Learning to Perceive the World as Articulated: An Approach for Hierarchical Learning in Sensory-Motor Systems

Jun Tani

Sony Computer Science Laboratory Inc.

Takanawa Muse Building, 3-14-13 Higashi-gotanda, Shinagawa-ku,Tokyo , 141 JAPAN

Tel 3-5448-4380, Fax 3-5448-4273, Email tani@csl.sony.co.jp

http://www.csl.sony.co.jp/person/tani.html

Stefano Nolfi

Institute of Psychology, National Research Council

15, Viale Marx - 00187, Rome, Italy

Abstract

This paper describes how agents can learn an internal model of the world structurally by focusing on the problem of behavior-based articulation. We develop an on-line learning scheme – the so-called mixture of recurrent neural net (RNN) experts – in which a set of RNN modules becomes self-organized as experts on multiple levels in order to account for the different categories of sensory-motor flow which the robot experiences. Autonomous switching of activated modules in the lower level actually represents the articulation of the sensory-motor flow. In the meanwhile, a set of RNNs in the higher level competes to learn the sequences of module switching in the lower level, by which articulation at a further more abstract level can be achieved. The proposed scheme was examined through simulation experiments involving the navigation learning problem. Our dynamical systems analysis clarified the mechanism of the articulation; the possible correspondence between the articulation mechanism and the attention switching mechanism in thalamo-cortical loops is discussed.

The original version of the manuscript was published in Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior, R. Pfeifer, B. Blumberg, J.A. Meyer and S.W. Wilson (Eds.), MA: MIT Press, 270-279. The revised version is in press. in Neural Networks, Special Issue 1999.

1 Introduction

How can sensory-motor systems attain an internal representation of the world in structurally organized ways? The consensus in cognitive science and artificial intelligence is that complex worlds can be represented efficiently utilizing modular and hierarchical structures of symbol systems (Newell, 1980). However, it is still not well-understood how such modular and hierarchical representations, if employed, become self-organized in analog neural systems by means of their iterative sensory-motor interactions.

The difficulty lies in the question of how the continuous sensory-motor flow can be perceived as being articulated into sequences of meaningful representative modules. Kuniyoshi (1994) addressed this articulation problem in the robot learning context. In his experiment with an assembling robot, the robot recognizes the various task performances by decomposing them into sequences of modular representations. Subsequently, the robot is able to learn various tasks in terms of combinations of the reusable modular representations obtained. For attaining such a modular representation, the task performance was temporally segmented by means of detecting "meaningful changes" in the observed sensory flow. The problem, however, is that the definitions of these "meaningful changes" were predetermined by designers. Our investigation focuses on how a robot can define "meaningful changes" by itself and perceive a continuous task performance as segmented into reusable modules.

Robot navigation learning, which has a quite long research history, faces the same type of problem. There are basically two types of approach. One is the neural network learning approach. Krose and Eecen (1994), Zimmer (1995) and Nehmzov (1996) showed that for relatively simple workspaces, localization problems for robots can be solved using the topology preserving map scheme (Kohonen, 1982). It is, however, difficult to scale-up this scheme since the very plain representation by a single neural network hardly organizes the modular and hierarchical structure of the learned contents. The other approach is the machine learning approach, which is used in landmark-based navigation (Kuipers, 1987; Mataric, 1992). In this approach, the travel of the robot is temporally segmented by means of landmarks such as turning at corners, encountering junctions, or going straight along corridors. This temporal segmentation enables the abstraction of robot experiences into a simple chain representation of these landmark types. The scheme can be scaled-up much more readily than the neural network learning approach since the landmarks play the roles of the representative modules. However, the problem is that the landmark types, which are defined by designers, are not necessarily intrinsic to the perceptions of a robot. The representative modules such as corners, junctions, or corridors, if necessary to the problem's solution, ought to be generated from the robot's experiences.

In this paper, we attempt to explain the problems of articulation and structural formation of modules and hierarchy from the dynamical systems perspective (Pollack, 1991; Beer, 1995; Schoner, Dose, & Engels, 1995; Smith & Thelen, 1994; van Gelder, 1998) by focusing on the structural coupling between the internal neural dynamics and the environmental dynamics. We propose a novel neural architecture which has been inspired by a modular and hierarchical learning method using neural nets, namely the mixture of experts proposed by Jacobs et al (1991). The proposed scheme is examined by conducting simulation experiments of robot navigation learning, where the mechanism of articulation is clarified qualitatively using dynamical systems concepts such as self-organization, coherence and phase transitions. We will discuss briefly the possible correspondence between the mechanism of the articulation and the mechanism of the attention switching which has been proposed to take place in thalamo-cortical loops.

2 Prediction Learning Using Sensory-Motor Flow

The paper introduces robot navigation learning as a prototype problem: our simulation experiments will illustrate how a set of representational primitives or "concepts" emerge and how they enable the construction of "concepts" in the higher level in a dynamic fashion. Our hierarchical learning approach is developed in combination with the prediction learning scheme which is described below.

Learning to predict the next sensation means that the system must acquire some analogical model of the target observed. Elman (1990) was the first to show that a recurrent neural network (RNN) can learn to predict word sequences by extracting regularity hidden in example sentences. Tani (1996) applied RNN prediction learning to the navigation learning problem. In this scheme, a robot learns to predict how its future sensory sequences will depend on the action sequences taken in a given workspace. It was shown that a real mobile robot with a range sensor learned the structure of the workspace hidden in the sensory-motor flow. The structure of the environment was represented as the attractor dynamics of the RNN in the course of prediction learning. It was further shown that the internal model obtained as the RNN attractor dynamics was utilized to generate mental plans for goal-directed behavior. However, a crucial criticism of this scheme is that the prediction of sensory input is made in a temporary discrete manner by means of the predefined branching mechanism. Branching plays the role of landmarks and invokes the temporal segmentation of the sensory-motor flow. Our new experiment is an attempt to eliminate these types of predefined mechanisms of temporal segmentation in the hope that the robot itself will find them.

One possible way to implement temporal segmentation of the sensory-motor flow is to focus on the magnitude of its change in time (Billard, 1996; Nolfi & Tani, 1999). For example, while a robot travels by following a straight wall using the range image, the image will be almost invariant. However, the sensory-motor state will change dramatically when the robot encounters a corner and starts turning left or right. This rapid change can be used as a signal for segmentation between the two behaviors of following straight wall and turning at a corner. However, the difficulty in this scheme is that the cornering behavior can be segmented several times since the sensory-motor state probably changes rapidly all through the cornering process. It is clear that the change of the sensory-motor state at a single moment provides only partial information about the on-going behavioral process. A specific mechanism is required whereby a meaningful time interval of the behavioral process, such as a cornering behavior, can be recognized as a unique event through extracting its specific spatio-temporal structure from the sensory-motor flow.

3 New Scheme

Our new proposal in this paper is to use multiple module RNNs, each of which competes to become an expert at predicting the sensory-motor flow for a specific behavior. The experts achieve their status through learning processes. For example, one module

RNN would win in predicting the sensory-motor flow while traveling around a corner; another would win while following a straight wall. The switching between the winning RNN modules actually corresponds to the temporal segmentation of the sensory-motor flow. The essential point in this scenario is that the segmentations take place by means of pronounced changes in the observed dynamical structure in the sensory-motor flow, rather than just in temporal differences in the sensory-motor state. These highly pronounced changes correspond to switching between the dynamical functions, each of which is embedded in an RNN through having learned the specific sensory-motor flow. One might ask how each RNN can choose to learn its corresponding sensory-motor flow. The speciality of each module is determined during the processes of on-line learning. The competition between the modules during the simultaneous processes of recognition and learning result in generating their specialties. The next section will introduce a new architecture called the mixture of RNN experts which has been extended from the original idea of the mixture of experts first expounded by Jacobs et al (1991).

3.1 Architecture

Figure 1 shows the proposed architecture for the mixture of RNN experts (MRE) which is used for the prediction-learning of the sensory-motor flow.

– Figure 1 –

Fig. 1(a) shows a hierarchical architecture consisting of two levels; more levels are possible in general.

Each RNN module in the lower level receives the sensory-motor inputs, $X_t : (s_t, m_t)$, and outputs the prediction of the sensory-motor inputs at a time Δt afterwards in the form $X_{t+1} : (s_{t+1}, m_{t+1})$, as shown in Fig. 1(b). The total output of the network is obtained from the weighted average of each output with its associated value of gate opening at the time g_t^i for all modules. The gate opening is computed dynamically with time using the prediction errors of each module, which are obtained from the difference between the prediction (s_{t+1}, m_{t+1}) and the outcome (s^*_{t+1}, m^*_{t+1}). The gate opens more if its module produces a relatively lower prediction error than the other modules. The module with the lowest error over a suitable time interval becomes the winner. The original work on the mixture of experts (Jacobs et al, 1991) used a gating network which selected the module with the closest correspondence to the target outputs. In our architecture, without using a gating network the module is activated autonomously as the result of dynamical competition between all modules over some time interval, utilizing on-line monitoring of the prediction errors. The winning module changes from one module to another as the profile of the sensory-motor flow changes with time.

The higher level network learns the gate opening dynamics of the lower level network. More specifically, each RNN module in the higher level samples the gate opening state of the lower level in the current time step $G_T : (g_T^1, g_T^2....g_T^n)$ and makes a prediction for the next time step G_{T+1}, as shown in Fig. 1(c). T denotes the time step in the higher level; the higher level sampling interval ΔT is much larger than that in the lower level. The modules in the higher level compete for gate opening g'^i_T, in the same way as shown for the lower level, and the resultant gate opening can be sent to yet higher levels in a recursive manner. The higher level network observes the lower level activities by means of perceiving its gate opening dynamics while the lower level network perceives the sensory-motor flow. In this manner, the signal is "bottom-up" as abstracted from one level to the next.

3.2 Algorithm

This subsection describes the mathematical formulae for the proposed scheme of the MRE. Suppose a single level network consists of n RNN modules, where x_t^i, y_{t+1}^i, y_{t+1}^* and g_t^i are the inputs, the outputs, the target outputs for teaching and the gate opening of the i-th module RNN, respectively. x_t and y_{t+1} correspond to the sensory-motor state or the gate opening state depending on the levels of the network.

The "soft-max" activation function is used to represent the i-th gate opening g_t^i given by:

$$g_t^i = \frac{e^{s_t^i}}{\sum_{j=1}^{n} e^{s_t^j}} \tag{1}$$

where s_t^i is the current internal value of the i-th gate opening. The total output of the network is y_{t+1}, given by:

$$y_{t+1} = \sum_{i=1}^{n} g_t^i \cdot y_{t+1}^i \tag{2}$$

We define the following likelihood function L which is maximized for prediction learning: it has been obtained by modifying the original definition of Jacobs et al (1991).

$$ln\, L = ln \sum_{i=1}^{n} g_t^i \cdot e^{\frac{-1}{2\sigma^2}\|y_{t+1}^* - y_{t+1}^i\|^2} \tag{3}$$

σ denotes a scaling parameter.

Both the weight of each RNN and the gate opening are updated simultaneously such that the likelihood function is maximized. This point is essential for the on-line learning scheme. In order to obtain the update rules for these two processes, we consider the partial derivatives of the logarithm of the likelihood function with respect to the internal value s^i and with respect to the output of the i-th RNN y^i given by:

$$\frac{\partial ln\, L}{\partial s_t^i} = g(i \,|\, x_t, y_{t+1}^*) - g_t^i \tag{4}$$

$$\frac{\partial ln\, L}{\partial y_t^i} = g(i \,|\, x_t, y_{t+1}^*) \frac{(y_{t+1}^* - y_{t+1}^i)}{\sigma^2} \tag{5}$$

where $g(i \,|\, x_t, y_{t+1}^*)$ is the *a posteriori* probability that the i-th module RNN generated the target vector y_{t+1}^*, in terms of x_t. Explicitly, this is given by

$$g(i \,|\, x_t, y_{t+1}^*) = \frac{g_t^i \cdot e^{\frac{-1}{2\sigma^2}\|y_{t+1}^* - y_{t+1}^i\|^2}}{\sum_{j=1}^{n} g_t^j \cdot e^{\frac{-1}{2\sigma^2}\|y_{t+1}^* - y_{t+1}^j\|^2}} \tag{6}$$

where $\|y_{t+1}^* - y_{t+1}^j\|^2$ represents the square of the error of the current prediction. Eq. (4) denotes the direction of update for the internal gate opening value s_t^i. The differentiation of $ln\, L$ with respect to y_{t+1}^i involves the error term $y_{t+1}^* - y_{t+1}^i$ weighted by the *a posteriori* probability associated with the i-th module RNN as shown in Eq. (5). Thus the connective weights of the RNN are adjusted to correct the error between the output of the i-th RNN and the global target vector, but only in proportion to the *a posteriori* probabilities. By this means, the individual RNN which is the expert for the on-going input sequence tends to learn exclusively. The error distributed to each module RNN is:

$$error_{t+1}^i = g(i \,|\, x_t, y_{t+1}^*) \cdot (y_{t+1}^* - y_{t+1}^i) \tag{7}$$

The details of the derivation of Eq. (4) to Eq. (7) are given in Ref. (Jacobs et al, 1991).

Upon obtaining the mathematical formulae, the actual update of the gate opening and the connective weights for each RNN are computed through the use of the back-propagation through time (BPTT) algorithm (Rumelhart, Hinton, & Williams, 1986). In this computation, the sequence of the sensory-motor inputs as well as the gate internal states for the previous l steps are stored temporally in the window memory. When new sensory-motor inputs are received, the window memory is shifted one step forward, the forward and backward computation by means of the BPTT are iterated for N^l times, and finally the sequence of l steps of the gate internal states as well as the connective weights for each RNN module are updated. The update for s_k^i, which is the ith gate internal state in the kth step in the window memory, is obtained as being:

$$\Delta s_k^i = \epsilon_g \cdot \frac{\partial ln\, L}{\partial s_k^i} - \eta_g \cdot (s_k^i - s_{k-1}^i) \qquad (8)$$

The first term in the right-hand side of the equation represents the direction of the update obtained in Eq. (4); the second term represents the damping term which suppresses abrupt changes in the gate opening; ϵ_g and η_g are parameters. This update is computed in the forward direction in the window memory from k=1 to k=l. The error obtained from Eq. (7) is back-propagated (Rumelhart, Hinton, & Williams, 1986) through the window memory for each RNN; the update of the connective weights is obtained by means of the steepest descent method utilizing parameters for the learning rate ϵ and for the momentum α.

4 Simulation Experiments

4.1 The environment

The scheme proposed above was investigated in the context of the navigation learning problem by simulation. We assumed a mobile robot with a sensor belt on its forward side holding 20 laser range sensors. The robot, upon perceiving the range image of its surrounding environment, maneuvers in a collision-free manner using a variant of the potential method (Khatib, 1986). (For further details of this maneuvering scheme, see Ref. Tani (1996).)

For our simulations, we adopted two different rooms, namely Room A and Room B connected by a door, as shown in Fig. 2 (a).

– Fig. 2 –

Fig. 2 (b) shows an example of the sensory-motor flow which corresponds to the robot travel indicated by the dotted line in Fig. 2 (a). In this workspace, the robot travels around one room three times, then enters the other room going through the newly opened door and travels around the other room three times. The on-line learning experiment was conducted while the robot moved between rooms for a total of 5 room encounters. The entire travel of the robot in this simulation took about 2100 Δt steps. The lower level network, which consists of 5 RNN modules each of which has 6 inputs, 6 outputs, 4 hidden units and 2 context units, learns to predict the sensory-motor state in the next step. (It is noted that only 6 out of 20 range sensor values are used for the RNN learning for the purpose of the reduction of the computation time.) The higher level network, which consists of 5 RNN modules each of which has 5 inputs, 5 outputs, 4 hidden units and 2 context units, learns to predict the gate opening state in the lower level network in the next step. Other parameter settings for the networks are $\epsilon = 0.002$, $\alpha = 0.9$, $\epsilon_g = 0.007$, $\eta_g = 0.02$. These settings are the same for both levels. The sampling interval in the higher level is 10 times longer than that in the lower level ($\Delta T = 10 \cdot \Delta t$). We observed how modules become self-organized in a hierarchical manner by looking at the gate opening dynamics taking place during the prediction learning of the two levels.

4.2 Results

We recorded gate opening dynamics both in the lower and in the higher levels during the entire learning process. First, let us consider the gate opening processes in the lower level network. Figure 3 shows the time development of each gate opening state and of the motor input in the lower level for three different periods.

– Figure 3 –

Figure 4 illustrates when and which module wins in the lower level network along the course travelled for each of the three different periods.

– Figure 4 –

Fig. 3 (a) shows the profiles for the period from step 130 to step 300 while the robot travelled in Room A for the first time. It can be seen that gate4 and gate3 open in turn as the profile of the motor command changes. In Fig. 4 (a), it is seen that the opening of gate4 corresponds to following a straight wall, while the opening of gate3 corresponds to both a left turn at a corner and to passing a T-junction. Fig. 3 (b) shows the profiles for the period from step 380 to step 550, when the robot experienced Room B for the first time. One can see that gate4, gate2 and gate3 open in turn. Fig. 4 (b) shows that these opening events corresponded to following a straight wall, making a right turn at a corner and making a left turn at a corner, respectively. Fig. 3 (c) shows the profiles for the period from step 820 to step 990, when the robot travelled around Room A for the second time. A remarkable finding is that the gate opening dynamics for this period differ from those observed during the first encounter with Room A. From Fig. 4 (c), one can see that the opening of gate3, which corresponded to both making a left turn at a corner and passing a T-junction in the previous encounter, now corresponds only to making a left turn at a corner, and that the opening of gate1 now corresponds to passing a T-junction. After this period, the learning processes in the network appeared to have stabilized and no further dramatic changes in the correspondence of the gate openings were found. By the end of the simulation, four types of meaningful concepts were generated using 4 RNN modules out of the 5 modules available to the lower level network. An important observation is that the process of generating concepts is totally dynamic in the sense that the correspondence between the RNN modules and their associated behavior is not static during the on-line learning process.

Next, we describe the gate opening dynamics in the higher level network. Figure 5 shows the opening of the 5 gates for the whole period of on-line learning. (The step number in this graph denotes the sensory-motor step number in the lower level, for clarity.)

– Figure 5 –

One can see that the stable switching of the gate opening between gate4 and gate1 takes place after 800 steps. This switching actually corresponds to the movement between rooms during the travel, where the open state of gate4 and gate1 correspond to travel in Room A and in Room B, respectively. We observe that gate0 opened only in the beginning while the robot traveled in Room A for the first time. The dynamic replacement of module0 by module4 for the representation of Room A evidently took place because the module representation in the lower level network also changed, as noted above. It is readily understood that the dynamics in higher level network can be stabilized only after stabilization occurs in the lower level network.

From the results we have obtained in the simulation experiments, it can be concluded that the proposed MRE architecture was successful in learning about the environment in a hierarchical way through the sensory-motor interactions of the robot. The lower level network learned to predict the row profile of the sensory-motor flow by organizing the modular representation of specific behavior. The higher level network did likewise for the sequences of segmented behavior by creating the higher concept of a room. Therefore, it can be said that the robot is not just perceiving the current sensory-motor flow but that it is also recognizing the background context of its behavior and situation.

We repeated this learning experiment for five times with different initial conditions including the starting position of the robot in either Room A or Room B and with different random initial connective weights of the networks. By looking at structures self-organized in the higher level network in these five experiments, equivalent module structures to those in the previous results, representing Room A and Room B, were found in four cases out of the five. Following this, we observed the lower level structures for these four cases and found that equivalent module structures to the previous result appeared in three cases, while the structures were different in one case. In the case where we did not observe clear module structure corresponding to two separate rooms in the higher level, it was observed that the lower level structures continued to change gradually which prevented the higher level structures from stabilizing. The stability in the higher level depends substantially on that in the lower level. These results reveal that the self-organization processes do not always arrive at one optimal solution. They can generate unstable and non-optimal structures by chance. Further studies are required for detailed understanding of “stability and diversity” problems in the hierarchical learning scheme.

5 Analysis and Discussion

5.1 On the dynamic mechanism for articulation

We have seen that building blocks for representing specific sensory-motor structures are self-organized in the lower level; the building blocks in the higher level are constructed by combining those in the lower level. The results may be interpreted as being the emergence of internal "symbols". However, the definition of our "symbols" is quite different to that used in traditional cognitive science studies (Newell & Simon, 1976; Newell, 1980). The "symbols" in our scheme are articulated not by the external designer's views but by the view intrinsic to the robot through its own experiences. In fact, the articulation emerges through the interactions between the system and its environment. For the purpose of putting forward this argument, we will now attempt to explain the qualitative mechanism of articulation from the dynamical systems perspective.

In order to understand how the switching of modules takes place dynamically as corresponding to room entering, we examined the time series of prediction outputs by each RNN module in the higher level network. Figure 6 shows the time series of prediction outputs and the corresponding error for each RNN module in the higher level network recorded from step 1400 to step 2000 during which period the robot moved from Room B to Room A.

– Figure 6 –

In Fig. 6 (a), the upper five rows represent the sequence of prediction outputs by the five RNN modules. The five squares aligned vertically in each five column row represent the values predicted for the five coming inputs (the sampling of the five gate openings in the lower level network) by their size. The largest square area corresponds to the value of 1.0 and zero area corresponds to a value of 0.0. The bottom row represents the sequence of five inputs. The robot moved from Room B to Room A at around step 1770 (denoted by a hatched line in the figure). Fig. 6 (b) shows the time series of the prediction error for each module. By looking at Fig. 6 (b), it is observed that the prediction error by RNN 1 is continuously minimized relative to the other RNNs until step 1770. During this period, it is seen, from Fig. 6 (a), that the output sequence by RNN 1 keeps coherence with the input sequence to a certain extent; RNN 0, RNN 2 and RNN 3 are not activated at all, while RNN 4 is activated but is incoherent with the input sequence. On the other hand, the output sequence by RNN 4 becomes coherent with the input sequence after step 1770 through minimizing its error, while the output sequence of RNN 1 loses coherence with the input. This switching of the winning RNN modules takes place rather quickly within several iteration steps of the RNNs. We speculate that the robot recognizes its surrounding in a particular room by means of achieving temporal coherence between the dynamics of a particular RNN module and the environmental dynamics.

In order to investigate the essential dynamical structure of the organization of the coherence and the incoherence appearing in the coupling between the internal and the environmental dynamical systems, we compared the phase trajectory of each RNN module for its intrinsic dynamics and for its coupled dynamics with Room A and with Room B. Figure 7 shows such a comparison.

– Figure 7 –

The five plots in the upper row in this Figure show the phase trajectory for the intrinsic dynamics of each RNN. Each RNN can be iterated autonomously without the input sequences by installing feed-back loops from the prediction outputs to the inputs. (This autonomous iteration of each RNN corresponds to mental rehearsing of the learned temporal patterns.) We plotted the activation trajectory for two context units in X and Y during the autonomous iteration using fixed connection weights for each RNN which had been obtained in step 2000 in the previous experiment. It is observed that the trajectories of RNN 0, RNN 2 and RNN 3 converge to a fixed point and that those of RNN 1 and RNN 4 do not converge to a point but generate invariant closures in the phase space. Further analysis showed that the dynamical structure of RNN 0, RNN 2 and RNN3 is characterized by a fixed point attractor, while the dynamical structure of RNN 1 and RNN 4 is characterized by a quasi-periodic attractor. The five plots in the middle row and the bottom row show the phase trajectories of the coupled

RNN dynamics with the Room A and Room B environments, respectively. The phase trajectories are plotted for the context units' activation states in response to receiving the input sequences while the robot travelled around the two rooms. In these plots, it is seen that RNN 0, RNN 2 and RNN 3 each generated the same fixed point attractor from their intrinsic dynamics regardless of whether the RNN was coupled with Room A or with Room B. The plots for RNN 1 and RNN 4 are different. RNN 1 exhibits a similar phase trajectory to its intrinsic one when coupled with Room B; RNN 4 does likewise when coupled with Room A. This observation coincides with the previous observation that the prediction sequence by RNN 1 and by RNN 4 became coherent with the input sequence while traveling around Room B and Room A, respectively.

In our previous work (Tani, 1996), we studied how a mobile robot learns to predict the sensory sequence during the navigation learning. It was shown that the RNN is able to learn the structure hidden in the environment as embedded in attractor dynamics by means of of sensory-motor interactions. Our analysis (Tani, 1996) showed that the prediction process is successful when coherence is achieved between the internal RNN dynamics and the environmental dynamics. Even when the coherence is perturbed by noise, the coherence can be re-established by means of the entrainment (Endo & Mori, 1978; Beer, 1995) of the internal RNN dynamics by the environmental dynamics (Tani, 1996). The same mechanism can explain the autonomous switching of modules in the current study: one module is activated in a manner inhibitory to the others by achieving coherence between its intrinsic dynamics and with a specific dynamical structure hidden in the input flow. When the essential dynamical structure in the input flow changes, the currently activated module loses coherence with the flow, while another module is activated through gaining coherence with the flow. This winning module switching takes place rather quickly as a phase transition of the dynamical state from one stable attractor to another. By means of switching of coherence among the repertoire of the intrinsic dynamics, this phase transition actually results in the articulation which the system perceives internally for the structural changes in the sensory-motor flow (see the schematic diagram for this explanation in Figure 8.)

– Figure 8 –

5.2 Recursive chains from state to function

Another important aspect which should be discussed is the relationship between state and function in the hierarchical learning. The direct observation of the sensory-motor state provides only non-robust information about its present process since the state can evolve in many ways. What should be focused on is rather the spatio-temporal structure hidden in the time development of the state, since such structures could be similar in many cases even when the state is changing quantitatively. The RNN, which is basically an adaptive type dynamical function, is used for capturing such similar structure from the observed time development of the state. This time development of the state is, eventually, represented by one of the RNN functions. The higher level observes that which RNN function is currently activated in the lower level in terms of its gate opening state. This gate opening state can vary as the result of structural change in the lower level. The resultant time development of the gate opening state is again captured by the RNN functions in the higher level. Here, we see that the aim of the hierarchical learning is to organize such recursive chains from the state to the function, and from the function to the state, through the level of abstraction.

5.3 Articulation and attentional switch

The observed module switching mechanism might be related to the proposed attentional switch mechanism in the thalamo-cortical loop which has been explained in terms of the search-light metaphor (Crick, 1984). In Crick's model, $windows of an attention$ are created by means of a gating mechanism in the nucleus reticularis (NR). The cortical feedback to the NR gates thalamic transmission of subcortical data; hence, the process allows the cortex to attend to part of these data selectively. Taylor and Alavi (1993), Baars (1997) and Newman (1997) proposed that gating by the NR induces a global winner-take-all competition among thalamo-cortical loops as a result of which only one of many competing sensory streams reaches consciousness. In a macroscopic sense, our explanation of the dynamical mechanism of articulation agrees with the above scenario whereby the gates in MRE correspond to the NR and the RNN experts correspond to the cortical parts. Therefore, it is hypothesized that the attentional switch among the set of thalamo-cortical loops with their accompanying NR gates play the essential role in articulating the bottom-up streams in the multiple levels and therefore in creating symbols and concepts. We suggest that "a consciousness" arises at the very moment of the articulation which is accompanied by dynamical switching among thalamo-cortical loops.

5.4 Related works

The current study was inspired by other adaptive agent research based on the dynamical systems approach by, for example, Beer (1995), Yamauchi and Beer (1994) and Smithers (1996). However, it is notable that the dynamical systems approach, which especially emphasizes coherence and structural coupling (Varela, Thompson, & Rosch, 1991) of the internal system with its environmental dynamical systems, has explained only the low level behavior mechanisms– for instance, walking by a legged agent (Beer, 1995), or simple visually guided behavior (Beer, 1996). In the current research, we attempted to apply the dynamical systems approach to the question of higher-order cognition by introducing mental processes of anticipation in the agents. Indeed, it has been found that anticipation plays an essential role for the cognitive mechanism of articulation. As we have illustrated in the preceding sections, the articulation of the input flow is triggered when the prediction of the next future event fails, causing temporal incoherence in the structural coupling. It is plausible that the higher-order cognition is achieved by utilizing not only the coherence but also the incoherence appearing in the interaction between the internal and the environmental dynamics (Tani, 1998).

During the initial reviewing process of this paper the authors were informed of other related studies. These studies are discussed below. Pawelzik et al. (1996) applied their scheme of annealed competition of experts to segmentation of time series from complex system. The basic idea in this scheme was quite similar to ours; segmentation of the time series is achieved by means of the dynamical competition among experts without employing the central gating network. A single layered network which consists of a set of Radial Basis function for each module was trained in an off-line manner using an annealing technique. The learning process seems to be much more stabilized compared to our cases because of the relative easiness in training of Radial Basis function in the off-line manner. Their study, however, did not address the issues of the hierarchical organization of modules. Several other authors extended the mixture of expert systems for the purpose of modeling temporal processes (Bengio & Frasconi, 1995; Cacciatore & Nowlan, 1994; Melia & Jordan, 1998). Wolpert and Kawato (1998) also proposed such extensions where multiple pairs of inverse and forward models in a single layer organization are utilized for the purpose of controlling motor outputs and predicting sensory inputs. They suggested from the anatomical observation that the motor control functions in the cerebellum could be modeled by the proposed multiple pairs of inverse and forward models.

Schmidhuber (Schmidhuber, 1992) proposed a self-organizing multi-level hierarchy of RNNs which learns to decompose sequences recursively. In Schmidhuber's architecture, sequences are segmented by means of detecting the prediction error of an RNN at each level. If the error is larger than a threshold value at a certain time step, the sequence is segmented at that step and a key symbol is sent to the higher level which composes another sequence in the succeeding segmentation in the lower level. The main difference between our scheme and Schmidhuber's is that in ours, sequences are segmented by means of the dynamical competition between experts without using any predefined threshold values. We speculate that this dynamical competition mechanism is much more natural and even robust compared to Schmidhuber's scheme. Recently Hochreiter and Schmidhuber (1997) proposed a method called "Long Short-Term Memory" (LSTM). LSTM learns to bridge minimal time lags in long time steps by enforcing constant error flow through "constant error carrousels" within special units. Multiple gate units learn to open and close access to the constant error flow: this mechanism seems to be analogous to the gate opening mechanism of the mixture of experts given in this paper. LSTM does not address directly the issue of the multiple levels of articulation on which we have focused in the current paper. Ring (1994) proposed a hierarchical learning scheme in which sequential tasks are learned by combining already known tasks into new ones by means of reinforcement learning. Since the hierarchy in Ring's system is built on the discrete representation of sensation and action as system primitives, his scheme cannot be applied directly to the problem of articulating continuous sensory-motor flow.

6 Conclusion

In this paper, we have proposed a novel scheme of hierarchical learning for sensory-motor systems using the mixture of RNN experts. The scheme was examined through simulation experiments concerning on-line navigation learning. The results indicate that the robot learns to articulate a continuous sensory-motor flow dynamically while the modular and hierarchical structures are self-organized internally in a recursive manner across multiple levels. We explained the observed mechanism of articulation qualitatively by using the dynamical systems language and we have discussed its correspondence with the attentional switch mechanism.

The proposed approach can be developed in many ways in the future. The example considered in this paper was limited to the prediction learning of the sensory-motor flow. One future development could be the inclusion of motor or action-learning mechanisms. The scheme should be extended to cover both prediction-learning and behavior-learning to ensure that "concepts" can also be self-organized for the purpose of action generation. Then, goal-directed planning can be carried out by utilizing the acquired internal representation embedded in the intrinsic dynamics of the RNNs, as had been described in Ref. Tani (1996). One limitation of the present study is that all the interactive processes were undertaken using only the bottom-up pathway in the architecture. A more general model is one in which top-down processes interact with bottom-up processes such that a module can be activated by means of of bi-directional interactive dynamics between the top-down prediction from the higher level network and the bottom-up signals from the lower level network. In future research, we will study how goal-directed behavior can be generated in the extension of the proposed scheme by considering the shortcoming mentioned above.

References

[1] Baars, B.J. (1997). *In the theater of consciousness: the workspace of the mind*. Oxford: Oxford University Press.

[2] Beer, R. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1), 173–215.

[3] Beer, R. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes, M. Mataric, J. Meyer, J. Pollack and S. Wilson (Eds.), *From Animals to Animats 4*(pp.421–429), Cambridge, MA: MIT press.

[4] Bengio, Y., & Fransconi, P. (1995). An input-output HMM architecture. In J.D. Cowan, G. Tesauro and J. Alspector (Eds.), *Advances in Neural Information Processing Systems 7* (pp.427–434), Cambridge, MA: MIT Press.

[5] Billard, A. (1996). Do you follow me? or learning to speak through imitation for social robots. Master's thesis, Dept. of AI, University of Edinburgh, UK.

[6] Cacciatore, T.W., & Nowlan, S.J. (1994). Mixtures of controllers for jump linear and non-linear plants. In G. Tesauro, D.S. Touretzky and T.K. Leen (Eds.), *Advances in Neural Information Processing Systems 6* (pp.719–726), San Mateo, CA: Morgan Kaufmann.

[7] Crick, F. (1984). Function of the thalamic reticular complex: the searchlight hypothesis. *Proc. of the National Academy of Sciences, USA*, 81, 4586–90.

[8] Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14, 179–211.

[9] Endo, T., & Mori, S. (1978). Mode analysis of a ring of a large number of mutually coupled van der Pol oscillators. *IEEE Trans. Circuits Syst.*, CAS-25(1), 7–18.

[10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[11] Jacobs, R., Jordan, M., Nowlan, S. & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1), 79–87.

[12] Jordan, M., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2), 181–214.

[13] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The Int. J. of Robotics Research*, 5(1), 90–98.

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

[15] Krose, B., & Eecen, M. (1994). A self-organizing representation of sensor space for mobile robot navigation. *Proceedings Int. Conf. on Intelligent Robotics and Systems '94*, 257–263.

[16] Kuipers, B. (1987). A qualitative approach to robot exploration and map learning. *Proceedings AAAI Workshop Spatial Reasoning and Multi-Sensor Fusion*, 774–779.

[17] Kuniyoshi, Y., Inaba, M., & Inoue, H. (1994). Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE. Trans. on Robotics and Automation*, 10(6), 799–822.

[18] Mataric, M. (1992). Integration of representation into goal-driven behavior-based robot. *IEEE Trans. Robotics and Automation*, 8(3), 304–312.

[19] Meila, M., & Jordan, M.I. (1998). Estimating dependency structure as a hidden variable. In M.I. Jordan, M.J. Kearns and T.K. Leen (Eds.), *Advances in Neural Information Processing Systems 10*, Cambridge, MA: MIT Press.

[20] Nehmzov, U. (1996). Flexible control of mobile robots through autonomous competence acquisition. *Measurement and Control*, 28(2), 48–54.

[21] Newell, A., & Simon, H. (1976). Computer science as emprical inquiry: symbols and search. *Communication of the ACM*, 19, 113–126.

[22] Newell, A. (1980). Physical symbol systems. *Cognitive Science*, 4, 135–183.

[23] Newman, J. (1997). Putting the puzzle together, part II: towards a general theory of the neural correlates of consciousness. *Journal of Consciousness Studies*, 4(2), 100–121.

[24] Nolfi, S. & Tani, J. (1999). Extracting regularities in space and time through a cascade of prediction networks: the case of a mobile robot navigation in a structured environment. *to appear.*

[25] Pawelzik, K., Kohlmorgen, J. & Muller, K.-R. (1996). Annealed competition of experts for a segmentation and classification of switching dynamics. *Neural Computation*, 8(2), 340–356.

[26] Pollack, J. (1991). The induction of dynamical recognizers. *Machine Learning*, 7, 227–252.

[27] Ring, M. (1994). Continual learning in reinforcement environment. Ph.D thesis, University of Texas at Austin.

[28] Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing*. Cambridge, MA: MIT Press.

[29] Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2), 234–242.

[30] Schoner, G., Dose, M., & Engels, C. (1995). Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16, 213–245.

[31] Smith, L., and Thelen, E. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge, MA: MIT Press.

[32] Smithers, T. (1996). On what embodiment might have to do with cognition. *Proceedings AAAI Fall Symposium, Embodied Cognition and Action*, 113–118.

[33] Tani, J. (1998). An interpretation of the 'self' from the dynamical systems perspective: a constructivist approach. *Journal of Consciousness Studies*, 5(5-6), 516–42.

[34] Tani, J. (1996). Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective. *IEEE Trans. System, Man and Cybernetics Part B*, 26(3), 421–436.

[35] Taylor, J.G. & Alavi, F.N. (1993). Mathematical analysis of a competitive network for attention. In J.G. Taylor(Ed.), *Mathematical Approach to Neural Networks*. New York: Lawrence Erlbaum.

[36] van Gelder, T. The dynamical hypothesis in cognitive science. *Behavior and Brain Sciences*, in press.

[37] Varela, F., Thompson, E., and Rosch, E. (1991). *The embodied mind*. Cambridge, Mass.: MIT Press.

[38] Wolpert, D. & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks* 11, 1317–1329.

[39] Yamauchi, B. M., & Beer, R. D. (1994). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior* 2(3), 219–246.

[40] Zimmer, U. R. (1996). Robust world-modeling and navigation in a real world. *Neuro-Computing* 13, 247–260.

List of Figures

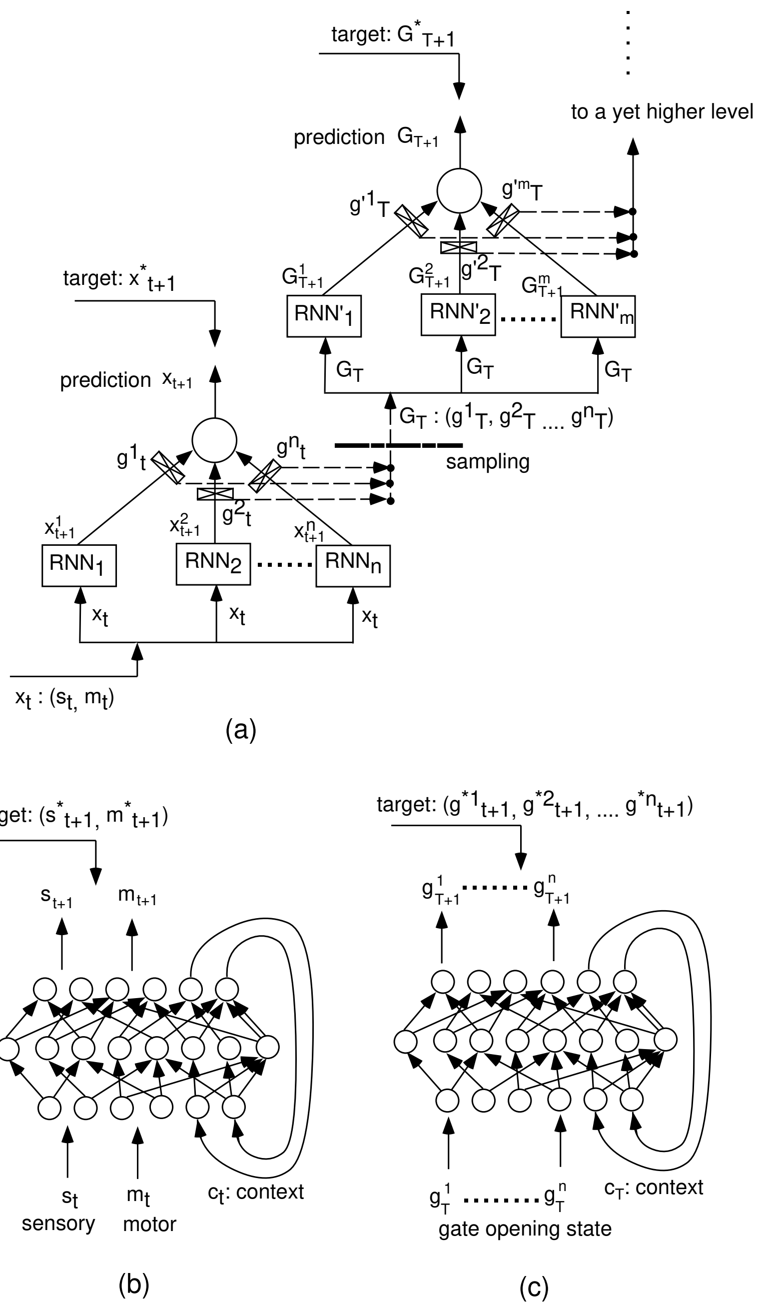


Figure 1: The complete architecture of the mixture of RNN experts for prediction learning. (a) hierarchical learning architecture, (b) details of each RNN module for learning the sensory-motor flow in the lower level, and (c) RNN module for learning the gate opening dynamics in the higher level.

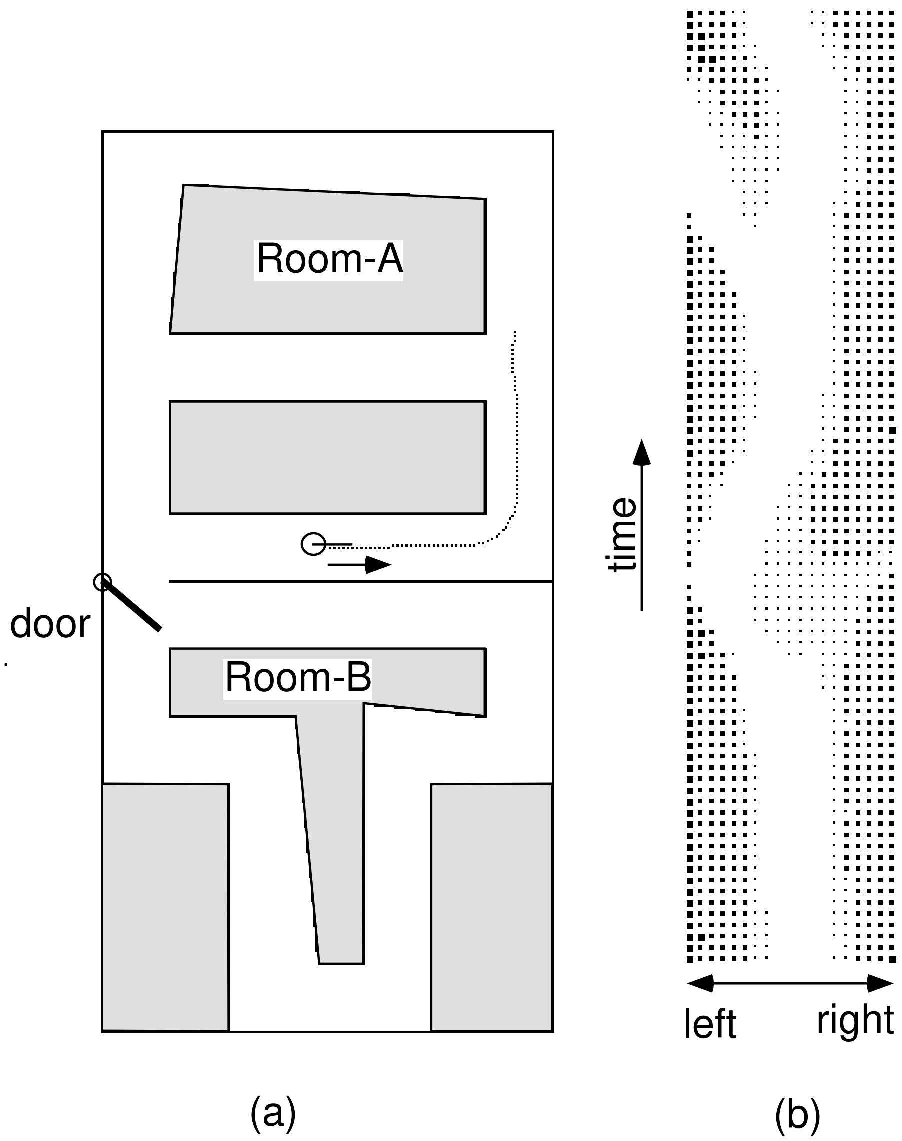


Figure 2: (a) Simulation workspace consisting of two rooms connected by a door. (b) the time development of the simulated range image while the robot traveled.

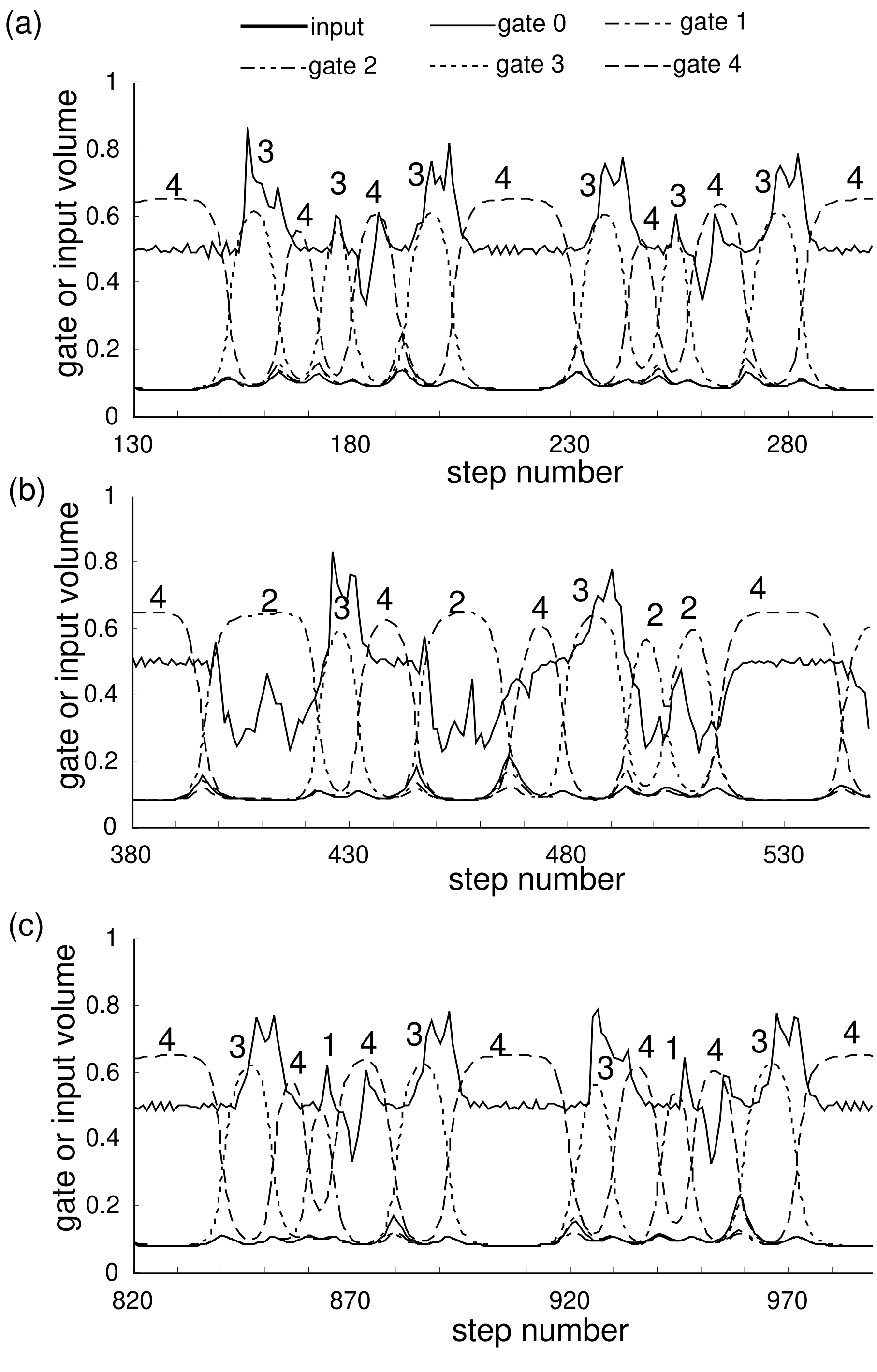
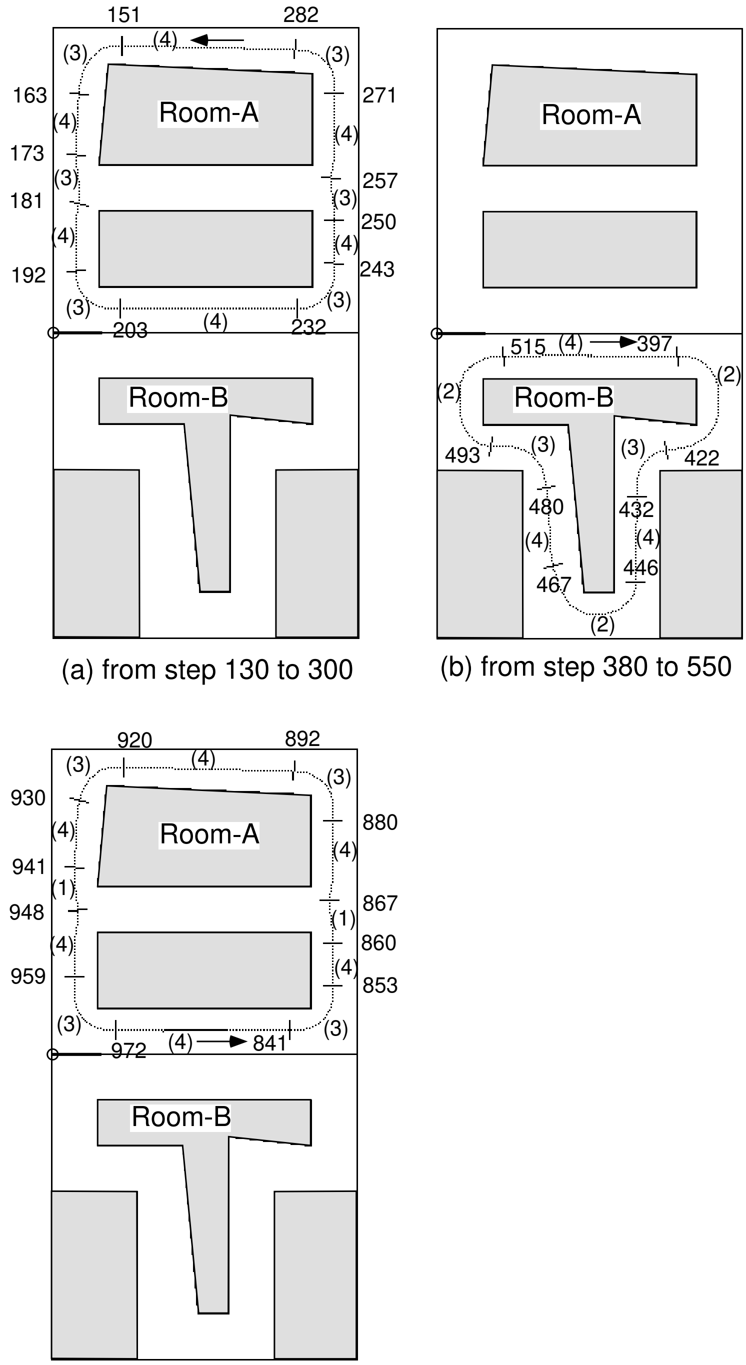


Figure 3: Time development of the opening of 5 gates and of a motor input in the lower level network for three different periods. The number near the data denotes the current winning gate.



(a) from step 130 to 300

(b) from step 380 to 550

(c) from step 820 to 990

Figure 4: Along the trajectory of the robot travel indicated by the dotted lines, the number in parentheses indicates which module wins and the number without parentheses denotes the step number when the module switching took place.

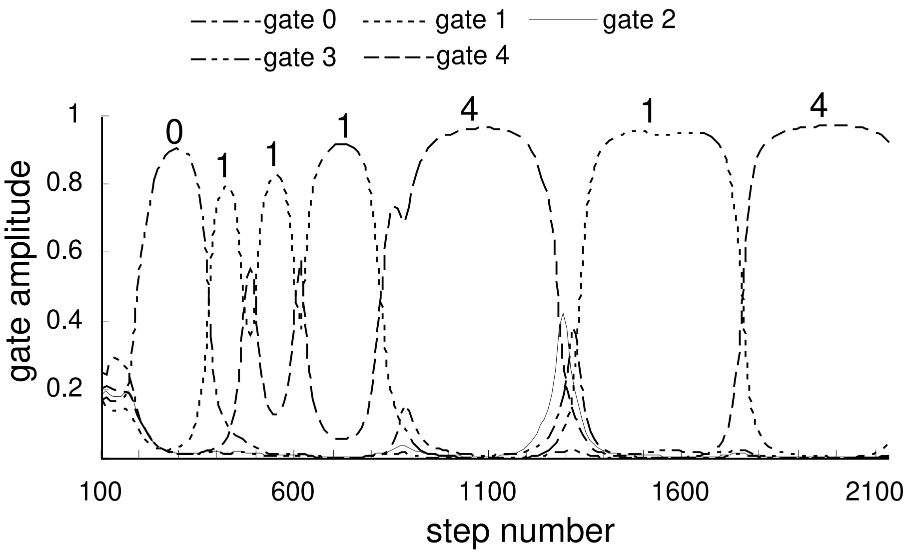


Figure 5: Gate opening dynamics in the higher level network during the whole process of learning.

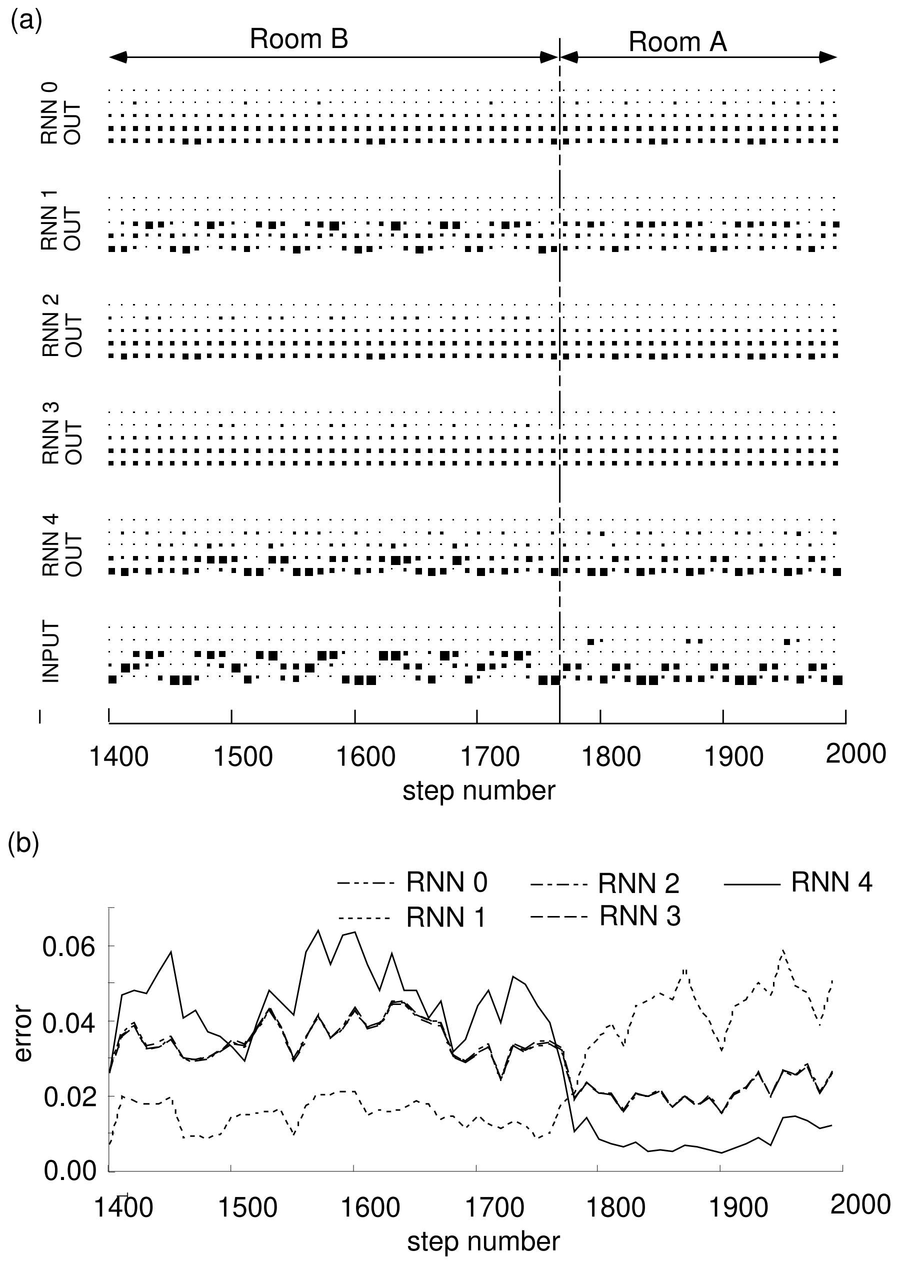


Figure 6: (a) shows the sequence of inputs (shown in the bottom row) and the corresponding prediction outputs by each RNN module (shown in the upper five rows) in the higher level network. (b) indicates the time dependence of the prediction error by each RNN module.

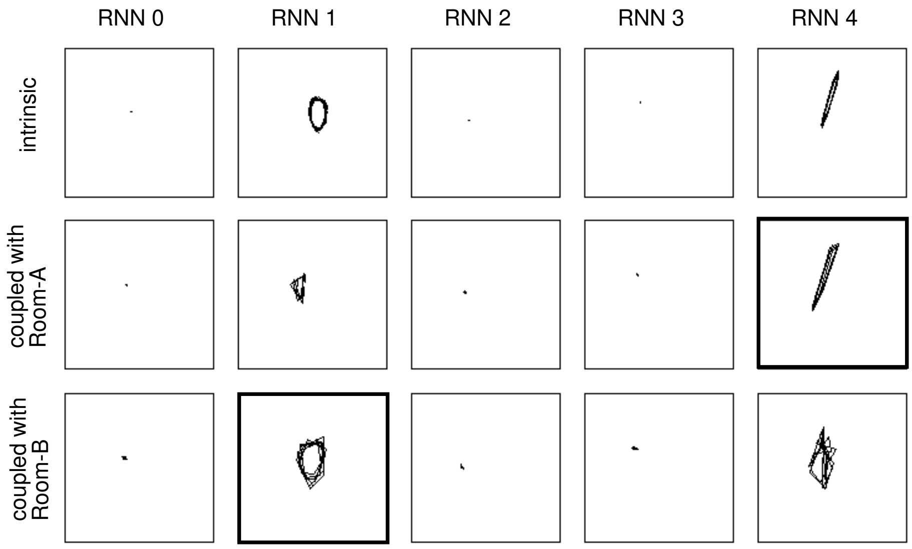


Figure 7: The phase trajectory for each RNN module in the higher level is shown for its intrinsic dynamics and for its coupled dynamics with Room A and Room B, in the upper row, middle row and bottom row, respectively.

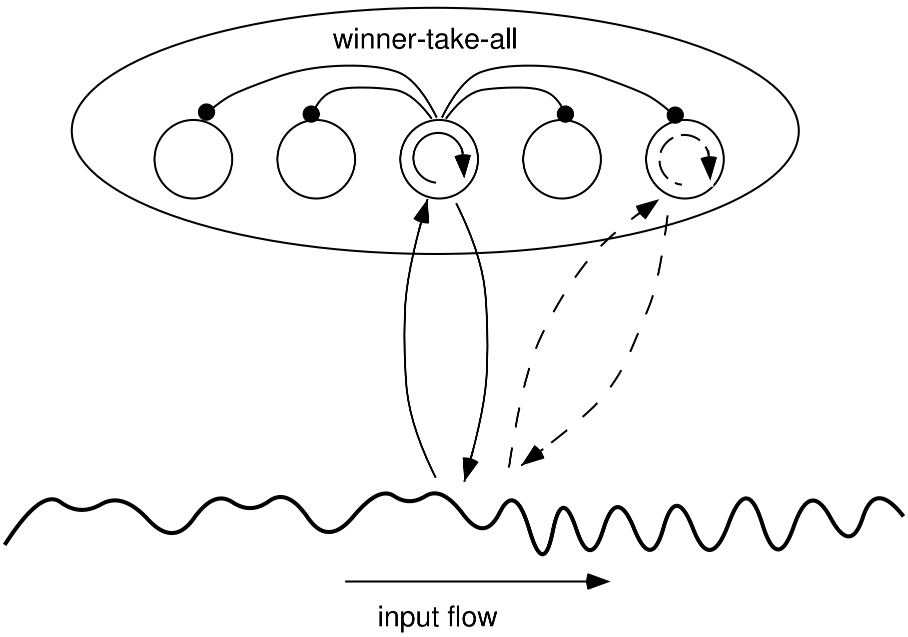


Figure 8: Dynamic mechanism of articulation. The input flow is articulated when coherence with the flow is switched among the repertoire of the intrinsic dynamics contained in multiple modules.