From Motor Babbling to Purposive Actions: Emerging Self-exploration in a Dynamical Systems Approach to Early Robot Development

Ralf Der[1] and Georg Martius[2]

[1] University of Leipzig, Institute of Computer Science,
POB 920, D-04009 Leipzig, Germany
der@informatik.uni-leipzig.de
http://robot.informatik.uni-leizpig.de
[2] Bernstein Center for Computational Neuroscience, University of Goettingen,
Bunsenstr. 10, D-37073 Goettingen, Germany
georg.martius@ds.mpg.de

Abstract. Self-organization and the phenomenon of emergence play an essential role in living systems and form a challenge to artificial life systems. This is not only because systems become more lifelike, but also since self-organization may help in reducing the design efforts in creating complex behavior systems. The present paper studies self-exploration based on a general approach to the self-organization of behavior, which has been developed and tested in various examples in recent years. This is a step towards autonomous early robot development. We consider agents under the close sensorimotor coupling paradigm with a certain cognitive ability realized by an internal forward model. Starting from *tabula rasa* initial conditions we overcome the bootstrapping problem and show emerging self-exploration. Apart from that, we analyze the effect of limited actions, which lead to deprivation of the world model. We show that our paradigm explicitly avoids this by producing purposive actions in a natural way. Examples are given using a simulated simple wheeled robot and a spherical robot driven by shifting internal masses.

1 Introduction

Adaptation and survival in uncertain and ever changing environments are one of the key challenges in natural and artificial beings. The field has seen many impacts from life sciences, one of the directions being epigenetic and developmental robotics [11] trying to mimic natural ontogenesis. Moreover, the role of embodiment has become an important subject in the past decade under (i) the practical aspect of reducing computational efforts for control by exploiting the physical properties of the robot in its environment, see [12], [9], and (ii) the more conceptual aspect that embodied sensorimotor coordination is vital for the self-structuring of the sensor space necessary for categorization and higher level cognition, see [15], [10].

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 406–421, 2006.

The approaches are very diversified and often oriented towards specific goals, leading to nice results up to the humanoid level. Our work aims more towards an approach from first principles. We consider agents under the close sensorimotor coupling paradigm, controlled by a neural network. Moreover, the robot disposes of a certain cognitive ability realized by an internal forward model (world model) predicting future observations on the basis of present observations and controls. Such models are speculated to play an important role for human motor control, cf. [17] as an example.

In the engineering sense the world model is learned by trying random actions. This is also assumed to take place in human development and is called motor babbling. However this approach is infeasible in high dimensional systems. This problem is called the curse of dimensionality in statistical learning theory and was realized to be a serious problem in learning sensorimotor tasks by Bernstein [1] long ago. Moreover, usually it is even not necessary to try all actions, but just those that contribute most to the information gain of the model. Our approach aims at the realization of self-exploration with emerging purposive actions instead of motor babbling.

The concomitant learning of both, the controller and the model, faces among others the cognitive bootstrapping problem. Starting at a "do nothing" and "know nothing" initialization of the controller and the internal model, respectively, the robot does not have any information on the structure and dynamics of its body so that the world model has to learn this from scratch. However, in order to learn effectively, the controls have to be informative or purposive so that the world model is provided with the sensorimotor patterns necessary for its improvement. On the other hand, these actions require a certain knowledge of the reactions of the body – information is acquired best by informed actions. This bootstrapping situation in principle reappears on all stages of the developmental process. We consider here a solution at a level, which is essentially based on the feed-back of proprioceptive sensors, i.e. self-exploration of the physical properties of the body. We understand this as early robot development, i.e. the first step of a self-organized development towards ever increasing behavioral competencies and understanding of the behavior of the body in its environment.

In recent years we have derived a systematic approach to the self-organization of behavior which has proven its practical applicability in a number of examples, see Refs. [4], [8] or the videos on [7]. This has been achieved not only for wheeled robots in a cluttered environment, see the video [2] and others on our video page, but also for high dimensional snake like robots, see the zoo videos on [7]. These creatures have no program (set of rules defining behavior), no aims, and no purpose. Yet they deploy activities by itself which are rooted in their bodies and related to the environment in which they "live". We will discuss in the present paper how this is related to the solution of the bootstrapping problem by emerging self-exploration.

The paper is organized as follows. In Sec. 2 we give a brief introduction to our general control paradigm. We demonstrate on a theoretical basis how purposive actions, necessary for self-exploration, emerge in a natural way in Sec. 3. These theoretical findings are verified in the following section with two examples, a wheeled robot and a spherical robot driven by internally shifting masses.

2 Controller Learning – Between Sensitivity and Predictability

The learning of the controller is based on the papers [3], [5]. We give here only the basic principles of our approach. We start from the information the robot gets by way of its sensor values.

2.1 The Sensorimotor Dynamics

Let us consider a robot which produces in each instant $t = 0, 1, 2, \ldots$ of time the vector of sensor values $x_t \in \mathbb{R}^n$. 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 $y \in \mathbb{R}^m$

$$y = K(x)$$

all variables being at time t. In the example of a two-wheeled robot we have $y_t = (y_{t1}, y_{t2})^\top$, y_{ti} being the target wheel velocity of wheel i. In the cases considered explicitly below, the controller is realized by a one layer neural network defined by the pseudolinear expression (omitting the time index)

$$K_i(x) = g(z_i) \tag{1}$$

where $g(z) = \tanh(z)$ and

$$z_i = \sum_j C_{ij} x_j + h_i \tag{2}$$

This seems to be overly trivial concerning the set of behaviors which are observed in the experiments. Note, however, that in our case the behaviors are generated essentially also by an interplay of neuronal and synaptic dynamics (see Eq. 11 below), which makes the system highly nontrivial.

Our robot is equipped with a world model which is a function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ predicting the current sensor values in terms of the earlier sensor and motor values, i.e.

$$x_t = F(x_{t-1}, y_{t-1}) + \xi_t \tag{3}$$

where ξ is the modeling error. In practical applications, F may be represented by a neural network with parameter vector w, which might be learnt by standard back propagation. The world model realizes the cognitive abilities of the robot. Cognition is understood on a very low level, meaning essentially the ability to predict the future consequences of the actions undertaken by the robot. This is actually what the world model does.

Introducing Eq. 1 into the equation for the world model, we get the dynamical system representing the dynamics of the SM loop as

$$x_t = \psi(x_{t-1}) + \xi_t \tag{4}$$

The dynamics for the parameters of the controller are derived from the following two objectives. We aim on the one hand, at a maximum sensitivity of the effects of the controls to the current sensor values. This induces a self-amplification of changes in the sensor values and thus is the source of activity. On the other hand, we require a maximum predictability of these effects, which are represented by future sensor values. This keeps the behaviour in "harmony" with the physics of the body and the environment.

The first objective is realized by requiring a high sensitivity of the map ψ of the sensorimotor loop towards small changes in its inputs. In more detail, we require that ψ realizes the new vector of sensor values x_t by applying a small shift to the inputs, i.e. we put

$$x_t = \psi\left(x_{t-1} + v_{t-1}\right) \tag{5}$$

or

$$\psi\left(x\right) + \xi = \psi\left(x + v\right) \tag{6}$$

where v is the input shift. This equation has a unique solution if ψ is invertible. If not, convenient approximations must be used. This question has to be solved in order to find a stable algorithm but we are not going into these details in the present paper.

At each time step we can find the value of v and define the error (omitting the time index)

$$E = \left\|v\right\|^2 = v^T v \tag{7}$$

where$\|\ldots\|$ means the Euclidean norm. The quantity $\hat{x}_{t-1} = x_{t-1} + v_{t-1}$ is the vector of previous sensor values as reconstructed from the current ones. We may therefore call E the reconstruction error. Moreover, from the point of view of time step $t-1$ the vector $\hat{x}_{t-1}$ is obtained by going one step forward in time by the true dynamics and then back to time $t-1$ by the inverse world model dynamics given by ψ. This is why we also call E the time loop error.

In order to get a more explicit expression we use Taylor expansion, which in leading order yields

$$\xi = L\left(x\right) v$$

where L is the Jacobian matrix defined as

$$L_{ij}\left(x\right) = \frac{\partial}{\partial x_j}\psi_i\left(x\right)$$

which is a direct measure of the stability of the dynamical system, see below for a discussion. If L exists we immediately find

$$v = L^{-1}\xi$$

so that

$$E = \left\|L^{-1}\xi\right\|^2 = \xi^T\left(LL^T\right)^{-1}\xi \tag{8}$$

which is the error function used in the algorithm for adapting C, see Eq. 9 below.

The Gradient Flow of the Parameters. The adaptation of the parameters of the controller can be realized by gradient descending the error function E as usual

$$\Delta C = -\varepsilon \frac{\partial}{\partial C} E \tag{9}$$

The resutling dynamics of the parameters can, at a formal level, be argued to produce the desired properties of the system. In fact, since LL^T is symmetric we may decompose it as

$$LL^T = \sum_{i=1}^{n} \lambda_i P_i$$

where $P_i = n_i n_i^T$ is the projector on the eigenvector n_i with λ_i the corresponding eigenvalue. Then

$$E = \sum_i \lambda_i^{-2} \xi_i^2$$

with $\xi_i = n_i^T \xi$ being the projection of the model error into the subspace spanned by n_i, both λ_i and P_i depending on the parameters C of the controller in an intricate way. This expression is only valid if the $n \times n$ matrix $Q = LL^T$ is of full rank, so that none of the λ_i is equal to zero. However, we also note that, if we start with an L of full rank, the parameter dynamics will drive Q away from impending singularities due to the divergence of E for any $\lambda_i \to 0$.

In more detail, writing the gradient rule as

$$\varepsilon^{-1} \Delta C = \sum_{i=1}^{n} \left(\frac{\xi_i^2}{\lambda_i} \frac{\partial \lambda_i}{\partial C} - \xi_i \frac{\partial \xi_i}{\partial C} \right) \lambda_i^{-2} \tag{10}$$

we see that the gradient flow is driven by two objectives. The first term on the right hand side obviously tends to increase each of the eigenvalues λ_i and hence the instability in the corresponding subspace. The interesting point is in the prefactors ξ_i^2/λ_i which mean that the update is strong where λ_i is small (high stability) and/or ξ_i^2 is large (high modeling error component in this subspace). This can be interpreted as the tendency of the parameter dynamics to produce in all directions the same degree of instability with subspaces of higher modeling error being destabilized even more strongly. Destability corresponds to a higher rate of noise amplification, such that one may say that those subspaces are explored more intensively, which are less well represented by the model. This is the effect which is relevant for the present paper and will be discussed by way of example in Sec. 3.2 below.

The second term in Eq. 10, the strength of which is modulated by ξ_i, essentially counteracts the overshooting destabilization of large error subspaces caused by the first term. It is to be noted, that the error components ξ_i not only depend on the quality of the model, but in an essential way on the behavior of the robot. Hence, both the ξ_i and eigenvalues λ_i change with changing parameters.

Altogether we may say, that our parameter dynamics generates an explorative behavior of the robot (by the first term), which however is related to the environmental reactions by the second term. This has been demonstrated in many applications realized in recent years, see for instance [5] and our video page.

Expicit Learning Rule. In the present paper we consider the one-layer neural network controller given by Eq. 1 so that the Jacobian is

$$L_{ij} = \sum_k A_{ik} g' (z_k) C_{kj}$$

where

$$A_{ik} = \frac{\partial}{\partial x_k} F_i (x, y)$$

and F is learnt concomitantly with the controller by supervised learning on the basis of the new sensor values. We use $g(z) = \tanh(z)$ where $g'' = -2gg'$ so that we get the explicit expressions (omitting the time indices everywhere)

$$\begin{aligned} \varepsilon^{-1} \Delta C_{ij} &= \zeta_i v_j - 2\zeta_i \rho_i y_i x_j \\ \varepsilon^{-1} \Delta h_i &= -2\zeta_i \rho_i y_i \end{aligned} \tag{11}$$

where $v = L^{-1}\xi$, $\mu = A^\top Q^{-1}\xi$, $\zeta_i = g_i' \mu_i$, and $\rho = Cv$. The inversion of the matrix $Q = LL^\top$ is done by standard techniques, and has proven in many applications to be feasible and not time critical with up to 20 independent degrees of freedom.

Note that the parameter ε is chosen such that the parameters change at about the same time scale as the behavior. The interplay between synaptic and state dynamics of the controller induces a high dynamical complexity of the sensorimotor loop. The resulting robot behaviors are of a much larger complexity than the pseudolinear expression with fixed parameters might ever realize.

3 Model Learning – Problems and Challenges

Internal models are one of the prerequisites for a robot to become a cognitive system. In the case of human motor systems the role of internal models has in particular been emphasized by the work of Wolpert [18], [16]. In the present paper we are concerned with forward models as given by Eq. 3 which are learnt in a supervised way on a training set of sensorimotor patterns (x_{t+1}, y_t). However, in order to learn the relevant information about the world, the training instances must be guaranteed to sufficiently sample not only the sensor space, but also the action space. In practice it is complicated to ensure this sampling property. In case of on-line learning there is always only a part of the state action space covered in a restricted interval of time. This fact actually is widely recognized but we will demonstrate it in an extremely simple situations in order to work out explicitly the bootstrapping problem involved.

3.1 The Deprivation Effect

Let us consider a very simple example of a sensorimotor loop given by a robot with two wheels. The only sensor values are given by the current velocities which can be measured by a wheel counter, i.e. we have only proprioceptive sensors in this case. Assuming that the reactions of the wheels are largely independent of each other, we expect the model to be given as $x_{t+1} = Ay_t + \xi$ where $x \in \mathbb{R}^2$ and $y \in \mathbb{R}^2$ are the measured and target wheel velocities, respectively, and ξ is the modeling error. In the case given $A = \alpha\mathbf{E}$ is essentially the unit matrix. This is what one expects to be learnt by for instance gradient descending the error $E = \xi^\top \xi$ with learning rule

$$\Delta A_t = \varepsilon_A \xi_t y_{t-1}^\top - \beta A_t \tag{12}$$

where $\xi_t \in R^n$, $y_{t-1} \in R^m$, and $\left(\xi y^\top\right)_{ij} = \xi_i y_j$. The small damping term $-\beta A$ has to be introduced in order to damp away the influence of the initial conditions. The scaling factor α is a hardware constant.

However, convergence to the correct solution $A = \alpha\mathbf{E}$ is guaranteed only if the training instances (x_t, y_{t-1}) cover the full state-action space. Now let us assume that the behavior is restricted to a certain subspace of the action space. Under our closed loop control paradigm, behavior is parameterized by the matrix C of the controller. A restricted behavior is produced by assuming the C matrix of the controller as

$$C = \gamma p p^\top \tag{13}$$

where p is a normalized vector, $pp^\top$ is the projector onto p, and γ a constant with $\gamma > 0$ and $\gamma\alpha > 1$. The sensorimotor dynamics[1]

$$x_t = Ag\left(Cx_{t-1} + h\right) + \xi_t$$

converges towards a fixed point. The controller will produce the vector $y = g\left(sp + h\right)$ defining the wheel velocities, where s is obtained from the solution of the fixed point equation. In particular if ($h = 0$ for the moment)

$$p = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{14}$$

the robot will move either straight forward or backward if $s > 0$ or $s < 0$, respectively. Choosing instead $p = (1, \ -1)^T / \sqrt{2}$ the robot will rotate on site. The behavior can still be further modified by changing h.

The point now is, that instead of converging towards the unit matrix, A is learnt as

$$A = \alpha p p^\top \tag{15}$$

so that A is essentially the projector on the subspace given by the degenerate controller. This is a correct solution in the space covered by y_t which of course is completely wrong in the complementary subspace of the motions of the robot.

[1] Consider $g(z)$ as a vector function, i.e. $g_i(z) = g(z_i)$.

This effect makes the learning unstable if the controller changes the motor vector on a slow time scale since the matrix A will follow this change. Of course this result hinges on the time scales. In fact the problem will not arise so strongly if the time scales for the learning are much larger than the intervals of persistent directions of the robot.

3.2 The Bootstrapping Scenario

As explained in the introduction, the aim of our approach is the concomitant learning of the controller and the model from scratch. The toy example given above has demonstrated that, if the actions $y \in R^m$ (with $m = 2$ in the example above) are restricted to a certain subspace, the model will degenerate to a projector onto that subspace with the effect that it will be completely wrong in the orthogonal subspace. The challenge is that the controller needs to "feel" this deprivation of the world model and to issue motor commands which provide the world model with the state-action pairs (x_t, y_{t-1}) necessary for learning in the orthogonal subspace neglected so far.

This is exactly what happens in our approach for the learning of the controller. We will now demonstrate this theoretically in terms of the above model with degenerate $C = \gamma pp^\top$. With A from Eq. 15 we get in the linear (low z) case

$$L = \gamma\alpha pp^\top$$

so that the Jacobian matrix is singular, hence E has a singularity, and the degenerate C is seen to be an instable fixed point of the gradient dynamics. Without loss of generality we may use the specific form Eq. 14 for p. Now let us assume that C has a small deviation δC which corresponds to the projector into the orthogonal subspace, i.e. we put

$$C = \gamma pp^T + \mu p_\perp p_\perp^T$$

where $p_\perp$ is orthogonal to p, i.e. $p_\perp p_\perp^T$ is the projector onto the orthognal complement of p, and μ is arbitrarily small. With a noisy input or with a random motor event (motor babbling) the action may be

$$y = sp + \sigma p_\perp \tag{16}$$

where $|\sigma|$ is small. We are now going to show now that this small fluctuation leads to a strong amplification of the $p_\perp p_\perp^T$ component in C.

If the robot is executing this action, we get a model error (we assume the p subspace is already learnt correctly and neglect other noisy events)

$$\xi = \alpha\sigma p_\perp \tag{17}$$

since A is still the degenerate matrix $A = \alpha pp^T$. The learning step for A produces

$$\Delta A = \varepsilon\xi y^T = \varepsilon\alpha\sigma s \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} + \varepsilon\alpha\sigma^2 pp_\perp^T$$

(we drop the damping term because it only contributes to the degenerate part of A). In the small z case considered we have now $L = (A + \Delta A)\, C$. In leading order (small μ and $\varepsilon\sigma$) we find with simple matrix algebra that

$$\left(LL^T\right)^{-1} = \frac{1}{4\alpha^2\varepsilon^2\sigma^4\mu^2} p_\perp p_\perp^T$$

which is the projector on just that subspace which is not well covered by the model so far, i.e. in this order the learning so to say concentrates fully onto the subspace not "understood" by the model. Using Eq. 17 the error is obtained as

$$E = \frac{1}{2\sigma^2\varepsilon^2\mu^2} \tag{18}$$

This shows that the learning will rapidly increase the strength μ of that part of C which projects into the orthogonal subspace. However, in this way also the contribution of the orthogonal actions is increasing so that the constant μ in Eq. 3.2 is increasing as long as the model is still wrong. We may interprete this by saying that the controller tries more and more actions which force the model to learn also the behavior in the orthogonal subspace. This is a kind of purposive behavior, the purpose being to feed the model with the necessary input-output pairs for complete learning. The process has to be started by some fluctuation in the output of the controller which may be called motor babbling.

The difference of this behavior to the usual strategy of issuing random motor commands consists in the fact that the novel motor commands are directed into the unknown regions of the state-action space. This of course is of relevance for high dimensional systems where random commands face the curse of dimensionality. This has been clearly demonstrated in our experiments with high dimensional (up to 20 independent motors) systems, see our videos of the snake robots, where collective modes are excited by this bootstrapping phenomenon. The background behind the high dimensional scenario is that the paradigm ensues spontaneous symmetry breaking and creating low dimensional searching modes in high dimensional search spaces, which will be demonstrated in a later paper.

4 Experiments

In the sections before we have studied deprivation of the world model and we have seen how purposive actions can efficiently eliminate this effect. In order to illustrate this in practice, we will consider different experiments. First, we consider the rather artificial setting as described in section 3.1 theoretically with a simulated two-wheeled robot. Second, the self-explorative character of you controlling paradigm is analyed using the same robot. Third, a simulated spherical robot is considered on a flat surface to show self-explorative behavior at a more complex system. Finally the spherical robot is considered in a basin like environment, where deprivation occurs naturally.

4.1 Experiment I – Two-Wheeled Robot: Deprivation and Bootstrapping

The idea of this experiment is to show, that first in the case of limited motor commands deprivation of the world model occurs, and second that our controller effectively produces purposive actions. We use the physics engine ODE (open dynamic engine [14]) for the computer simulation experiments. A simulated two-wheeled robot is controlled with motor commands within a subspace of the action space. Motor commands y are understood as wheel velocities and the sensors values x are the read back wheel velocities obtained from the wheel counters. Please recall the controller function $K(x) = y = \tanh(Cx + h)$ (Eq. 2). As decribed in section 3.1 the controller matrix is degenerated as $C_{ij} = 0.6 + \lambda u_{ij}$, where u_{ij} are random numbers and $0 < \lambda \ll 1$. We modulate h such that the robot drives backward and forward periodically.

As expected, we observe a degeneration of the world model, see Figure 1. After the model learning is basically converged, the learning of the controller according to Eq. 11 was switched on (at time 4550). After a short break down of the activity one observes the emergence of motor commands which live mainly in the orthogonal subspace. This means rotational behavior of the robot. Later on both, straight and rotational modes, are equally visited so that the model gets the necessary information. A is converging towards the unit matrix as it should be. The behavior and the parameter dynamics are displayed in Figure 1.

4.2 Experiment II – Two-Wheeled Robot: Frequency Wandering

Besides the effects discussed so far there is more to the self-exploration properties of our approach. In particular the fact that the error $E = v^{\top}v$ is invariant to rotations of v introduces a certain invariance of the state dynamics against frequency changes (in a linear approximation). This leads to the effect that the robot self-regulates the frequencies of its motor values.

In the experiments we use the simulated two-wheeled robot as in the previous section. Most of the time the robot moves by sequences of straight and rotational motion primitives. This corresponds to the exploration of the physical space and is what one would call an explorative behavior in the usual sense. However, occasionally the controller gradually increases the frequency of the dynamics in the sensorimotor loop so that rather complex trajectories emerge in the physical space. With even higher frequencies we observe a jiggling of the body where physical effects due to inertia, swing, and even gyro effects come into play. We may say that this is the phase of the self-exploration of these physical properties of the body. However, the simple world model does not understand the high frequency modes very well, so that they are left after some time. The robot returns to its "normal" behavior with a succession of rotational and straightforward driving modes. This play repeats more or less forever with a strong influence of the noise. In Figure 2 the short-time fourier transform of the motor values are displayed, which reflect the frequency in the sensorimotor dynamics.

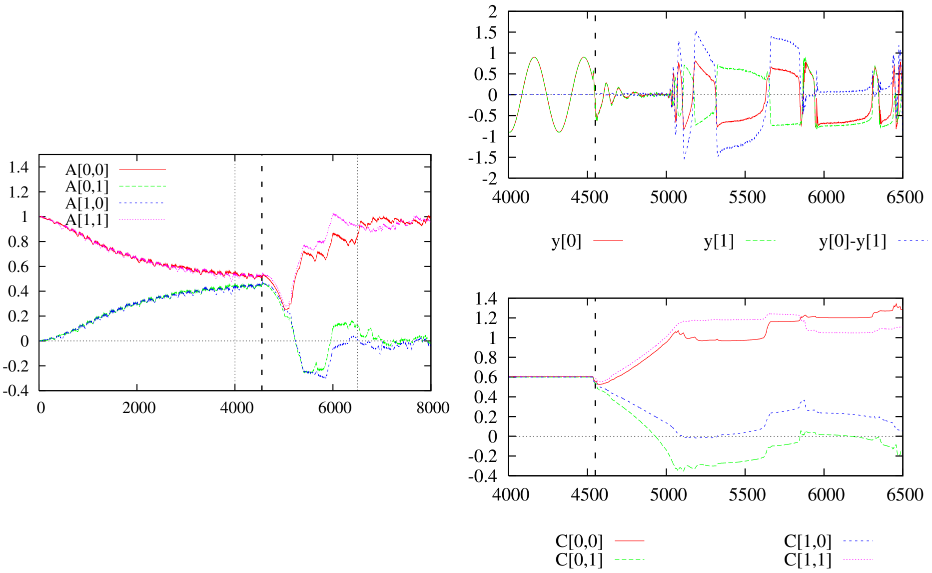

Fig. 1. Model deprivation and recovery in case of a two-wheeled robot. From time 0 – 4550 the robot was controlled by fixed C and modulated h (oscillating forward/backward), and after time 4550 the learning of the controller is enabled. The time scale is 1/50 sec, i.e. the whole run is 160 sec long. Left: Model matrix A. Degenerates until controller learning is enabled. After that it learns towards a unit matrix. Upper right: Motor commands y_i and steering $y_0 - y_1$. One can clearly see that the controller performs rotational actions in a dedicated manner after the activation of the learning (> 5000). Later on the motion consists of straight and rotational modes leading to the full deployment of the world model. Lower right: Controller matrix C.

This scenario actually reminds one of the fact that the controller with its learning dynamics does not know about the physical space so that everything it does is the exploration of the properties of the body and the exploration of the space is only in the eye of the beholder. A relation to the space would emerge if we include sensors informing about positions in space. The emergence of the concept of space will be the subject of a later paper.

4.3 Experiment III – Spherical Robot: Emerging Self-exploration on Flat Surface

The wheeled robot is a rather simple example of a sensorimotor loop. In order to show the emergence of sensorimotor coordination by self-exploration we demonstrate the above phenomena with a more complicated robotic object. The object of study is a simulated spherical robot see Figure 3, inspired by Julius Popp [13].

The motor commands y are the nominal positions of the masses along the axes. The sensor values are in this case the components of the vector of the z-axis of the robot in the world coordinate system. In this way, the controller has only very restricted information about the physical state of the sphere. Nevertheless, our

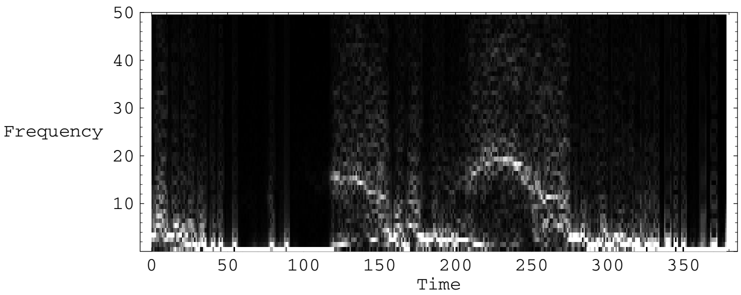

Fig. 2. Power spectra of the motor values (speeds) over time. Each column is the lower frequency part of the discrete Fourier transform of a 2 seconds time window of the motor values (wheel velocities). Subsequent lines are overlapping, so that the time scale is in units of 0.5 seconds. Dark pixels correspond to low energy and bright to high energy in the corresponding frequency band. High energy in a certain frequency band means that changes between forward and backward driving occur at about that frequency. Note that at times 120 and 200 a jump in the frequency occurs meaning that the robot suddenly changes to a highly complex motion pattern which then gradually decays towards the mentioned low frequency regime.

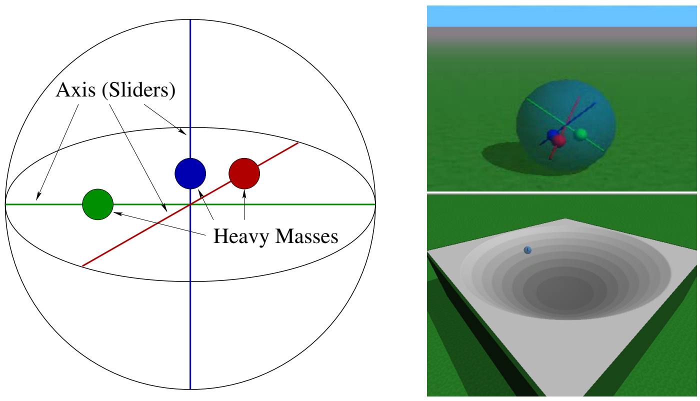

Fig. 3. Simulated spherical robot used in the experiment. Left: Sketch of a spherical robot. Inside the robot there are three orthogonal axes equipped with sliders. To each slider a heavy mass is attached which can be shifted along the axis. There is no collision or interaction of the masses at the intersection point of the axes. Upper right: Picture of a spherical robot on the ground. Lower right: Picture of a spherical robot in a basin.

learning algorithm manages to produce highly coordinated sensorimotor patterns corresponding to different rolling modes in the course of time.

Let us consider the case of the sphere on a flat surface, see the video [6]. In the beginning the controller and world model is initialized in the "do nothing" and "know nothing" situation (C and A are small random matrices). The parameter dynamics given by Eq. 11 drives C until noise amplification sets in and the inner masses start moving so that the world model also starts learning. After some time the sphere starts to roll slowly. Different movements are probed and later a nice and constant rolling mode emerges. Such periods stay for quite a long period of time. What happens in that time is that the world model gets restricted information and the deprivation sets in which leads to the sudden appearance of new controller actions which lead to a kind of explorative periods followed by a new stable rolling period about a different axis.

Besides of the rolling mode we also observe a kind of jumping mode and rolling modes with all three axes involved. These behaviors demonstrate the self-exploration of the body and show how our algorithm manages to close the sensorimotor loop in order to excite stable behavioral modes adequate to the physical properties of the body.

This effect is also demonstrated with a different sensor set. In this experiment we equipped the spherical robot with six infra-red sensors, which are installed in each point of intersection of the axes with the surface of the sphere with the direction along the axis and range of about two diameters of the sphere. The six sensors values are fed directly to the controller. No other sensors are in use. The sensor characteristic was chosen nonlinear as $x = s^{\alpha}$, where s is the primary sensor value (distance) and $\alpha = 1.5$. The effect is that the sensor characteristic is a smoother function of the angular position of the robot. Still the sensor information is extremely unreliable and related to the position of the sphere in a very complicated way. Nevertheless, starting with the “do nothing” and “know nothing” initialization, we observe many different rolling modes which are visited in the course of time. In Figure 4 the power spectra of the sensor values over time are displayed. High frequency means here high velocity. We observed different behavioral modes. For example rolling with different velocities around one of the slider axis or also the tumbling mode involving all three axis.

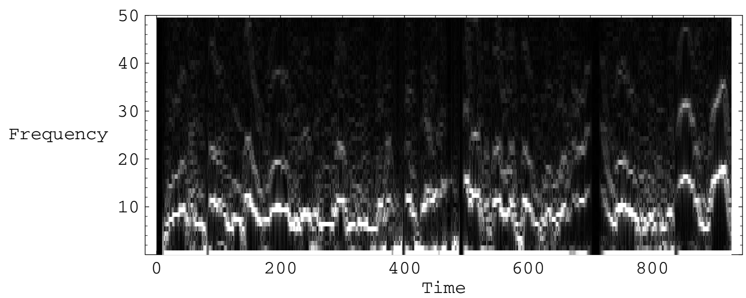

Fig. 4. Power spectra of the infra-red sensor values of the spherical robot on flat ground over time based on 10 s time windows. The bright pixels indicate that there is a dominating frequency of the sensor values which means that the robot is in a rolling mode, the rolling velocity being roughly proportional to the frequency. Periods of stable rolling modes of different velocities are seen to sometimes change rapidly into a resting mode (frequency zero) or to other velocities.

4.4 Experiment IV – Spherical Robot: Deprivation in a Basin

An interesting effect is produced if we put the sphere into a circular basin. We use as sensors the projections of the z-axis of the body coordinate system on the z-axis of the world coordinates. The motor commands are the nominal position of the mass points on the inner axis. In Fig. 5 the behavior of the robot in a basin is shown using the power spectra of the sensor values and the determinant of the world model. The initial phase is of the same nature as on the flat surface, i.e. from time 0 – 80 one can see self-explorational modes, where different frequencies are probed. Then a stable rotational mode emerged (time 80 – 120), which is the circulation in the basin at a constant height. The circulation mode is manifest in the power spectrum by the low frequency excitation, the high frequency excitations being the motions of the axes of the robot due to the rolling motion.

The circulation mode is a behavior which is not so easily realized with the internally shifting masses. Contrary to the rolling on the flat surface the circulation in the basin permanently changes the direction of the axes and hence of the sensor vector. Nevertheless, the controller finds a strategy, such that the circulation mode is stable over many laps. Interestingly this stable sensorimotor pattern is realized by a trajectory which directly reflects the specific geometry of the world. In a certain sense one might say that the robot by its behavior recognizes this geometry.

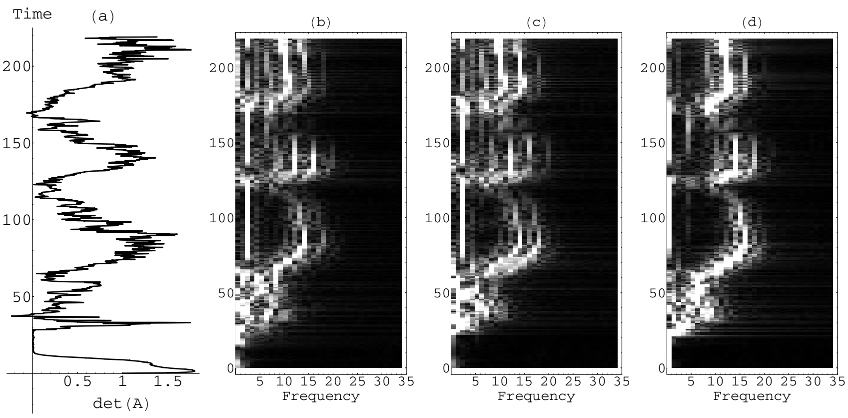

Fig. 5. Behavior of a spherical robot in a basin. (a): Determinant of the model matrix A, (b – d): Power spectra of the sensor values (x, y, z) over time. In the time intervals 80 – 110 and 140 – 170 there are components of stable low frequency in sensor 1 (x) and 2 (y), which correspond to the circulation in the basin at constant height. The higher frequencies reflect the rolling of the sphere as in the flat surface case. One can see that the value of the determinant of A decreases while the robot stays in one mode of behavior (80 – 110, 140 – 170). This is an indication for the deprivation of the model arising from the restriction to a specific mode of behavior. Once the deprivation reaches a certain measure the bootstrapping of new actions sets in which leads to the recovery of the model (increasing determinant).

The behavior in the basin also shows that staying in a stable mode for a longer time leads in general to a deprivation of the world model. This is seen from the time plot of $\det A$, which is taken as a crude measure for the degeneracy. In Fig. 5 the deprivation can be seen at the behavior of $\det A$. It is seen to be indeed decreasing until it reaches nearly 0 at time 115, which means A is closed to a singularity. An explorational period follows, which effectively explores the orthogonal subspace, and the determinant of A is seen to increase rapidly. The same starts again at time 140 and so forth.

5 Conclusions and Outlook

The results presented in the present paper can be considered as a step towards autonomous early robot development, meaning the scenario where an unbiased robot might learn the essential sensorimotor coordination by self-exploration. The important point of our approach is, that it is completely domain invariant, so that the emerging behaviors are dictated by the physical properties of the body and the environment. This has a direct bearing for embodied AI in the sense, that our controller learns to excite certain physical modes of the body, which are qualified by the fact that they can be understood by the world model in easy terms. Hence, we may understand these modes as behavioral primitives which may be used in more complex behavioral architectures.

We have given a theoretical approach to the deprivation problem which arises in the interplay between the world model and the controller. The system does not have any information on the structure and dynamics of the body, so that the world model has to learn this from scratch. This involves the so called bootstrapping problem, meaning that on the one hand the controls have to be such, that the world model is provided with the necessary information. On the other hand, these actions require a certain knowledge of the reactions of the body – information is acquired best by informed actions. The concerted manner by which both the controller and the world model evolve during the emergence of the behavioral modes seems to be a good example of this process.

We consider our approach as a novel contribution to the self-organization of complex robotic systems. At the present step of our development the behaviors, although related to the specific bodies and environments, are without goal. As a next step we will realize a so called behavior based reinforcement learning. When watching the behaving system one often observes behavioral sequences which might be helpful in reaching a specific goal. The idea is to endorse these with reinforcements in order to incrementally shape the system into a goal oriented behavior.

Acknowledgement

This study was supported by a grant from the german BMBF in the framework of the Bernstein Center for Computational Neuroscience Goettingen, grant number 01GQ0432.

References

1. N. A. Bernstein. *The Co-Ordination and Regulation of Movements.* Pergamon Press, 1967.
2. Ralf Der. Video of robot in cluttered environment. http://robot.informatik.uni-leipzig.de/Videos/Pioneer/2004/maze2.wmv, 2004.
3. Ralf Der, Frank Hesse, and René Liebscher. Contingent robot behavior generated by self-referential dynamical systems. *Autonomous robots*, 2005. submitted.
4. Ralf Der, Frank Hesse, and Georg Martius. Learning to feel the physics of a body. In *Proceedings CIMCA'2005*, 2005.
5. Ralf Der, Frank Hesse, and Georg Martius. Rocking stumper and jumping snake from a dynamical system approach to artificial life. *J. Adaptive Behavior, submitted*, 2005.
6. Ralf Der, Frank Hesse, and Georg Martius. Video of spherical robot in free space. http://robot.informatik.unileipzig.de/Videos/SphericalRobot/2005/spherical_floor_trace.mpg, 2005.
7. Ralf Der, Frank Hesse, and Georg Martius. Videos of self-organized creatures. http://robot.informatik.uni-leipzig.de/Videos, 2005.
8. Ralf Der, Georg Martius, and Frank Hesse. Let it roll – emerging sensorimotor coordination in a spherical robot. In *Proceedings AlifeX*, 2006.
9. F. Iida, G. J. Gomez, and R. Pfeifer. Exploiting body dynamics for controlling a running quadruped robot. In *International Conference of Advanced Robotics (ICAR 05)*, 2005.
10. Yasuo Kuniyoshi, Yasuaki Yorozu, Yoshiyuki Ohmura, Koji Terada, Takuya Otani, Akihiko Nagakubo, and Tomoyuki Yamamoto. From humanoid embodiment to theory of mind. In *Embodied Artificial Intelligence*, pages 202–218, Berlin, Heidelberg, New York, 2003. Springer.
11. Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 0(0):1–40, 2004.
12. Rolf Pfeifer and Fumiya Iida. Embodied artificial intelligence: Trends and challenges. In *Embodied Artificial Intelligence*, pages 1–26, 2003.
13. Julius Popp. Sphericalrobots. http://www.sphericalrobots.com, 2004.
14. Russel Smith. Open dynamics engine. http://ode.org/, 2005.
15. Rene te Boekhorst, Max Lungarella, and Rolf Pfeifer. Dimensionality reduction through sensory-motor coordination. In *Kaynak, Okyay (ed.) et al., Artificial neural networks and neural information processing – ICANN/ICONIP 2003. Joint international conference ICANN/ICONIP 2003, Istanbul, Turkey, 26-29, 2003. Proceedings. Berlin: Springer. Lect. Notes Comput. Sci. 2714, 496-503* .
16. D. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1999.
17. D. M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9), 1998.
18. Daniel M. Wolpert, R. Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2:338 – 347, 1998.