Pergamon

Neural Networks, Vol. 10, No. 1, pp. 153–159, 1997
Copyright © 1996 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0893–6080/97 $17.00+.00

PII: S0893-6080(96)00066-4

*CONTRIBUTED ARTICLE*

# Self-organizing Internal Representation in Learning of Navigation: A Physical Experiment by the Mobile Robot YAMABICO

JUN TANI AND NAOHIRO FUKUMURA

Sony Computer Science Laboratory Inc.

(*Received* 24 *January* 1994; *revised and accepted* 20 *May* 1996

**Abstract**—*This paper discusses a novel scheme for sensory-based navigation of a mobile robot. In our previous work (Tani & Fukumura, 1994, Neural Networks, 7(3), 553–563), we formulated the problem of goal-directed navigation as an embedding problem of dynamical systems: desired trajectories in a task space should be embedded in an adequate sensory-based internal state space so that a unique mapping from the internal state space to the motor command could be established. In the current formulation a recurrent neural network is employed, which shows that an adequate internal state space can be self-organized, through supervised training with sensorimotor sequences. The experiment was conducted using a real mobile robot equipped with a laser range sensor, demonstrating the validity of the presented scheme by working in a noisy real-world environment. Copyright © 1996 Elsevier Science Ltd.*

**Keywords**—Neural networks, Sensory motor system, Mobile robot, Navigation Learning.

## 1. INTRODUCTION

Conventionally, the scheme of sensory-based navigation has been formulated with the assumption of a global representation of the world. Given a detailed map of the workspace described in the global coordinate system as *a priori* knowledge, the robot navigates to the specified goal by following it (Elfes, 1987; Durrant-Whyte & Leonard, 1989; Asada, 1990; Freyberger *et al.*, 1990).

However, in the situation where the robot itself learns to acquire navigational knowledge through its behavioral experiences, it might be more natural if the knowledge could be represented in a localized form, from the view point of the robot itself. The problem to consider is how the task could be comprehended internally by the robot through association with the temporal inflow of the sensory information.

In our previous study (Tani & Fukumura, 1994) we proposed a vector field approach, in which the internal state of the robot, corresponding to past sequences of sensory inputs, was assumed. The manoeuvring direction (motor command) at each time could be determined as a unique mapping from this internal state, which, however, imposed a condition: the internal state space should be defined such that the vector flow in the task space (the desired trajectories) can be successfully embedded in that.

Here, how to construct internal state space is an essential problem. In the previous formulation, sensory regression of a fixed length was taken as the internal state, and the actual mapping from this regression vector to the output (motor command) was realized by means of a time-delay neural network (TDNN). A set of simulations showed that learning tasks by supervised training, such as homing and sequential routing of limited complexities, was successfully achieved.

The formulation, however, is not yet general because it predetermined the essential structure of the internal state space (as a fixed length of sensory regression), which caused inflexibility in adaptation for more complex tasks. In this paper, we consider a more general scheme, aiming that an adequate state space can be self-organized without any assumptions of *a priori* temporal structure. We also show that such

Acknowledgements: The authors would like to thank S. Yuta and M. Tokoro for useful discussion on this study.

Requests for reprints should be sent to Jun Tani, Sony Computer Science Laboratory, Inc., Takanawa Muse Building, 3–14–13 Higashi-gotanda, Shinagawa-ku, Tokyo, 141 Japan; Tel: 3–5448–4830; Fax: 3–5448–4273; e-mail: tani@csl.sony.co.jp

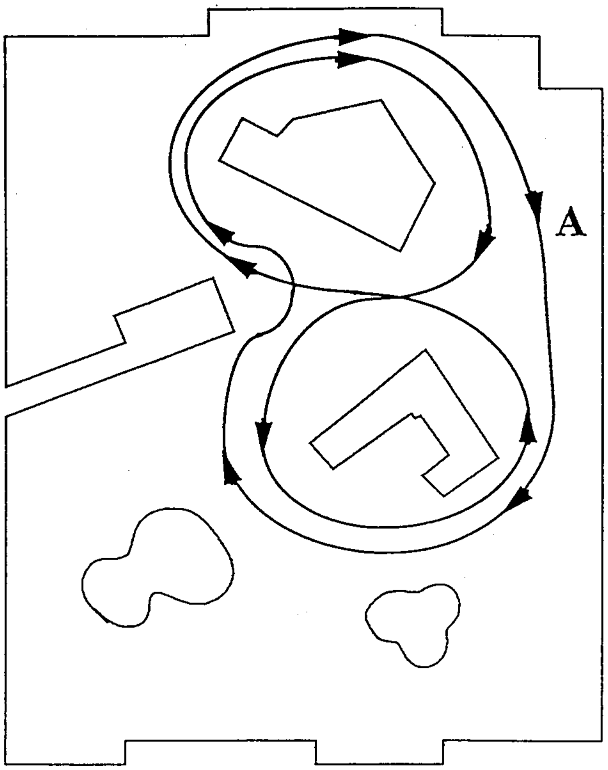



FIGURE 1. An example of the navigation task of sequential routing. The robot has to loop around figures of "8" and "0" in sequence, with *A* as the branching point.

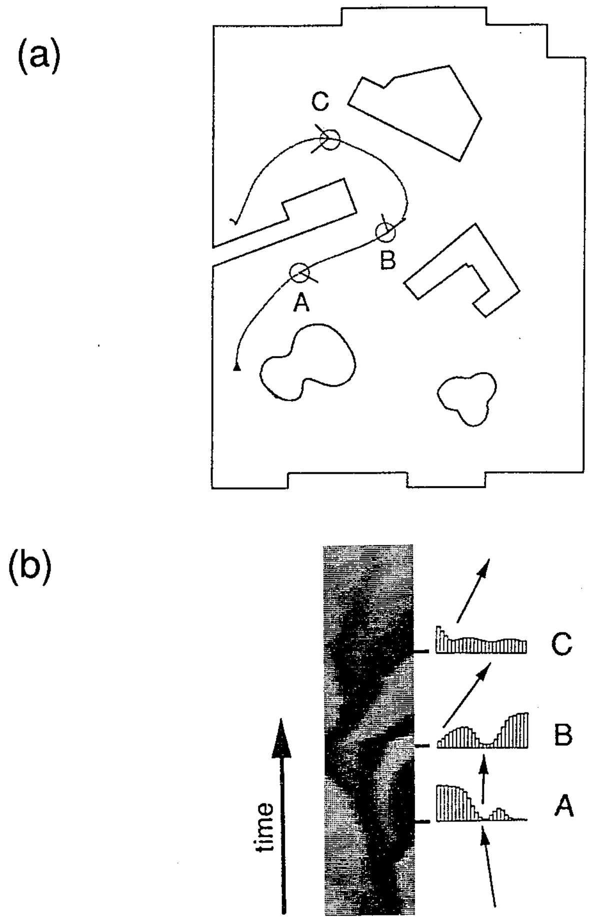



FIGURE 2. An example of the trajectory and corresponding bifurcation sequence in the sensory data flow. (a) The trajectory contains three bifurcation points (A), (B) and (C). (b) In the spatio-temporal sensory data, the brighter area indicates that its range is closer. The exact range profile at each bifurcating point is shown at the side. The arrows indicate the branching decisions of "transit" to a new branch or "stay" at the current one.

a scheme is robust enough for noisy real-world environments, by conducting a physical experiment with the mobile robot YAMABICO (Yuta, 1990).

## 2. HIDDEN STATE PROBLEM

Local sensing at each moment gives only partial information of the true world state that must be identified for determination of optimal action. This is well known as the *hidden state problem* (Lin & Mitchell, 1992). To deal with this problem, we utilize historical information in sensing, which can uncover hidden features.

Figure 1 shows an example of the navigation task, (which is adopted for the physical experiment in a later section). The task is for the robot to repeat looping of a figure of "8" and "0" in sequence. The task is not trivial because at the branching position *A* the robot has to decide whether to go "8" or "0" depending on its memory of the last sequence. A TDNN architecture is difficult to use in this situation because we have to decide the optimal regression form by *a priori* such that it can attain the desired memory structure enabling this sequential task.

Our new attempt, here, is to utilize a recurrent neural network (RNN) with feedback loops of context units (Jordan, 1986; Rumelhart *et al.*, 1986; Pineda, 1987), aiming for the required temporal structure that is self-organized, utilizing its internal state space during training.

## 3. NAVIGATION ARCHITECTURE

This section reviews the proposed navigation architecture (Tani & Fukumura, 1994), which consists of two levels: a control level and a navigation level, and also shows the current implementation of RNN into this.

We place a constraint on the trajectories to be generated: each trajectory should be a smooth one, avoiding collisions with obstacles. This condition reduces degrees of freedom in the navigation and simplifies the problem dramatically. The control level employs a scheme similar to the potential method (Khatib, 1986) in order to realize this constraint. Incorporating this scheme, the task of the navigation level is simplified to decisions of branching directions

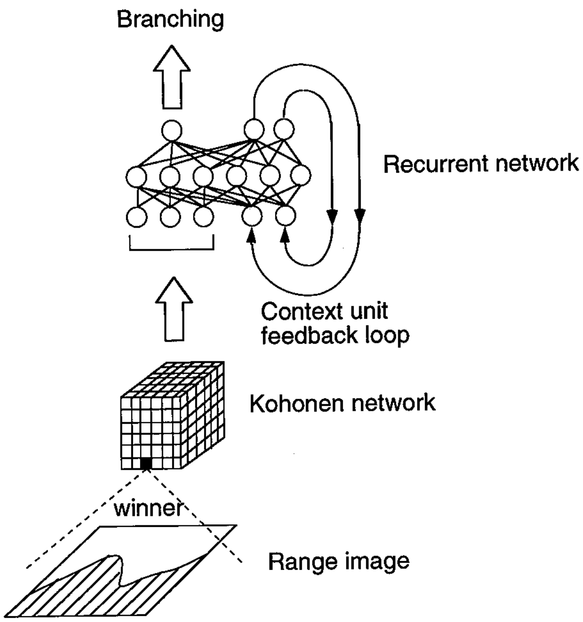



**FIGURE 3. A composite neural system: Kohonen net and RNN. The output by the Kohonen net (the winner position in 3D output space) is fed into the RNN.**

at finite points in the task space. An approximation of the desired vector field (the direction of desired movement) in the task coordinate system can be reconstructed only by acquiring *the topological trajectory*, consisting of those representative vectors at branching points.

Figure 2 shows an example of the robot's travel measured in the later described experimental workspace. The upper figure is its test trajectory and the lower figure its corresponding sensory flow. The time history of the range image, covering 160 degrees of the frontal side, is expressed as upward in the time direction by the shaded sequence. The darker part denotes far distance to obstacles in its direction and the brighter part its close distance. In this test travel, *A, B* and *C* on the trajectory become the branching points. We can see that those correspond to bifurcations in the range image in the lower figure. The exact range image profile at each bifurcation point is also shown at the side. In this example travel, the navigation level decides to branch at *B* and *C* but not at *A* (staying at the same branch). A more precise description is given in Tani and Fukumura (1994).

The branching decision for *the topological trajectory* is made by the neural architecture shown in Figure 3. This neural architecture consisting of a Kohonen network (Kohonen, 1982) and RNN is invoked only at the branching point. The range image, consisting of 24 range values at the branching, is fed into a Kohonen network and it is compressed into the output of a three dimensional vector. This output (the three-dimensional position of the cube of the Kohonen network in the figure) is fed into the input of RNN, generating output of the branching decision. This output is either of 0 (move straight) or 1 (move to a new branch), which is determined by the current sensory inputs and the context units values at the last branching. Our expectation is that the essential temporal structure associated with the task can be extracted by means of this context activity.

The employed recurrent network, in the later experiment, consists of three input units, eight hidden units, two context feedback units and one output unit. The Kohonen network consists of $6 \times 6 \times 6$ units as output and 24 units as input.

## 4. ROBOT HARDWARE

The physical experiment was conducted using the mobile robot YAMABICO (Yuta, 1990). The range image is obtained by a range finder consisting of three CCD cameras and laser projectors (see Figure 4). A range at a direction is calculated by triangulation: the height of a laser-projected horizontal line on a obstacle measured by a camera with a fixed tilt angle denotes the range. The ranges at 24 angular directions covering 160 degrees of the frontal side of the robot are measured at every 0.15 s. The range is measurable from 0.2 m to 5.0 m.

A computation loop between the robot and a host computer was constructed. The robot transmits the range image to the host computer by radio, the navigation architecture in the host generates the corresponding motor command and transmits it back to the robot. The robot moves at a constant speed of 0.3 m/s.

## 5. PROCEDURE OF RECURRENT TRAINING

The robot learns the navigation tasks through supervision by a trainer who is assumed to know the optimal paths. In the training of cyclic looping navigation, we repeatedly guided the robot to the desired loop from a set of arbitrarily selected initial locations. In actual training, the robot moves by the navigation of the control lever and stops at each bifurcation point, where the branching direction is taught by the trainer. The sequence of range images and teaching branching commands at those bifurcation points are fed into the neural architecture as training data.

The objective of training RNN is to find the optimal weight matrix that minimizes the mean square error of the training output (branching decision) sequences associating with sensory inputs (outputs of Kohonen network). The weight matrix can be obtained through an iterative calculation of back-propagation through time (BPTT) (Rumelhart *et al.*, 1986).

In this calculation the RNN is transformed into a

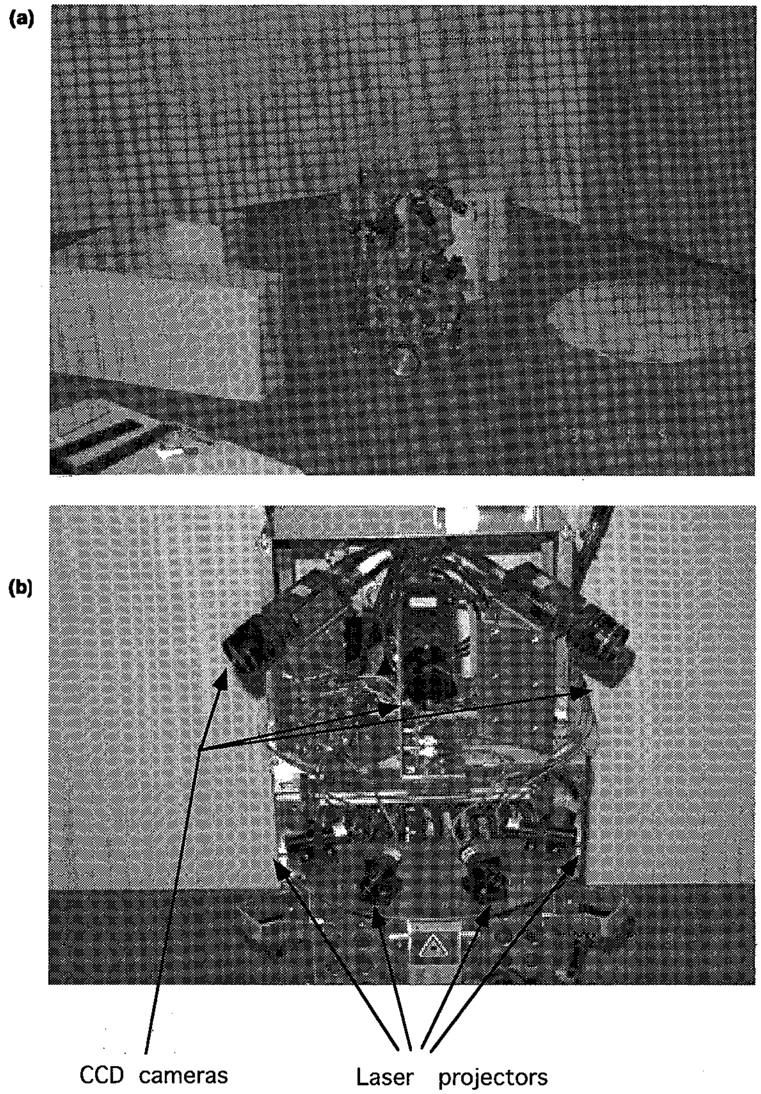



**FIGURE 4. (a) The mobile robot YAMABICO in the adopted workspace of 5 m × 7 m area. (b) YAMABICO is equipped with a range finder covering its frontal side The range consists of three CCD cameras and four laser projectors.**

cascaded feed forward network without loops by duplicating the original three-layered network in the time direction. The generalized delta rule (Rumelhart *et al.*, 1986) is applied to the cascaded network in order to find the weight update vector at each sequence.

The topological structure of the internal state space as well as its mapping to the output space are modulated through the learning process. When the learning error becomes asymptotically close to zero, it can be said that the task is embedded into a certain internal state space on the RNN.

## 6. EXPERIMENT

The experimental task is shown in Figure 1 of a previous section. The robot, at a position $A$, has to switch to the route of "8" or "0" by turns in the adopted workspace of a 5m × 7m area. We repeated the training of the robot, starting from 10 arbitrarily selected positions. Figure 5 shows the LED trace of the training trajectories. After the training, the robot was started from arbitrary initial positions, with setting initial context values to random ones.

Figure 6a and b show examples of test travels. The result showed that the robot always converged to the desired loop regardless of its starting position. Its convergence, however, took a certain period that depended on the case. Before convergence, the robot, in some cases, made a wrong branching. This was due to noise as well as the effects of the context activations of the RNN. The RNN initially could not output normally because of arbitrary initial setting of the context units. As the robot moves around the workspace encountering a sequence of known sensory inputs, the context activation starts to converge.

Noises affects the navigation performance remark-

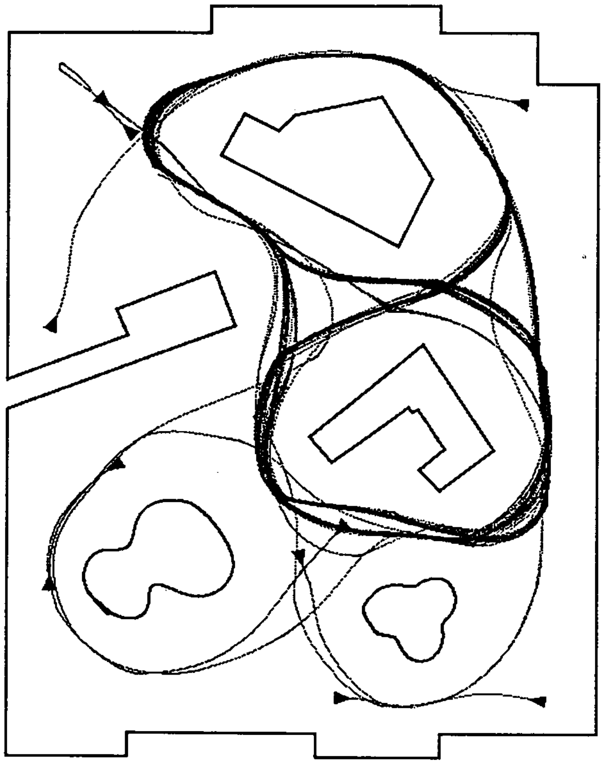
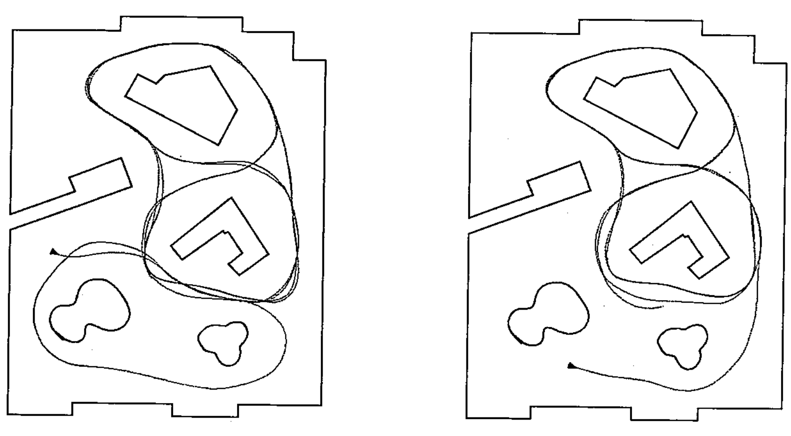


**FIGURE 5. LED trace of the training trajectories.**

ably. When miscellaneous noise such as mechanical, sensory, and radio noise is present, the bifurcation points sometimes become unstable. Thus, even after convergence, the robot could by chance go out of the loop, as the context activations are perturbed by larger noise. However, the robot always comes back to the loop after a while in the convergence of the context activations. Though the actual navigation contains an emergent property in its local decisions, it can be said that the global structure of convergence is quite stable in terms of the global attractor dynamics.

It is interesting to know how the task is represented inside the network. We investigated the activation patterns of RNN after the convergence into the loop, and the result is shown in Figure 7. The input and the context at each branching point is shown with three white and two black bars, respectively. One cycle of these (completing two routes of "0" and "8") are aligned upward as one column. The figure shows those of four continuous cycles. It can be seen that the robot navigation is exposed to much noise; the sensing input vector becomes unstable at particular locations, and the number of branchings in one cycle is not constant (i.e., 16 or 17 times). The row labeled (A) and (A′) are the branches to the routes of "0" and "8" respectively. In this point, the sensor input receives noisy chattering of different patterns independent of (A) or (A′). The context, on the other hand, is completely identifiable between (A) and (A′), which shows that the task sequence between two routes is rigidly encoded internally, even in the noisy environment. The robot's branching actions are determined by two factors, the internal dynamics of the context activations and the external force by the sensory inputs. During the transient behaviour, i.e., the convergent process to the cycling behavior, the external force by the sensory inputs tends to "entrain" the internal dynamics to be coherent with the environmental dynamics. Once the coherence is achieved in the stational behavior of the cycling, the internal dynamics can suppress a certain range of perturbations invoked by the external sensory noise.

We also investigated the effects of the dimensionality of the internal state on the performance by changing the number of the context units, while preserving other structures as the same. Learning with one context unit did not minimize the learning error, which could mean that the desired task requires



(a) (b)

**FIGURE 6. LED trace of the test travels after the training. Two examples of (a) and (b) starting from different positions, are shown. The black triangle indicates each starting position.**

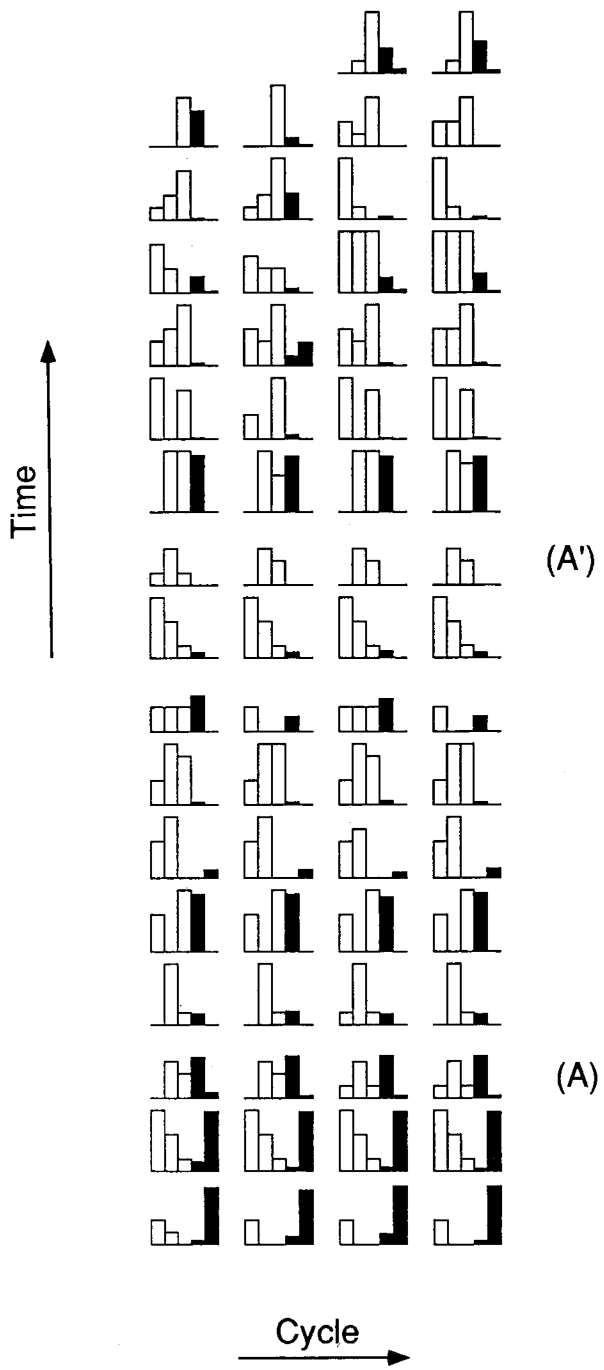



**FIGURE 7. Input and hidden state patterns after convergence into the desired loop. White and black bars represent input and hidden stages, respectively at branching points. Four complete cycles are shown by four columns, in which the row labeled (A) and (A′) are the branches to the routes of "0" and "8", respectively.**

an internal state space of larger dimensionality. As the number of the context units is increased, the learning tends to converge easier. By employing five context units, we tested the navigational performance. The robot converged into wrong loops before achieving the correct one in some cases. From this, it an be inferred that an internal state space with an excessive dimensionality could have generated harmful local attractors as well as the desired attractor. It can be said that the generalization is possible with an internal representation having the required, yet smaller dimensionality.

## 7. RELATED WORK

Brooks (1987), Mataric (1992) and others recently proposed the approach of *behavior based representation*, aiming to achieve a decentralized control scheme for autonomous agents. They employ rather symbolic representation in their formulas. Mataric (1992) considered a finite state machine (FSM)-type representation of the world model for the navigation problem. In her approach, the robot moves around an indoor space by wall-following, acquiring a map represented by a chain of predefined landmark types based on its sequential experiences.

It might be interesting to consider the difference between two formulas, a symbolic one and that of a dynamical system in our problem domain. One obvious advantage of symbolical representation is its explicitness as seen from the outside. Users can check inside, and can also tune the content if necessary. It is, on the other hand, assumed that system performance will not be sufficiently robust. A problem can arise when the robot fails to recognize an oncoming landmark because of some noise. The robot will be lost because it has received erroneous sensory inputs which are different from the one expected using the FSM. The FSM simply halts upon receiving those illegal inputs.

Our approach based on dynamical systems has no explicit representation to be seen from the outside. The knowledge of the navigation is submerged in a rather redundant description of neural network dynamics capable, however, of handling stochastic properties in its interactions with the environment, as shown in our experiment. When the context is lost by receiving erroneous sensory inputs, the context is recovered autonomously as long as the neural dynamics continues to iterate. The actual performance based on the dynamical system's approach can be inherently robust as long as the task is embedded in the global attractor dynamics.

## 8. SUMMARY

A RNN was employed to self-organize the required internal representation for sensory-based, goal-directed navigation of a mobile robot. A physical experiment was conducted with the mobile robot YAMABICO, in which the supervised training of a complex sequential routing task (an example of the hidden state problem) was tested. The results showed that a RNN could embed the task into its stable global attractor, even in a noisy environment.

We are currently studying to extend our dynamical systems approach to the model-based navigation so that the robot can plan action sequences for flexibly changed goals. A part of this study is shown elsewhere (Tani, 1996).

## REFERENCES


Asada, M. (1990). Map building for a mobile robot from sensory data. *IEEE Transactions on Systems, Man and Cybernetics,* **37–6,** 1326–1336.

Brooks, R. (1987). Intelligence without representation. *Artificial Intelligence,* **47,** 139–159.

Durrant-Whyte, H., & Leonard, J. (1989). Navigation by correlating geometrical sensor date. *Proceedings of IEEE/RSJ International Workshop on Intelligent Robotics and Systems '89* (pp. 440–447).

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation,* **3,** 249–265.

Freyberger, F., Kampmann, P., & Schmidt, K. (1990). Constructing maps for indoor navigation of a mobile robot by using an active 3D range image device. *Proceedings of IEEE/RSJ International Workshop on Intelligent Robotics and Systems '90* (pp. 143–148).

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequence machine. *Proceedings of Ninth Annual Conference of Cognitive Science Society* (pp. 531–546). Lawrence Erlbaum.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research,* **5**(1), 90–98.

Kohonen, T. (1982). Self-organized formation of topographically correct feature maps. *Biological Cybernetics,* **43,** 59–69.

Lin, L. J. & Mitchell, T. (1992). Reinforcement learning with hidden states. *Proceedings of the Second International Conference on Simulation of Adaptive Behavior* (pp. 271–280).

Mataric, M. (1992). Integration of representation into goal-driven behavior-based robot. *IEEE Transactions on Robotics and Automation,* **8,** 304–312.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters,* **59,** 2229–2232.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McLelland (Eds.) *Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems Man and Cybernetics, Part B,* Special issue on robot learning **26**(3), 421–436.

Tani, J., & Fukumura, N. (1994). Learning goal-directed sensory-based navigation of a mobile robot. *Neural Networks,* **7**(3), 553–563.

Yuta, S., & Iijima, J. (1990). State information panel for inter-processor communication in an autonomous mobile robot controller. *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems '90* (pp. 223–229).