

Representing Robot-Environment Interactions by Dynamical Features of Neuro-Controllers *

Martin Hülse, Keyan Zahedi, Frank Pasemann

Fraunhofer Institute for Autonomous Intelligent Systems (AIS)
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany

Abstract

This article presents a method, which enables an autonomous mobile robot to create an internal representation of the external world. The elements of this internal representation are the dynamical features of a neuro-controller and their time regime during the interaction of the robot with its environment. As an examples of this method the behavior of a Khepera robot is studied, which is controlled by a recurrent neural network. This controller has been evolved to solve an obstacle avoidance task. Analytical investigations show that this recurrent controller has four behavior relevant attractors, which can be directly related to the following environmental categories: free space, obstacle left/right, and deadlock situation. Temporal sequences of those attractors, which occur during a run of the robot are used to characterize the robot-environment interaction. To represent the temporal sequences a technique, called macro-action maps, is applied. Experiments indicate that macro-action maps allow to built up more complex environmental categories and enable an autonomous mobile robot to solve navigation tasks.

*in: M.V. Butz, O. Sigaud, P. Gérard (Eds.), *Anticipatory Behavior in Adaptive Learning Systems*, LNAI 2684, Springer, Berlin 2003, pp. 222-242.

1 Introduction

There are now many attempts to increase the intelligence or cognitive abilities of autonomous systems like physical mobile robots. This seems to be desirable for many tasks one expects these systems to solve. Equipped with several types of sensors and with enough actuators they should be able to navigate and act in non-trivial changing environments. Sometimes they are assumed to develop also communication skills and some kind of social behavior which allows cooperative interactions - possibly with humans. In some sense they are often expected to mimic living systems.

This is of course a challenging perspective, and in general it is assumed that a better understanding of how such systems build up internal representations of their environment, how these representations can be modified during interaction with the environment, and how it can be adapted to a dynamical task management, are the prerequisites of this desired development.

On the other hand, using advanced dynamical neural networks for behavior control, it is quite unclear how an internal representation in a neuro-controller will look like, i.e. on which level it will be implemented. It could be implemented as specific connectivity structure, as a weight matrix, as stationary states, or in terms of attractors of internal dynamical processes [25]. To approach these problems, it will be interesting to know, for instance, how motor commands can be mapped onto their sensory consequences or how a desired stream of sensor inputs can be accomplished by an appropriate sequence of motor commands [26], [15].

Following a modular neuro-dynamics approach, the basic assumption of this work is that cognitive performance is based on internal dynamical properties, which are provided by a recurrent connectivity structure of neural subsystems [2],[5],[11], [21]. An evolutionary robotics approach [18] is used to develop appropriate recurrent neuro-controller. But if we assume that higher cognitive abilities (e.g. planning tasks) need some kind of internal representation of the external world, one suggests to use those dynamical properties as basic elements for internal representations. The interaction of the robot is not only purely triggered by environmental conditions but is determined by specific internal dynamical features of its neural control system. This means in different situations different dynamical properties become active. Thus, it is suggested that those dynamical properties are the basic entities for a description of the robot's environment. Following this argumentation future prediction or expectations of robot behavior are state anticipations. If a state is interpreted as an specific "attractor of the robot-environment system", this implies that any goal-directed behavior can only be developed in a sensori-motor loop. Thus, the main focus of this work is to demonstrate

how different dynamical features of neuro-controllers can be extracted during the interaction of the robot with its environment, how they can be used for classification of environmental properties, and how such categories can be utilized to encode and produce goal-directed behavior.

In the following investigations we concentrate, as a demonstration of method, on a simple example of robot behavior. An evolved recurrent neuro-controller is introduced which is able to endow miniature Khepera robots [16] with a robust obstacle avoidance behavior (section 2). The prominent feature of the used evolutionary algorithm *ENS*³ [23] is its ability to evolve neural networks of general recurrent type without a specific connectivity structure determined in advance, which makes this algorithm similar to the GNARL algorithm [1]. It is thus mainly used for structure development, but it optimizes parameter values, like weights and bias terms at the same time. Only the number of input and output neurons of a controller are fixed according to a given sensor-motor configuration. Therefore resulting networks can have any kind of connectivity structure, including feedback-loops and self-connections.

In section 3 an implementation of a standard Braitenberg controller (BC) [7] and of an evolved neural network, called the minimal recurrent controller (MRC) [9], are used to analyze how the interaction of a robot with its different environments can be represented in terms of dynamical controller features. For this purpose methods like the *first return map* (FRM) of appropriate sensor and motor data are applied. In section 4 a method called *macro-action maps* (MAM) is introduced to represent temporal sequences of specific dynamical features during the interaction of the robot. The specific dynamical features can be interpreted as “attractors of the robot-environment system”, and the sequence of successively visited domains can be seen to represent relevant aspects of the environment and of the task, respectively. Some experiments are presented which illustrate how macro-action maps represent the robot-environment interaction and further on, how the elements of those maps can be used to build up more complex environmental categories. It is shown that those categories can enable an autonomous mobile robot to solve navigation tasks.

2 The Task

The task is to control a miniature robot, the Khepera, such that it can move collision free in a given environment with scattered objects; i.e., the classical obstacle avoiding task. To solve this task, eight infrared sensors can be used as proximity sensors, six at the front, two at the rear, and there are two

wheels driven by two motors. The controllers only use two inputs, I_0 and I_1 . At each time step t they serve as buffers for the average of the current values of the three left, respectively the three right front sensors. Data from the rear sensors of the robot are not used. Controllers will have two output units, O_0 and O_1 , providing the signals driving the left, respectively right motor. The neurons of the controllers will be of the additive type; i.e. their dynamics is given by

$$a_i(t+1) = \theta_i + \sum_{j=1}^n w_{ij} \cdot f(a_j(t)) , \quad (1)$$

where the activation a_i of neuron i at time $t+1$ is the sum of its bias θ_i and the weighted sum of the outputs $f(a_j)$ of the other neurons at time t , and w_{ij} denotes the strength of the connection from neuron j to neuron i . The transfer function f will be defined differently for both controllers. For the BC it is implemented as follows:

$$f(x) = \begin{cases} 0 & : x < -1 \\ \frac{1}{2}(x+1) & : -1 \leq x \leq 1 \\ 1 & : x > 1 \end{cases}$$

For the MRC we use $f(x) = \tanh(x)$ as transfer function. The connectivity of the controllers and a typical path of a simulated robot in one of the environments is shown in Figure 1.

2.1 The Braitenberg-Controller

In the case of the BC the sensor signals are pre-processed in such a way that controllers get input values $I_{0,1}$ between 0 and +1. They increase with decreasing distance between sensor and obstacle. In order to realize separate back- and forward movements for each wheel one needs positive and negative signals M_0 and M_1 , driving the left and right motor. They are provided by a post-processing of controller outputs O_0, O_1 according to: $M_{0,1} = 2 \cdot O_{0,1} - 1$. Wiring and bias terms of the BC are inspired by [17]. Each controller output has excitatory connections to the sensors on its own side and inhibitory connections to the sensors on the opposite side. Thus, this implementation of the BC is a simple feed-forward network. The two bias terms realize a positive offset value for each motor, and therefore a forward motion is generated when the robot receives no sensor inputs. An input signal, I_0 say, steers the robot away from obstacles, as it inhibits the output unit O_1 and at the same excites the output unit O_0 . 1.

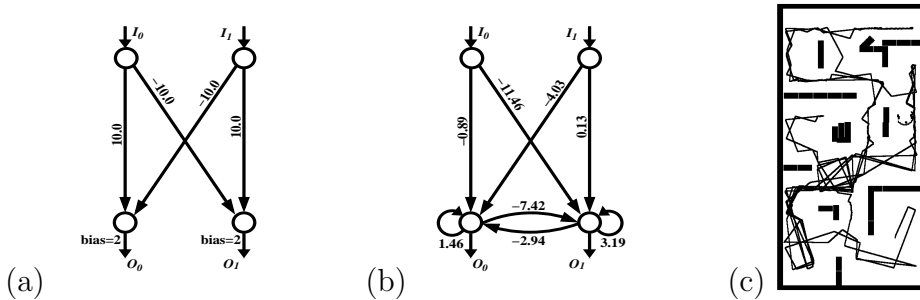


Figure 1: Structure of the two controllers (a) BC and (b) MRC, which solve the obstacle avoidance task. Figure (c) indicates the behavior of a simulated Khepera robot controlled by the MRC.

2.2 The Evolved Minimal Recurrent Controller

The sensor signals for the MRC are pre-processed such that input values for I_0 and I_1 are mapped to the interval $[-1; +1]$. They increase with decreasing distance of an obstacle from the sensors. The interval $[-1; +1]$ for input values is chosen for consistency, because the MRC uses \tanh as transfer function for the two output units. There are no bias terms, and no post-processing for the motor signals is necessary in this case.

Given these two input and output units, the evolutionary ENS^3 -algorithm [23] is used to develop a controller achieving the desired properties without employing an internal unit. One of the results, shown in Figure 1(b), is used for the following discussions. This controller, the MRC, has positive self-connections (1.46 and 3.19) for its output units, which interact recurrently by inhibitory connections -7.42 and -2.94 . It is well known from analytical investigations [20], [22] that single units with self-connections larger than $+1$ and 2-neuron loops with an even number of inhibitory connections can have co-existing fixed point attractors providing a hysteresis effect. The dynamical interplay of these three structures generates the advanced behavior of robots [9] as described in the following section.

2.3 Robot Behavior Controlled by BC and MRC

The BC and the MRC were introduced as networks which solve an obstacle avoidance task. But, the actual behavior is different. Figure 1(c) indicates a typical path of the robot generated by the MRC. It shows both, obstacle avoidance and exploration behavior. The behavior of the physical robot controlled by this network is comparable to that of the simulated one. Especially sharp corners and dead ends can be handled correctly, as indicated in the

upper right corner of the environment in Figure 1c. In contrast to that the behavior of the BC is very different. Because it realizes an obstacle avoidance task in the sense that it turns to the right if an obstacle is detected on the left and the other way round. But in the case of deadlocks provoked by sharp corners or impasses it gets stuck and has no chance to escape autonomously from such situations. Hence one can say, the MRC shows a qualitatively improved behavior, since its interaction with the environment is context-sensitive in contrast to the pure reactive behavior of the BC. This better performance of the MRC is produced by the interplay of three different hysteresis effects generated by the already mentioned recurrent connectivity structure of the MRC. The hysteresis effect i.e. evoked by the positive self-connection on output unit O_0 (in the following denoted by LHE for *left-side hysteresis effect*) provides an appropriate turning angle for avoiding an obstacle on the right side. Whereas the self-connection on output unit O_1 generates a *right-side hysteresis effect* (RHE), which produces a right turn to avoid an obstacle on the left. The escape of the robot from dead ends is caused by the *extended hysteresis effect* (EHE), which lets the robot turn until it has a free space in front of it. The EHE results from an interplay of the LHE and the RHE which is mediated by the inhibitory ring of units O_0 and O_1 [9].

3 Sensori-Motor First Return Maps

In this section we will utilize the so called *first return maps* of sensor signals and motor signals, respectively, to represent the robot-environment interactions. These maps are defined as follows: The *sensor first return map* (S-FRM) plots the difference $\Delta_I := (I_0 - I_1)$ of the two inputs I_0, I_1 at time $t + 1$ over the difference at time t . Correspondingly, the *motor first return map* (M-FRM) plots the difference $\Delta_O := (O_0 - O_1)$ between the output signals O_0, O_1 to the left and right motors at time $t + 1$ over the difference at time t . We also make use of the sensori-motor map (SMM), which plots $\Delta_O(t)$ over $\Delta_I(t)$.

For these plots there are three characteristic points on the main diagonal: the lower left corner with coordinates $L = (-2, -2)$, the origin $(0, 0)$, and the upper right corner $R = (2, 2)$. For the S-FRM the points L and R represent a near obstacle at the right, and left side of the robot, respectively. The origin represents in general obstacle free space. For the M-FRM the points L and R represent left and right turns on the spot with maximal angular velocity, points on the main diagonal constant circular motion, and the origin represents straight movement along a line. Finally, a SMM represents the action following a sensor stimulus. The point L stands for a fast left turn

of the robot if there is a near object at its right side, and R stand for a fast right turn if an obstacle appears at the left side. The origin codes the situation where there are identical left and right sensor inputs (they may, of course, be zero) and the robot reacts with a straight movement. Points in the upper left and lower right quadrants represent impossible situations like almost instantaneous jumps of objects from one side to the other (S-FRM), or instantaneous switching between left and right rotations (M-FRM), or undesirable reactions turns toward an object (SMM). Relevant quadrants for the discussion are therefore the upper right ($x > 0, y > 0$) and lower left quadrant ($x < 0, y < 0$), called R-quadrant and L-quadrant, respectively. Of course paths in the R-quadrant correspond to right turns, those in the L-quadrant to left turns.

For comparison, in Figure 2 these three types of first return maps are depicted for the Braitenberg controller and the minimal recurrent controller. These representations of sensori-motor signals clearly show the different control techniques applied by the two networks. Looking at the S-FRMs (figures (a)), one realizes that points accumulate around the main diagonal; i.e. successive differences in inputs change only gradually during time steps. According to the pre-processing of sensor data one observes that the BC receives smaller differences of input values than the MRC. Next consider the M-FRMs of the two controllers (figures (b)). The M-FRM of the BC appears as a more or less stretched version of its S-FRM in Figure 2(a), indicating roughly a pure reactive response of the controller to its sensor inputs. Whereas the M-FRM of the MRC shows a significantly different pattern. Both relevant quadrants show a sequence of points which can be described as leaf-shaped. These curves can be divided into upper and lower parts; i.e., parts above, respectively below the main diagonal. We will first concentrate on the L-quadrant.

Due to the obstacle avoidance task, for which the fitness function rewards straight movements of the robot, we expect that the evolved MRC tries to minimize the difference of the outputs as fast as possible. But a somewhat different behavior is observed and can be explained as a hysteresis effect. Whenever the output difference $\Delta_o(t)$ at time t grows (upper part), the output will grow steadily until it reaches its maximum. In fact, there are no shortcuts between the upper and lower path. Once it reaches the maximum difference, the MRC will steadily reduce the output difference until it reaches zero. Thus the origin and the point L are the only intersections of the upper and lower paths in the L-quadrant. The reason for this can be read from Figure 2(c) and will be discussed later in this section. If one looks at the R-quadrant, a very similar behavior is observed. The MRC steadily increases or decreases the difference of its output until it reaches one of the intersections

points, R or the origin. But one can observe also an additional feature. The MRC allows a much stronger growth of differences in the R-quadrant. As will be shown later, it also keeps the maximum positive difference Δ_O for a longer time than the negative maximum difference. This is indication for the strategy to leave dead ends and sharp corners.

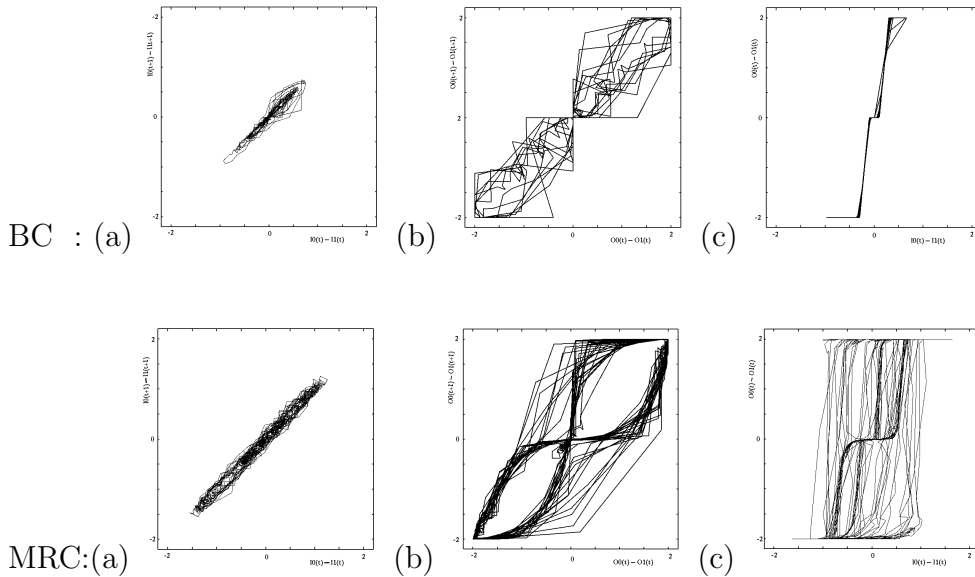


Figure 2: First return maps of the two controllers taken while the physical robot was moving in its environment: (a) first return map of the difference Δ_I of input values, and (b) of the difference Δ_O of output values; (c) the sensori-motor map plotting $\Delta_O(t)$ over $\Delta_I(t)$.

Next the SMMs of the controllers are compared (Figures 2(c)). The Braitenberg controller shows mainly three different activities corresponding to output differences $\Delta_O = 1, 0$, and -1 . We will start the analysis at the origin. If the input difference Δ_I increases to a certain threshold, the output difference Δ_O will jump to its maximum. If the difference of the input Δ_I is large and then slowly decreases until it falls below a positive threshold, then Δ_O will jump to 0. The same holds for negative input differences and the negative threshold value.

Analyzing the MRC, hysteresis effects are observed. The inner narrow loops in the R- and the L- quadrant correspond to hysteresis intervals for positive and negative input differences Δ_I . They appear as a “widening” of the jump lines of the BC. The figure shows, that for the MRC one has $\Delta_O \approx 0$ for a plateau of input differences Δ_I . If the positive input difference exceeds a certain positive threshold value, the Δ_O will jump to its maximum. If

then Δ_I decreases, Δ_O will stay constant until Δ_I falls below a second, lower threshold value. For L-quadrant a similar behavior is observed. But there is a third hysteresis seen in Figures 2(c) for the MRC, which is remarkable. To see this, assume the system is in a state where obstacles are at equal distance from the robot and therefore it moves straight forward; i.e., we are in the origin of the SMM and the robot moves into a deadlock situation. If now the input difference Δ_I increases steadily the output difference Δ_O will then jump to a high value at some threshold, as discussed before. But if Δ_I is now decreasing, Δ_O can still remain stable at the high value, even if the input difference falls below the negative threshold value for the inner hysteresis. It is exactly this feature which is needed to handle dead ends and sharp corners. If the robot is in a sharp corner, it needs to turn in one direction, even if the difference in inputs decreases significantly. Else the robot would try to avoid the side on which its input is higher, thereby turning towards the other side with the lower sensor input. Then the higher input value would decrease, while the lower would increase, until the situation is reversed and the direction of rotation, correspondingly. The resulting behavior is shown by the Braitenberg controller. Hysteresis is the solution to this deadlock situations, and it is provided by the recurrent structure of the MRC as demonstrated in [9]. To emphasize this result a 3-dimensional plot of the SMM is plotted adding the time axis (Figure 3). The first two maximum differences (corresponding to L- and R-points) show how the MRC reacts to corners. The last maximum difference shows how the MRC reacts to a dead end or a sharp corner. The MRC provides maximum output difference Δ_O for a longer time (as can be seen from the 2-dimensional projection in the lower figure) until the robot can go straight ahead again. Observing the physical robot, one can see clear 180-degree turns in such dead end situations. Figure 4 demonstrates the robustness of the MRC. During the interaction of the robot most of the time the control signals lie in those domains, which produce precise actions like moving straight (40%), left turns (15%), and right turns (5%).

The comparison of the control strategies, as reflected by Figure 2, explains and supports our hypothesis, that the BC shows a purely reactive behavior, whereas the MRC shows a qualitatively improved behavior.

4 Macro-Action Maps

As outlined in the last section, for the MRC there are four discernable robot-environment interactions which are clearly related to the internal dynamics of the controller. The two small hysteresis effects, corresponding to the

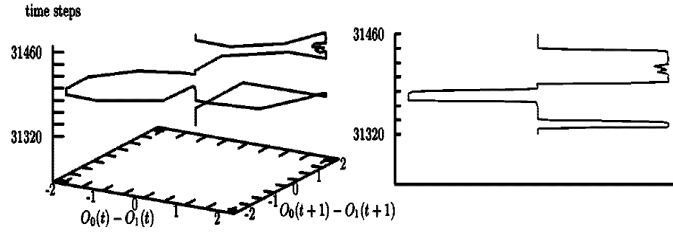


Figure 3: The time development of the MRCs first return map of output differences $\Delta_O(t)$, and its projection onto the $(\Delta_O(t), t)$ plane (right).

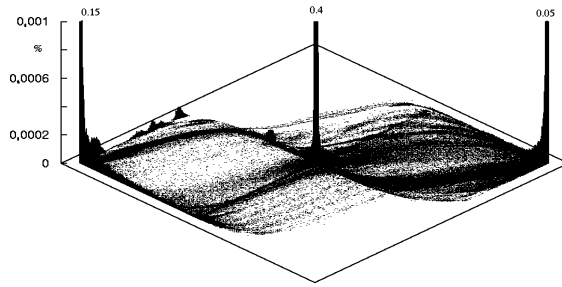


Figure 4: The relative appearance of points in the MRCs first return map of output differences Δ_O , indicating three main actions represented by the points $\Delta_O = +2, 0, -2$.

co-existence of two fixed point attractors, become active during simple left or right turns. The third hysteresis effect with its extended input domain is activated in typical deadlock situations, whereas simple straight forward movements in obstacle free space indicate a unique stable fixed point. Figures 2(b), 3, and 4 illustrate that these different interaction states are easy to distinguish if in addition to the difference of the motor outputs Δ_O also its time development is plotted. The following table shows the relations between features of the internal dynamics, difference in controller output, and the external situation:

4.1 Definition and Implementation

One possible way to implement a temporal segmentation of different robot-environment interaction states and its graphical representation is based on the following definitions.

A *macro-action* $[\{+/-\} d, s]$, $d, s > 0$, describes a rotation of d time steps to the right (+) or left(-) followed by a positive straight movement of

| internal dynamics | Δ_0 | time steps | external world |
|---------------------------|------------|------------|-------------------------------|
| unique stable fixed point | 0 | any | no obstacles |
| LHE | -2 | < 20 | obstacle on the right |
| RHE | +2 | < 20 | obstacle on the left |
| EHE | ± 2 | > 20 | sharp corners, impasses, etc. |

Table 1: Relation between internal dynamics of the MRC and the external world.

s time steps. We call $\{+/-\}$ the *sign of the macro-action*. A sequence of turns with no straight forward movement between them are summarized to one turn. \mathcal{A} is the set of all macro-actions. A *macro-action map* (MAM) is defined as a directed graph with nodes representing macro-actions and the directed edges indicate the temporal predecessor-successor relation between two macro-actions.

The development of such a MAM is inspired by the work on landmark-based navigation [12]. The robot segments its path through the environment according to landmarks. These landmarks are kept in a simple, self-organizing chain representation. Here, instead of landmark macro-actions are used. During interaction the emerging macro-actions are appended to the list of macro-actions. Usually this list would be temporally ordered. But before a new macro-action is added, it is tested if there exists a similar node in the list. If a similar node is found, values of the already existing node in the list will be updated with a value, which is the mean of the new and the already existing node. If there are more than one similar nodes in the list, only the first in the list will be changed. The predicate $S : \mathcal{A} \times \mathcal{A} \rightarrow \{0, 1\}$ of similarity between two macro-action $[d_1, s_1]$ and $[d_2, s_2]$ is defined as follows:

$$S([d_1, s_1], [d_2, s_2]) = 1 \iff (\text{sign}(d_1) \cdot \text{sign}(d_2) = 1) \wedge (|d_1 - d_2| \leq 5) \\ \wedge \left(|s_1 - s_2| \leq 5 \vee |s_1 - s_2| \leq \frac{1}{2}(s_1 + s_2) \right).$$

Finally a predecessor-successor relation will be established between the last node in the list and the already existing similar node. In such a way predecessor-successor relations between macro-actions are established. Iterating this procedure while the robot is moving in its environment, a MAM unfolds, which represents, on the level of attractor sequences, the interaction of the robot. This procedure is fulfilled as long the number of nodes does not increase anymore. Then the robot is stopped and nodes and relations in MAM are deleted if they are updated relatively seldom with respect to the total number of nodes in the MAM. Updates are counted for each node and relation.

In the following experiments we delete nodes if their update rate was less than 5 %, and relations if their update rate is less than 2.5 %.

4.2 Experiments on Building up MAMs

The following experiments will make clear how specific features of the robot environment are represented by the macro-action maps. Some examples of such MAMs and the related interactions are shown in Figure 5, 6, 7 and 9. The nodes of the graphs in these figures are labeled by the corresponding macro-action and a number, which is the absolute number of its occurrences during which the MAM was developed. Correspondingly, the numbers on the edges indicate the absolute number of occurrence of this sequence of macro-actions.

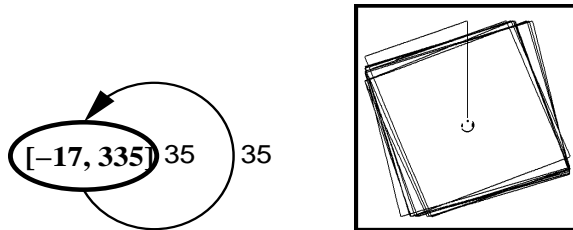


Figure 5: Right: robot path controlled by the MRC in square shaped world. In this simple world a very simple macro-action maps consisting of only one node and one edge is developed (left side).

For the first experiment (Figure 5) the MAM consists of only one node and one edge. In this case the world is square shaped with no obstacles, and it can be argued, that the constant rotation angles lead to constant driving distances and vice versa. Therefore the interaction of this world with this MRC is characterized by a temporal constant sequence of two interaction states: turning and driving straight. They are characterized by the left hysteresis effect (LHE) and the unique fixed point attractor (compare Table 1).

The same holds for the second experiment (Fig. 6) for which the world is chosen to be rectangular. In contrast to the first experiment there are now two different distances which the robot can move straight forward after a simple turn. Like in the first experiment the turning angle remains constant, but in this rectangular world the robot has to cover a short and a long distance until the next obstacle. Therefore the macro-action map representing the robot interaction in this experiment consists of two nodes and edges. This can be interpreted as the alternate appearance of the two macro-actions

$[-18, 57]$ and $[-17, 423]$ during the interaction. The first indicates a turn to the left followed by a short forward move and the second also a turn to the left but followed by a longer forward movement.

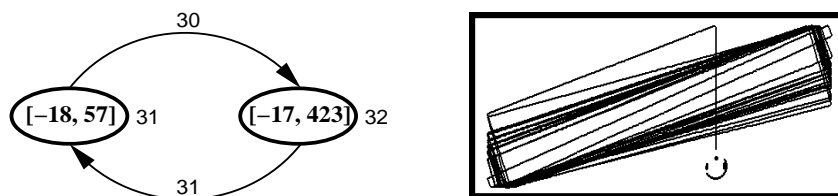


Figure 6: Robot path controlled by the MRC in rectangular world (right side). The two different distances which can be covered by robot before an obstacle appears are indicated by the two nodes in the corresponding macro-action map on the left.

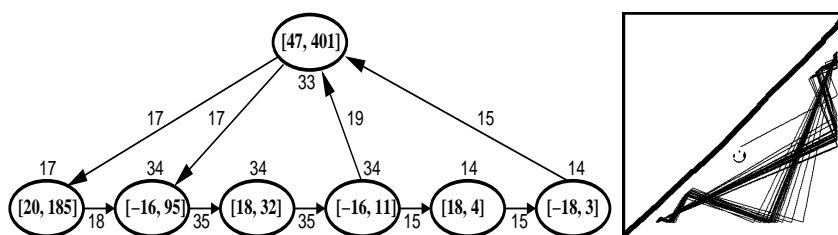


Figure 7: Robot path controlled by the MRC in triangular world (right side) and the corresponding macro-action map on the left side. See text for explanations.

A more interesting experiment is shown in Figure 7. Here the robot is confronted with two sharp corners: the bottom left corner and the upper right corner. In the corresponding MAM these two corners are represented by one and the same node ($[47, 401]$). This is because in the sharp corners only the EHE becomes active which produces the large turning angle with the same sign. Referring only to this single node the robot therefore can not discriminate by itself in which of these two sharp corners it is located.

But if the temporal sequence of the macro-actions is taken into account then these two sharp corners became distinguishable. As it is pictured in Figure 8 the macro-action sequence $[47, 401] [20, 185]$ indicates the robots way out of the bottom left sharp corner and the sequence $[47, 401] [-16, 95]$ the way out of the upper right sharp corner. Because after the escape of the bottom left corner the next obstacle is always avoided by a turn to the left. Whereas after an escape of the upper right corner the next obstacle is always avoided by a turn to the right.

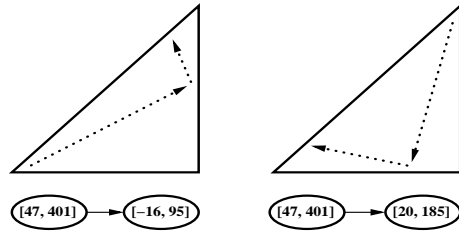


Figure 8: Subpaths of the robots interaction in the triangular world and the corresponding macro-action sequences, which are also subsequences of the MAM in Figure 7. The first node / macro-action refers to both sharp corners, but the successor node indicates which corner exactly the robot comes from.

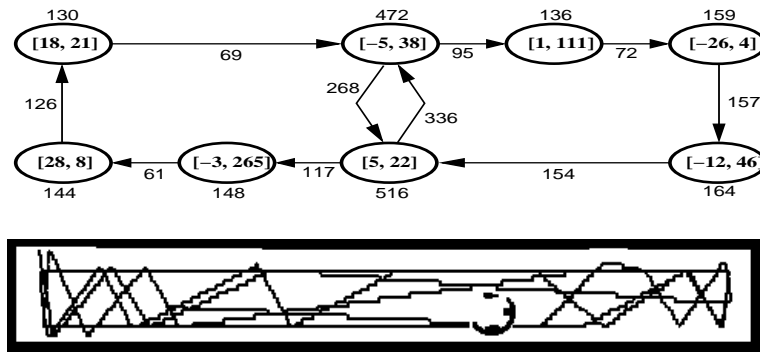


Figure 9: Robot path controlled by the MRC in a small corridor (bottom) and the corresponding macro-action map on the top. See text for explanations.

The last experiment in this section is an example for the fact that simple worlds can nevertheless produce complex MAMs. Figure 9 shows the environment of this experiment, which is a small corridor. One might suggest that the corresponding MAM contains only one node, as in the first experiment. But, the robot controlled by the MRC moves hardly parallel to the walls, and therefore it will oscillate between the two walls of the corridor. This means many turns and therefore many macro-actions are performed during the interaction. And finally it can be observed at the end of the corridor that the robot usually turns back in two steps. In the corresponding MAM this is represented by the two macro-action sequences $[28, 8]$ $[18, 21]$ and $[-26, 4]$ $[-12, 46]$.

Traveling through the corridor is characterized by oscillations as it is depicted in the macro-action map of Figure 9 by the loop between the two macro-actions $[-5, 38]$ and $[5, 22]$. The relation between these two nodes and the movement through the corridor is demonstrated in Figure 10. The

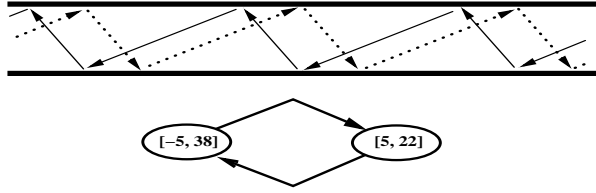


Figure 10: Traveling through the small corridor is characterized by a zigzag move, because the robot moves hardly parallel to the walls. So, the robot-environment interaction in this experiment is mainly described by small turning angles followed by short straight forward moves. This is related to the substructure of the MAM in Figure 10 (bottom) with only two nodes and edges, because left and right turns are alternately performed.

most frequent occurrence of this oscillation in this experiment is indicated by the large numbers of the corresponding nodes and edges in Figure 9.

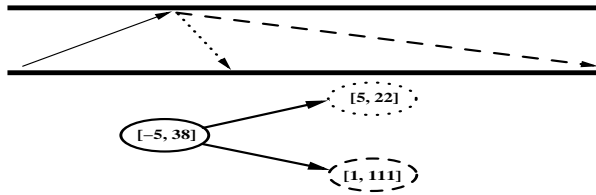


Figure 11: In the MAM of the experiment with the small corridor (Fig. 9) the node $[-5, 38]$ has the two successor nodes $[5, 22]$ and $[1, 111]$. The paths which correspond to these macro-actions are shown, line style refers to the same node / macro-action in the subgraph at the bottom of this figure. The two successor nodes or macro-actions are mainly distinguished by their "turning angles" 5 and 1. It is easy to see in this figure, that a smaller turning angle leads in a corridor to a longer straight forward movement.

Another point of the MAM in this experiment is, that the two nodes $[-5, 38]$ and $[5, 22]$ representing the zigzag move through the corridor are the only nodes, which have two successor nodes. It is easy to see in Figure 11 and 12, that for each node the second successor node reflects the "experiences" of the robot, that in narrow corridors smaller turning angles to avoid collisions lead to longer straight forward movements.

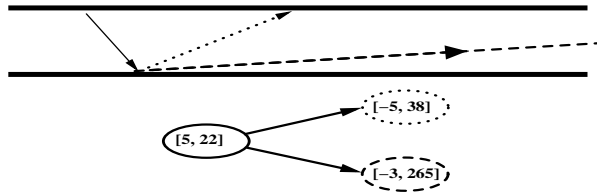


Figure 12: Like Figure 11, but according to node $[5, 22]$ the successor node $[-3, 265]$ in comparison with successor node $[-5, 38]$ reflects the fact that smaller "turning angles" lead to longer straight forward movements.

4.3 Experiments on Using MAMs for Exploration, Homing and Navigation

In the previous section the reader saw how MAM can be used to characterize the robot–environment interaction. The next three experiments demonstrate how MAMs can be used to encode behavioral sequences, which overcome the limited interaction of the MRC. In the following it is shown that a MAM based categorization of environmental states can provide exploration, homing and navigation capabilities.

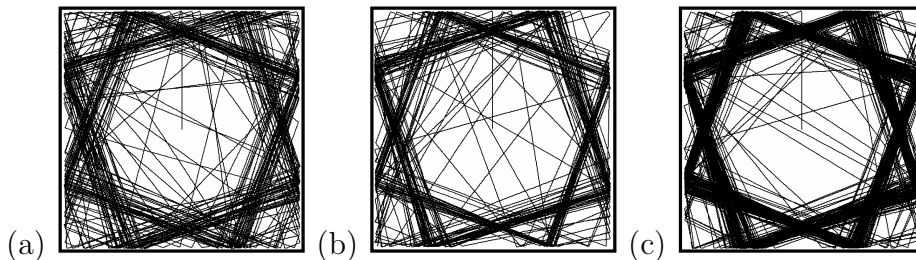


Figure 13: Robot path controlled by the MRC, but including a random turn if the last two (a) / four (b) / ten (c) macro-actions are similar. (100,000 time steps are plotted in Fig. (a) and (b) and 250,000 in (c).)

First of all a few definitions must be introduced in this section. As it was pointed out earlier, a MAM is a directed graph $G = (V, E)$, with $v \in V$ representing the macro-actions and $e \in E$ representing the predecessor-successor relationships between the macro-actions. A *path* of length k from a macro-action v_0 to a macro-action v_k in the directed graph $G = (V, E)$ of the MAM is a sequence $\langle v_0, v_1, \dots, v_k \rangle$ of macro-actions such that $(v_i, v_{i+1}) \in E$ for $i = 0, \dots, k - 1$. The length of the path is the number of relations. We also have to define the similarity of macro-actions and paths of macro-actions. Two macro-actions are *similar* ($v_0 \approx v_1$) if they satisfy the relation given in equation 2. A path p of macro-actions is a subset of a MAM $G = (V, E)$ if

$\forall v_p \in p \exists v_i \in V : v_p \approx v_i$ and for each relation between two macro-actions in the path, there exists a corresponding relation between two similar macro-action in the MAM. A corresponding relationship does not have to have the same occurrence label; i.e.

$$p \subset G \iff \forall (v_p, v_{p+1}) \in p \exists e = (v_i, v_{i+1}) \in E : v_p \approx v_i, v_{p+1} \approx v_{i+1}, v_i \in V.$$

The first experiment in this section demonstrates how the robot controlled by the MRC can develop an exploration behavior. We assume, that a good exploration behavior is shown, if the robot's path through the world covers up most of the free space. This is done using the square shaped environment as shown in Figure 5. As it was already mentioned the robots interaction in this world is represented by a MAM with only one node and one edge. That means, during the interaction only one macro-action emerges. During the interaction of the robot with the environment, a new path of macro-actions (in this case with $v_p = v_q, \forall v_p, v_q \in p$) is generated. To equip the robot with exploratory abilities the control of the interaction by the MRC is interrupted, if a path p of a certain length k of newly recorded macro-actions is a subset of the MAM. The interruption in this case means that the robot will turn a random angle and then be controlled by the MRC again. Figure 13 reflects the results of three experiments of this kind. The difference between these three experiments is the value, which determines how many of the last macro-actions have to be similar before a random turn is triggered.

The results of the three experiments show that after the robot is rotated by a random angle it does not take a long time to reach the old pathway, as shown in Figure 5. But in contrast to this figure the paths plotted in the Figures 13(a) – (c) indicate that there are two such stable pathways in this square shaped world. Depending from the start position and orientation the robot runs clockwise or counterclockwise through this world after a while. This determines in which of this two stable pathways the robot will end. Going back to the exploration abilities, one can see that in all three experiments the robot covers the whole area. The difference is according to the length k of the subset of macro-actions, that have to occur, before a random turn can be performed.

In the next experiment a homing behavior using MAM was implemented. In a triangular world (like Figure 7) the robot has to recognize only one of the two sharp corners as its home and wait there for a while. As it was already argued in the corresponding experiment of the pervious section, each sharp corner in this world is represented by the same node in the MAM. Therefore the robot has to distinguish the two sharp corners by a precedent sequence of macro-actions. In detail, additionally to the pure MRC control a *attention*

mode is implemented, which is deactivated by default but becomes active, if a specific sequence of two macro-actions occurs. Finally, if the attention mode is active and a macro-action corresponding to an extended hysteresis effect (EHE) occurs, then the robot stops for a certain time. After the waiting time is over, the attention mode is deactivated and the robot is controlled again by the MRC. Thus, “home” is not determined by the single node representing both sharp corners but is distinguished by a macro-action sequence activating the attention mode. For example, if the upper right sharp corner is defined as “home” (compare Fig. 7), then a macro-action sequence similar to [47, 401] [-16, 95] activates the attention mode. The sequence [47, 401] [20, 185] defines the the bottom left corner as “home”.

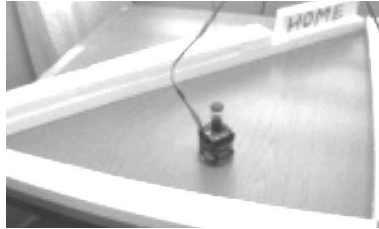


Figure 14: Homing behavior experiment with the miniature robot Khepera in a triangular world, like Fig. 7. One of the two sharp corners in this world is defined as home. If the robot moves in this ”home corner”, it has to stay there for a certain time.

This experiment has been done with a simulated and a real Khepera robot testing each corner separately as home. A mpeg-video, which can be downloaded from [10], shows two experiments with the real Khepera robot. Figure 14 is a screen shot of the first experiment in this movie where the ”upper right” corner (according to Figure 7) is defined as home. Both experiments show the following fact: the robot only stops in the correct corner, but it also passes the correct corner. This is because, to recognize the sharp corner, which is defined as home, a similar sequence of the corresponding macro-action has to occur. In detail this means, that the robot can not take a look at the currently appearing sensor values, because they are ambiguous. In fact, a sequence or a path of macro-actions must be recorded in order to find the correct corner. This means that a certain level of robot-environment interaction has to be performed, as the MRC has a limited sensory system. A robust discrimination between the two sharp corners can only be based on the anticipated macro-action sequence, which emerge from a specific robot-environment interaction.

In the last experiment of this section the MAM is applied to solve a navigation problem. Figure 15(a) shows the path of a robot through a small

corridor, which is identical to the experiment in Fig. 9. As it was already mentioned in the pervious section, the movement through the corridor is mainly characterized by zigzag movements. But the MAM of this experiment reflects also "experiences" of the robot that small turning angles lead to longer distances before a new avoidance action has to be triggered. In this experiment it was tried to reduce the zigzag movements and bring the robot to straight movements through this corridor using the corresponding MAM in the following way.

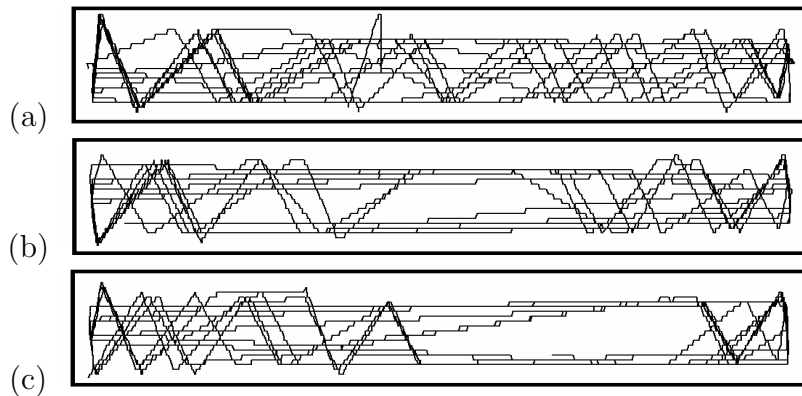


Figure 15: Three robot paths through the small corridor. Figure (a) shows the incisive zigzag movements of the robot. The two other figures show more straight movements through the small corridor - mainly in the middle.

Like in the two experiments before the occurring macro-actions are monitored during the interaction. The control of the robot by the MRC is interrupted, if a macro-action occurs, which is similar to those macro-actions representing the oscillational behavior in the corridor (see Fig. 10). During this interruption the robot is turned 1 step to the right, if the last macro-action is similar to $[-5, 38]$. This is defined according to its second successor node $[1, 111]$ in the MAM (Fig. 9 and 11). Because this second successor node refers to a longer distance before an avoidance behavior is triggered. Accordingly the robot is turned 3 steps to the left, if the last macro-action is similar to $[5, 22]$. Because in this case those 3 turn steps to the left refer to the second successor node $[-3, 265]$ of node $[5, 22]$ (Fig. 9 and 12), which again refers to a longer straight movement distance. After this rotation the control is given back to the MRC.

Figure 15(b) and (c) show the results of two runs of this experiment. The zigzag movements are still there, mainly at the ends of the corridor. This is not surprising, because first of all the specific macro-action must occur to "recognize" the small corridor and to trigger the smoother turns. In detail,

the dynamics of the controller has to be in or near enough to the corresponding attractor. But in the middle of the corridor straight movements are the majority. Whereas in the experiment, where the robot is only controlled by the MRC (Fig. 15(a)), the zigzag movements are predominant.

5 Discussion

In this article a basic robot behavior - moving in a given environment by avoiding scattered objects - was chosen to demonstrate two methods for representing robot-environment interactions by dynamical features of neuro-controllers. The first method, based on first return maps of sensori-motor data, revealed a clear difference in control techniques between a standard feed-forward type of neuro-controller, the BC, and a more sophisticated one, the MRC, which makes extensive use of the recurrent connectivity of its motor neurons. The resulting differences in robot behavior becomes obvious especially in deadlock situations. In particular, the first return map (M-FRM) of the motor data and the combination (SMM) of sensor and motor data appeared to be most instructive. They already indicate the existence of four fundamental types of robot-environment interactions, which are constituted by the neural structure of the MRC.

Based on Table 1 the MAMs can be seen to indicate the temporal sequence of internal dynamical features of the MRC, which become effective during the interaction of the robot with its environment. Basically each macro-action for the MRC relates to a hysteresis interval of sensor inputs followed by a dynamics determined by a global fixed point attractor. The number of time steps of the actual rotation allows to distinguish between hysteresis effects over small and extended input domains. Finally, the sign of a macro-action indicates left and right turns; i.e., left and right hysteresis effects. Therefore one can use the MAMs as a representation of the objects or specific features of the world as they occur during the interaction of the robot with its environment. According to the four different dynamical properties of the MRC this representation can only refer to categories like *obstacle on the left*, *obstacle on the right*, *deadlocks*, and *free space*. This is an essential point, with respect to the landmark-based navigation approach. Landmark types are defined by human designers [12] whereas the MAM approach only takes into account the perception of the robot.

But the experiments in section 4 show that other categories of environmental conditions can be built, if substructures of MAMs are taken into account. The experiments in section 4.3 indicated that MAMs can provide homing and navigation tasks. The homing experiment demonstrated

a method for distinguishing different objects in the environment based on macro-action sequences of a MAM. In addition the implementation of this experiment on a physical mobile robot shows that MAMs are able to handle noisy and discretized environments. Furthermore the navigation experiment introduced a MAM based classification of “small corridors”. In addition, this classification was used to trigger specific actions in such a way that the number of turning actions (or zigzag movements) was decreased. Although in this case the improved action-selection process is “hard-wired”, the trigger conditions and the triggered action were defined according to the underlying MAM. Therefore it is argued, that MAMs can be seen as basic entities for an anticipation process, on which planning tasks can be realized.

It was shown that the MAM approach implies the usage of regularities arising from the sensori-motor interaction of the robot-environment system. That is, because the basic entities of the MAMs, the macro-actions, are the behavior relevant dynamical features of the MRC, which become active in different environmental situation. But it was also shown that this implies, any object categorization (like a specific sharp corner or a small corridor in section 4.3) is a result of a performed interaction. An interaction is always specified by the environment, the robot as an embodied systems, and a controller. Therefore, the MAM approach is conform with approaches, which assume that perception and interaction can not be separated, and the generated behavior influences the perception process ([3], [4], [13], [15], [19]).

As was mentioned in the introduction, this contribution is intended to be a demonstration of method. In fact the presented MAM approach is here based only on the MRC. This controller was evolved for obstacle avoidance only and therefore the MAM is strongly dependent on this system and task. Especially the definition of the mapping (Table 1) between internal dynamical properties of the MRC and its output signals depend on this system. Using our modular neuro-dynamics approach to behavior control of autonomous agents this work is understood as a first step to equip autonomous agents with self-generated internal representations of behavior relevant properties of the external world. Other tasks and other robot platforms will in general require other controllers with different dynamical properties. Hence these systems will need other mappings to generalize temporal sequences of dynamical features during the interaction with the environment.

The experiments have shown that MAMs can become very complex, when representing the full sequences of the interactions. But the homing and the navigation experiments suggest also that a representation of the whole interaction sequence is not necessarily needed to solve a specific problem. In fact, homing and navigation tasks use only specific substructures of the underlying MAM. Therefore future work on MAMs has to include mechanisms

which can focus on or separate relevant substructures of the MAM.

But the question which substructures of a MAM or which internal representation of the environment will be necessary and sufficient for the agent is then of course task dependent. Therefore the agent must be able to evaluate the MAM by itself with respect to a given problem or task. Following [6] we argue, that this evaluation of the agents internal representation has to include the agents anticipated and the actual interaction outcomes. The elements of the MAM are directly related to the external world and accessible by the agent and therefore a promising starting point to endow autonomous agents with self-generated internal representation to improve their behavioral repertoire.

References

- [1] Angeline, P.J., Saunders, G.B. and Pollack J.B. (1994), An Evolutionary Algorithm that Evolves Recurrent Neural Networks, *IEEE Transactions on Neural Networks*, **5**, 54-65.
- [2] Arbib, M.A., Erdi, P. and Szentagothai, J. (1998), *Neural Organization: Structure, Function, and Dynamics*, Cambridge, MA, MIT Press.
- [3] Bajcsy, R. (1988), Active Perception, *Proceedings of the IEEE*, 76: 996 – 1005, 1988.
- [4] Ballard, D.H. (1991), Animate vision, *Artificial Intelligence*, 48:57 – 86.
- [5] Beer, R., (1995) A dynamical systems perspective on agent-environment interaction, *Artificial Intelligence*, 72(1), pp. 173 – 215, 1995.
- [6] Bickhard, M. H., Treveen, L. (1995) *Foundational Issues in Artificial Intelligence and Cognitive Science*, Elsevier Scientific, Amsterdam, 1995.
- [7] Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, MIT Press, Cambridge, MA.
- [8] Harnard, S. (1990), The symbol grounding problem, *Physica D*, 42, pp. 335–346.
- [9] Hülse, M., Pasemann, F. (2001), Dynamical Neural Schmitt Trigger for Robot Control, J. R. Dorronsoro(Ed.): ICANN 2002, LNCS 2415, Springer Verlag Berlin Heidelberg New York, pp. 783–788, 2002.

- [10] Hülse, M.: Implementation of homing behavior based on a recurrent neuro-controller and macro-action maps, MPEG video, <http://www.ais.fraunhofer.de/INDY/aml/X/MRChoming.mpeg>, 12.12.2002.
- [11] Krichmar, J. L., Edelman, G. M., (2002), Machine Psychology: Autonomous Behavior, Perceptual Categorization, and Conditioning in a Brain-Based Device, *Cerebral Cortex*, vol. 12, pp. 818 – 830, 2002.
- [12] Mataric, M.J. (1994), Navigating With a Rat Brain: A Neurobiologically-Inspired Model for Robot Spatial Representation, *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, 282–290, 1994.
- [13] Metta, G., Fitzpatrick, P. (2002), Better Vision Through Manipulation, In: *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Prince, C. G., Demiris, Y., Marom, Y., Kozima, H. and Balkenius, C. (Eds.), Lund University Cognitive Studies, 94, 97 – 104, 2002.
- [14] Michel, O., *Khepera Simulator* Package version 2.0: Freeware mobile robot simulator written at the University of Nice Sophia-Antipolis by Oliver Michel. Downloadable from the World Wide Web at <http://wwwi3s.unice.fr/~om/khep-sim.html>
- [15] Möller, R. (1999), Perception Through Anticipation - A Behavior-Based Approach to Visual Perception. In: *Understanding Representation in the Cognitive Sciences* (A. Riegler; A. von Stein; M. Peschl, eds.), Plenum Press, New York, 1999.
- [16] Mondada, F., Franzi, E., Ienne, P. (1993), Mobile robots miniaturization: a tool for investigation in Control Algorithms. In *Proceedings of ISER' 93*, Kyoto, October 1993.
- [17] Mondada, F., Floreano, D. (1995), Evolution of neural control structures: Some experiments on mobile robots. *Robotics and Autonomous Systems*, 16:183 – 195.
- [18] Nolfi, S., Floreano, D. (2000), *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, Cambridge.
- [19] Nolfi, S., Marocco, D., Active Perception: A Sensorimotor Account of Object Categorization, *Proceedings of the 7th International Conference*

- on Simulation of Adaptive Behavior: From Animals to Animats 7*, 266–271, 2002.
- [20] Pasemann, F. (1993), Dynamics of a single model neuron, *International Journal of Bifurcation and Chaos*, 2, 271–278.
 - [21] Pasemann, F. (1995), Neuromodules: A dynamical systems approach to brain modelling. In Herrmann, H., Pöppel, E. and Wolf, D. (eds.), *Supercomputing in Brain Research - From Tomography to Neural Networks*, Singapore: World Scientific, pp. 331-347.
 - [22] Pasemann, F. (1995), Characteristics of periodic attractors in neural ring networks, *Neural Networks*, 8, 421-429.
 - [23] Pasemann, F., Steinmetz, U., Hülse, M., and Lara, B. (2001), Robot control and the evolution of modular neurodynamics, *Theory in Biosciences*, 120, 311–326.
 - [24] Tani, J. (1998), An Interpretation of the “Self” From the Dynamical Systems Perspective: A Constructivist Approach, *Journal of Consciousness Studies*, 5(5-6), 1998.
 - [25] Tani, J., Sugita, Y. (1999), On the Dynamics of Robot Exploration Learning, *Proc. of 5th European Conf. of Artificial Life (ECAL99)*, 279-288, 1999.
 - [26] Wolpert, D.M., Ghahramani, Z., Flanagan, J.R. (2001), Perspectives and problems in motor learning, *Trends in Cognitive Science*, 5(11): 487-494.