pp. 376-380 in Neural Networks for Computing, American Institute of Physics Publication, Conference Proceedings 151, (New York, 1986)

Topology Conserving Mappings for Learning Motor Tasks

H. Ritter and K. Schulten
Physik-Department
Technische Universität München, D-8046 Garching

Abstract

Topology conserving mappings play an important role for biological processing of sensory input. We suggest that principles found capable of establishing such maps can also be applied to organize the learning of motor tasks. As an example we consider the task of learning to balance a pole.

Introduction

In this contribution we like to suggest that principles found capable of establishing topology conserving mappings between sensory input and cortical brain areas can also organize the learning of motor tasks. This is demonstrated for an example, the teaching of a robot device to balance a pole. Our study is motivated by the observation that the aquisition of many higher motor skills requires a period of consciously controlled generation of the respective movements until their execution becomes automatic. We model this behaviour by a control system which consists of three components: i) a controller C with output f^C capable of solving the control task at hand and acting as a teacher for ii) an initially unorganized array A of units y , which modifies the controller's output f^C to a value f^A and passes it to iii) the system S to be controlled (Fig.1). Both controller C and array A receive information about the current state x of the system S and the task of the array shall be to learn gradually to take over the function of the controller by establishing a topology conserving mapping between the states of the system and suitable control responses. For this purpose we employ a modified form of an algorithm which originally goes back to Kohonen[1,2,3,4] and which we have investigated recently in the context of somatotopic mappings[5].

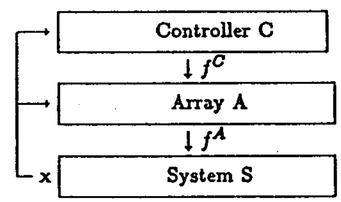

Figure 1.

The Model

The evolution of the system dynamics and the topology conserving mapping will be described in discrete timesteps $t_n = n\Delta t$. The array A will play the most important role in the following. It serves two purposes: First it monitors at each time step t_n the state x_n of the system S associating a particular unit $y^* \in A$ with the current state. Second, it associates responses $f^A(y)$ with each of its units y. The response $f^A(y^*)$ of the unit y^* selected at time t_n constitutes then the force which controls the system S until one timestep has elapsed and a new system state x_{n+1} is obtained.

Let X denote the state space of the system S to be controlled, A the array of units $y \in A$ and F the space of admissible forces to control S. With each unit $y \in A$ we associate at time t_n two vectors $\phi_n(y) \in X$ and $f_n^A(y) \in F$. The force for a given state $x \in X$ shall then be given by $f^A(y^*)$, where y^* is that particular unit for which $\phi_n(y^*)$ is closest to x, i.e. $|\phi_n(y^*) - x| = \min_{y \in A} |\phi_n(y) - x|$. This prescription specifies a (discretized) map Φ_n between states $x \in X$ and control responses $f \in F$ at time t_n. The initial map Φ_0 is chosen arbitrarily, e.g. is random. This map shall now be gradually transformed into a map sending state vectors x into adequate control actions f.

The desired final map should fulfill two demands: (1) Each of the units y shall be devoted to a small subregion of X and to a response $f^A(y)$, such that neighbouring units in the array belong to neighbouring subregions in X and to similiar responses f^A. (2) The resolution of the discretized map shall be fine in those regions of the state space X which are often realized by S and may be coarser in those regions which are less frequently assumed by the system. These are just the demands met by algorithms capable of establishing topology conserving maps between a sensory source and a cortical brain area[1-5]. The role of sensory signals for such maps is played in the present application by the sequence of pairs (x_n, f^C) produced by the time evolution of the state of S and the respective action of the "teacher" C. However, for a good mapping the dimension of X should not exceed the dimension of A. Although this is not so severe a restriction for a technical application, it is a difficulty in the biological case, where one would expect A to correspond to a two dimensional neural sheet thus allowing only a mapping of the two most relevant degrees of freedom. This problem seems to be overcome in the biological organism by using several suitably interacting neural sheets or by compressing higher dimensional spaces into 2-dimensional sheets at the price of discontinuities.

The goal of our algorithm is the gradual refinement of the map Φ_n from an initial random choice to a state where Φ_n can take over the control of the system S. For this purpose we suggest the following refinement procedure for Φ_n at each time step n:

1) Search for unit y^* with

$$\|\phi_n(y^*) - x_n\| = \min_{y \in A} \|\phi_n(y) - x_n\| \qquad (1)$$

where x_n is the state of S at time t_n.

2) Update Φ_n via

$$\begin{aligned} \phi_{n+1}(y) &= \phi_n(y) + h(y - y^*, t_n) \cdot (x_n - \phi_n(y)) \\ f^A_{n+1}(y) &= f^A_n(y) + h(y - y^*, t_n) \cdot (f^C - f^A_n(y)) \end{aligned} \qquad (2)$$

where f^C is the output of the controller C for state x_n and $h(y,t)$ is a function of Gaussian type centered at zero in its first argument and of width and amplitude decreasing with increasing second argument t.

3) Act upon S with a control force

$$f = \alpha(t_{n+1}) f^A_{n+1} + (1 - \alpha(t_{n+1})) f^C \qquad (3)$$

until the next time step. Here $\alpha(t)$ is a function which gradually increases from $\alpha = 0$ at t=0 to $\alpha = 1$ at the end of the learning.

Steps 1) and 2) have been shown to lead to a topology conserving map if the pairs (x_n, f^C) can be considered as a series of independent stationary random variables[1,2,4]. However here the developing map itself feeds back onto the source of its input by controlling the time evolution of S. This can modify the precise dynamics of the evolution of the map, but our simulations suggest that the properties of this process to converge to the topology conserving map remain preserved.

The advantage of the above algorithm over a control rule given by a fixed table of values ϕ and corresponding forces f^A lies in its capability to distribute the pairs (ϕ, f^A) over the space $X \otimes F$ with regard to the density of control actions required by the given motor task. This adaptation occurs automatically in the course of learning without the need for prior knowledge. Furthermore the spreading of the local adjustments into the immediate neighbourhood of a selected unit y^* in step 2 of the above refinement procedure brings about the topology conserving property of the resulting map and can be considered as a rudimentary form of generalization which facilitates convergency. The values f^A need not necessarily specify directly a control force. They can as well be input parameters of a lower level control law which serves to calculate the proper response actions from the parameter f^A and the system state x. From the view of the algorithm this amounts to replacing the original system S by a new system S', which is the concatenation of this control law and S.

Simulation Results

In the following we will show the results of the algorithm for the case of learning to balance a pole. Here the system S represents a massless rod with two point masses 1 and m attached to its upper and lower end, respectively. The rod's motion is restricted to a vertical plane with the mass m confined to glide along the horizontal x-axis and gravity pointing downward along the negative z-axis. θ denotes the counterclockwise angle subtended by the rod and the positive x-axis and balancing can be achieved by exerting a horizontal force f upon the mass m (Fig.2). The pole motion was simulated in time steps of 0.1s.

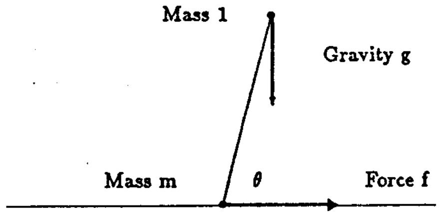

Figure 2

The state vector of the system S was represented by the last two successive pole inclinations, i.e. $x_n = (\theta_n, \theta_{n-1})$. The force delivered by the controller was given by

$$f^C = const \cdot (1.4 \cos\theta - 0.1\dot{\theta}). \qquad (4)$$

A two dimensional square array of 25x25 units was assumed for A. Learning was achieved by a series of trials, each trial starting with an initial value for θ drawn randomly from the interval $[60°, 120°]$ and lasting until either θ had left the interval $[0°, 90°]$ or 60 timesteps had elapsed.

Figures 3–6 below show the development of the map Φ_n during the simulation. Each of the left diagrams shows the distribution of the units y in the state space X of pairs of successive pole inclinations θ_n, θ_{n-1}. Each unit y is depicted at the location in X given by its associated vector $\phi_n(y)$. In order to illustrate the emerging topology conserving character of Φ_n we have connected those points $\phi_n(y^{(1)}) = (\theta_n^{(1)}, \theta_{n-1}^{(1)})$ and $\phi_n(y^{(2)}) = (\theta_n^{(2)}, \theta_{n-1}^{(2)})$ for which $y^{(1)}$ and $y^{(2)}$ are neighbouring units in A. The increased order of these connections in going from Figure 3 (initial map Φ_0) to Figure 6 (final map Φ_n, n=10 000) attests to the emerging topology conserving property of the $\phi_n(y)$.

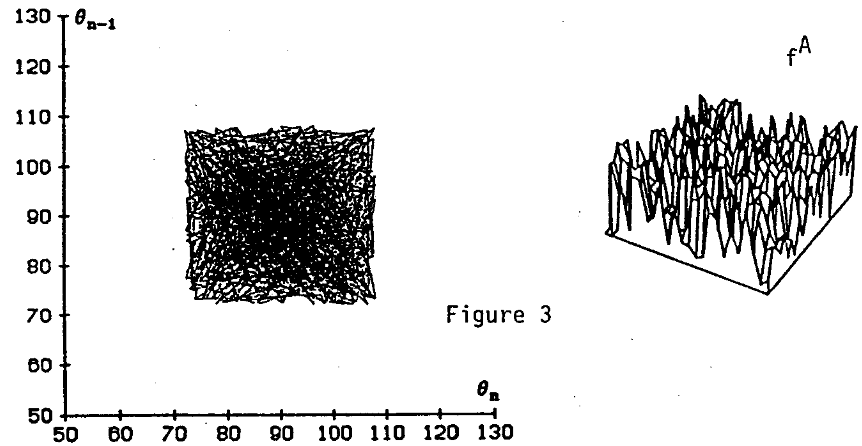

Figure 3

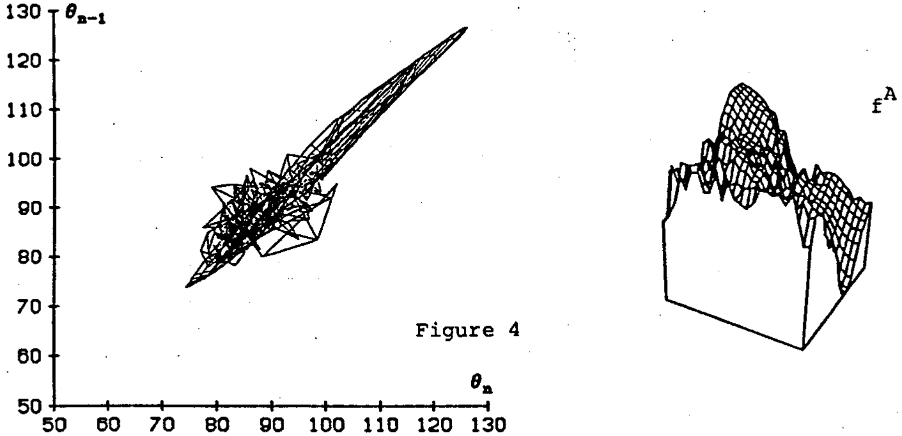

Figure 4

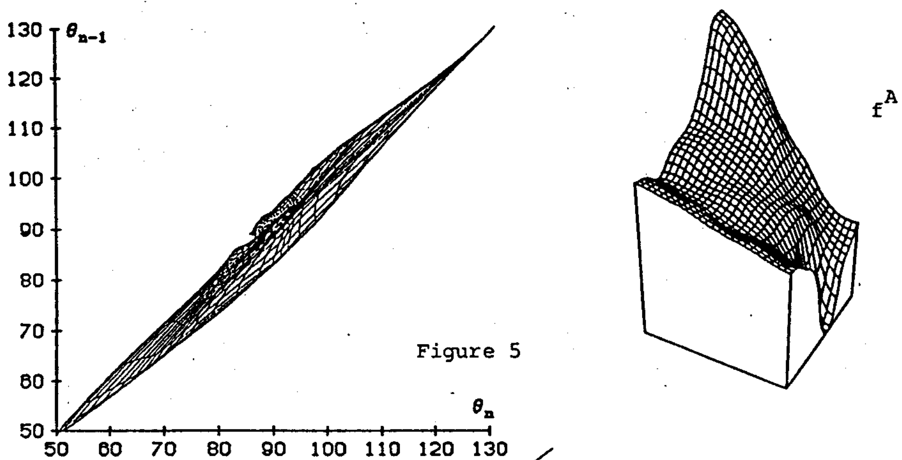

Figure 5

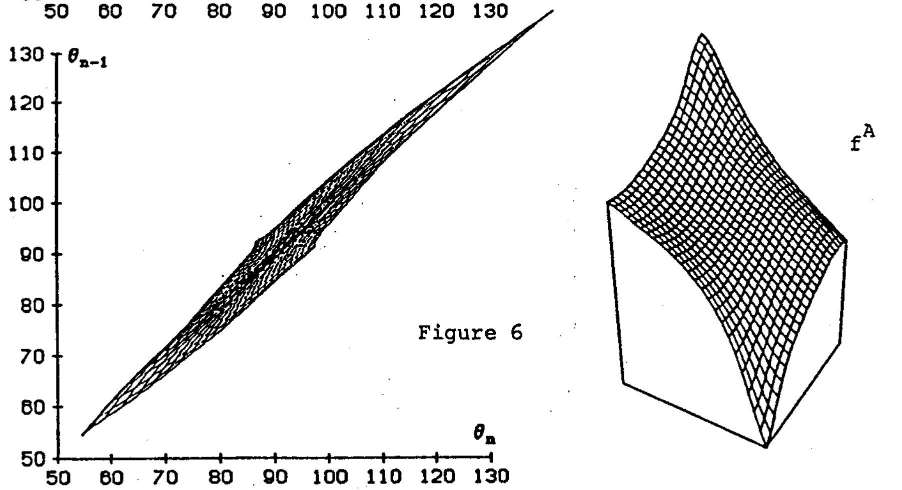

Figure 6

The right diagrams in Figs. 3–6 show the value of the force $f^A(y)$ as a mesh surface above the array A. Figure 3 presents the initial state of Φ. The values for both $\phi(y)$ and $f^A(y)$ for each unit y were chosen randomly from fixed intervals. The following figures 4 and 5 show the gradual development of the map after 100 and 1000 timesteps respectively. The units rapidly begin to "tune in" along the line $\theta_n = \theta_{n-1}$, which reflects the fact that successive pole inclinations θ usually differ only by a small amount and which therefore constitutes the essential region of X for the problem. Finally Fig.6 shows the asymptotic state reached after 10 000 time steps.

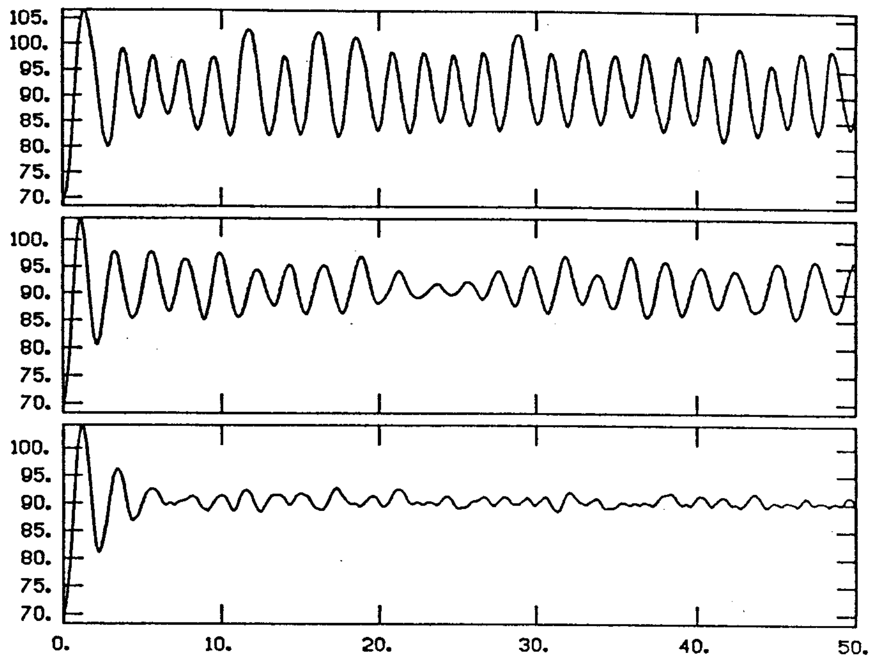

Figure 7

A smooth mapping has evolved and the controller now may be disposed of without significantly affecting the balancing performance of the system.

Actually the array is capable of performing the balancing task on its own considerably earlier, but at a correspondingly reduced level of performance. This is shown in Fig.7, which compares the time evolution of the inclination θ for a pole initially inclined at an angle of $\theta = 70^o$ under the control of Φ after 1000, 5000 and 10 000 time steps. During the time displayed in Fig.7 the controller had been disabled.

References

1. T.Kohonen, Biol.Cybern. 43, 59-69 (1982).
2. T.Kohonen, Biol.Cybern. 44, 135-140 (1982).
3. T.Kohonen, Proc. 6th Int. Conf. on Pattern Recognition 114-128 (1982).
4. T.Kohonen, Self-Organization and Associative Memory (Springer, Berlin 1984).
5. H.Ritter, K.Schulten, Biol.Cybern 54 (1986) (to appear).