研究会報告

Dynamical Systems Approach in Learnable Autonomous Robots

Jun Tani
Sony Computer Science Laboratory Inc.
Takanawa Muse Building, 3-14-13 Higashi-gotanda,
Shinagawa-ku,Tokyo, 141 JAPAN
email: tani@csl.sony.co.jp
Fax +81-3-5448-4273
Tel +81-3-5448-4380

March 23, 1995

Abstract

This paper studies the essential dynamical structure that arises in two different classes of learning of the sensory-based navigation, namely *skill-based learning* and *model-based learning*. In *skill-based learning* a robot learns navigational skills for a fixed navigational task such as *homing*, while in *model-based learning* a robot learns a model of the environment, then conducts planning on the model to reach an arbitrary goal. We formulated that the former is achieved by learning the state-action map, and the latter does by learning the forward model of the environment, using recurrent neural learning scheme. The analysis of the dynamical structure from the coupling of the internal neural dynamics and the environment showed that generation of the global attractor is crucial for both learning cases. Experiments were conducted using a mobile robot with a laser range sensor, which verified our assertions in a simple obstacle environment.

1 Introduction

Recently, many have discussed how the knowledge should be represented internally for a mobile robot that navigates based on its local sensory inputs. Conventionally, the navigation problem 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 [2]. 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. The problem to consider is how the task knowledge can be represented as intrinsic [3] to the robot, and how such representations can be obtained through its behavioral experiences.

Others [7, 13] 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. This scheme enables the robot to acquire the internal model of the obstacle environment by a local representation scheme. It is, however, considered that the representations by the FSM are still "parasitic" since symbols manipulated in the FSM are in the 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.

This paper addresses the above problems by using the dynamical system's approach [1, 4, 10], expecting that this approach would provide other effective representational. The approach focuses on the fundamental dynamical structure that arises from coupling the internal and the environmental dynamics [1]. Here, the objective of learning is to adapt the internal dynamical function such that the resultant dynamical structure might generate the desired system behavior. The system's performance becomes structurally stable if the dynamical structure maintains a sufficiently large basin of attraction against possible perturbations. The most advantage of this approach is that we are able to conduct structural analyses of the system by fact that the internal representations, embedded into attractor dynamics, share the same metric space with the physical environment.

We investigate two classes of task learning, namely *skill-based learning* and *model-based learning.* The *skill-based learning* aims to ensure that a robot will acquire skills (represented as a state-action map) for a fixed navigational task, such as homing or cyclic routing, under the supervision of a trainer. In the *model-based learning* the robot learns the internal model of the environment rather than the direct state-action map so that the robot may adapt flexibly to different goal tasks. An important difference between two is that, in the former approach, the action is determined in a reactive way just by looking up the map while it is determined in a deliberative way through *mental simulation* of the model in the latter approach. We study how the state-action map or the environmental model can be represented by means of neural dynamical functions. Then we will explicate the conditions that each learning scheme should satisfy from the view point of dynamical systems.

2 Navigation Architecture

We review the navigation architecture [11] which is applied to the *YAMABICO* mobile robot [14]. *YAMABICO* can obtain the range image, covering a 160 degree arc in front of the robot, by a laser range finder in real-time.

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 field method [5]—i.e. the robot simply proceeds towards a particular potential hill in the range profile (direction toward an open space). 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 peak or to remain with the current one. The navigation level functions only at the branching point that appears in unconstructed environment. Hereafter, our discussions focus on how to determine the branching sequences.

3 Skill-Based Learning

The objective of *skill-based* learning is that the robot learns a fixed navigational task on the topological trajectory comprising branch points. We consider two specific tasks, namely homing and cyclic routing as examples. In the homing task, the robot has to travel back to a fixed branch point starting from an arbitrary position in the workspace. In the cyclic routing task, the robot have to travel into a fixed cyclic loop comprising branch points with starting from an arbitrary position.

3.1 Learning state-action map

The neural adaptation schemes are applied to the navigation level so that it can generate an adequate state-action map for a given task. Although some might consider that such map can be represented by using a layered feed-forward network with the inputs of the sensory image and the outputs of the motor command, this is not always true. The local sensory input does not always correspond uniquely to the true state of the robot (the sensory inputs could be the same for different robot positions). Therefore, there exists an ambiguity in determining the motor command solely from sensory inputs. This is a typical example of so-called non-Markovian problems which have been discussed by Lin and Mitchell [6]. In order to solve this ambiguity, a representation of contexts which are memories of past sensory sequences is required. For this purpose, a recurrent neural network (RNN) [4, 8] was employed since its recurrent context states could represent the memory of past sequences. The employed neural architecture is shown in Figure. 1. The sensory input p_n and the context units c_n determine the appropriate motor command x_{n+1}. The motor command x_n takes a binary value of 0 (staying at the current branch) or 1 (a transit to a new branch). The RNN learning of sensory-motor (p_n,x_{n+1}) sequences, sampled through the supervised

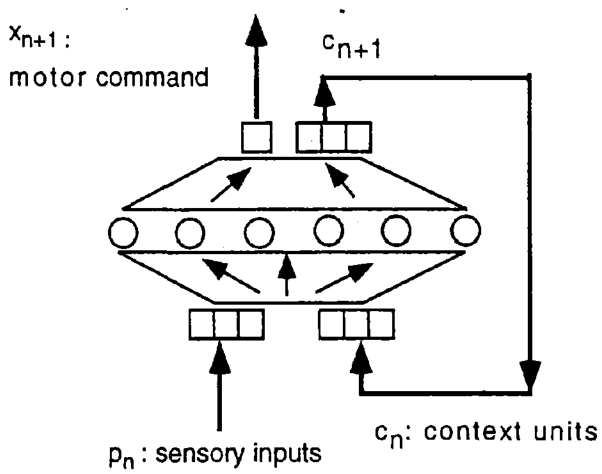

Figure 1: Neural architecture for skill-based learning.

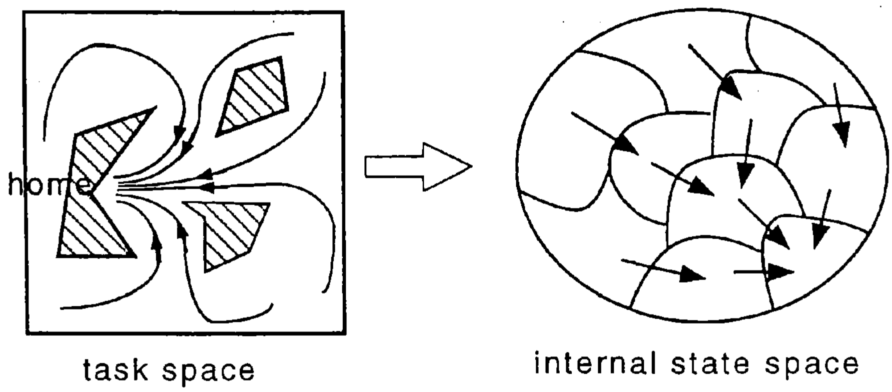

Figure 2: The desired trajectories in the task space and its mapping to the internal state space.

training, can build the desired state-action map by self-organizing adequate internal representation in time.

3.2 Embedding problem

The objective of the neural learning is to embed a task into certain global attractor dynamics which are generated from the coupling of the internal neural function and the environment. Figure 2 illustrates this idea. We define the internal state of the robot by the state of the RNN. The internal dynamics, which are coupled with the environmental dynamics through the sensory-motor loop, evolve as the robot travels in the task space. We assume that the desired vector field in the task space forms a global attractor, such as a fixed point for a homing task or limit cycling for a cyclic routing task. All that the robot has to do is to follow this vector flow by means of its internal state-action map. This requires a condition: the vector field in the internal state space should be self-organized as being topologically equivalent to that in the task space in order that the internal state determine the action (motor command) uniquely. This is the embedding problem from the task space to the internal state space, and RNN learning can attain this, using various training trajectories. This analysis conjectured that the trajectories in the task space can always converge into the desired one as long as the task is embedded into the global attractor in the internal state space.

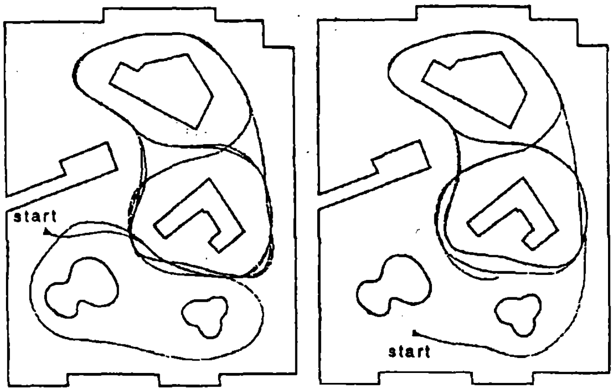

Figure 3: Trace of test travels for cyclic routing.

3.3 Experiment

An experiment of learning a cyclic routing task is presented. The assigned task is to repeat looping of a figure of '8' and '0' in sequence. In the training the robot moved by collision-free control, and branching of the navigation level was taught by the trainer. The employed RNN consists of three input units, eight hidden layer units, two context units and one output unit. The trainer guided the robot back to the target loop from arbitrary selected starting position. The travel of the training was repeated until the robot is assured of being capable of achieving the given task when started from an arbitrary position.

It was found that robot could achieve the task in a stable way after 10 times repetitions of the training. In the test travel, the robot was started from arbitrary initial positions, with setting the initial values of context units as random. Figure 3 shows examples of the test travels. The result appeared that the robot can converge to the desired loop from any position in the workspace. Its convergence, however, takes a certain period depending on the case. The RNN initially cannot output normally until the context units catch up the context. As the robot moves around the workspace, encountering a sequence of known sensory input, the orbit in the internal state space starts to converge from the initial transient one. Noises affects the navigation performance remarkably. When miscellaneous noise such as mechanical, sensory and radio noise is present, the branching sometime become unstable. Thus, even after convergence, the robot could by chance go out the loop, perturbed by such noise. However, it always comes back to the loop after while. Although the actual navigation contains stochastic property in its local decisions, it can be said that the structure of convergence is quite stable in terms of the global attractor dynamics generated.

4 Model-Based Learning

In this learning, the 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 [4] 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 4. The mapping function of the RNN can be written as;

$$\begin{aligned} c_{n+1} &= f_c(p_n, x_n, c_n, W_c) \\ \hat{p_{n+1}} &= f_p(p_n, x_n, c_n, W_p) \end{aligned} \tag{1}$$

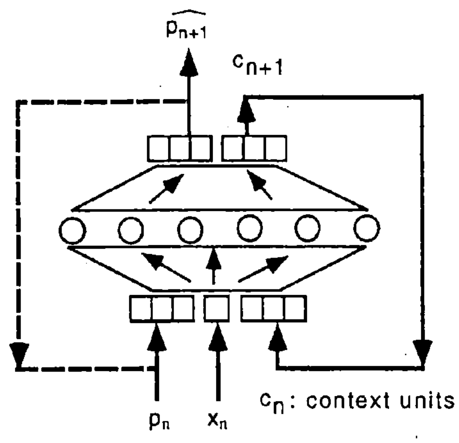

Figure 4: Forward model by RNN architecture.

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. 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 [9].

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 4, 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 Dynamical mechanism of *situatedness*

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 respect 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

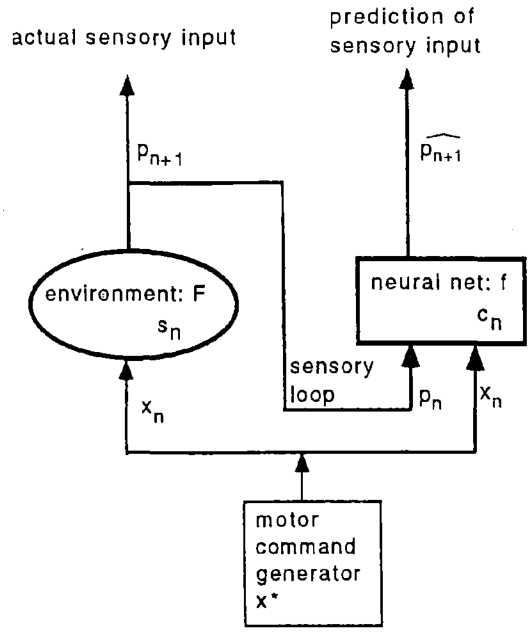

Figure 5: Entrainment of internal dynamics by environment.

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 $\hat{\underline{p}}*$. 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:

$$\exists \underline{c}_0, \exists \underline{s}_0 \Rightarrow \hat{\underline{p}}* = \underline{p}* \tag{2}$$

The conclusion 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 (2). 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 5. 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 harmonized 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.

4.3 Experiment

4.4 Learning and lookahead prediction

We conducted experiments on the scheme using *YAMABICO*. 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. The adopted RNN architecture

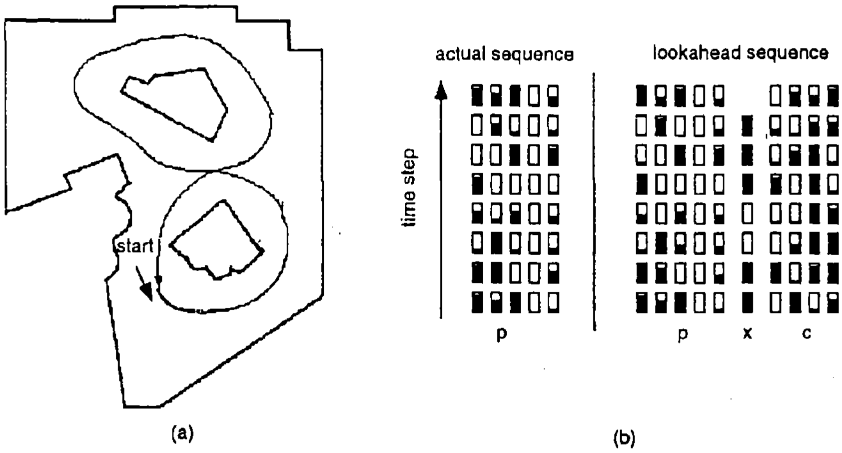

Figure 6: Lookahead prediction for a given motor program.

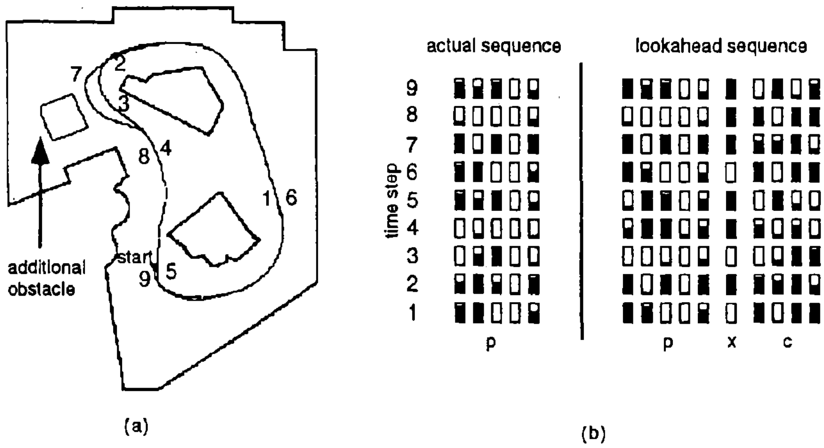

Figure 7: Auto-recovery from an addition of an obstacle.

is three-layered having 10, 12 and 9 units for the input, hidden, and output layers respectively. It has four context units. After learning 193 sampled data, it was observed that lookahead predictions became accurate except in cases with certain noise effects.

An example of a lookahead prediction test is shown in Figure 6. In (a) an arrow denotes the branching point where the robot conducted a lookahead prediction of a motor program given by 1100111 with switched to the closed-loop mode (after get *situated*). 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 lookahead, the motor program and its context values. The values are indicated by the bar heights. It can be seen that the lookahead for the sensory inputs agrees very well with the actual values.

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 shown in Figure 7 demonstrates an example of auto-recovery 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 7 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

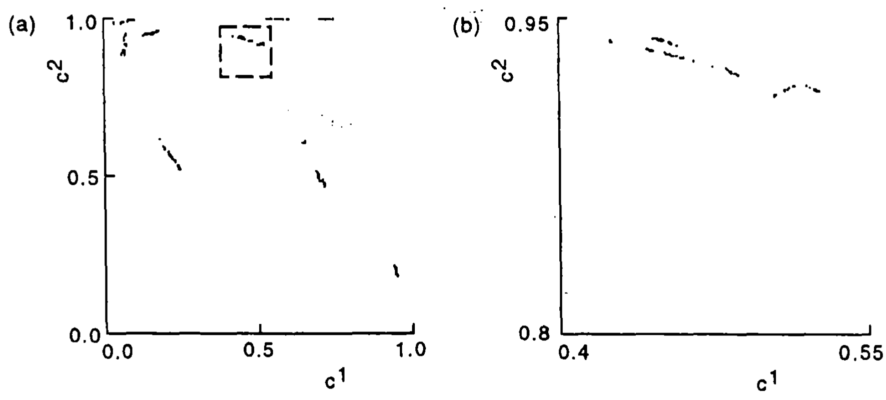

Figure 8: Attractor observed in the internal dynamics.

part 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.

The repeated experiments showed that the mechanism of the auto-recovery is general. This implies that the learning of the RNN might have created the global attractor. In order to confirm this, we analyzed the dynamical structure self-organized in 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. 8(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 of learning simple grammatical descriptions [8, 12].

5 Discussion and Conclusion

We have investigated the dynamical structure that arises in the coupling of the internal neural function and the environment, and have shown that the generation of the global attractor is essential for the embedding as well as the entrainment discussed in the *skill-based learning* and *model-based learning* respectively.

Our formulations have also clarified essential differences in their dynamical compositions. In the *skill-based* scheme, the dynamical structure arises only from the coupling of the internal and the environmental dynamics when the robot actually travels. In the *model-based* scheme, attractor dynamics exists in the internal dynamics even when it is decoupled from the environmental one. This decoupling allows the robot to have a "symbolic process" which accounts for its cognitive activities of *mental simulation* or planning. This symbolic process can be grounded to the physical world by means of *situatedness* when coupled with the environmental dynamics.

References

[1] R.D. Beer. A Dynamical System's Perspective on Agent-Environment Interaction. *Artificial Intelligence*, in press.

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

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

[4] M.I. Jordan and D. E. Rumelhart. Forward models: Supervised Learning with a Distal Teacher. *Cognitive Science*, Vol. 16, pp. 307–354, 1992.

[5] O. Khatib. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90–98, 1986.

[6] 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.

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

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

[9] 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.

[10] L. Steels. Mathematical Analysis of Behavior Systems. In *proc. of From Perception TO Action*, 1994.

[11] J. Tani and N. Fukumura. Learning Goal-directed Sensory-based Navigation of a Mobile Robot. *Neural Networks*, Vol. 7, No. 3, pp. 553–563, 1994.

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

[13] B. M. Yamauchi and R. D. Beer. Spatial Learning for Navigation in Dynamic Environment. *IEEE Trans. on Systems, Man, and Cybernetics*, in press.

[14] S. Yuta and J. Iijima. State Information Panel for Inter-Processor Communication in an Autonomous Mobile Robot Controller. In *proc. of the IEEE International Workshop on Intelligent Robots and Systems (IROS'90)*, 1990.