference is that these spinning modes are less predictable and therefore quickly left in the unreinforced setup. One can see in the traces, 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. Note, that spinning at the place (high peeks) is not persistent because of friction. So the robot tends to gain some speed by rolling along the ground.
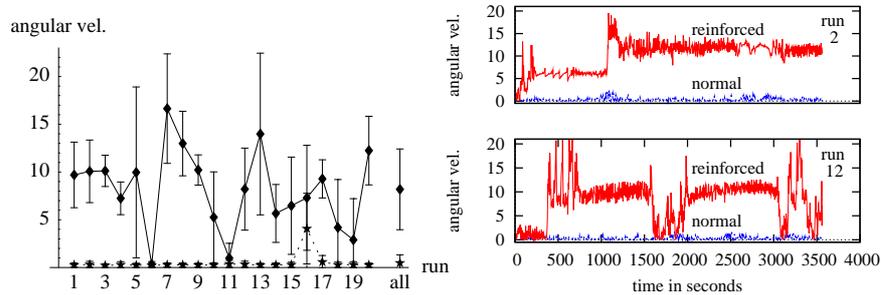
Fig. 4. *Left:* Mean and std. deviation of the angular velocity ω_z of the Sphere for 20 runs each 60 minutes long with (*diamonds/solid line*) and without (*stars/dotted line*) spin reinforcement; *Right:* Time course of the angular velocity during 2 runs, i.e. 4 independent simulations (*upper:* run 2, *lower* run 12).

6 Discussion

We demonstrated in the present paper a simple method by which the otherwise freely self-organised behaviour, generated by the general homeokinesis paradigm, can be guided towards desired behaviours. First we studied an emergent exploratory behaviour in form of a velocity sweep using a two degree of freedom rolling barrel robot. This shows that different behaviours are exhibited in course of time. We integrated a reinforcement signal defined by an external observer into the learning rule of the controller. In essence the original time loop error is multiplied by a strength factor, obtained from the reinforcement signal. The approach is applied to a spherical robot in two scenarios, fast motion reinforcement and spin reinforcement. In both cases the performance was significantly increased and it was shown that the robot was guided towards rewarded behaviours. Nevertheless, the exploratory character of the paradigm stays still intact.

We consider our approach as a contribution to autonomous robot development [18, 19] and see potential applications in this field. With the presented reinforcement mechanism we are now able to guide the development of behaviours. However, in the current setup the internal world model will forget past behaviours, so that there is no long term effect of the reinforcement. This can be achieved with multiple internal models and will be subject of a future paper.

References

1. Neal, M., Timmis, J.: Once More Unto the Breach: Towards Artificial Homeostasis? In Castro, L.N.D., Zuben, F.J.V., eds.: Recent Developments in Biologically Inspired Computing. Idea Group (January 2005) 340–365

2. Paolo, E.D.: Organismically-inspired robotics: Homeostatic adaptation and natural teleology beyond the closed sensorimotor loop. In Murase, K., Asakura, T., eds.: Dynamical Systems Approach to Embodiment and Sociality, Adelaide, Advanced Knowledge International (2003) 19 – 42

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

4. Der, R., Steinmetz, U., Pasemann, F.: Homeokinesis - a new principle to back up evolution with learning. In: Computational Intelligence for Modelling, Control, and Automation. Volume 55 of Concurrent Systems Engineering Series., Amsterdam, IOS Press (1999) 43–47

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

6. Der, R., Herrmann, M., Liebscher, R.: Homeokinetic approach to autonomous learning in mobile robots. In Dillman, R., Schraft, R.D., Wn, H., eds.: Robotik 2002. Number 1679 in VDI-Berichte. (2002) 301–306

7. Der, R., Hesse, F., Martius, G.: Rocking stamper and jumping snake from a dynamical system approach to artificial life. Adaptive Behavior **14**(2) (2006) 105–115

8. Der, R., Martius, G., Hesse, F.: Let it roll – emerging sensorimotor coordination in a spherical robot. In Rocha, L.M., Yaeger, L.S., et al., eds.: Artificial Life X : Proc. X Int. Conf. on the Simulation and Synthesis of Living Systems, International Society for Artificial Life, MIT Press (August 2006) 192–198

9. Der, R., Martius, G., Hesse, F.: Videos of self-organized creatures. http://robot.informatik.uni-leipzig.de/research/videos (2007)

10. Der, R., Martius, G.: From motor babbling to purposive actions: Emerging self-exploration in a dynamical systems approach to early robot development. In Nolfi, S., Baldassarre, G., et al., eds.: From Animals to Animats 9, Proc. SAB 2006. Volume 4095 of Lecture Notes in CS., Springer (2006) 406–421

11. Stout, A., Konidaris, G., Barto, A.: Intrinsically motivated reinforcement learning: A promising framework for developmental robotics. In: The AAAI Spring Symposium on Developmental Robotics. (2005)

12. Oudeyer, P.Y., Kaplan, F., Hafner, V.V., Whyte, A.: The playground experiment: Task-independent development of a curious robot. In Bank, D., Meeden, L., eds.: Proceedings of the AAAI Spring Symposium on Developmental Robotics, Pages 42-47, Stanford, California. (2005)

13. Schmidhuber, J.: Completely self-referential optimal reinforcement learners. In: ICANN (2). (2005) 223–233

14. Popp, J.: Sphericalrobots. http://www.sphericalrobots.com (2004)

15. Smith, R.: Open dynamics engine. http://ode.org/ (2005)

16. Martius, G., Der, R., Hesse, F., Güttler, F.: Leipzig robot simulator. http://robot.informatik.uni-leipzig.de/software (2006)

17. Hamed, N.: Self-Referential Dynamical Systems and Developmental Robotics. PhD thesis, University of Leipzig (2006) In preparation.

18. Lungarella, M., Metta, G., Pfeifer, R., Sandini, G.: Developmental robotics: a survey. Connect. Sci. **15**(4) (2003) 151–190

19. Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., Thelen, E.: Autonomous mental development by robots and animals. Science **291** (2001) 599 – 600