Extending Kohonen's Self-Organizing Mapping Algorithm to Learn Ballistic Movements

Helge Ritter and Klaus Schulten
Physik-Department
Technische Universität München
D-8046 Garching
Federal Republic of Germany

Abstract: Rapid limb movements are known to be initiated by a brief torque pulse at the joints and to proceed freely thereafter (ballistic movements). To initiate such movements with a desired starting velocity **u** requires knowledge of the relation between torque pulse and desired velocity of the limb. We show for a planar two-link arm model that this relationship can be learnt with the aid of a self-organizing mapping of the type proposed earlier by Kohonen. To this end we extend Kohonen's algorithm by a suitable learning rule for the individual units and show that this approach results in a significant improvement in the convergency properties of the learning rule used.

1. Introduction

A most important task of biological or robot motor systems is the precise execution of movements of the limbs. In view of the great variability of body parts across individuals of biological species and over age it seems unlikely that this capability can be completely prewired into the nervous system. Instead it seems necessary for biological motor control systems to be adaptive and to rely to a considerable extent on learning. This same adaptive property would also be advantageous for robots which experience alterations of their limb characteristics through wear, may be outfitted with new limb parts during their life time, or need to adjust to new loads.

In this contribution we shall focus on the case of so-called ballistic movements of multi-jointed robot arms. Such motions are initiated by brief torque pulses acting on the joints. The pulses cannot be controlled through long-loop sensory feedback and, instead, need to be known before execution. We will show that the knowledge of the relationship between torques and the desired velocity of the arm's end effector can be acquired through suitable learning rules for the formal neurons of a computational network. This demonstration will be carried out for the most simple, non-trivial arm movement, namely that of a two-jointed robot arm confined to a plane (Fig.1, Section 2).

The robot learns during an exploratory phase the relationship between the torques

τ_1, τ_2 and the desired end effector velocity $\mathbf{u}_{desired}$. The learning procedure applied is based on Kohonen's algorithm [2-4] for the formation of topology (neigborhood) conserving mappings between a continuous feature space F (here the tuple of joint angles θ_1, θ_2) and a discrete net N of formal neurons and represents an extension of our previous work on the pole balancing problem [6]. We applied Kohonen's algorithm to map arm configurations to a net N, however, extended the algorithm by appending to each neuron $\mathbf{y} \in N$ a tensor $\mathbf{A}(\mathbf{y})$ which connects $\mathbf{u}_{desired}$ and τ_1, τ_2 by the relationship $\vec{\tau} = \mathbf{A}\mathbf{u}_{desired}$. These tensors are learnt through a comparison of the desired velocity $\mathbf{u}_{desired}$ and the velocity $\mathbf{v}_{actual}$ actually achieved at a particular state of the learning cycle. A most important feature of the extension is that the neighborhood conserving aspect of Kohonen's algorithm is applied to the learning of $\mathbf{A}(\mathbf{y})$: when a neuron updates its own tensor, its neighbors also participate in the adjustment. This cooperation results in a significant increase of the speed of convergency of the learning rule for $\mathbf{A}$ and its robustness to poor starting values.

2. Model and algorithm

We consider a planar two-jointed arm in the absence of gravity. For suitable ranges of the two joint angles θ_1, θ_2 each cartesian position $\mathbf{x} = (x_1, x_2)$ corresponds uniquely to a pair of joint angles. This enables us to equivalently use either the end effector coordinates or the joint variables to describe the arm configuration unambiguously. This property does no longer hold for an arm model with redundant degrees of freedom, the discussion of which shall be postponed to a subsequent paper.

The arm can be actuated by applying suitable torques at its two joints. If a torque $\mathbf{d}(t)$ is applied, the arm moves according to the equation of motion ([1]):

$$d_i(t) = \sum_j A(\mathbf{x})_{ij}\ddot{x}_j + \sum_{jk} B(\mathbf{x})_{ijk}\dot{x}_j\dot{x}_k. \tag{1}$$

Here $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are configuration-dependent matrices containing the information of the dynamics of the arm. We want to address the question, how the necessary values for these matrices can be learnt to be able to accelerate the end effector of the arm from any given initial rest position to a specified velocity $\mathbf{u} = (u_1, u_2)$ by applying a suitable short, but intense torque pulse (ballistic movement). The torque pulse is given by

$$d_i(t) = \tau_i \cdot \delta(t). \tag{2}$$

For the velocity $\mathbf{u}$ attained immediately after this torque pulse, Eq.(1) yields

$$u_i = \sum_j \mathbf{A}^{-1}(\mathbf{x})_{ij}\tau_j, \tag{3}$$

i.e. for a δ-shaped torque pulse the resultant velocity does not depend on the B_{ijk}'s. For a realistic torque pulse of finite width this result is only approximately valid, but for a sufficiently narrow pulse the error involved can be kept very small.

For any fixed starting position $\mathbf{x}$ the matrix $\mathbf{A}$ in the linear relationship (3) can be estimated by a simple learning rule of error-correction type ([4],[7]). Each movement trial generates a new approximation $\mathbf{A}(\mathbf{x}, t+1)$ using the following adjustment rule:

$$\mathbf{A}(\mathbf{x}, t+1) = \mathbf{A}(\mathbf{x}, t) + \epsilon\Big(\vec{\tau} - \mathbf{A}(\mathbf{x}, t)\mathbf{v}\Big)\mathbf{v}^T. \tag{4}$$

Here $\vec{\tau}$ is the amplitude of the torque pulse amplitude of the trial and $\mathbf{v}$ is the resulting actual velocity of the end effector (which during learning may differ from the desired $\mathbf{u}$). $\epsilon \in [0,1]$ is a fixed parameter (see below). This learning rule yields a sequence of successive approximations $\mathbf{A}(\mathbf{x}, t), t = 1, 2, ...$ to $\mathbf{A}(\mathbf{x})$ and is described and analysed in Section 4. However, $\mathbf{A}(\mathbf{x})$ must be learnt for all configurations $\mathbf{x}$. To this end we employ a set of units (formal neurons) arranged in a planar grid and labeled by indices $\mathbf{y} = (i, j)$. Each unit shall accomplish two things: it shall assign itself to a small subregion of the configuration space of the arm, and it shall learn the correct relationship between a desired velocity and the torque pulse amplitude for the arm being in this subregion. To achieve this, two quantities $\vec{\theta}(\mathbf{y}, t)$ and $\mathbf{A}(\mathbf{y}, t)$ are associated with each unit $\mathbf{y}$ at trial t. They represent two different kinds of data: whenever the current configuration $\vec{\theta}^*$ is closer to $\vec{\theta}(\mathbf{y}^*, t)$ than to any $\vec{\theta}(\mathbf{y}, t), \mathbf{y} \neq \mathbf{y}^*$, then unit $\mathbf{y}^*$ is selected to take over control for that configuration. For each movement, the tensor $\mathbf{A}(\mathbf{y}^*, t)$ associated with the selected unit $\mathbf{y}^*$ is used to calculate for the desired velocity $\mathbf{u}$ the necessary torque amplitude $\vec{\tau}$ according to Eq.(3). After each movement, the velocity $\mathbf{v}$ actually achieved is used to adjust the values $\vec{\theta}(\mathbf{y}, t)$ and $\mathbf{A}(\mathbf{y})$ for all units $\mathbf{y}$ in the neighborhood of the selected unit $\mathbf{y}^*$. The adjustment involves three steps: i) all $\vec{\theta}(\mathbf{y}^*, t)$ are shifted towards the configuration $\vec{\theta}^*$ from which the movement was started; ii) the error correction rule (4) is used to calculate an improved estimate $\mathbf{A}^*$ of the correct transformation between $\vec{\tau}$ and $\mathbf{u}$, based on the actual outcome of the movement; iii) all $\mathbf{A}(\mathbf{y}, t)$ for $\mathbf{y}$ in the neighborhood of $\mathbf{y}^*$ are shifted towards the improved estimate $\mathbf{A}^*$.

This procedure is summarized in the following equations:

o) Select unit $\mathbf{y}^*$ which satisfies

$$||\vec{\theta}(\mathbf{y}^*, t) - \vec{\theta}^*|| = \min_{\mathbf{y}} ||\vec{\theta}(\mathbf{y}, t) - \vec{\theta}^*||. \tag{5}$$

i) Adjust $\vec{\theta}$ for all units $\mathbf{y}$

$$\vec{\theta}(\mathbf{y}, t+1) = \vec{\theta}(\mathbf{y}, t) + h_1(\mathbf{y} - \mathbf{y}^*, t)\Big(\vec{\theta}^* - \vec{\theta}(\mathbf{y}, t)\Big). \tag{6}$$

ii) Choose a desired velocity $\mathbf{u}$ and execute movement with $\vec{\tau} = \mathbf{A}(\mathbf{y}^*, t)\mathbf{u}$. Use velocity $\mathbf{v}$ actually achieved to obtain the improved estimate $\mathbf{A}^*$

$$\mathbf{A}^* = \mathbf{A}(\mathbf{y}^*, t) + \epsilon\Big(\tau - \mathbf{A}(\mathbf{y}^*, t)\mathbf{v}\Big)\mathbf{v}^T. \tag{7}$$

iii) Adjust $\mathbf{A}$ for all units $\mathbf{y}$

$$\mathbf{A}(\mathbf{y}, t+1) = \mathbf{A}(\mathbf{y}, t) + h_2(\mathbf{y} - \mathbf{y}^*, t)\Big(\mathbf{A}^* - \mathbf{A}(\mathbf{y}, t)\Big). \tag{8}$$

The functions $h_i(\mathbf{y}-\mathbf{y}^*,t)$ in i) and iii) determine the neighborhood of $\mathbf{y}^*$ which receives significant adjustments. To this end $h_i(\mathbf{s},t)$ is taken to be a positive function of gaussian type with respect to s, centered at $\mathbf{s}=0$, whose width and height are slowly decreasing with iteration number t.

Performing step i) alone yields a neighborhood conserving mapping between the units y and the configuration space of the arm, i.e. close configurations are mapped to neighboring units in the ij-grid of the units. The generation of such mappings by the above procedure was first suggested and analysed by Kohonen ([2-4]) in the context of sensory mappings. Steps ii) and iii) are a natural extension of this algorithm: during the process of choosing a region in configuration space, each unit simultaneously learns a mapping between a desired movement (here given by velocity $\mathbf{u}$) and the required motor command (here represented by torque amplitudes $\vec{\tau}$) valid for this region. This approach has already been applied successfully to the pole-balancing-problem ([6]). As a further extension of Ref.[6], which dealt with supervised learning, we here supplement Kohonen's algorithm by the learning rule (4) which allows unsupervised learning. In Section 4 we will derive some properties of this learning rule in order to show in Section 5 that rules i)-iii) considerably improve both speed and range of convergency of each unit's tensor $\mathbf{A}$ to its correct value, compared to the case when each unit has to learn its tensor independently by means of Eq.(4).

3. Simulation of the Model

Before proceeding to a formal analysis in Secs.4 and 5 we shall illustrate the algorithm. We simulated for this purpose a network of 100 units arranged in a 10 × 10-grid. The arm providing the input to the algorithm consisted of two massless links of lengths 1 and 0.9 respecively, with unit point masses at its distal joint and end effector, respectively. During the learning phase we cycled repeatedly through steps o)-iii) above, each cycle providing one learning step. Before each cycle the end effector was positioned at a location $\mathbf{x}$ in the region R shown in Figure 1. For each cycle $\mathbf{x}$ was chosen randomly with uniform distribution over R, thereby providing the value $\vec{\theta}$ required for steps o) and i). To perform steps ii) and iii), a vector $\mathbf{u}$ for the desired velocity was chosen. $\mathbf{u}$ was chosen of unit length ($||\mathbf{u}||=1$) and pointing in a random direction with all directions equiprobable. The parameter ϵ in (7) was 0.25 and the functions h_1 and h_2 were taken to be Gaussians with respect to $||\mathbf{y}-\mathbf{y}^*||$ with identical initial amplitudes of 0.99 and widths of 2 lattice spacings.

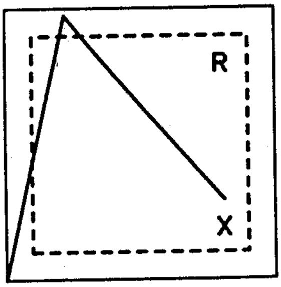

Figure 1

The width of both functions h_1 and h_2 was slowly decreased to a final value of 0.8 lattice spacings during the 3000 learning steps of the simulation. In addition the amplitude of h_1 was also decreased to a final value of 0.1. This had the effect of gradually diminishing the degree of plasticity of the network. The initial values for $\mathbf{A}$ were obtained by adding a random error of amplitude 2 to each of the matrix elements of the correct matrices.

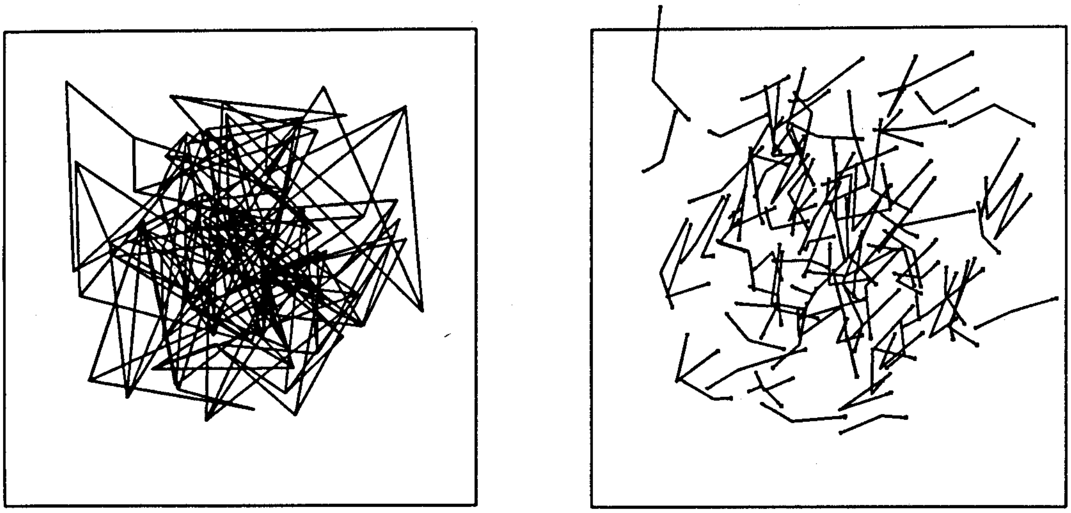

Figure 2

This resulted in the initial state shown in Fig.2. On the left of the figure the initial mapping between units $\mathbf{y}$ and end effector positions is shown. Each unit is drawn at the end effector position corresponding to $\vec{\theta}(\mathbf{y},0)$, positions belonging to lattice neighbors being connected by lines to make the neighborhood relations visible. Obviously there is very little initial regularity in the correspondence between units and end effector positions.

As for the matrices $\mathbf{A}$ a similiar convenient representation cannot be given, the right diagram instead shows the initial reaction of the system on movement requests.

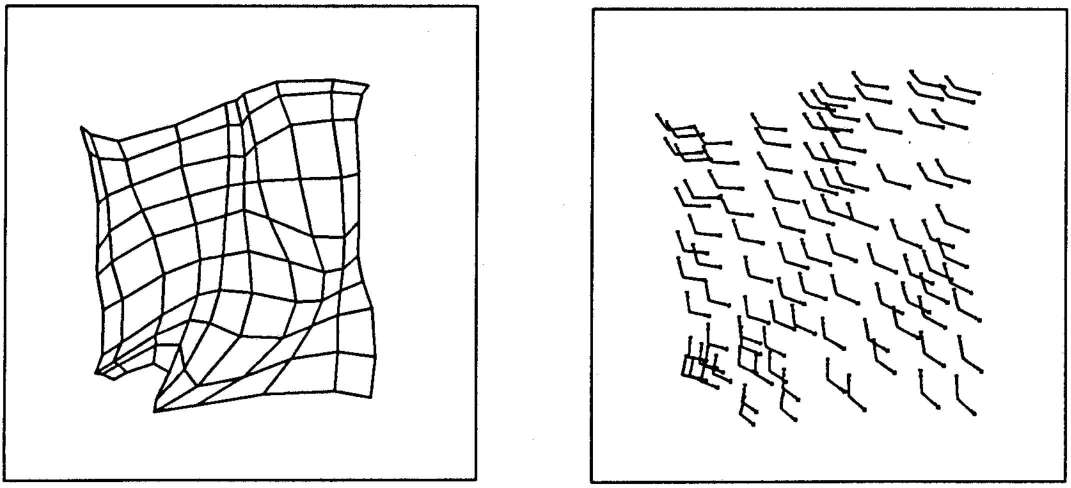

Figure 3

For each end effector position represented by a unit, the diagram shows the end effector velocity actually achieved by this unit, if either an upward or a rightward movement with unit velocity is required from the system: due to the random errors introduced

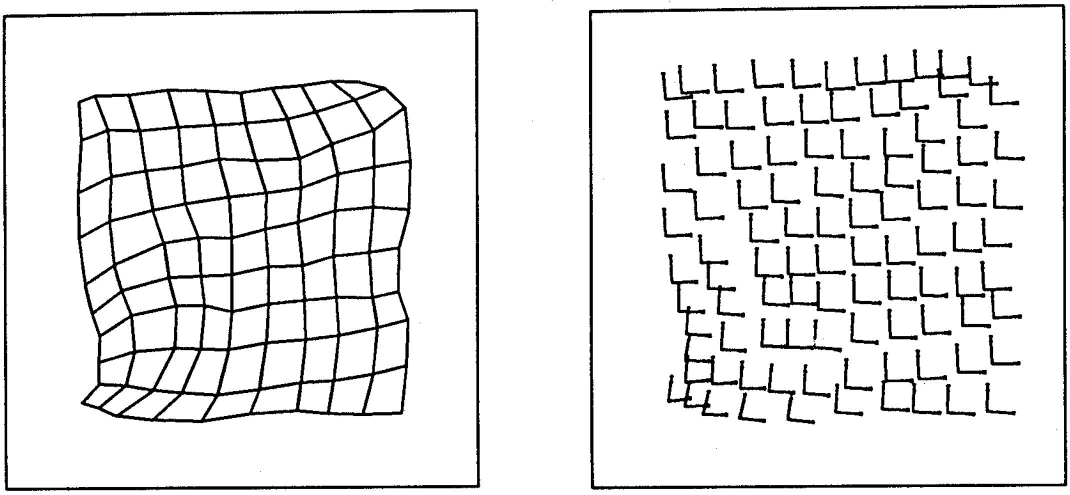

Figure 4

in the values provided for the matrices $\mathbf{A}(\mathbf{y},0)$ the responses are far from the desired ones. The situation resulting after only 300 learning steps is shown in Fig.3. Both, the mapping between units and end effector locations (left) and the execution of the test movements (right) has markedly improved. Finally after 3000 learning steps (Fig.4), the algorithm has found a satisfactory mapping between units and end effector positions and the units have learnt to respond quite accurately to the test movement requests [perhaps it should be stressed, that neither the learning steps nor the performance at any stage of learning is restricted to these test movements, which serve only as a convenient means to visualize the degree of convergency achieved by the matrices $\mathbf{A}(\mathbf{y},t)$].

To demonstrate the cooperation of neighboring units in the adjustments iii) of the matrices $\mathbf{A}(\mathbf{y},t)$, we have run a simulation from the same initial conditions as above, but with the width of the neigborhood for the adjustments iii) decreased to zero, i.e. $h_2(\mathbf{y}-\mathbf{y}^*,t) = \delta_{\mathbf{y},\mathbf{y}^*}$. In this case each unit learns in isolation from all others, relying only on the learning rule (4). As a result, only very few units manage to converge to the correct result, which is shown in Fig.5: only a minority of the units react correctly whereas most of them have been attracted to a wrong solution. The resulting state cannot be amended by providing more learning steps, and we shall show in the next Section that it is due to the presence of additional stationary points of the learning rule (4), from which, once reached, no escape is possible.

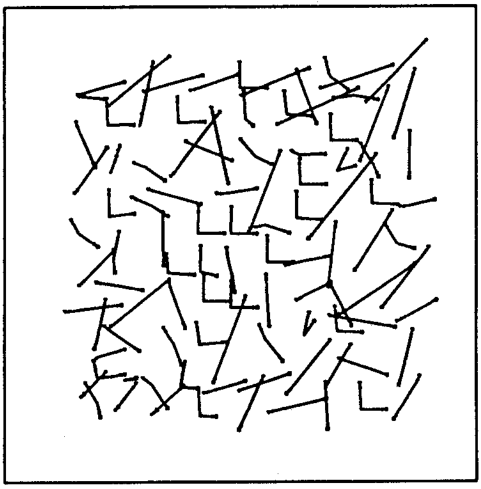

Figure 5

4. Analysis of the Learning Rule

In this section we shall analyse the convergency properties of the learning rule Eq.(4), when used to learn $\mathbf{A}(\mathbf{x})$ for a single configuration $\mathbf{x}$. For an analysis of more general aspects of this type of learning rule see e.g. [4],[7]. In the following we will drop the argument $\mathbf{x}$, i.e. in this section we denote the correct transformation by $\mathbf{A}_0$ and the estimate after t iterations by $\mathbf{A}(t)$. Thus Eq. (4) reads now

$$\mathbf{A}(t+1) = \mathbf{A}(t) + \epsilon\Big(\vec{r} - \mathbf{A}(t)\mathbf{v}\Big)\mathbf{v}^T. \tag{9}$$

As in (4), $\mathbf{v}$ is the actual velocity of the trial and $\vec{r}$ obeys $\vec{r} = \mathbf{A}_0\mathbf{v} = \mathbf{A}(t)\mathbf{u}$ with $\mathbf{u}$ denoting the desired velocity of the trial. Hence we have

$$\mathbf{v} = \mathbf{A}_0^{-1}\mathbf{A}(t)\mathbf{u}. \tag{10}$$

It is actually mathematically more convenient to consider the the convergence of a matrix $\mathbf{B}$ defined through $\mathbf{B}(t) := \mathbf{A}_0{}^{-1}\mathbf{A}(t) - \mathbf{1}$. We obtain from (9) and (10):

$$\mathbf{B}(t+1) - \mathbf{B}(t) = -\epsilon\mathbf{B}(t)\Big(\mathbf{1} + \mathbf{B}(t)\Big)\mathbf{u}\mathbf{u}^{\mathbf{T}}\Big(\mathbf{1} + \mathbf{B}(t)^T\Big). \tag{11}$$

For the following we assume that for each trial the desired velocity $\mathbf{u}$ is given by a bounded random variable independent of prior trials and with stationary distribution. Let $||.||$ denote the euclidean matrix and vector norms respectively, i.e. $||\mathbf{B}|| = (\sum_{ij} B_{ij}^2)^{1/2}$ and $||\mathbf{u}|| = (\sum_i u_i^2)^{1/2}$ and define $\alpha := \sup||\mathbf{u}||^2$. We then obtain for $\Delta\mathbf{B} := \mathbf{B}(t+1) - \mathbf{B}(t)$

$$\Delta||\mathbf{B}||^2 = \Delta\mathrm{Tr}\ (\mathbf{B}\mathbf{B}^T) = \mathrm{Tr}\ (2\Delta\mathbf{B}\ \mathbf{B}^T + \Delta\mathbf{B}\Delta\mathbf{B}^T),$$

where Tr denotes the trace operation. Inserting (11) we obtain

$$\Delta||\mathbf{B}||^2 = -\epsilon\ \mathrm{Tr}\ \Big(\mathbf{B}(\mathbf{1}+\mathbf{B})\mathbf{u}\Big[2 - \epsilon\mathbf{u}^T(\mathbf{1}+\mathbf{B})^T(\mathbf{1}+\mathbf{B})\mathbf{u}\Big]\mathbf{u}^T(\mathbf{1}+\mathbf{B})^T\mathbf{B}^T\Big), \tag{12}$$

or

$$\Delta||\mathbf{B}||^2 = -\epsilon\Big[2 - \epsilon||(\mathbf{1}+\mathbf{B})\mathbf{u}||^2\Big]||\mathbf{B}(\mathbf{1}+\mathbf{B})\mathbf{u}||^2. \tag{13}$$

If at step t the condition

$$0 < \epsilon < 2/\alpha(1 + ||\mathbf{B}(t)||)^2 \tag{14}$$

is fullfilled, Eq.(13) together with the inequality $||(\mathbf{1}+\mathbf{B})\mathbf{u}|| \leq (1 + ||\mathbf{B}||)||\mathbf{u}||$ tells us that $||\mathbf{B}(t+1)|| \leq ||\mathbf{B}(t)||$ and therefore (14) remains valid with t replaced by $t+1$. Thus the norms $||\mathbf{B}(t)||$, $t = 0, 1, 2..$ form a strictly monotonously decreasing sequence whose only stationary points can be at either $\mathbf{B} = 0$ or any nonzero $\mathbf{B}$ obeying $\mathbf{B} = -\mathbf{B}^2$. The latter possibility is ruled out if we require $||\mathbf{B}(0)|| < 1$. Thus we have shown

Theorem 1: Let $||\mathbf{B}(0)|| < 1$ and $0 < \epsilon < 2/\alpha(1 + ||\mathbf{B}(0)||)^2$ with $\alpha := \sup||\mathbf{u}||^2$. Then (11) entails $\lim_{t\to\infty}\mathbf{B}(t) = 0$, i.e. $\lim_{t\to\infty}\mathbf{A}(t) = \mathbf{A}_0$.

The stationarity condition $\mathbf{B}^2 = -\mathbf{B}$ shows that besides $\mathbf{B} = 0$ a whole manifold M of additional, non-vanishing stationary points exists. In the following we will demonstrate that at least a large submanifold of this manifold has an attractive neighborhood and, therefore, can be reached for suitable initial values not obeying the condition $||\mathbf{B}|| < 1$ of theorem 1.

The manifold M is formed by all matrices $\mathbf{B} \neq 0$ obeying $||\mathbf{B}(\mathbf{B}+\mathbf{1})|| = 0$. This motivates considering the change of the quantity $d(\mathbf{B}) = ||\mathbf{B}(\mathbf{B}+\mathbf{1})||^2 = \mathrm{Tr}\ \mathbf{B}(\mathbf{B}+\mathbf{1})(\mathbf{B}+\mathbf{1})^T\mathbf{B}^T$ under (11). Assuming ϵ sufficiently small to neglect terms of order ϵ^2 and using the cyclic properties of the trace, we obtain

$$\Delta d(\mathbf{B}) = -2\epsilon \mathrm{Tr}\ \mathbf{B}(\mathbf{1}+\mathbf{B})\Big[\mathbf{B}\mathbf{u}\mathbf{u}^T(\mathbf{1}+\mathbf{B})^T + \mathbf{u}\mathbf{u}^T(\mathbf{1}+\mathbf{B})^T(\mathbf{1}+\mathbf{B})\Big](\mathbf{1}+\mathbf{B})^T\mathbf{B}^T. \quad (15)$$

The general expression (15) is not very amenable to further analysis. Therefore we will focus on the case where (15) may be replaced by its average over the random variable $\mathbf{u}$ and assume $\mathbf{u}$ to be isotropically and independently distributed in every component, such that (after a suitable rescaling) we have $\langle \mathbf{u}\mathbf{u}^\mathbf{T} \rangle = \mathbf{1}$. This leaves us with

$$\begin{aligned}\langle \Delta d(\mathbf{B}) \rangle &= -2\epsilon \mathrm{Tr}\ \mathbf{B}(\mathbf{1}+\mathbf{B})\big(\mathbf{1} + 2\mathbf{B} + \mathbf{B}^T + \mathbf{B}\mathbf{B}^T + \mathbf{B}^T\mathbf{B}\big)(\mathbf{1}+\mathbf{B})^T\mathbf{B}^T \\ &= -2\epsilon \mathrm{Tr}\ \mathbf{B}(\mathbf{1}+\mathbf{B})\big(\mathbf{1} + \frac{3}{2}\mathbf{B} + \frac{3}{2}\mathbf{B}^T + \mathbf{B}\mathbf{B}^T + \mathbf{B}^T\mathbf{B}\big)(\mathbf{1}+\mathbf{B})^T\mathbf{B}^T \qquad (16) \\ &= -2\epsilon \mathrm{Tr}\ \mathbf{B}(\mathbf{1}+\mathbf{B})\mathbf{H}(\mathbf{B})(\mathbf{1}+\mathbf{B})^T\mathbf{B}^T.\end{aligned}$$

where the matrix $\mathbf{H}(\mathbf{B})$ is defined by

$$\mathbf{H}(\mathbf{B}) = \mathbf{1} + \frac{3}{2}\mathbf{B} + \frac{3}{2}\mathbf{B}^T + \mathbf{B}\mathbf{B}^T + \mathbf{B}^T\mathbf{B}. \quad (17)$$

We will now show that there are regions of the stationary manifold M where $\mathbf{H}$ is a strictly positive definite matrix and, therefore, $\langle \Delta d(\mathbf{B}) \rangle < 0$. Hence any point $\mathbf{B}$ sufficiently close to these regions is on the average attracted towards M. A condition for this to happen follows from

Theorem 2: Let $\mathbf{B}_0 := \sum_{i=1,n} \mathbf{p}_i\mathbf{q}_i^T$ where $\mathbf{p}_i, \mathbf{q}_i$ are $2n$ vectors whose scalar products obey the conditions

$$\mathbf{p}_i \cdot \mathbf{p}_j = 0, \quad \mathbf{q}_i \cdot \mathbf{q}_j = 0, \quad (i \neq j);$$

together with $||\mathbf{p}_i|| \cdot ||\mathbf{q}_i|| \geq 3/2$, $i = 1..n$. Then for any $\mathbf{B}$ sufficiently close to $\mathbf{B}_0$ we have $\langle \Delta d(\mathbf{B}) \rangle < 0$.

Proof: Define for $i = 1..n$:

$$\begin{aligned}\alpha_i :=& ||\mathbf{q}_i||; \\ \beta_i :=& \frac{3}{2||\mathbf{q}_i||} \leq ||\mathbf{p}_i||; \\ \mathbf{w}_i :=& \alpha_i\mathbf{p}_i + \beta_i\mathbf{q}_i;\end{aligned}$$

This yields

$$\mathbf{H}(\mathbf{B}_0) = \mathbf{1} + \sum_{i=1..n} \mathbf{w}_i\mathbf{w}_i^T + \sum_{i=1..n} (||\mathbf{p}_i||^2 - \beta_i^2)\mathbf{q}_i\mathbf{q}_i^T. \tag{18}$$

This shows explicitly the strict positivity of $\mathbf{H}(\mathbf{B}_0)$. As $\mathbf{H}$ depends continuously on its argument, this is valid in a whole neighborhood around $\mathbf{B}_0$ entailing $\langle \Delta d(\mathbf{B}) \rangle < 0$ there.

Two remarks may be necessary. First, there are many matrices $\mathbf{B}_0$ for which Theorem 2 holds, but which are far away from the manifold M, i.e. for which $||\mathbf{B}_0(\mathbf{1}+\mathbf{B}_0)||$ is large. For these starting values convergency towards M cannot be guaranteed on grounds of the above theorem, as successive iteration steps, though diminishing $||\mathbf{B}(\mathbf{1}+\mathbf{B})||$ on the average, may leave the neigborhood of the starting value for which this property holds. A sufficient condition for $\mathbf{B}_0 \in M$ is given by $\mathbf{p}_i \cdot \mathbf{q}_j = -\delta_{ij}$. Second, there are points in M, for which $\langle \Delta d(\mathbf{B}) \rangle < 0$ cannot be guaranteed for a whole neighborhood so that an approach to M cannot be ensured near these points. It can be shown that e.g. $\mathbf{B} = -\mathbf{1}$ constitutes such a case.

Summarizing this section we have proven the convergency of the learning rule (9) to the correct value, provided that the initial estimate for $\mathbf{A}$ is not too poor. The basin of attraction comprises at least the region $||\mathbf{A}_0^{-1}\mathbf{A} - \mathbf{1}|| < 1$ (Theorem 1). In addition there exists a whole manifold of different undesired fixed points, towards which the learning rule can converge (Theorem 2).

5. Convergency Improvement due to Network Properties

In this Section we will focus on the effect of the adjustment rule (8) on the convergence properties of the network as a whole. The essential property of (8) is, that the adjustments prescribed by the learning rule of the previous section are not only confined to the particular unit $\mathbf{y}^*$ optimally matching the current configuration, but in addition are partially spread also to all units within a certain neighborhood of unit $\mathbf{y}^*$. We shall show that this feature results in at least two benefits: First, it increases the rate of convergency and, second, it leads to an increased robustness of the network to poor initial values for the mappings to be learnt, i.e. even for initial values from which part of the units would not be able to converge to the correct mapping if they had to learn in isolation, successful convergency for all units is achieved with (8).

In order to carry out the analysis, we must make some simplifying assumptions. First, we assume that the ordering of the variables $\vec{\theta}(\mathbf{y},t)$ has already occurred and reached an asymptotic distribution such that the probability to be selected at step o) is equal for each unit. It has been shown elsewhere [3,4,5] that to approximate such a state is one of the prominent features of the adjustment rule i). Second, we will restrict ourselves to the case where the correct mapping $\mathbf{A}_0$ is independent of the configuration $\vec{\theta}$ and, therefore, is the same for each unit. We expect that the results of our analysis will not be affected in an essential way by dropping this condition, as the variation of $\mathbf{A}_0$ over the adjustment region given by the function h_2 in (8) usually will be only small. Further we shall take the function h_2 to be time-independent and assume that all adjustments are sufficiently small to drop quadratic and higher terms when necessary.

Taking these conditions and using the matrix $\mathbf{B}$ introduced in Section 4 instead of $\mathbf{A}$, we have to consider the algorithm:

o) Select unit $\mathbf{y}^*$ from the grid.

i) Choose a random desired velocity $\mathbf{u}$ and set

$$\mathbf{B}^* = \mathbf{B}(\mathbf{y}^*, t) + \Delta_L \mathbf{B}(\mathbf{y}^*, t) \tag{19}$$

where $\Delta_L \mathbf{B}(\mathbf{y}^*, t) = -\epsilon \mathbf{B}(\mathbf{y}^*, t)(\mathbf{1} + \mathbf{B}(\mathbf{y}^*, t))\mathbf{u}\mathbf{u}^T(\mathbf{1} + \mathbf{B}(\mathbf{y}^*, t))^T$ is the change of $\mathbf{B}(\mathbf{y}^*, t)$ if we apply only the learning rule of Section 4.

ii) Adjust all $\mathbf{B}(\mathbf{y}, t)$ according to

$$\mathbf{B}(\mathbf{y}, t+1) = \mathbf{B}(\mathbf{y}, t) + h_2(\mathbf{y} - \mathbf{y}^*)\big(\mathbf{B}^* - \mathbf{B}(\mathbf{y}, t)\big) \tag{20}$$

and go to step o).

Consider now the quantity

$$S(t) := \sum_{\mathbf{y}} ||\mathbf{B}(\mathbf{y}, t)||. \tag{21}$$

For each cycle o)-ii) we have

$$\begin{aligned} \Delta||\mathbf{B}(\mathbf{y}, t)||^2 &= 2\mathrm{Tr}\ \Delta\mathbf{B}(\mathbf{y}, t)\mathbf{B}(\mathbf{y}, t)^T \\ &= 2h_2(\mathbf{y} - \mathbf{y}^*)\mathrm{Tr}\ \Big[\big(\mathbf{B}^* - \mathbf{B}(\mathbf{y}, t)\big)\mathbf{B}(\mathbf{y}, t)^T\Big] \\ &\leq 2h_2(\mathbf{y} - \mathbf{y}^*)\big(||\mathbf{B}^*|| - ||\mathbf{B}(\mathbf{y}, t)||\big)||\mathbf{B}(\mathbf{y}, t)|| \\ &= 2h_2(\mathbf{y} - \mathbf{y}^*)\big(\Delta_L||\mathbf{B}(\mathbf{y}^*, t)|| + ||\mathbf{B}(\mathbf{y}^*, t)|| - ||\mathbf{B}(\mathbf{y}, t)||\big)||\mathbf{B}(\mathbf{y}, t)||, \end{aligned} \tag{22}$$

where we have written $||\mathbf{B}^*|| - ||\mathbf{B}(\mathbf{y}^*, t)|| =: \Delta_L||\mathbf{B}(\mathbf{y}^*, t)||$. Equation (22) gives us

$$\Delta||\mathbf{B}(\mathbf{y}, t)|| \leq h_2(\mathbf{y} - \mathbf{y}^*)\big(\Delta_L||\mathbf{B}(\mathbf{y}^*, t)|| + ||\mathbf{B}(\mathbf{y}^*, t)|| - ||\mathbf{B}(\mathbf{y}, t)||\big), \tag{23}$$

where we must keep in mind that $\Delta_L||\mathbf{B}(\mathbf{y}^*, t)||$ still depends on the random variable $\mathbf{u}$, which again shall be distributed with identity correlation matrix as in Section 4. Inserting (23) into (21), averaging over both $\mathbf{u}$ and the selected unit $\mathbf{y}^*$ and making use of the symmetry of h_2 we then arrive at

$$\begin{aligned} \langle \Delta S(t) \rangle_{\mathbf{y}^*, \mathbf{u}} &\leq \frac{1}{N} \sum_{\mathbf{y}, \mathbf{y}^*} h_2(\mathbf{y} - \mathbf{y}^*)\big(||\mathbf{B}(\mathbf{y}^*, t)|| - ||\mathbf{B}(\mathbf{y}, t)|| + \Delta_L||\mathbf{B}(\mathbf{y}^*, t)||\big) \\ &= \frac{h}{N} \sum_{\mathbf{y}^*} \langle \Delta_L||\mathbf{B}(\mathbf{y}^*, t)|| \rangle_{\mathbf{u}} \leq 0, \end{aligned} \tag{24}$$

where N denotes the number of units and we have set

$$h = \sum_{\mathbf{y}} h_2(\mathbf{y} - \mathbf{y}^*), \tag{25}$$

which is independent of $\mathbf{y}^*$ if we neglect edge effects. If at each time step any adjustment were exclusively confined to the unit $\mathbf{y}^*$ selected at that time step, i.e. $h_2(\mathbf{y}-\mathbf{y}^*) = \delta_{\mathbf{y}\mathbf{y}^*}$, we would have obtained the result (24) with h replaced by a value of unity. Therefore, the "lateral adjustments" increase the change of $||\Delta\mathbf{B}||$ per iteration step, i.e the rate of convergency, by a factor of h over the value without them. As h is a measure of the size of the neighborhood region over which the adjustments are spread, this neighborhood region should be made as large as possible. However, the conditions of this derivation require restraining the neighborhood to a region over which $\mathbf{A}$, respective $\mathbf{B}$, do not vary too much.

This result concerning the rate of convergency is still fairly general, as we hitherto have not used any special properties of the learning rule prescribing $\Delta_L\mathbf{B}$. These will be invoked in the subsequent paragraph to show that in addition to an increased rate of convergency the lateral adjustments lead also to an increased radius of convergency, i.e. to an enhanced robustness to poor starting values for the mappings of the individual units.

To this end we shall show the following lemmas:

Lemma 1: Let $h_2(\mathbf{y} - \mathbf{y}^*)$ be non-negative, symmetric with respect to interchange of $\mathbf{y}$ and $\mathbf{y}^*$ and non-vanishing at least for all nearest neighbor pairs $\mathbf{y}$ and $\mathbf{y}^*$ from the lattice. Then for the average change of the network per time step to vanish all norms $||\mathbf{B}(\mathbf{y}, t)||$ must be equal.

Proof: We consider the quantity $Q(t) := \sum_{\mathbf{y}} ||\mathbf{B}(\mathbf{y}, t)||^2$. For the change ΔQ between two consecutive time steps we have from (22) and $\Delta_L||\mathbf{B}(\mathbf{y}^*, t)|| \leq 0$:

$$\Delta Q \leq 2 \sum_{\mathbf{y}} h_2(\mathbf{y} - \mathbf{y}^*)\Big(||\mathbf{B}(\mathbf{y}^*, t)|| - ||\mathbf{B}(\mathbf{y}, t)||\Big)||\mathbf{B}(\mathbf{y}, t)||. \tag{25}$$

Averaging over $\mathbf{y}^*$ and using the symmetry of h_2 we finally obtain

$$\langle\Delta Q\rangle_{\mathbf{y}^*} \leq -\frac{1}{N} \sum_{\mathbf{y},\mathbf{y}^*} h_2(\mathbf{y} - \mathbf{y}^*)\Big(||\mathbf{B}(\mathbf{y}^*, t)|| - ||\mathbf{B}(\mathbf{y}, t)||\Big)^2. \tag{26}$$

Together with $h_2(\mathbf{y}-\mathbf{y}^*) > 0$ for all nearest neighbor pairs $\mathbf{y}, \mathbf{y}^*$ this proves the lemma.

Therefore all matrices $\mathbf{B}(\mathbf{y}, t)$ share the same fate with respect to their convergency to the desired fixed point $\mathbf{B} = 0$: either all of them reach $\mathbf{B} = 0$ or they all settle on the manifold M of undesired fixed points described in the previous Section. But once the average value of $||\mathbf{B}(\mathbf{y}, t)||$ in the lattice has fallen below a value of unity, Eq.(24) together with Theorem 1 of Sect.3 shows us that at least some units will converge to $\mathbf{B} = 0$ and as a consequence all others will have to follow, no matter how poor their initial starting values $\mathbf{B}(\mathbf{y}, 0)$ may have been. Without lateral adjustments, i.e. $h_2(\mathbf{y} - \mathbf{y}^*) = \delta_{\mathbf{y},\mathbf{y}^*}$, this consequence does not obtain. In this case Eq.(25) places no restriction on the norms $||\mathbf{B}(\mathbf{y}, t)||$ and the above Lemma does not apply. Therefore we have shown that the lateral adjustments enable units with starting values accidentally well within the basin of attraction of the desired fixed point to effectively increase the convergency zone for all other units of the array. As a result a fair proportion of units with very poor starting values can be tolerated before the capability of the array to globally converge to the correct solution is affected.

The bound on the critical average norm $||\mathbf{B}(\mathbf{y},t)||$ for convergency can be further refined. To this end we shall need

Lemma 2: For sufficiently small ϵ, the expectation value $\langle d(\mathbf{B}(t))\rangle_{\mathbf{u}}$ of the function $d(\mathbf{B}) = ||\mathbf{B}(\mathbf{B}+\mathbf{1})||^2$ subject to Eq. (15) obeys the inequality

$$\langle d(\mathbf{B}(t))\rangle_{\mathbf{u}} \geq d(\mathbf{B}(0)) \cdot \exp(-2\epsilon\lambda t), \tag{27}$$

where λ is any constant majorizing the matrix $\mathbf{H}$ of Eq.(17) over its whole time trajectory:

$$\lambda \geq \sup_{\mathbf{B}(t)} ||\mathbf{H}(\mathbf{B}(t))||. \tag{28}$$

(This is always possible, as $||\mathbf{H}||$ can be bounded by a polynomial in $||\mathbf{B}||$, which itself remains bounded).

Proof: From (16) and Tr $\mathbf{AB} \leq ||\mathbf{A}|| \cdot ||\mathbf{B}||$ we obtain

$$\frac{\langle \Delta d(\mathbf{B})\rangle_{\mathbf{u}}}{d(\mathbf{B})} \geq -2\epsilon||\mathbf{H}(\mathbf{B})|| \geq -2\epsilon\lambda. \tag{29}$$

For sufficiently small ϵ we may replace (29) by

$$\langle \Delta \ln d(\mathbf{B})\rangle_{\mathbf{u}} \geq -2\epsilon\lambda. \tag{30}$$

Thus

$$\begin{aligned} \langle d(\mathbf{B}(t))\rangle_{\mathbf{u}} \geq\ & \exp\Big(\langle \ln(d(\mathbf{B}(t))\rangle_{\mathbf{u}}\Big) \\ & \geq d(\mathbf{B}(0)) \cdot \exp(-2\epsilon\lambda t), \end{aligned}$$

proving the lemma.

We now have

$$\begin{aligned} \langle \Delta_L ||\mathbf{B}(\mathbf{y}^*,t)||\rangle_{\mathbf{u}} &= \frac{1}{2}\langle \Delta_L ||\mathbf{B}(\mathbf{y}^*,t)||^2\rangle_{\mathbf{u}}/||\mathbf{B}(\mathbf{y}^*,t)|| \\ &= -\frac{\epsilon}{2}\langle ||\mathbf{B}(\mathbf{y}^*,t)(\mathbf{B}(\mathbf{y}^*,t)+\mathbf{1})\mathbf{u}||^2\rangle_{\mathbf{u}}/||\mathbf{B}(\mathbf{y}^*,t)|| \\ &= -\frac{\epsilon}{2} d(\mathbf{B}(\mathbf{y}^*,t))/||\mathbf{B}(\mathbf{y}^*,t)||. \end{aligned} \tag{31}$$

Combining (24), (31) and Lemma 2 yields

$$\begin{aligned} \langle \Delta S(t)\rangle_{\mathbf{y}^*,\mathbf{u}} \leq & -\frac{\epsilon h}{2N}\sum_{\mathbf{y}^*}\frac{d(\mathbf{B}(\mathbf{y}^*,t))}{||\mathbf{B}(\mathbf{y}^*,t)||} \\ \leq & -\frac{\epsilon h e^{-2\epsilon\lambda t}}{2N}\sum_{\mathbf{y}}\frac{d(\mathbf{B}(\mathbf{y},0))}{||\mathbf{B}(\mathbf{y},t)||} \end{aligned}. \tag{32}$$

From this we can see that $||\mathbf{B}(\mathbf{y},t)||$ decreases on the average. Therefore the replacement of the denominator $||\mathbf{B}(\mathbf{y},t)||$ by $||\mathbf{B}(\mathbf{y},0)||$ should not destroy the inequality, leaving us with

$$\langle \Delta S(t)\rangle_{\mathbf{y}^*,\mathbf{u}} \leq -\frac{\epsilon h e^{-2\epsilon\lambda t}}{2N}\sum_{\mathbf{y}}\frac{d(\mathbf{B}(\mathbf{y},0))}{||\mathbf{B}(\mathbf{y},0)||}. \tag{33}$$

Integrating (33) gives our final result

$$\lim_{t\to\infty}\langle S(t)\rangle_{\mathbf{y}^*,\mathbf{u}} \leq S(0) - \frac{h}{2\lambda}D_0, \tag{34}$$

where

$$\begin{aligned} D_0 = & -\frac{1}{2N}\sum_{\mathbf{y}}\frac{d(\mathbf{B}(\mathbf{y},0))}{||\mathbf{B}(\mathbf{y},0)||} \\ = & -\frac{1}{N\epsilon}\sum_{\mathbf{y}}\langle \Delta_L ||\mathbf{B}(\mathbf{y},0)||\rangle_{\mathbf{u}}. \end{aligned} \tag{35}$$

The quantity $-D_0$ can be interpreted as the initial average change of $||\mathbf{B}||$ of a unit due to the learning rule of Section 4, normalized to $\epsilon = 1$.

From Eq.(34) we see that on the average each $||\mathbf{B}(\mathbf{y},0)||$ moves by an amount of at least $hD_0/2N\lambda$ closer to the desired fixed point $\mathbf{B} = 0$. This increases our previous bound of unity on the critical value for the average norm $||\mathbf{B}(\mathbf{y},t)||$ required for global convergency to $\mathbf{B} = 0$ to the new value of $1 + hD_0/2N\lambda$. Remarkably, the increment is again proportional to the strength of the lateral adjustments as measured by h.

6. Conclusion

We have presented an extension of the self-organized mapping algorithm proposed earlier by Kohonen [2-4] and have shown that the new algorithm can be applied to the unsupervised learning of ballistic movements. The novel features of the algorithm are i) the representation of the desired mapping as a collection of locally valid linear approximations to be learnt by Kohonen's original algorithm and ii) the use of a learning rule of error correction type to obtain values for the adjustment steps in Kohonen's algorithm on the basis of trial movements. We further have demonstrated, that employing the error correction rule in this fashion results in an increase of both its convergency rate and its range of convergency.

Finally we should like to remark that the applicability of the presented algorithm is not resticted to ballistic movements. We envisage its use also for other computationally similiar tasks, such as e.g. learning to control compliant robot arm motions from force feedback.

1. Brady M., Hollerbach J.M., Johnson T.L., Lozano-Perez T., Mason M.T. (eds): Robot Motion: Planning and Control, Cambridge Massachusets: MIT-Press 1984

2. Kohonen T.: Self-organized Formation of Topologically Correct Feature Maps. Biol. Cybern. **43**, pp. 59-69 (1982)

3. Kohonen T.: Analysis of a Simple Self-organizing Process. Biol. Cybern. **44**, pp. 135-140 (1982)

4. Kohonen T.: Self-Organization and Associative Memory, Heidelberg, Springer Series in Information Sciences 8 , 1984

5. Ritter H., Schulten K.: On the stationary state of Kohonen's Self-Organizing Sensory Mapping. Biol.Cybern. **54**, pp. 99-106 (1986)

6. Ritter H., Schulten K.: Topology Conserving Mappings for Learning Motor Tasks. In J.S. Denker (Ed.), Neural Networks for Computing, AIP Conf. Proceedings 151, Snowbird/Utah 1986

7. Rumelhart D.E., McClelland J.L.: Parallel Distributed Processing (Vol.1), Cambridge Massachusets: MIT-Press 1984