Self-Organization of Symbolic Processes through Interaction with the Physical World

Jun Tani
Sony Computer Science Laboratory Inc.
Takanawa Muse Building, 3-14-13 Higashi-gotanda,
Shinagawa-ku,Tokyo, 141 JAPAN
tani@csl.sony.co.jp

Abstract

We study how an autonomous robot can attain a cognitive process that accounts for its symbolic manipulation of acquired knowledge without generating fatal gaps from the reality. The paper focuses on two essential problems; one is the symbol grounding problem and the other is how the internal symbolic processes can be *situated* with respect to the behavioral contexts. We investigate these problems by applying a dynamical system's approach to the robot navigation problem. Our formulation, based on a forward modeling scheme using recurrent neural learning, shows that the robot is capable of learning grammatical structure hidden in the geometry of the workspace from the local sensory inputs through its navigational experiences. Furthermore, the robot is capable of *mentally simulating* its own action plans using the acquired forward model. Our assertion is that the internal representation obtained is grounded, since it is self-organized solely through interaction with the physical world. We also show that structural stability arises in the interaction between the neural dynamics and the environmental dynamics, which accounts for the *situatedness* of the internal symbolic process.

1 Introduction

The recent successes of behavior-based robotics [Brooks, 1986; Maes, 1991] have led to an underestimation of the necessity of internal representation. The behavior-based robots are characterized by their direct sensory-motor maps, through which they can react rapidly to the dynamical environment. Although the resultant reactive-type behavior of these robots can be quite complex depending on the adopted environment, the emergence of such complex behavior does not necessarily account for all aspects of intelligence. We consider that some intelligent activity should involve deliberative internal computation rather than merely reactive interaction between the environmental and the internal systems. An intelligent robot should be capable of *mentally simulating* its own potential action plans through manipulating the knowledge of its internal model in a flexible way, before choosing a course of action. Such internal computation should maintain certain combinatorial powers especially when the acquired knowledge contains grammatical complexity. We consider that the deliberative thinking paradigm of the traditional AI itself is not misleading. However, the paradigm faces two essential problems. One is the "symbol grounding problem" as Harnad [Harnad, 1990] has discussed, namely "*How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads ?*" The other is how the symbolic process can be *situated* to the current context—that is determined solely from the history of interacting with the environment. This paper discusses our novel approaches to address the above issues, in which we have used a mobile robot as an experimental platform.

Conventionally, the problem of mobile robot navigation has been approached in rather straightforward manner. A global representation formula is employed: a robot builds an environmental map, represented in global coordinates, by gathering geometrical information as it travels [Elfes, 1987; Freyberger *et al.*, 1990]. Although a variety of methodologies has been proposed in this context, potential problems still remain, especially in robot localization. The localization is not always robust enough in the noisy environments of the real-world since there exist gaps between the knowledge of the global map and the information provided by the local sensory inputs.

Kuipers [Kuipers, 1987], Mataric [Mataric, 1992], and others have developed an alternative approach based on landmark detection. In this approach, the robot acquires a graph-type representation of landmark types. This representation is equivalent to a finite state machine (FSM), as a topological modeling of the environment. In navigation, the robot can identify its topological position by anticipating the landmark types in the FSM representation. Although this scheme enables the robot to acquire grammatical knowledge of the obstacle environment by a local representation scheme, its stability is not clear in circumstances where incorrect landmark matching happens to take place. A FSM would halt if fed an illegal symbol. This navigation strategy is susceptible to such a crash if the landmark type is misread.

Although robustness can be enhanced through improving the landmark detection scheme by combining, for example, global positioning (as conducted in ref.[Mataric, 1992]) or other sensor-fusion techniques, it would remain limited as long as the model is represented symbolically.

The above discussion has demonstrated that approaches using global maps or FSM cannot provide representation intrinsic to the robot. In this paper, we focus on the dynamical system's approach [Beer, 1995; Jordan, 1988] as an alternative, with the expectation that its language can be utilized to build an effective representational and computational framework for behavior-based robots. It is known from the theory of symbolic dynamics [Crutchfield, 1989] that some classes of dynamical systems, for example chaos and fractals can provide combinatorial and linguistic power. Therefore, there is the possibility that knowledge, which requires its own grammatical handling, could be represented as being embedded into such intrinsic dynamical functions. Another characteristic, which we can take advantage of, is the phenomenon of entrainment that takes place between different dynamical systems which are coupled together. We will show that the internal symbolic process is naturally *situated* to the system's context by means of entrainment of the internal dynamics from the environment, through its interaction with the physical world.

2 Navigation Problem

We consider how a mobile robot learns to navigate in an unstructured environment under the following conditions:

- The robot cannot access its global position, but it must navigate based on its local sensory (range image) input.
- There are no explicit landmarks accessible to the robot in the adopted workspace.
- No *apriori* knowledge of the workspace geometry is given.

Previously, we have formulated the *skill-based learning* of navigation [Tani and Fukumura, 1994a; 1994b]. This scheme aims to ensure that a robot will acquire skills (a state-action map) for a fixed navigational task, such as homing or cyclic routing, under the supervision of a trainer. The current paper presents the formulation of *model-based learning*. The benefit of this type of learning is that the process of planning with the internal model enables the robot to adapt flexibly to different goal tasks. The specific application, shown later in this paper, is that after the learning process, the robot *mentally simulates* its action plans using the acquired model i.e. it conducts lookahead predictions of future sensory input for arbitrary motor programs.

3 Architecture

The *YAMABICO* mobile robot [Iida and Yuta, 1991] was used as an experimental platform. We briefly review the navigation architecture [Tani and Fukumura, 1994b; 1994a]. *YAMABICO* can obtain the range image by a laser range finder in real-time. The ranges for 24 directions, covering a 160 degree arc in front of the robot, are measured every 150 milli seconds by triangulation. The robot maneuvers by differentiating the rotation velocity of the left and right wheels, and it normally moves with a speed of 0.3 m/s.

In our formulation, maneuvering commands are generated as the output of a composite system consisting of two levels. The control level generates a collision-free, smooth trajectory using a variant of the potential method [Khatib, 1986], while the navigation level directs the control level in a macroscopic sense, responding to the sequential branching that appears in the sensory flows. The navigation level can be adapted through learning; the control level, on the other hand, is fixed.

Firstly, let us describe the control level. The robot can sense the forward range readings of the surrounding environment in robot-centered polar coordinates given by $r_i (1 \leq i \leq N)$. The angular range profile R_i is obtained by smoothing the original range readings through applying an appropriate Gaussian filter. The maneuvering focus of the robot is a local peak (the angular direction of the largest range) in this range profile. The robot proceeds towards a particular potential hill (an open space in the environment) by targeting its peak with a constant control gain. This control scheme is implemented as follows:

$$V_{dif} = k_p \cdot \theta_f \tag{1}$$

where V_{dif} is the differential rotational velocity between the left and right wheels, θ_f is the angular displacement of the focus point from the center, and k_p is a constant gain.

The navigation level focuses on the topological changes in the range profile as the robot moves. As the robot moves through a given workspace, the profile gradually changes until another local peak appears when the robot reaches a branching point. At this moment of branching the navigation level decides whether to transfer the focus to the new local maximum or to remain with the current one. This is the essential point of our architecture. The navigation level functions only at branching point that appears in unconstructed environment. Therefore the navigation decisions are made on the topological trajectory that is determined by the dynamics of collision-free maneuvering applied to the environment.

The navigation level makes decisions for each branch by utilizing the sensory input at that moment. Two types of sensory inputs are used, one is the range image and the other is the local travel distance measured from the previous branch to the current one. The range image is compressed by the vector quantization technique known as the Kohonen network [Kohonen, 1982]. The N-dimensional vector, describing the range profile R at each branching sequential time n, is fed into the network, and the resultant mapping into an output vector p_n of fewer dimensions l is obtained. More details of this application of the Kohonen's net should refer to [Tani and Fukumura, 1994b]. Hereafter, all discussion focuses on the schemes for the navigational level, and we will describe "branching decisions" simply as "motor com-

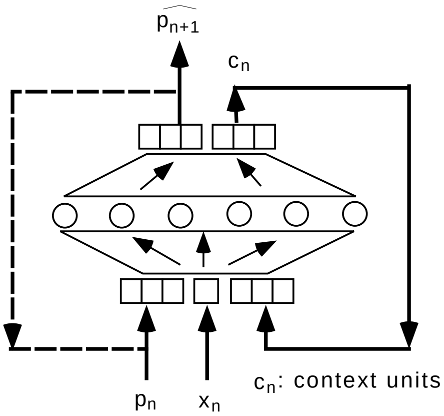

Figure 1: Forward model by RNN architecture.

mands".

In the actual implementation, the robot sometimes drops into a concave dead end from which the robot cannot escape with its current maneuvering scheme. In this terminal point, the robot turns through 180 degrees and receives sensory inputs, then starts again.

4 Model-Based Learning

Our main concern is how a robot can acquire the internal model as an intrinsic function which enables the *mental simulation* of its own actions in the obstacle environment. Here, we attempt to apply the scheme of forward modeling [Jordan, 1988] to the problem.

4.1 Forward modeling

The objective is to build a forward model through which a robot can conduct lookahead prediction of the sensory input sequence (as the distal output) as a result of the given motor program (of the proximal input) in branching sequence. (Hereafter, the term "motor program" denotes a sequence of motor commands.) The objective forward model is embodied using a standard discrete time RNN architecture, as shown in Figure 1. The mapping function of the RNN can be written as;

$$c_{n+1} = f_c(p_n, x_n, c_n, W_c) \quad (2)$$
$$\hat{p_{n+1}} = f_p(p_n, x_n, c_n, W_p)$$

where f_c and f_p are the nonlinear maps from the current branching step to the next branching step, and W_c and W_p denote parameter sets of connective weights. This RNN architecture receives the current sensory input p_n, the current motor command x_n, then outputs the prediction of the next sensory input $\hat{p_{n+1}}$. We employ the idea of the context loop [Elman, 1990] which enables the network to obtain a certain temporal internal representation. (In Figure 1, there is a feedback loop from the context units in the output layer to those in the input layer.) The current context input c_n (a vector) is a copy of the context output in the previous time: by this means the context units remember the previous internal state. The navigation problem is an example of a so-called "hidden state problem" [Lin and Mitchell, 1992]: a given sensory input does not always represent a unique situation/position of the robot. Therefore, the current situation/position is identifiable, not by the current sensory input, but by the memory of the sensory-motor sequence stored during travel. Adequate temporal internal representation of the travel history, by taking advantage of the context loop, can achieve just such a memory structure. The forward model is acquired in the learning phase; the robot travels around the workspace with sampling the sensory-motor sequence in the branching, then the network is trained as off-line by using back-propagation through time algorithm [Rumelhart *et al.*, 1986].

After the learning phase is completed, the robot is operated in the so-called open-loop mode: the robot travels in the workspace by an arbitrary motor program while conducting the one-step lookahead prediction (predicts next sensory input as the result of the current motor command). The RNN predicts the next sensory input $\hat{p_{n+1}}$ by inputting the current sensory input p_n and the current motor command x_n to the network. The RNN, in the beginning of the travel, cannot predict the next sensory input correctly since the initial context value is set randomly. However, the context value can get *situated* as the RNN continues to receive the sensory-motor sequence during the travel, then the RNN begins to predict correctly.

After the robot is *situated* to the environment, the RNN can be switched into the closed-loop mode with stopping the robot at a branch point. Now, a lookahead prediction of an arbitrary length for a given motor program can be made by copying the previous prediction of the sensory input to the current sensory input. (As indicated by a dotted line in Figure 1, the closed-loop for the sensory input is made.) Let us denote the motor program as $x*$. Then the lookahead prediction of the sensory input sequence $\hat{p*}$ can be obtained by recursively applying $x*$ to the RNN mapping function, with using the initial values of context units c_0 and the sensory input p_0 which have been obtained in the open-loop mode.

4.2 Situatedness by entrainment

This sub-section investigates the mechanism of *situatedness* by focusing on the coupling between the internal neural dynamics and the environmental dynamics.

First, we will define the term "attractor" for both of the environmental and the internal dynamics. Let us consider the environmental dynamics F. We consider an infinite length of randomly generated binary sequences (the motor program $x*$) to be fed into the robot. Let $s*$ be the resultant state transitions of the environmental state in the branching sequence. The environmental state s can be represented by the robot's position (including the orientation) upon branching. In the ideal case with no noise in the environment, the infinite travel of the robot forms an invariant set $\underline{s}*$, since the trajectory of the robot is limited to be in a subspace of the entire workspace after an initial transient period. We define this invariant set as the attractor of F with re-

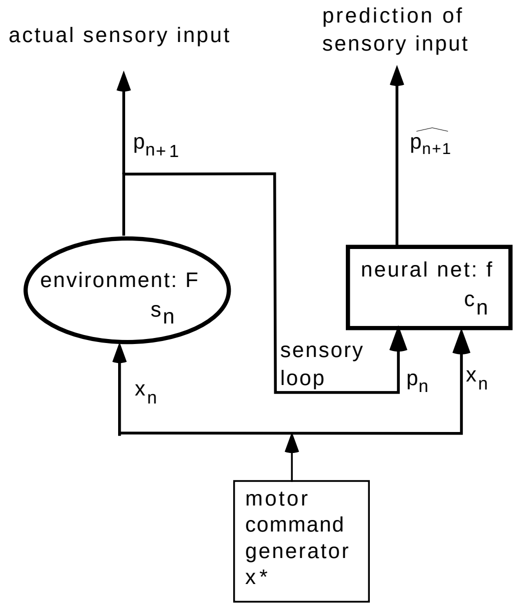

Figure 2: Entrainment of internal dynamics by environment.

spect to the excitatory input $x*$. Also, we define an invariant set $\underline{p}*$ for the sequence of the sensory input which $\underline{s}*$ corresponds to. It is important to note that this attractor is the global attractor, since the robot's travel starting from any position in the workspace results in the same invariant set. For the neural dynamics f, let us consider a lookahead prediction of the RNN with respect to a motor program $x*$ of an infinite length which is randomly generated. This generates an infinite sequence of the transitions of the context $c*$. When this infinite sequence forms an invariant set, this invariant set $\underline{c}*$ is defined as the attractor of f. The sensory sequence which corresponds to $\underline{c}*$ is indicated as $\underline{\hat{p}}*$. Depending on the learning process, the generation of the global attractor is not assured for f. Since the objective of learning is to make the neural dynamics f to emulate the environmental dynamics F by means of the sequence of the sensory input, f in the limit of a learning process satisfies, for an arbitrary motor program $x*$, that:

$$\exists \underline{c}_0,\ \exists \underline{s}_0 \ \Rightarrow\ \underline{\hat{p}}* \, = \, \underline{p}* \tag{3}$$

The idea here is that there is, at least, one attractor for f by which the lookahead prediction of the sensory input can be made correctly, as satisfying (3). Now let us consider the coupling of these two dynamics. In the open-loop mode, the RNN predicts the next sensory inputs $\hat{p_{n+1}}$ using the current sensory inputs p_n while the robot travels following the motor program $x*$. This coupling is schematically shown in Figure 2. In this coupling, it is conjectured that two sequences $p*$ and $\hat{p}*$ converge into the same sequence for all the initial states of s_0 and c_0 if f has been formed as global attractor dynamics. This implies that the internal dynamics, with arbitrary setting of the initial state, always become coherent with the environmental dynamics and predict the sensory inputs correctly, as long as the internal model is embedded in the global attractor dynamics.

This feature of the entrainment of the internal dynamics by the environmental one assures an inherent robustness of the robot's behavior against temporal perturbations. The robot, during its travel, could lose its context if perturbed by noise. The robot, however, can get *situated* again by means of the entrainment as long as it continues to interact with the environment.

5 Experiment

We conducted experiments on the scheme presented above using the mobile robot *YAMABICO*. The robot learns the forward model through trial and error. The robot samples the data of the sensory-motor sequence while it wanders around the adopted workspace for a certain period, then it learns the forward model of the navigation level using the data obtained off-line. After learning, the capability of lookahead predictions is statistically measured in order to examine how the robot learns the internal model. If its knowledge is found to be insufficient, the above process of learning is repeated through sampling more data.

5.1 Learning and look-ahead prediction

Learning was repeated for rounds with increasing number of the sampled data sets. The sampled data set was fed into the RNN for off-line learning for each round, in which the network was re-trained with randomly set weight values. The training of the RNN was conducted for 20,000 steps, which are repeated if the mean square learning error per unit output cannot be decreased below 0.01. The adopted RNN architecture is three-layered having 10, 12 and 9 units for the input, hidden, and output layers respectively. It has four context units. After each round of learning, the test of a given lookahead prediction is conducted for different 10 travels. Each travel starts from an arbitrary free space in the workspace. The robot travels using random branching with the RNN switched in the open-loop mode until the RNN becomes able to predict the next sensory input correctly. The robot is stopped when the prediction error, for all sensory input units, becomes less than 0.15 twice in succession. Then, lookahead prediction is conducted, with the RNN switched in the closed-loop mode, for an arbitrary motor program which comprises seven steps branching. Thereafter, the robot is directed by the motor program in order to test the lookahead prediction. After 10 travels, the mean square prediction error per sensory input unit (MSPE) is calculated.

In the first round of the learning, the robot sampled 49 input data and learned them. In the test, it took long steps (often more than 15 steps) until the RNN in the open loop mode supplied good predictions. In the ensuing lookahead predictions in the closed loop mode, the RNN could hardly predict more than three steps ahead. It seemed that the RNN learned only particular instances of the sampled sequences but not in a more general way. In the second round with learning 102 input data, the steps to capture the context were shortened, and the lookahead prediction often went smoothly for several steps. However, once the prediction failed in the

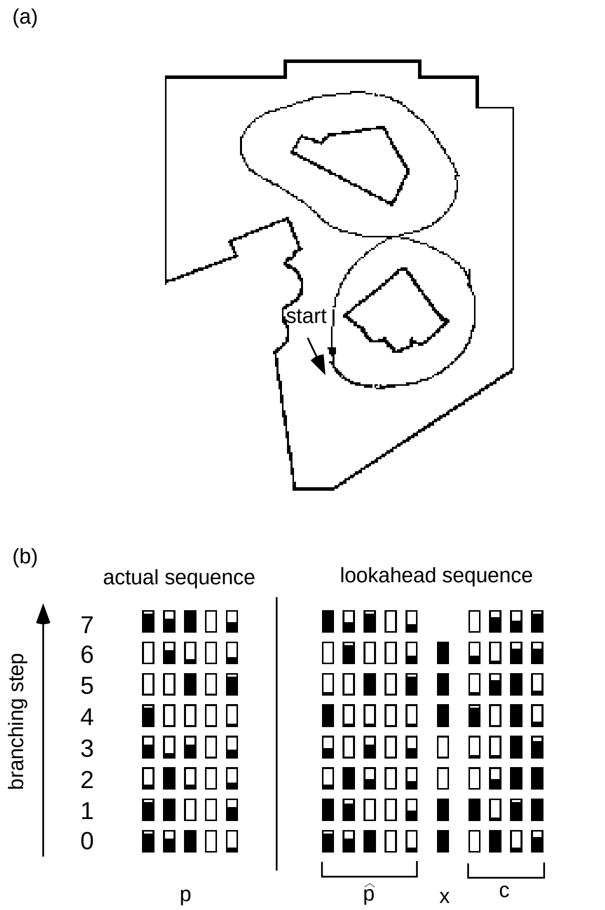

Figure 3: Lookahead prediction for a given motor program.

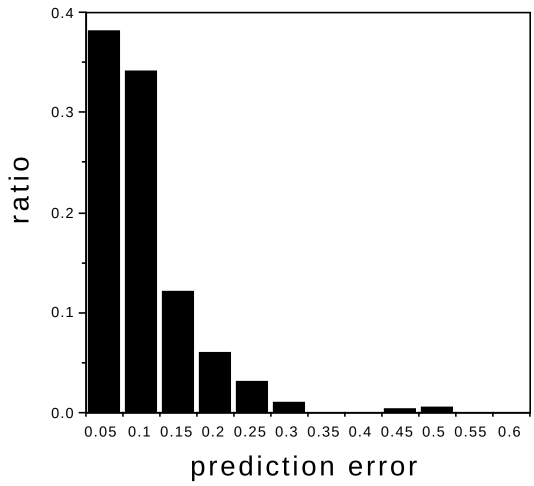

Figure 4: Distribution of the prediction error.

middle of the sequence, it continued to fail for subsequent steps. In the third round with learning 193 input data, it was observed that the context could be recovered within several steps, and also that lookahead predictions became accurate except in cases with certain noise effects. Since the RNN could predict correctly for sequences it had never learned exactly, it can be said that the RNN succeeded in extracting the necessary rules in the form of generalized ones. An example of the comparison between a lookahead prediction and its sensory sequence during travel is shown in Figure 3. In (a) an arrow denotes the branching point where the robot conducted a lookahead prediction of a motor program given by 1100111. The robot, after conducting the predictions, traveled following the motor program, generating the trajectory of a "figure of eight", as shown. In (b) the left side shows the sensory input sequence, while the right side shows those of the look-ahead, the motor program and its context values. The values are indicated by the bar heights. It can be seen that the look-ahead for the sensory inputs agrees very well with the actual values. Figure 4 shows the distribution of the prediction error for a single unit in the third round. It is shown that the fraction of "good" predictions with an error of less than 0.1 is more than 70 percent. Since the robot could capture the context and then achieve good lookahead prediction regardless of the initial setting of the position and the context values, it is assumed that the robot succeeded in learning the forward model as embedded into the global attractor dynamics, through trial and error.

In order to confirm the formation of the global attractor in the experiment, we conducted the phase space analysis for the internal dynamics of the RNN. The RNN, switched to the closed loop mode, was activated for two thousand forward steps using input sequences of random motor commands. The phase diagram was plotted as a two-dimensional projection using the activation state of two context units, excluding 100 points from the initial transient steps. Fig. 5(a) shows the resulting phase diagram, while (b) shows an enlargement of part of (a) in which a one-dimensional structure is seen. We repeated this several times with different initial values of the internal states, and found that they all resulted in the same attractor structure. It confirmed that the internal dynamics are self-organized in the form of the global attractor dynamics. Although any theory has not been established to explain the creation of low-dimensional global attractor in the recurrent neural learning, its tendency is suggested in other numerical experiments [Pollack, 1991; Tani and Fukumura, in press].

We have stated that the global attractor provides an inherent robustness for context dependent navigation as a natural consequence of coupling between the internal and the environmental dynamical systems. The following experiment demonstrates an example of autorecovery from temporal perturbation. The robot traveled in the workspace while predicting the next sensory inputs with the RNN switched to the open-loop mode. During this travel, an additional obstacle was introduced. The upper part of Figure 6 shows the trajectory of the robot's travel; the lower part shows the comparison of the actual sensory inputs and corresponding one-step lookahead prediction. The branching sequence number is indexed beside the trajectory; this number corresponds to the prediction sequence in the lower part

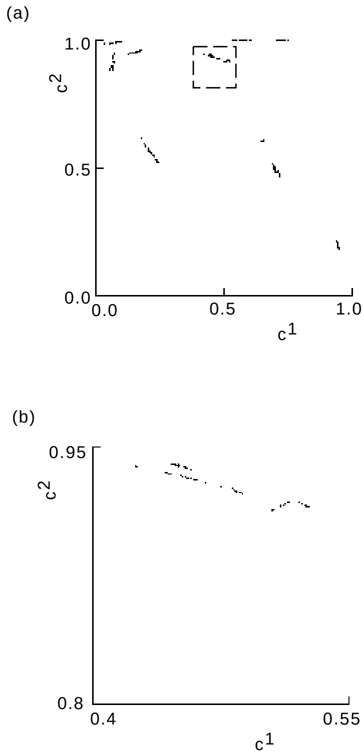

Figure 5: Attractor observed in the internal dynamics.

of figure. The prediction starts to be incorrect once the robot passes the second branching point, as it encounters the unexpected obstacle. The robot, however, continues to travel and meanwhile the obstacle is removed. After the sixth branching point, as the lost context is recovered by means of the regular sensory feed, the prediction returns to the correct evaluation. It is noted that the values of the context units in this branch are almost the same as those of the first branch. This shows that the robot recognized its returning to the same branching point by capturing the context of the travel again.

6 Symbol Grounding Process

A primitive conceptualization of the symbol grounding process is conjectured as the result of our experiments. Figure 7 illustrates the concept. As the robot travels around the workspace, clusters of the sensory inputs are collected in the sensory space arising from its branching sequences. Meanwhile the dynamical mapping is self-organized in the internal state space such that it accounts for the transitions among the clusters of the collected sensory inputs. If different symbols are assigned to each cluster of sensory inputs, the *mental simulation* process carried out by the internal dynamics might be equivalent to the symbolic process of manipulating the symbols. Here, our primitive symbols are not in the arbitrary shape of usual symbol tokens[1], but in the nonarbitrary shape based on the physical interaction between

[1]The discussion inherits Harnad's [Harnad, 1990] claim: Symbol manipulation would be governed not just by the arbitrary shapes of the symbol tokens, but by the nonarbitrary

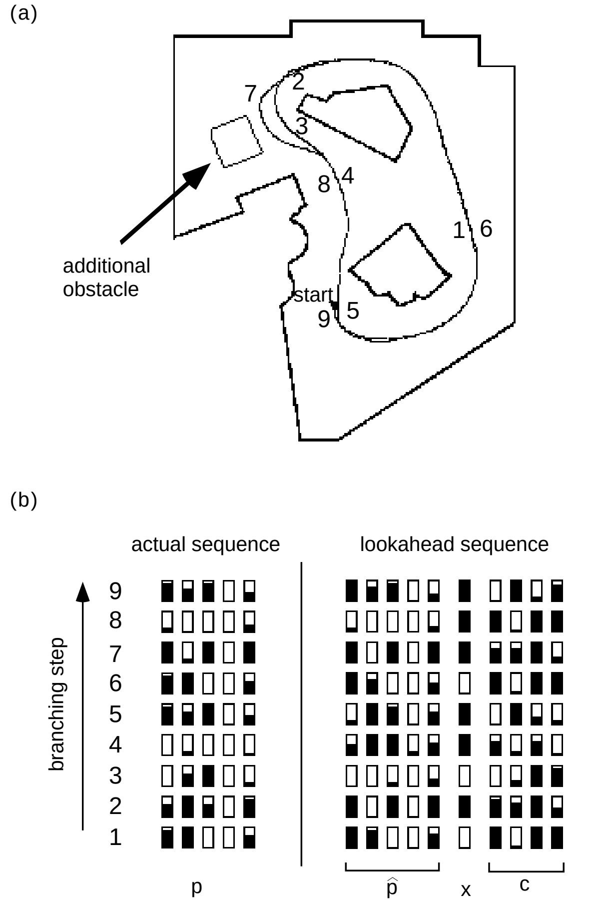

Figure 6: Auto-recovery from an addition of an obstacle.

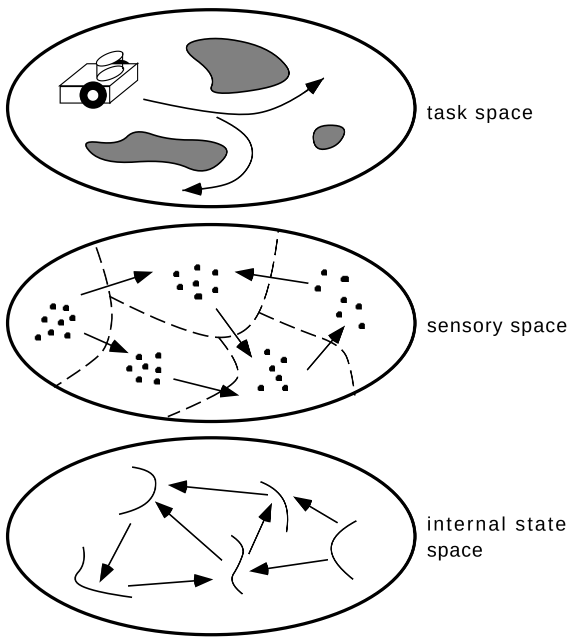

Figure 7: The symbol grounding process.

the robot and the environment.

One might consider that such symbolic processes can be represented in the form of a FSM more easily. We, however, consider that the internal representations by a FSM are still "parasitic" since symbols are manipulated into an arbitrary shape regardless of their meaning in the physical world. A crucial gap exists between the actual physical systems defined in the metric space and their representation in the non-metric space, which makes the discussion of the structural stability of the whole system difficult. In contrast to this state of affairs, the representation in our scheme can be said to be intrinsic to the system since it is embedded in attractor dynamics which share the same metric space with the physical environment. In this meaning, the structural stability arises in the interaction between the internal and environmental systems, which accounts for the *situatedness* of the internal symbolic process.

7 Conclusion

This paper has described how symbolic processes are self-organized in the navigational learning of a mobile robot. Our study, based on a dynamical system's approach, has shown that the forward modeling scheme based on RNN learning is capable of extracting grammatical structure hidden in the geometry of the workspace from navigational experience. The robot was capable of *mentally simulating* its own actions using the acquired forward model. We have shown that such mental process by the RNN can naturally be *situated* with respect to the behavioral contexts, provided that the forward model learned is that embedded on the global attractor. Finally it is concluded that the dynamical system's approach enables the robot to construct its symbolic process as grounded to the physical world.

References

[Beer, 1995] R.D. Beer. A Dynamical Systems Perspective on Agent-Environment Interaction. *Artificial Intelligence*, 72(1):173–215, 1995.

[Brooks, 1986] R. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Trans. Robotics and Automation*, RA-2:14–23, 1986.

[Crutchfield, 1989] J.P. Crutchfield. Inferring statistical complexity. *Phys Rev Lett*, 63:105–108, 1989.

[Elfes, 1987] A. Elfes. Sonar-based Real-world Mapping and Navigation. *IEEE Journal of Robotics and Automation*, 3:249–265, 1987.

[Elman, 1990] J.L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[Freyberger *et al.*, 1990] F. Freyberger, P. Kampman, and G. Schmidt. Constructing Maps for Indoor Navigation of a Mobile Robot by Using an Active 3D Range Imaging Device. In *proc. of the IEEE International Workshop on Intelligent Robots and Systems (IROS'90)*, 1990.

shapes of the icons and category invariants in which they are grounded.

[Harnad, 1990] S. Harnad. The Symbol Grounding Problem. *Physica D*, 42:335–346, 1990.

[Iida and Yuta, 1991] S. Iida and S. Yuta. Vechicle Command System and Trajectory Control for Autonomous Mobile Robots. In *Proc. of the IEEE/RSJ Int. Workshop on Intelligent Robots and Systems '91*, pages 212–217, 1991.

[Jordan, 1988] M.I. Jordan. Indeterminate Motor Skill Learning Problems. In M. Jeannerod, editor, *Attention and Performances, XIII.* MIT Press, Cambridge, MA, 1988.

[Khatib, 1986] O. Khatib. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[Kohonen, 1982] T. Kohonen. Self-Organized Formation of Topographically Correct Feature Maps. *Biological Cybernetics*, 43:59–69, 1982.

[Kuipers, 1987] B. Kuipers. A Qualitative Approach to Robot Exploration and Map Learning. In *AAAI Workshop Spatial Reasoning and Multi-Sensor Fusion (Chicago)*, 1987.

[Lin and Mitchell, 1992] Long-Ji Lin and T.M. Mitchell. Reinforcement Learning with Hidden States. In *proc. of the Second International Conference on Simulation of Adaptive Behavior*, 1992.

[Maes, 1991] P. Maes. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back.* MIT Press, Cambridge, MA, 1991.

[Mataric, 1992] M. Mataric. Integration of Representation into Goal-driven Behavior-based Robot. *IEEE Trans. Robotics and Automation*, 8:304–312, 1992.

[Pollack, 1991] J.B. Pollack. The Induction of Dynamical Recognizers. *Machine Learning*, 7:227–252, 1991.

[Rumelhart *et al.*, 1986] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. Mclelland, editors, *Parallel Distributed Processing.* MIT Press, Cambridge, MA, 1986.

[Tani and Fukumura, 1994a] J. Tani and N. Fukumura. Embedding Task-Based Behavior into Internal Sensory-Based Attractor Dynamics in Navigation of a Mobile Robot. In *proc. of the IEEE Int. Conf. of Intelligent Robots and Systems 94' (IROS94)*, pages 886–893, 1994.

[Tani and Fukumura, 1994b] J. Tani and N. Fukumura. Learning Goal-directed Sensory-based Navigation of a Mobile Robot. *Neural Networks*, 7(3):553–563, 1994.

[Tani and Fukumura, in press] J. Tani and N. Fukumura. Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning. *Biological Cybernetics*, in press.