PHYSICAL REVIEW E VOLUME 55, NUMBER 1 JANUARY 1997

Learning feature constraints in a chaotic neural memory

Shigetoshi Nara[1] and Peter Davis[2]
[1]The Division of Mathematical and Information Sciences, Faculty of Integrated Arts and Sciences, Hiroshima University, Kagamiyama 1-7-1, Higashi-Hiroshima 739, Japan
[2]ATR Adaptive Communications Research Laboratories, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
(Received 26 April 1996)

We consider a neural network memory model that has both nonchaotic and chaotic regimes. The chaotic regime occurs for reduced neural connectivity. We show that it is possible to adapt the dynamics in the chaotic regime, by reinforcement learning, to learn multiple constraints on feature subsets. This results in chaotic pattern generation that is biased to generate the feature patterns that have received responses. Depending on the connectivity, there can be additional memory pulling effects, due to the correlations between the constrained neurons in the feature subsets and the other neurons. [S1063-651X(97)00901-X]

PACS number(s): 87.10.+e, 05.45.+b

I. INTRODUCTION

Investigating the adaptability of chaotic systems is important from the point of view of understanding behavior of natural adaptive systems, and also relevant to engineering highly adaptive multifunctional systems [1–6]. A number of studies have already been made on adaptation in chaotic devices and networks. In previous works, we have demonstrated adaptive behavior in a particular type of neural network model that has both nonchaotic and chaotic parameter regimes. In the nonchaotic regime, the network functions as a conventional associative memory [13]. In the chaotic regime, there are itinerant orbits that visit nearly all the memory patterns, and so can be used to search among them [7]. For example, we have shown such a network can be used to search interactively for a satisfactory memory recall, when the only input to the system is in the form of scalar responses, by an external environment or operator, to patterns generated by the network [7,8]. There we used adaptive bifurcation dynamics [7–12] in which the external response, or reaction signal, is fed back to a system parameter, specifically the connectivity parameter, to drive the system above or below the onset of chaos depending on whether the response was ''bad'' or ''good.'' There we emphasized that it can be functionally significant for a network to have (at least) two regimes of behavior, a stable associative regime and an itinerant chaotic regime, and switch between the two regimes.

In this paper we show how the chaotic state itself, i.e., chaotic attractor, can be adapted by reinforcement learning. We discuss how this is useful in the context of search-access of memory.

II. RECURRENT NEURAL NETWORK MODEL

We consider a neural network model described by the following set of equations:

$$s_i(t+1)=\mathrm{sgn}\left(\sum_j \epsilon_{r;ij}T_{ij}s_j(t)-\theta_i\right), \tag{1}$$

where each neuron is represented by discrete variables $s_i=\pm1$, T_{ij} is the synaptic connection matrix, and θ_i are threshold values. ϵ_r is a matrix of binary activity values $\epsilon_{r;ij}=1$ or 0 with $\Sigma_j\epsilon_{r;ij}=r$, where $r(0\leqslant r\leqslant N)$ is the connection number or fan-in: the number of neurons each neuron is connected to. The synaptic connection matrix is defined as

$$\mathsf{T}=\sum_{\mu=1}^{L}\sum_{\lambda=1}^{M}\vec{\xi}_\mu^{\lambda+1}\otimes\vec{\xi}_\mu^{\lambda}, \tag{2}$$

where $\vec{\xi}_\mu^{M+1}=\vec{\xi}_\mu^{1}$ and L and M are, respectively, the number of cycles and the period of each cycle. $\{\vec{\xi}\}$ is a set of memory patterns. We assume that the total number of patterns is much less than the numbers of neurons, $LM\ll N$, and that the memory patterns typically have small overlap, i.e., small value of $m_{\mu\lambda\mu'\lambda'}=\vec{\xi}_\mu^{\lambda}\cdot\vec{\xi}_{\mu'}^{\lambda'}$.

If connectivity r is large, $r\simeq N$, the network functions as a conventional associative memory. If $\vec{s}(t)$ is one of the memory patterns, $\vec{\xi}_\mu^{\lambda}$ say, then $\vec{s}(t+1)$ will be the next memory pattern in the cycle, $\vec{\xi}_\mu^{\lambda+1}$ If $\vec{s}(t)$ is near one of the memory patterns $\vec{\xi}_\mu^{\lambda}$, then it will be in the basin of $\vec{\xi}_\mu^{\lambda}$ and the sequence $\vec{s}(t+kM)(k=1,2,3,\dots)$ generated by the M-step map will converge to that memory pattern. [More specifically, for each memory pattern $\vec{\xi}_\mu^{\lambda}$, there is a set of states $B_{\mu\lambda}$, called memory basins, such that if $\vec{s}(t)$ is in $B_{\mu\lambda}$ then $\vec{s}(t+kM)(k=1,2,3,\dots)$ will converge to $\vec{\xi}_\mu^{\lambda}$.]

If connectivity r is reduced sufficiently, the memory patterns become unstable and $\vec{s}(t)$ chaotically wanders around in pattern space. Typically, the wandering $\vec{s}(t)$ repeatedly visits all the regions that are memory basins at large connectivity values.

III. ADAPTATION BY REINFORCEMENT LEARNING OF THRESHOLDS

Now as our adaptation problem, we consider the problem of adaptation of the output when we assume that the only input to the systems are a few scalar signals $E(t)$ that repre-

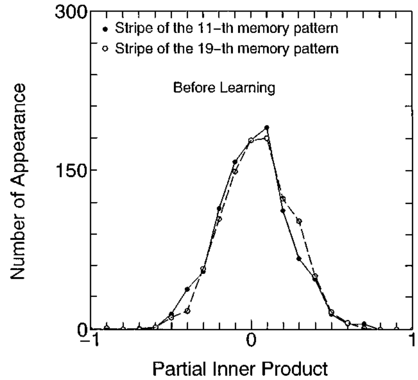

FIG. 1. Feature distributions before learning. A feature distribution is the histogram of values of the overlap (inner product) between *the state of the neurons in a feature subset and the corresponding target patterns*, obtained from a chaotic time series. Connectivity $r=10$. The two distributions in the figure are for target features corresponding to stripes in the 11th and 19th memory patterns.

sent external responses to the output patterns $\vec{s}(t)$. To be specific, we consider a response to indicate the presence (or absence) of a specific feature in the output. A feature is defined as the similarity of a subset of neurons (such as a stripe in a two-dimensional pattern) with a target pattern. (In the context of memory it is of particular interest to consider, as we do in the following, the case where target features correspond to features in memories.)

Feature, or feature-match, value of state $\vec{s}(t)$ is defined as

$$C_\alpha(t)=\frac{1}{N_f}\sum_{i\in G_\alpha} s_i(t)v_i^\alpha . \tag{3}$$

Here, v^α is a target feature pattern and G_α is the feature subset, the subset of N_f neurons whose outputs are used in the definition of feature. $C_\alpha(t)$ becomes $+1$ if the feature subset in the current output is identical with the search target and 0 if orthogonal. The external response is then assumed to be

$$E=H((1-\delta)-C), \tag{4}$$

where H is a unit step function

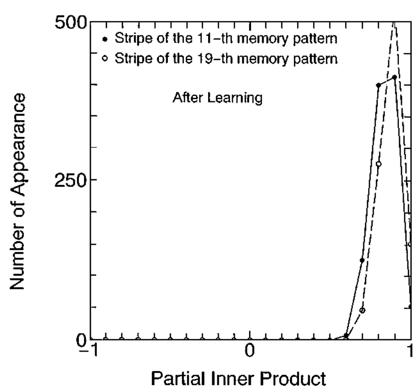

FIG. 2. Feature distributions after learning. The same distributions as in Fig. 1, obtained after learning.

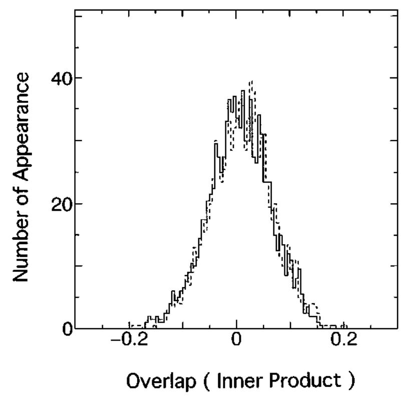

FIG. 3. Single-memory overlap distributions at $r=10$ before learning. A memory overlap distribution is the histogram of values of the overlap (inner product) between *the state of the neurons and a memory pattern*, obtained from a chaotic time series. The two distributions in the figure are for the 11th (solid line) and 19th (dashed line) memory patterns.

$$H(x)=\begin{cases}1 & \text{if} \quad x\geqslant 0\\ 0 & \text{otherwise,}\end{cases} \tag{5}$$

and $\delta\geqslant 0$ is a tolerance parameter. Specific values of $\delta>0$ are given in simulations shown later.

We will show how the itinerant behavior in the chaotic regime can be adapted by reinforcement learning using such external responses, i.e., we show that these responses can be used to adaptively constrain the chaos to concentrate on the subspace of patterns with the target features. We introduce the following learning rule,

$$\theta_i(t+1)=\theta_i(t)-\Delta s_i(t) \quad (i\in G_\alpha) \quad \text{if} \;\; E=0, \tag{6}$$

where $\Delta>0$ is a reinforcement constant. This modifies the threshold value of a neuron in the feature subset when the response is good, i.e., $E=0$, making it easier to regenerate the current neuron state.

Intuitively, adjusting thresholds is like adding external input in the sense that it biases or constrains the firing of neurons in the feature subset. If the number of neurons con-

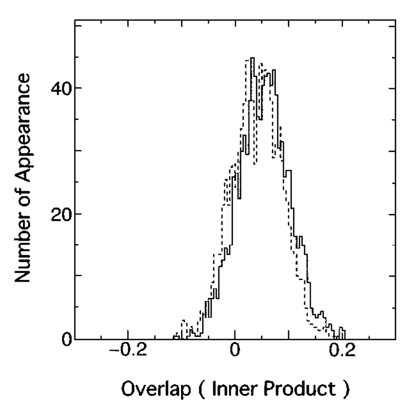

FIG. 4. Single-memory overlap distributions at $r=10$ after learning. The same distributions as in Fig. 3 [for the 11th (solid line) and 19th (dashed line) memory patterns], obtained after learning.

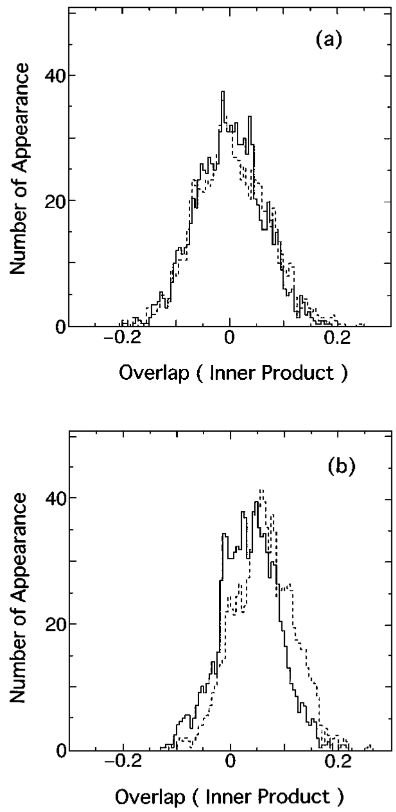

FIG. 5. (a) Single-memory overlap distributions at $r=20$ chaos before learning and (b) after learning, for the 11th (solid line) and 19th (dashed line) memory patterns.

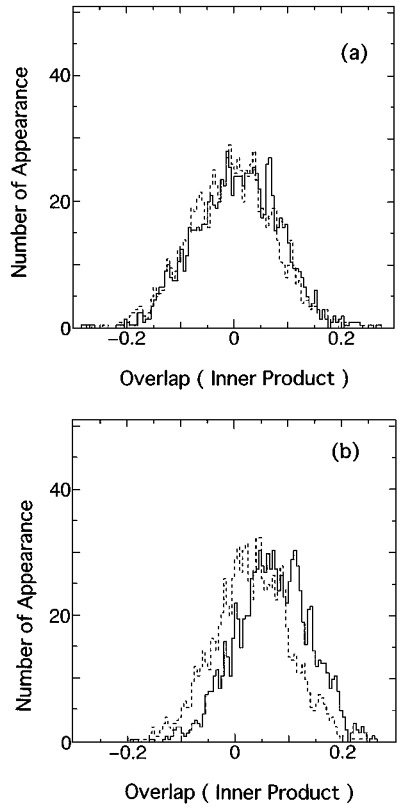

FIG. 6. (a) Single-memory overlap distributions at $r=30$ chaos before learning and (b) after learning, for the 11th (solid line) and 19th (dashed line) memory patterns.

strained, i.e., the number of neurons in the feature subset, is small compared to the total number of neurons, then the stability of memory at full connectivity will not be affected by the modification of the neurons in the feature set. Also, if the number of neurons constrained is small compared to the connectivity, then the constraining is not expected to ''kill'' the chaos at the reduced connectivity. So it is easy to imagine that it is possible to have arbitrarily constrained chaos for reduced connectivity and the same stable memories at full connectivity.

However, it is not so clear that we can achieve arbitrarily constrained chaos by reinforcement learning. It is easy to see that the one-shot ''obsessive'' case of large Δ will work so long as the chaotic itinerance visits a pattern that gets a good response $E=0$. However, what about the more general case when more than one pattern gets a good response (e.g., multiple different feature subset patterns in a neighborhood near the target feature pattern would get equivalent responses if tolerance δ is nonzero) and we want some averaging mechanism so the chaotic attractor shifts more gradually? Are the chaotic dynamics sufficiently robust and flexible to allow this type of adaptation?

One significant difference between the dynamics resulting from direct input bias or threshold settings, and that resulting from reinforcement learning, is that adjusting thresholds with reinforcement learning may result in nonuniform biases if more than one pattern receives reinforcement, the distribution of thresholds depending on the detailed history of the chaotic itinerance during the learning process.

In general, in stochastic (not necessarily chaotic) adaptation and reinforcement learning systems [14], it is expected that the choice of δ and Δ and the balance between them are significant to adaptation performance. To test the feasibility of the idea, we did some numerical experiments. We looked at the effect on statistical measures characteristic of single typical chaotic orbits, namely, the histogram of feature values, and the histogram of values of overlap with a particular memory, the ''single-memory overlap.'' We performed the experiments at connectivity values such that before the learning, both the single-memory overlap and feature overlap distributions resembled that expected from random pattern generation. (Though other measures, such as the projection on memory basins, may indicate that it is not uniform random pattern generation.)

Our main observations are as follows. It is indeed possible to do (gradual) adaptation, so that after a time the feature value distribution becomes centered on ''one.'' The single-memory overlap distribution may have suffered a shift, as a result of the constrained firing and dynamical correlations between neurons. This can be done without affecting the stability (apart from the details of basin boundaries) of the memories at full connectivity.

Also, we found that it is also possible to do adaptation in the more general case when *multiple* constraints are applied by providing responses to multiple feature sets. However, in the case of adaptation to multiple responses, whether learning is done sequentially or simultaneously can have significant effects. We will now illustrate these points with a specific numerical example.

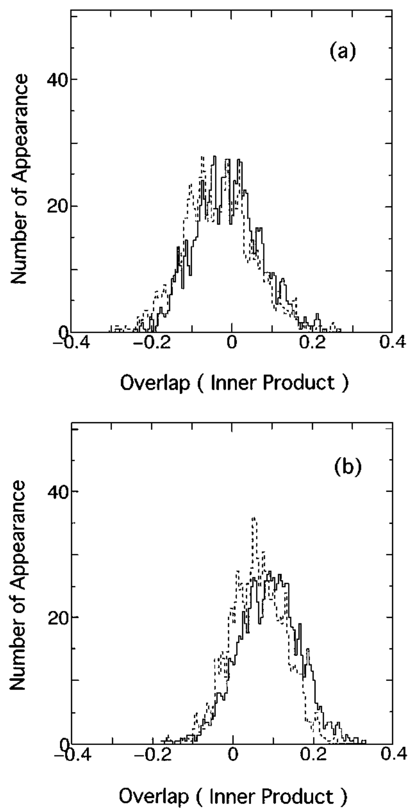

FIG. 7. (a) Single-memory overlap distributions at $r=40$ chaos before learning and (b) after learning, for the 11th (solid line) and 19th (dashed line) memory patterns.

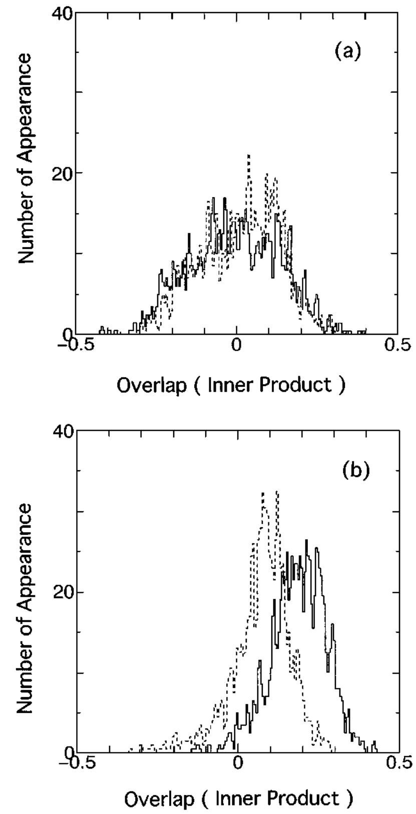

FIG. 8. (a) Single-memory overlap distributions at $r=50$ chaos before learning and (b) after learning, for the 11th (solid line) and 19th (dashed line) memory patterns.

IV. DEMONSTRATION OF LEARNING MULTIPLE CONSTRAINTS

In our numerical example, we choose two separate feature subsets, each corresponding to a single stripe in a $\sqrt{N}\times\sqrt{N}$ pixel pattern, where $\sqrt{N}\ll N$. This type of feature definition is relevant, for example, to searching for face patterns with a few specific facial features among an ensemble of face photographs. The dimension of the feature subsets $N_f=\sqrt{N}$ is much smaller than the dimension of the full pattern, which is the total number of neurons N. As target features, we choose a single-stripe pattern from one memory, and another stripe pattern from another memory. The total number of memories, a set of random patterns, is $LM\sim\sqrt{N}\ll N$.

The figures show the results for $N=400$, $LM=30$, and $N_f=20$. The memories with the (different) target features are memories number 11 and 19. The effect of learning is shown in Figs. 1 and 2, showing the statistics of feature subset pattern generation before (Fig. 1) and after (Fig. 2) learning, in the case where the connectivity parameter $r=10$. Figures 3 and 4 show the corresponding single-memory overlap distributions. The values of the learning parameters used were $\Delta=0.3$ and $\delta=0.2$. The dependence of learning on these learning parameters was not sensitive.

After learning, the network tends to generate patterns in which one feature stripe is almost the same as the corresponding stripe in the 11th memory and another feature stripe is similar to the corresponding stripe in the 19th memory, while the other $N-2N_f(=360)$ neurons not in the feature subsets continue to fluctuate wildly. The single-memory overlap distribution is still Gaussian but the peak position has shifted ~ 0.05. This is plausible because the dynamics has been constrained to have $N_f=20$ bits in common with each of the two memories (20/400 = 0.05). This amount of shift would be expected also from a random pattern generation in which there was no dynamical correlation between neurons and the remaining $N-2N_f$ fluctuated independently.

The dynamical correlations between neurons becomes apparent when we increase connectivity. Before learning, for instance, chaotic dynamics of connectivity $r=20$, 30, 40, and 50 all have similar Gaussian-shaped single-memory overlap distributions, as shown in Figs. 5(a), 6(a), 7(a), and 8(a). However, after learning, though the learning was done at the same connectivity $r=10$, the distributions differ, as shown in Figs. 5(b), 6(b), 7(b), and 8(b). Note that the shifts of each distribution are now larger than 0.05, which indicate that memory dynamics have moved closer to the 11th and 19th memories than would be expected from just clamping of the feature subset. The more we increase connectivity, the stronger the memory-pulling effect, and the more the memory overlap distribution is deformed from that before learning.

Next, we comment on the learning schedule. Generally speaking, since we have two constraints, there are three ways of learning, namely, (i) learn the 11th memory feature first, and then the 19th memory feature, or (ii) learn the 19th memory feature first, and then the 11th memory feature, or (iii) learn both features simultaneously. The results presented here correspond to the third case. The other cases did not seem to work as well, although we did not optimize learning by a thorough survey of parameters. However, in general it can be expected that the tendency to shift toward, or be pulled by, the memory with the target feature means that depending on the relative position of the memories in pattern space, it may become more difficult to learn one feature after another.

Another specific example of where adaptation was less successful, was for two feature subsets with 40 neurons each, all other parameters being equal. In this case, the chaos had a much stronger tendency to localize during adaptation, killing the fluctuation dynamics needed to drive the progressive adaptation.

V. DISCUSSION

We have shown that adaptation of chaos by reinforcement learning is possible. That is, the chaotic dynamics give a suitable sampling of the pattern space, so that target features can be found and moreover there is effective constraining of the chaos to the subspace of patterns with the target features.

It is not easy to give precise rules for what combination of the key parameters, such as number of neurons and memories, connectivity, feature subset size, tolerance, reinforcement constant, and adaptation, will be successful. But this is not a problem only of chaotic systems. The problem of how to optimize and maintain variance is a common difficult problem for any system using stochastic search methods for adaptation. However, it is significant to have specific examples that show how adaptation is possible.

An interesting issue in the context of constraining chaos in a network which at full connectivity has a set of memories, is the effect on the chaos of the memories. This is seen here in a tendency to shift toward the memories containing the target features observed at larger connectivities. That is, for the distribution to shift further than would be expected from just fixing the states of the subset of neurons relevant to the feature. In the case of multiple constraints there can be a simultaneous shift toward multiple memories. It is expected that if similar operations were done for two close memories (for example, in a hierarchical memory such as is possible with pseudoinverse embedding), the result would look like a constrained ''synthesis'' or dynamic montage of these memories. This implies interesting possibilities for interactive synthesis of memories.

Finally, we comment on the implications for memory search by adaptive bifurcation. In memory search by adaptive bifurcation [7,8], a system parameter is driven by external responses: for example, the connectivity parameter r can be driven as

$$r = N[1-(1-N_0/N)E] \tag{7}$$

(where $r=N_0$ is in the chaotic regime), so that a memory pattern receiving a good response $E=0$ will be stable, but memories or other states corresponding to reactions $E=1$ will be unstable and result in wandering in memory space. This allows the search among memories, in which the search time depends on the localization of the chaos on the neighborhoods of memories receiving good responses. The results of this paper show that with additional feedback to the thresholds of the specific neurons in the feature subsets, the network can learn from earlier searches to reduce search time in later searches by constraining the chaos to the subspace of patterns with the target features.

ACKNOWLEDGMENTS

This work has been partially supported by the Grant-in-Aid for Scientific Researches from the Ministry of Education, Science and Culture of Japan. Part of this work was done at ATR Optical and Radio Communications Research Laboratories.

[1] *Dynamic Patterns in Complex Systems*, edited by J. A. Kelso, A. J. Mandell, and M. F. Shlesinger (World Scientific, Singapore, 1988).
[2] J. S. Nicolis, Rep. Prog. Phys. **49**, 80 (1986).
[3] H. Haken, in *Neural and Synergetic Computers*, Vol. 42 of *Springer Series in Synergetics*, edited by H. Haken (Springer, Berlin, 1990).
[4] I. Tsuda, E. Koerner, and H. Shimizu, Prog. Theor. Phys. **78**, 51 (1987).
[5] P. Davis and S. Nara, in *Proceedings of the International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1990*, edited by Yamakawa (Fuzzy Logic Systems Institute, Iizuka, 1990), p. 617.
[6] P. Davis and S. Nara, in *Proceedings of the First Symposium on Nonlinear Theory and Its Applications*, edited by S. Oishi (EIC, Inakari, 1990), p. 97.
[7] S. Nara and P. Davis, Prog. Theor. Phys. **88**, 845 (1992).
[8] S. Nara, P. Davis, and H. Totsuji, Neural Networks **6**, 963 (1993).
[9] P. Davis, Jpn. J. Appl. Phys. **29**, L1238 (1990).
[10] T. Aida and P. Davis, in *OSA Proceedings on Nonlinear Dynamics in Optical Systems*, edited by N. B. Abraham *et al.* (Optical Society of America, Washington, D.C., 1991), Vol. 7, p. 540.
[11] P. Davis, in *OSA Proceedings on Nonlinear Dynamics in Optical Systems* (Ref. [10]), Vol. 7, p. 562.
[12] T. Aida and P. Davis, IEEE J. Quantum Electron. **30**, 2986 (1994).
[13] Y. Mori, P. Davis, and S. Nara, J. Phys. A **22**, L525 (1989).
[14] A. G. Barto, in *Vision, Brain and Cooperative Computation*, edited by M. A. Arbib and R. Hanson (MIT, Cambridge, 1987), p. 665.