Rocking stumper and jumping snakes from a dynamical systems approach to artificial life

Ralf Der, Frank Hesse and Georg Martius

University of Leipzig, PF. 920 D-04009 Leipzig, Germany
{der|fhesse|martius}@informatik.uni-leipzig.de

Abstract. Dynamical systems offer intriguing possibilities as a substrate for the generation of behaviour due to their rich behavioural complexity. However this complexity together with the largely covert relation between the parameters and the behaviour of the agent is also the main hindrance in the goal oriented design of a behaviour system. The paper presents a general approach to the self-regulation of dynamical systems so that the design problem is circumvented. We consider the controller (a neural network) as the mediator for changes in the sensor values over time and define a dynamics for the parameters of the controller by maximising the dynamical complexity of the sensorimotor loop under the condition that the consequences of the actions taken are still predictable. This very general principle is given a concrete mathematical formulation and implemented in an extremely robust and versatile algorithm for the parameter dynamics of the controller. We consider two different applications, a mechanical device called the rocking stumper and the ODE simulations of a "snake" with five degrees of freedom. From this and many other applications we conclude that our self-regulating parameter dynamics engenders artificial life forms of unknown so far dynamical complexity.

1 Introduction

Dynamical systems form a powerful tool for both the analysis and the realization of the behaviour of autonomous robots. The increased interest in using dynamical system theory for the analysis of the robot in its environment may be dated back to the seminal paper by Randall Beer [1] in 1995. About at the same time the book [12] by Port and van Gelder initiated a broad interest in the role of dynamical systems for understanding life and cognition. There are numerous applications of this approach so far. In particular, the dynamical system theory has been used to understand the functionality of evolved networks for robot control [10], see for instance [9].

Apart from providing analytical tools, dynamical systems offer intriguing possibilities as a substrate for the generation of behaviour. Let us consider a robot which is controlled by a neural network, say, transforming sensor values into motor commands. When using a recurrent network this transformation can be rather complex and reaches far beyond a simple reactive paradigm. This has been considered by several authors under varying contexts and with varying success. An elaborate behaviour based design system has been developed in the context of dual dynamics. The system has a layered structure of behavioural subsystems realized by ordinary differential equations, each layer having its own time constant. Communication between the subsystems is realized by specific interaction and "bifurcation-inducing" mechanisms which have to be designed by hand, cf. [2]. However applications so far are scarce. Dynamical systems can also be helpful in solving decision problems. For instance in [5] the authors show how successful route selection through a cluttered environment can emerge from on-line steering dynamics, without explicit path planning. Of particular interest is the dynamical system paradigm for walking machines where neural oscillators are used to generate the different gaits, see for instance [6], [14], [8] and [11].

The authors quoted have mainly tried to design dynamical systems such that they realize prescribed tasks, the smooth navigation through a cluttered environment being a prominent example. The main problem with this approach, however, is in the design of the dynamical systems in view of the largely covert relation between parameters and behaviour of the robot.

The main objective of our work is in fostering the self-organisation of such systems under a true emergentist paradigm. Central is the hope to find a mechanisms of self-regulation for the parameters so that in the rich reservoir of possible behaviours a working regime is stabilised which ensures the viability of the agent. Under this paradigm the aim is not the realization of a specific task given from outside but the emergence of organised motions.

Taking emergence at its roots means in our case to formulate the objective for the robot on a very general not domain related level. In the present paper we develop a dynamics for the parameters of the controller which is essentially driven by the requirement that the dynamical complexity of the sensorimotor loop is to increase moderated by the requirement that the consequences of the actions taken are still predictable. It is the message of the present paper that this very general statement can be given a concrete mathematical formulation and that the emerging behaviours are of an unknown so far complexity.

2 Principles of self-regulation

Based on the paper [4] we give here the basic principles of our approach. Basic to our approach is the dynamics of the sensor values. Let us consider a robot which produces in each instant $t = 0, 1, 2, \dots$ of time the vector of sensor values $x_t \in R^n$. By way of example we may consider a wheel driven robot where

$$x = (v_l, v_r, IR_1, \dots, IR_{n-2})^\top \quad (1)$$

with v_l and v_r are the wheel velocities of the left and right wheel, respectively, as measured by the wheel counters, IR_i is the value of the infrared sensor i with $0 \le IR_i \le 1$. We use closed loop control, i.e. the controller is given by a function $K : R^n \to R^m$ mapping sensor values $x \in R^n$ to motor values $y \in R^m$

$$y = K(x)$$

all variables being at time t. In the example we have $y = (y_1, y_2)^\top$, y_i being the control (target velocity) of wheel i. The controller may or may not depend on internal states realizing a proactive or a purely reactive behaviour, respectively.

Our controller is to be adaptive, i.e. it depends on a set of parameters $C \in R^P$. In the cases considered explicitly below the controller is given by the pseudolinear expression

$$K_i(x) = g(z_i) \tag{2}$$

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

$$z_i = \sum_j C_{ij} x_j + H_i \tag{3}$$

This seems to be overly trivial concerning the set of behaviours which are to be realized. Note however that in our case the behaviours are generated essentially also by an interplay of neuronal and synaptic dynamics which makes the system highly nontrivial.

2.1 World model and sensorimotor dynamics

We assume that our robot has a minimum ability for cognition. This is realized by a world model $F : R^n \times R^m \to R^n$ mapping the actions y and old sensor values x to the new sensor values, i.e.

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

where ξ_t is the model error. The model F can be learned by the robot using any learning algorithm of supervised learning. Let the model be a parameterised function (neural net) with parameters $a \in R^M$. The parameters a can be adapted by gradient descending the error function based on ξ. The structure of the model and the learning procedure define the passive cognitive abilities of the robot.

With these notions we may write the dynamics of the sensorimotor loop in the closed form

$$x_{t+1} = \psi(x_t) + \xi_t \tag{5}$$

where $\psi(x) = F(x, K(x))$. The function ψ can be visualised as a time series predictor for the time series of the sensor values x_t with the controller being known.

In the case considered below we have $x, y \in R^n$ and we assume that the response of the sensor is linearly related to the motor command, i.e. we write (dropping the time index at the matrix A here and in the following)

$$x_{t+1} = Ay_t + \xi_t \tag{6}$$

where A is a matrix and ξ the modelling error $\xi = x - x^{pred.}$ with

$$x_{t+1}^{pred.} = Ay_t \tag{7}$$

denoting the model values of the new inputs. The model can be learned by the delta rule as

$$\Delta A = \varepsilon_M \, \xi_t y^\top t \tag{8}$$

Then

$$\psi(x) = AK(x)$$

and the sensorimotor loop is

$$x_{t+1} = AK(x_t) + \xi_t \tag{9}$$

Again, this model seems to be oversimplified. However model learning will be seen to be very fast so that different world situations are modelled by relearning.

2.2 The paradigm of controlled sensitivity

As discussed in more detail in [4] the behaviour is defined by formulating a parameter dynamics for the controller so that a self-regulating system is obtained. The parameter dynamics is essentially driven by the requirements, that the dynamical complexity of the sensorimotor loop is to increase, and that the consequences of the actions taken are still predictable. The dynamical complexity is directly related to the sensitivity of the sensorimotor dynamics to changes in the sensor values. We claim that one can combine the two above requirements by introducing sensor values $\hat{x}$ defined by

$$\|x_{t+1} - \psi(\hat{x}_t)\| = \min \tag{10}$$

with a conveniently defined norm[1]. Explicitly the shift $v_t = \hat{x}_t - x_t$ is

$$v_t = \arg\min_u \|x_{t+1} - \psi(x_t + u)\| \tag{11}$$

Obviously v is small if both ξ (which measures the predictability) is small and the function ψ is sensitive to its arguments. Hence the two aims of getting a robot with both highly sensitive reactions and predictability of behaviour amounts to the requirement that the shift necessary to produce the new sensor values is as small as possible. Consequently we may define

$$E_t = v_t^2 \tag{12}$$

where (dropping the time index) $v^2 = v^\top v$ as our objective function for the behaviour of the robot. Using gradient descent the parameter dynamics is

$$\Delta C = -\varepsilon \frac{\partial E}{\partial C}(x, C) \tag{13}$$

Note that the parameter dynamics Eq. 13 is updated in each time step so that in practical applications the parameters may change on the behavioural time scale. This means that the parameter dynamics is constitutive for the behaviour of the robot.

[1] In general the choice of $\hat{x}$ is not unambiguous. In this case one may use the set of all possible solutions in order to create the learning signal.

2.3 Explicit expressions

The above equations define our approach in principle. However in order to better understand the nature of the parameter dynamics we study it in the approximation of small v. The definition Eq. 10 of the shift may be written as the requirement

$$\|\psi(x) + \xi - \psi(x+v)\| = \min \tag{14}$$

If v is small we may use Taylor expansion to write

$$\psi(x+v) = \psi(x) + L(x)\, v \tag{15}$$

where L is the Jacobian matrix of the sensorimotor loop defined as

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

Using Eq. 15 in Eq. 14 we find[2]

$$v = L^{-1}(x)\, \xi$$

and obtaining v means now "only" to find the (pseudo-) inverse of the matrix L. Introducing the positive semidefinite matrix $Q = LL^\top$ Eq. 12 may now be written as

$$E = \xi^\top Q^{-1} \xi \tag{16}$$

see [4] for further details. We used this expression in the parameter dynamics Eq. 13 in the examples given below. As explained above, in these examples we have

$$\psi_i(x) = \sum_{k=1}^{n} A_{ik}\, g(z_k)$$

so that

$$L_{ij}(x) = \sum_{k=1}^{n} A_{ik}\, g'(z_k)\, C_{kj} \tag{17}$$

Eq. 16 involves the inverse of the matrix Q which measures the sensitivity of the sensorimotor loop towards changes in the sensor values. Therefore, minimising E is immediately seen to increase this sensitivity. We have shown in many practical applications that in this way the robot develops an explorative behaviour which however is moderated by the fact that E is also small if the prediction error ξ is small. Behaviour may be understood as the compromise between these two opposing tendencies.

[2] If L is singular this is to be understood in the sense of the pseudoinverse.

3 Example I. The rocking stumper

One of the interesting phenomena observed under the parameter dynamics derived from Eq. 13 is the active closing of the sensorimotor loop so that the system is set into motion, see [4]. In order to demonstrate this phenomenon we consider here a system consisting of a stumper-like object with a pole mounted on it driven by two motors in orthogonal directions, see Fig. 1. The only sensors we have are two infrared sensors mounted at the two front ends of the trunk looking down and slightly sideways. Their values x_1 and x_2 depend on the distance to the ground in a highly nonlinear way. Our controller consists of two neurons with outputs y_1, y_2 controlling the angles of the pole relative to the trunk.

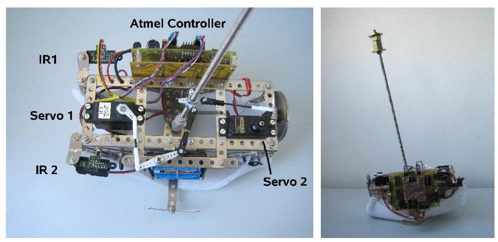

Fig. 1. Pole driven stumper. Left: close view from the top; Centre: pole to the back, Right: pole to the front

We use the linear world model with the learning step of Eq. 8 and the pseudolinear controller so that the gradient of the error $E = \xi^\top L^{-1T} L^{-1} \xi$ is easily evaluated since the inversion of the matrix L can be done explicitly.

The initialisation of the parameters C_{kl} can be done randomly starting with small values. However one should check whether the sign of the determinant of L is positive, if not reinitialise. The point here is that the error E diverges if L is singular and that the sign of the determinant defines the nature of the bifurcations taking place. If the determinant is negative, the feed-back strength in the sensorimotor loop is driven towards large negative values. Once beyond the flip bifurcation the signs of the controller outputs are inverted in each time step which is difficult to realise for the robot.

After initialisation we at first have subcritical values for the feed-back strength of the sensorimotor loop (see [4] for details) so that the influence of the noise (the prediction error ξ) is damped and we observe only small fluctuations of the pole position. With increasing values of the controller parameters C and therefore increasing feed-back strength the pole movements become stronger so that after some time a bifurcation point, typically the Neimark-Sacker bifurcation, see e.g. [7], [11] for details, is reached and an (irregular) oscillatory motion sets in. In Fig. 2 the behaviour, reflected by the sensor readings, and the parameter adaptation is displayed over time.

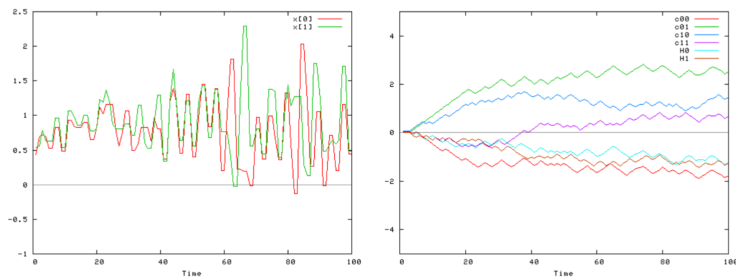

Fig. 2. Behaviour represented by sensor readings and controller parameters starting from low initialisation. Left: Sensor values from left and right infrared sensor over time. One can see clearly how the controller becomes sensitive and increases C_{kl}; Right: Controller parameter values over time. The controller matrix is adapted to map the difference of both sensors to servo 1 (y_0) and the sum of both sensors to servo 2 (y_1). The bias terms (upper two of the lower three lines) H_i are seen to be adapted such as to compensate for the positive average of the sensor values.

The interesting point in these experiments is that despite of the extremely nonlinear and nondeterministic behaviour of the mechanical system (the stumper) the controller learns to produce a motion which looks like the system is trying to probe into the possibilities of its body in a more or less controlled manner. In Fig. 3 the behaviour in a later stage of the experiment is shown.

We observed a rocking (oscillatory) as well as a walking like behaviour, the latter being caused by a rotational mode of the pole with suitable phase shift. The emergence of these modes is a direct consequence of the sensitisation paradigm. In fact, it is in these modes that the controller – based on the current sensor values – can evoke the maximum change in the sensor values over the time step. Ideally this would mean to "feel" the eigenfrequency of the mechanical system which is indeed about what happens.

In order to demonstrate the environment related nature of the emerging behaviours we put the performing robot into a corner where the infrared sensors measure a much shorter distance. As a result the robot became calm for a short time. Then the parameters were readapted to the new situation, so that an oscillatory behaviour sets in again. The same readaptation scenario occurred when moving the robot away from the corner by hand. We see that the robot is always sensitive to its environment and adapts to new situations quickly.

4 Example II Snakes

Systems with more degrees of freedom and of much higher complexity may be realised in ODE simulations, cf. [13]. We consider snakes as sketched in Fig. 4. In this application we use proprioceptive sensors only so that the sensorimotor loop now has n degrees of freedom where n is the number of joints. We assume again a linear world model in the form of Eq. 7.

Fig. 3. Environment sensitive behaviour. Top: Sensor values from left and right infrared sensor over time; Bottom: Parameter values over time; Until time 640 we observed rocking (oscillatory) motion with a short break at time 480. Then the robot was set into a corner. The infrared sensors measure much shorter distances since they see the walls. At time 870 the robot was pulled back into free space. After each change of the environment the robot was calm for a while (low sensor fluctuation) and probed the new environment, however after a short time the robot rocked again.

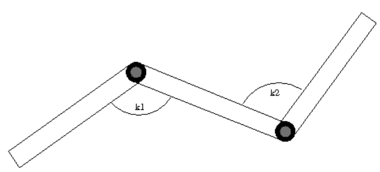

Fig. 4. A snake with two joints. Sensor values sent to the controller are the angular velocities of the joints, the controller outputs being the desired angular velocities. Note that the controller has no knowledge about the angles, the masses and geometry of the arm, and other environmental observables. The only information about the world is given by collisions and friction forces.

In the general n dimensional case the inversion of the matrix L does not make sense numerically. Instead we find v directly by solving the equation $\xi = Lv$ for v by some numerical method. A rather crude approximation turns out to be appropriate.

In the following experiments we used a snake with $n = 5$ joints on a plane, see Fig. 5. We initialise the matrix A as a diagonal matrix with the A_{ii} chosen such that the response of the joints is already coarsely modelled. The matrix C is also chosen diagonal but with very small random values for the C_{ii} so that in

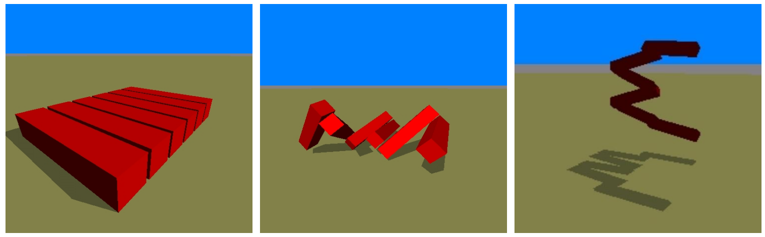

Fig. 5. Screenshots of snakes on a plane. Left: in initial position; Centre: crawling; Right: jumping.

the beginning the joints execute fluctuating motions only. In the beginning we have n decoupled feed-back loops due to this diagonal initialisation. As seen in Fig. 6 the parameter dynamics rapidly increases the diagonal elements of C so that the feed-back strength in each of the loops increases. After some time they reach the critical values where the fixed points are destabilised and an intensive motion sets in. In this regime the nondiagonal elements are also seen to develop so that the dynamics of the joints are coupled. This is on the one hand again an effect due to the sensitisation pressure which favours oscillatory modes. On the other hand the reaction of the joints to the applied forces are correlated due to collision, inertia, and friction effects. Therefore the motion of the snake is largely depending on the environmental conditions which is clearly born out by the experiments. The coherence in the motions of the joints is reflected by the nondiagonal elements of the matrix A, see Fig. 7.

The emerging dynamics is quite complex and rather difficult to analyse. However the degree of organisation of the motion can for instance be measured by the motion of the centre of the snake projected on the plane, see Fig. 8 (left). We find that in the beginning the centre is more or less stationary (in a time average picture) but after some time the snake covers increasingly larger regions of space. Apart from that, the altitudes of the snake segments also provide information about the type of behaviour. For analytical purpose we consider the centre of the highest and lowest segment over time. The difference between both can be interpreted as a measure for the current posture. Jumping behaviour is characterised by an altitude > 0.5 of the lowest segment. As shown in Fig. 8 (right) the snake sits up frequently and occasionally performs jumps. Note, that even on long time scales we observe qualitative changes in the parameters (Fig. 7), indicating a rich behaviour diversity. This is also seen directly when watching the snake over a long time (see the videos). We did this in many experiments in varying environments and also with two snakes in a cage. In all cases we observed an impressing variety of behaviours so that we are inclined to say that our self-regulation controller makes the snake "alive".

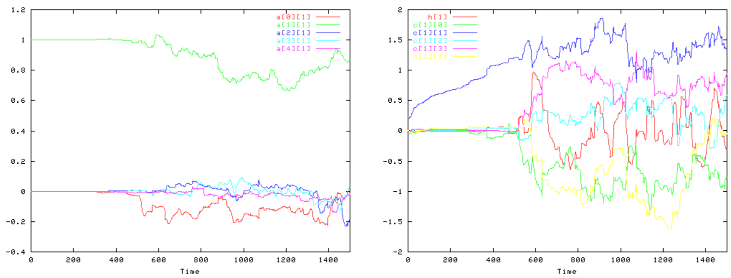

Fig. 6. Development of the parameters A_{i1} and C_{1j} associated with the neuron controlling joint 1 in the initial phase of an experiment. Left: The world model matrix A is initialised as the unit matrix reflecting the independence of the joints. The learning dynamics preserves this in the initial phase. Right: The diagonal elements of the matrix C are initialised with very small random values for C_{ii}. The diagonal elements increase until the supercritical feed-back strength is reached and the system starts to move (at about time 500). The development of the nondiagonal elements reflects the integration of contributions of the other segments. However, the self coupling C_{ii} is seen to stay still dominant (top line).

5 Discussion

We have demonstrated in the present paper that our general paradigm in applications to completely different agents yields in each case an environment related active behaviour. The emerging behaviours are are dictated by the body of the agent. Our stumper develops rocking or even "walking" modes with sometimes covering substantial regions of space. The snake which is mechanically completely different is seen to develop crawling and jumping modes which may be considered as emerging behavioural organisation where the snake learns to feel the possibilities of its body.

The emerging behaviours may be called environment related although they are generated by a completely domain invariant principle. For instance this is demonstrated by the stumper which when in a rocking or "walking" mode can be taken and put in a corner so that there is a completely new mechanical and sensorial situation. Nevertheless after some time it again finds back into its rocking behaviour so that eventually it gets out of situations where it is captured. Similarly we can put one or several of our snakes into a cluttered environment (work in progress) without the snakes being caught in corners. Moreover the snakes may entangle but in all situations find a way to disentangle. First results can be found in the videos in [3].

It is a further interesting property of our approach that the parameter dynamics never gets stuck in the saturation regions of the neurons or that the activity of the agents goes down for a longer time. Although we have taken some numerical precautions this is still an amazing property of the algorithm in

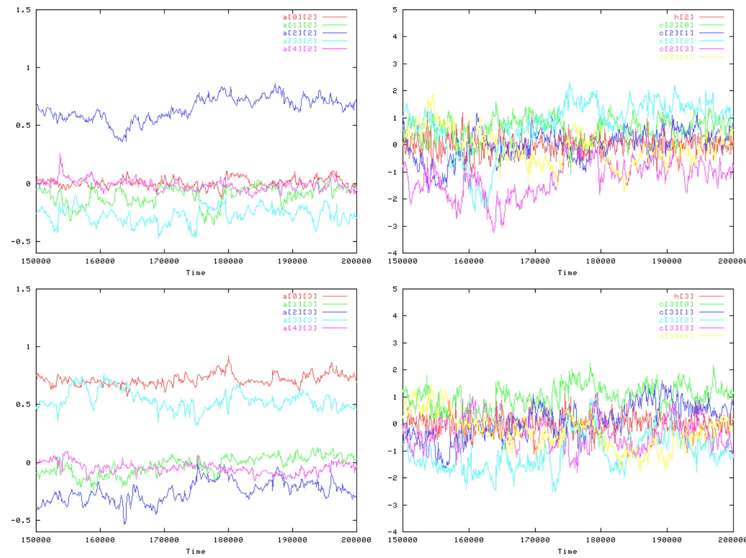

Fig. 7. Controller and model parameters for joints 2 (top) and 3 (bottom)during time step 150 000 to 200 000. (every 100th value plotted). Left: Model parameters; Right: Controller parameters; In accordance to our sensitisation paradigm the controller parameters are substantially changing over time but stay in a certain range, so that the neurons remain in a sensitive working regime. The model parameters A_{ij} describe the observed angular velocity at joint i as the response of the motor action applied to joint j. One would expect a diagonal matrix A, however some non-diagonal elements are non-zero, reflecting the correlations between different joints, for instance $a[0][3]$ in the lower left diagram.

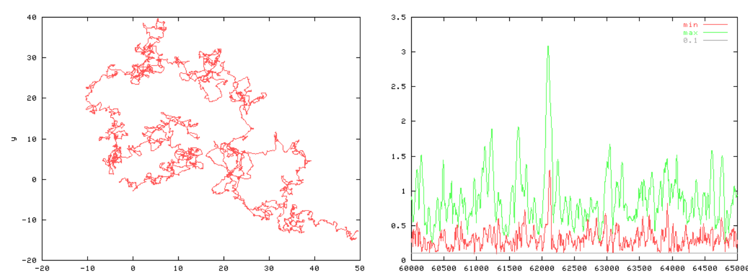

Fig. 8. Motion of snake with 5 joints (6 segments with length 1) during the experiment. Left: Position of the snakes centre projected on the plane over 165 000 time steps with starting point at (-3,0). Right: Altitudes of the centres of the highest (max) and the lowest (min) segment from time step 60 000 to 65 000. Segments laying completely on the ground have a altitude of 0.1, standing upright have a altitude of 0.5. One can see that the snake sits up and even jumps so that it exceeds the altitude of 0.5 with the lowest segment.

view of the fact that the parameter dynamics ultimately is driven by the noise (prediction error) which may change by orders of magnitude.

We consider our approach as a novel contribution to the realization of artificial life systems. In fact if life is an emerging property of complex systems in challenging environments then we may claim that we observe self-organised forms of (artificial) life of a rather high complexity. At the present step of our development the behaviours although related to the specific environments are without goal. As a next step we will realize a so called behaviour based reinforcement learning. When watching the behaving system one often observes behavioural 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 behaviour.

References

1. R. D. Beer. A dynamical systems perspective on agent-environment interaction. *Artif. Intell.*, 72(1-2):173–215, 1995.
2. A. Bredenfeld, H. Jaeger, and T. Christaller. Mobile robots with dual dynamics. *ERCIM News*, 42, 2001.
3. R. Der. Videos of self-organised robot behavior. http://www.informatik.uni-leipzig.de/˜ der/Forschung/videos.html, 2005.
4. R. Der, F. Hesse, and R. Liebscher. Contingent robot behavior generated by self-referential dynamical systems. *Autonomous robots*, 2005. submitted.
5. B. R. Fajen, W. H. Warren, S. Temizer, and L. P. Kaelbling. A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *Int. J. Comput. Vision*, 54(1-3):13–34, 2003.
6. M. D. G. Schöner and C. Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems,*, 16:213–245., 1995.
7. R. Haschke and J. J. Steil. Input space bifurcation manifolds of recurrent neural networks. *Neurocomputing*, 64C:25–38, 2005.
8. H. S. Hock, G. Schöner, and M. A. Giese. The dynamical foundations of motion pattern formation: Stability, selective adaptation, and perceptual continuity. *Perception & Psychophysics*, 65:429–457, 2003.
9. M. Hülse and F. Pasemann. *Dynamical Neural Schmitt Trigger for Robot Control*, volume 2415 of *Lecture Notes in Computer Science*. Springer, 2002.
10. S. Nolfi and D. Floreano. *Evolutionary Robotics*. MIT Press,Cambridge, MA, 2000.
11. F. Pasemann, M. Hild, and K. Zahedi. SO(2)-networks as neural oscillators. In J. Mira and J. Alvarez, editors, *Computational Methods in Neural Modeling*, pages 144–151. Springer, 2003.
12. R. F. Port and T. V. Gelder, editors. *Mind as Motion*. The MIT Press, Cambridge, MA, 1995.
13. R. Smith. Open dynamical engine. http://ode.org/, 2005.
14. A. Steinhage. *Dynamical Systems for the Generation of Navigation Behavior*. PhD thesis, Ruhr-Universit at Bochum, Germany, 1997.