A Neural Network Model for Temporal Sequence Learning and Motor Programming

Bernard Ans, Yves Coiton, Jean-Claude Gilhodes, Jean-Luc Velay

A Neural Network Model for Temporal Sequence Learning and Motor Programming

BERNARD ANS,[1] YVES COITON,[2] JEAN-CLAUDE GILHODES,[2] AND JEAN-LUC VELAY[2]

[1]Université Pierre Mendès France, Grenoble, and [2]Université de Provence, Marseille

Abstract—*A neural network model for fast learning and storage of temporal sequences is presented. The recall of a learned sequence is triggered by the occurrence of an item relating to its identity, and one of the main distinctive features of this model is that the speed at which a sequence is repeated can be freely modulated by a control subsystem. The possible applications of the model are illustrated by applying it to the production of motor forms. It is shown that any spatial shape memorized in exteroceptive terms can be reproduced in terms of movement by any of the effector systems of the body, and in particular by a simulated jointed arm, at any point in its working space and at any suitable size scale. Our theoretical approach reinforces the idea that the structures responsible for planning a movement in the central nervous system might be largely independent of the motor systems performing this movement.*

Keywords—Dynamic memory, Motor program, Motor shape, Neural network, Pointing movement, Temporal sequence, Temporary memory, Tracking movement.

1. INTRODUCTION

Most of the existing neural network models for associative memory (Hinton & Anderson, 1981; Rumelhart & McClelland, 1986; Grossberg, 1988; Kohonen, 1988) have been of a static type. They have generally been designed to produce a single learned response to a given trigger stimulus, even if several processing cycles may be necessary for the expected response to stabilize completely. On the other hand, there exists a family of memory models of a more dynamic type, in which a specific temporal sequence of expected events is recalled in response to a single trigger stimulus or from a "seeding" of the system. However, this family of neural network models for temporal sequence learning and recall (e.g., Kohonen, 1977; Willwacher, 1982; Jordan, 1986; Dehaene, Changeux, & Nadal, 1987; Elman, 1988; Ans, 1990b; Reiss & Taylor, 1991), or dynamic memories in short, is not as large as that of static models because the temporal modelling of neural processes is not easy. This is nevertheless a really fundamental line of re-

search, if only because in ecological situations, the controlled production of a coherent chain of events in response to the requirements of the environmental context is a regular and by no means an exceptional occurrence. At the more theoretical level, the exact knowledge of the working of a connectionist system consisting of "neurons" and "synapses," which is able to perform a coherent sequence of "instructions" in response to a specific execute command, should be of great potential value as a means of understanding how sequential programs are actually implemented at the neurobiological level.

The existing connectionist models for dynamic memories are varied in regard to their degree of biological realism and their performances. One model close to neurobiology is that developed by Dehaene et al. (1987). The architecture of this model is based on formal synaptic triads that are responsible for the heterosynaptic regulation (Heidmann & Changeux, 1982) and the synaptic efficacies change in accordance with Hebb's rule (Hebb, 1949). From the behavioural point of view this model is of the selectionist type, which means that although the network is initially capable of spontaneously producing a very large variety of sequences (pre-representations), only those reinforced by imitative learning will persist, and the remainder will tend to disappear. Behaviour of this type has been studied with special reference to the acquisition of bird songs (Konishi, 1985). Although this approach is particularly

attractive and well suited to the task in question, it is less appropriate when the sequences to be learned are not part of a "genetically" predetermined repertoire and when the model is constantly faced with novel situations that have to be memorized.

The model architecture designed by Kohonen (1977) is a prototype that has inspired most subsequent models in some way or another. Kohonen's model involves an associative memory that stores the relation between the current item in a sequence and the ordered succession of its K predecessors. This recent sequence history is obtained by delaying the output from the associative memory by introducing K lag registers that form a spatial buffer. The length of the temporal span, that is, the size of the time window encompassing the current sequence history within the spatial buffer, is therefore fixed and determined by the number of lag registers used. In addition, the associative layer records the general context surrounding the sequence being processed that will subsequently serve to trigger the sequence recall as well as to solve some of the problems inherent to distributed memory systems, centering on the interferences between the sequences. Kohonen clearly set out the main principles underlying his one-layer model, without specifying, however, exactly which type of learning rule he uses. It is hence difficult to assess the efficiency of his model.

Among the descendants of this model, in which the sequential production of items is governed by their recent history and their surrounding context, the one developed by Jordan (1986) is characterized by the method used for short-term storage. The current sequence history is set up within a single layer forming a feedback loop: the state of the layer at instant t depends on its own previous state at instant t_{n-1} and on the item in the sequence at that previous instant. By a recursive process, the time span encompasses a recent history, the length of which depends on the structure of the sequence being currently processed. The author uses a layer of hidden units in which the current sequence history is combined with the context involved in the processing, which is conceived of as a "plan." The weights of the connections in hidden and output layers change according to the back propagation algorithm (Le Cun, 1986; Rumelhart, Hinton, & Williams, 1986). The model proposed by Jordan, like that by Elman (1988), which is an interesting variation on the former, is efficient in view of its long-term storage capacity. This quality is in fact due to the learning algorithm on which the processing is based, which has been intensively used elsewhere precisely because of its efficiency, despite the fact that it is reputed not to be a very realistic representation of neurobiological processes (Grossberg, 1987b; Crick, 1989; Hinton & Shallice, 1991; Reiss & Taylor, 1991). And in the present paper, where we propose a *neuromimetic* model for temporal sequence learning and recall, we do not use

a learning method of this type because one of our main aims here is to keep as close as possible to the well-known basic neural processes.

This principle of neurobiological plausibility is particularly adopted in Reiss and Taylor (1991) where back propagation is not consequently used. The authors propose both a theoretical analysis and a set of simulations of a model for temporal sequence storage in which the main distinctive feature is that the short-term storage of the current sequence history is obtained without using spatial buffering or a single layer with recurrent connections. It is built up within a layer of leaky integrator neurons, with a range of time constants, that store directly on their membrane a representation of the succession of their recent inputs. This model features a number of required properties, in particular: the sequences are quickly learned because there are no hidden layers to slow down learning; the system leads to recall in which there is no distortion of the temporality of the patterns forming sequences (the lengths of time each pattern was presented during the learning phase) and in which disambiguation is achieved (in that the sequence ABCBD . . . can be reliably recalled in spite of the fact that the pattern B (ambiguous) could lead either to C or to D). The other purpose of this paper is to evaluate the relevance of the model to understanding hippocampal structure and activity.

There is one natural ability that surprisingly has rarely been taken into account in the literature, namely, the ability to recall a sequence at a speed that can be modulated in the course of time. This property is nevertheless perfectly commonplace: we are able, for instance, to sing a tune we have learned with a variety of different tempos, or we can even choose to hold a single note in the sequence for a fairly long time, and subsequently, resume the normal course of the recall. The model of Reiss and Taylor (1991) would be able to account for the recall of a sequence at a modulated speed, and to continue a sequence that had been "held" at some point, except at ambiguous pattern points. The neural network model for temporal sequence storage, which is presented here, possesses the same set of general required properties as found in the model of Reiss and Taylor (we note, in particular, the use of the same realistic learning rule that achieves the same storage properties); but furthermore, the model we propose can recall a sequence at a freely modulated speed even at ambiguous pattern points, and gives an explicit neural implementation of this commonplace ability. This can be achieved because all the layers of our model are made up of "winner take all" (WTA) clusters with robust self-sustained states whose updating frequency is regulated by a common control module. In the event of the output of the system being maintained during recall at a given pattern, then the specific history of this pattern is self-sustained and entirely preserved in a temporary memory layer. On the basis of this nonre-

stricted temporal context, recall can resume whether the held pattern was ambiguous or not. In addition one of the other aims of this paper is precisely to show that the explicit neural modelling of the output speed control of a temporal sequence may be crucial in understanding the basic processes that determine the size of the spatial forms produced by living sensorimotors systems.

In Section 2 we give a detailed description of our neural network model for temporal sequence storage, leaving aside its potential applications; this model is partly based on an original architecture briefly described in a previous paper (Ans, 1990b).

The general dynamic memory model will be applied to the production of motor sequences in Section 3.

Most of our everyday gestures require the participation of numerous effectors that are coordinated in both time and space. Some of these movements, such as those involved in locomotion are highly automatized (Shik, Severin, & Orlovskii, 1966): in this case they call on a specific set of effectors and are largely controlled by subcortical structures. Motor activities of other types, such as writing and drawing, for example, rely more strongly on cortical structures and are not as strictly linked to one particular set of effectors. Movements of the latter type are probably based on centrally represented motor programs (Keele, Cohen, & Ivry, 1990), the spatiotemporal characteristics of which are invariable (Viviani & Terzuolo, 1980). These "motor forms" can moreover be produced in various formats and sizes without undergoing any change in their characteristic basic geometrical proportions (Bernstein, 1967). Last, these motor sequences can be repeated using a different set of effectors from those with which the movement was initially learned. For example, a child who has learned to write on a horizontal sheet of paper, using mainly his fingers and wrist, will also be spontaneously able to write on a vertical blackboard, using his wrist, elbow, and shoulder muscles. These motor behaviours are in accordance with the hypothesis that the structure responsible for planning a movement may be largely independent of the motor system performing this movement.

Dynamic memories seem to provide a suitable representation for the structures that may be at the root of the movement planning. The undedicated dynamic memory model, described in Section 2, was subsequently coupled with a sensorimotor system capable of performing goal-directed arm movements. This system was given the neuromimetic sensorimotor architecture we developed separately in a previous modelling study on goal-directed movements (Coiton, Gilhodes, Velay, & Roll, 1991; Gilhodes, Coiton, & Velay, 1991). It consists of a Kohonen layer (Kohonen, 1988) where sensory information of two types is combined in a functional map. This sensory layer controls a motor layer that drives the effectors of either a numerically simulated arm or an artificial jointed arm. After a

learning phase, the arm is able to perform movements aimed toward either a fixed target (pointing movements) or a moving target (tracking movements). The coupling between the dynamic memory model and the sensorimotor system was achieved by means of a neuromimetic interfacing module.

2. A NEUROMIMETIC MODEL FOR DYNAMIC MEMORY

2.1. General Description of Model

The model, which is schematized in Figure 1, was designed to be able to learn, memorize, and recall temporal sequences. It consists essentially of five layers, the architecture and working principles of which will be described in detail in Section 2.2. The inputs and outputs of these layers consist of multidimensional patterns with components corresponding to the mean firing rate of real neurons. The architecture receives two main input patterns with either graded or discrete values: the first input, $Z(t)$, represents a temporal sequence of patterns that requires learning, and the second input, ID, relates to the identity of the ongoing sequence (e.g., its "name," or more generally, its surrounding context). This identification key has to be maintained constantly throughout the processing (learning and recall) of its associated sequence $Z(t)$.

During the learning phase, the input layers F and C recode the input patterns $Z(t)$ and ID, transforming them into binary valued patterns. The activity pattern
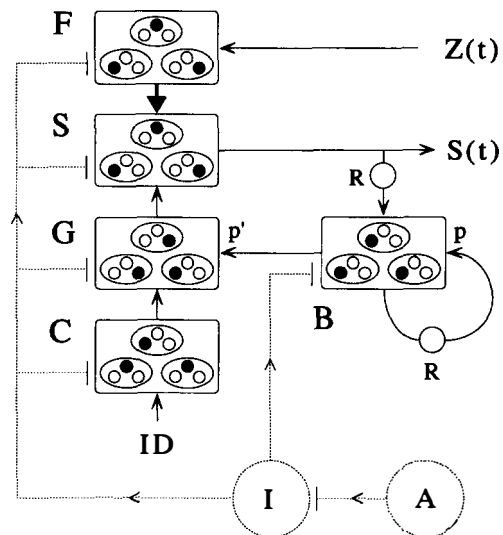
FIGURE 1. The architecture of the temporal sequence-learning model. Each rectangular module is a layer consisting of units (small circles) arranged in separate clusters (enclosed units). To illustrate this, the number of units per cluster and the number of clusters in each layer were arbitrary chosen. In each cluster, only one unit is active (black circle) and the others are silent (white circles). The input connectivity to all the layers is of the distributed type, except for that from F to S, which is of a specific type (heavy arrow).

$F(t)$ of layer F is transmitted topographically to layer S (these two layers both have the same structure) via strong specific excitatory connections, which leads to $S(t) = F(t)$. Hence the temporal sequence $S(t)$ occurring in layer S is a copy of $F(t)$ and conveys in a recoded form the original input sequence $Z(t)$. Due to the intervention of the time-lag modules R, the current state of layer B is a global representation $B(t)$ of the states preceding the current state $S(t)$ of layer S within a variable time window: layer B is a kind of temporary memory that preserves the current history of the sequence $S(t)$.

Layer G combines this current history $B(t)$ with the constant activity pattern C identifying the sequence being processed. Then layer S stores the association between its present state $S(t)$ and the pattern $G(t)$ representing its history in the light of the sequence identifier. Several sequences, each presented with its own separate identification key, can be stored in this way in the permanent memory S after being processed several times.

In the sequence recall phase, the input layer F is no longer activated and it is the occurrence of the identity pattern ID that triggers the recall of its associated sequence reproduced at the output $S(t)$ in a different format from that of the original sequence $Z(t)$. The permanent memory S progressively reconstructs the successive items by means of the maintained identifier and the temporary memory, which itself is built up dynamically, exactly as in the learning phase.

As we shall see below, the pattern of activity of each of the five main layers of the model is preserved in a robust self-sustained state that cannot be modified unless a resetting is induced by an inhibitory control process (A, I) that periodically updates the whole architecture.

2.2. Structure and Working Principles of Layers

Each of the layers F, S, G, C, and B (Figure 1) is composed of a set of nonoverlapping and unconnected *clusters* of elementary neuron-like units. The numbers of units inside the clusters are not necessarily the same and the number of clusters composing each layer may also change in the different layers. These numbers are structural parameters that have to be chosen in simulations. Each unit within a cluster activates itself and inhibits all other units in the cluster (see also Figure 2a showing an example of a cluster structure), and this fixed prewiring subserves a competitive process between the units inside each cluster. Grossberg (1973, 1987a) has developed a neurobiologically plausible mathematical model for this process where the continuous variation in time in the activity of the competing units can be obtained from a system of differential equations. This author (Grossberg, 1987a) has proposed that this competitive process, which is costly to model in detail, can be simulated using a simple choice algorithm: within a cluster only the unit receiving the largest external input activation (the winning unit) achieves its maximum value, whereas all other units in the cluster become silent (WTA cluster). We have adopted here this contrast-enhancement process that behaves approximately like a binary switching and is currently in use in neural network models, particularly in those with special references to neurobiological correlates (Dehaene, Changeux, & Nadal, 1987; Dehaene & Changeux, 1989, 1991; Strong & Whitehead, 1989). The temporal properties of the intracluster competitive process, in the scope of the time sequence processing, will be detailed in the case of the input layer F (see Section 2.2.1) that has the same basic working principles as that of the four other layers.

(a)   *(b)*
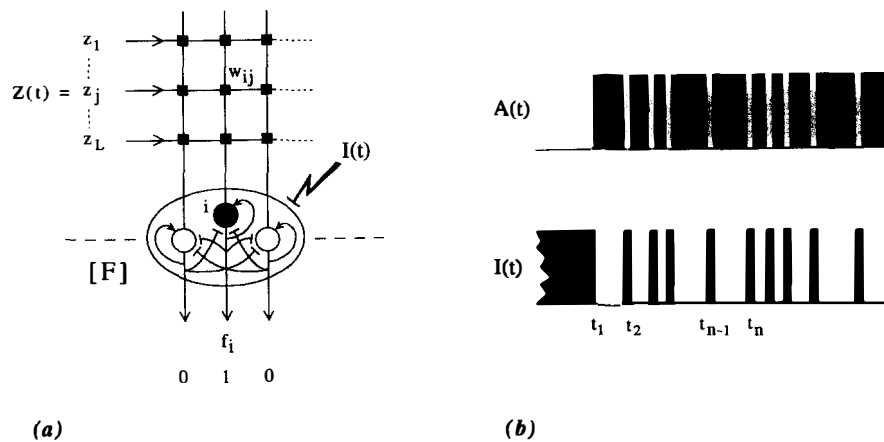
FIGURE 2. (a) Activation of one of the clusters in layer F by applying the multidimensional temporal sequence $Z(t)$. The vertical line above unit i is a simplified dendritic tree receiving the horizontal inputs z_j to which a synaptic weight w_{ij} has been assigned. The connectivity within the cluster subserves a competition mechanism. $I(t)$ is an inhibitory signal that serves to reset the whole layer. (b) Diagram showing the opposition between the adjustable frequency signal emitted by the control device A and the inhibitory resetting signal $I(t)$.

The input connectivity to all the layers, except that concerning the specific wiring from layer F to layer S, is of the distributed type, that is, each single unit in a given source layer (or each component of the multi-dimensional external inputs Z or ID) is connected to all the units in the corresponding target layer. It is in particular the same for the spreading connections to layer B, but in this case the activity patterns originating from the source layers (i.e., from layer S and from layer B itself through recurrent connections) are first delayed in time by lag modules R (see Section 2.2.2). In fact, the model does not necessarily require such a complete connectivity and a random partial wiring may suffice when the five layers contain large numbers of clusters and units per cluster.

The one exception referred to above is the connectivity between layers F and S, which is of a specific type. Here each unit belonging to a given cluster in F is connected only to the corresponding unit in the homologous cluster in S. These one-to-one excitatory links, which have the same fixed positive synaptic weights, mean of course that the two layers both have the same structure, that is, the same number of clusters and the same number of units in each cluster. The other interlayer distributed connections, and those involving the external inputs, are also all excitatory. They are weighted with positive synaptic efficacies initially chosen at random, and remain subsequently unchanged, except for the weights of the connections from G to S that can be modified according to a synaptic plasticity rule.

2.2.1. *Layer F*. Figure 2a shows one of the clusters constituting the input layer F. An input sequence Z is represented by a time-varying vector $Z(t) = [z_1(t), z_2(t), \ldots z_j(t), \ldots z_L(t)]$, whose particular forms of time variation will be discussed in Section 2.3.3. This input pattern activates all the units i in all the clusters constituting the layer, via links weighted with fixed, positive synaptic efficacies w_{ij}. In the simplest case, where vector Z is a single pattern with no temporality, and none of the units in the layer have yet been activated, the competition within the cluster can be expressed in more formal terms as follows. Within a cluster, only the unit receiving the strongest external input activation $a_i = \sum_j w_{ij} z_j$ (the winning unit) has an active output ($f_i = 1$ by convention), whereas the other units (the losing units) have a null output. Hence a given item Z is transformed into a binary valued pattern by all the clusters in layer F, so that for example $F = [(010)(001) \ldots (010)(100)]$, assuming by convention (as we shall do throughout Section 2) that a layer and its output have the same name. Provided that the size of input Z and the number of clusters in the layer are not too small, the minimum hypothesis consisting of choosing the weights w_{ij} at random should yield a

wide range of distinct states F in response to a large set of different vectors Z.

In the most general case, that where an input $Z(t)$ changes constantly with time, it is necessary to introduce an external means of controlling the recoding process in layer F. The competitive mechanism is a very fast one, and at time $t = t_1$ for instance, layer F "captures" the instantaneous state $Z(t_1)$ of the input according to its specific format $F(t_1)$. The winning unit in each cluster subsequently freezes, however, into a robust, self-sustained state, and the layer becomes insensitive to any further changes in the sequence $Z(t)$: for $t > t_1$, layer F remains in the steady state $F(t_1)$. This insensitivity has been tested on low-level simulations using the system of differential equations proposed by Grossberg (1973, 1987a), which expresses competition. For the clusters to become receptive again to the input sequence, they have to be reset by an input $I(t_2)$ that at $t = t_2$ delivers a brief but intense inhibitory signal to all the units in the layer. Just after the end of this short inhibition, the layer becomes able to quickly pick up the current state $Z(t_2)$ in the sequence in the form $F(t_2)$. The process then continues in the same way, so that the sequence $Z(t)$, whether its values are graded or not, is sampled at each instant $t = t_n$ whenever the updating signal $I(t_n)$ occurs: $F(t) = \text{constant} = F(t_n)$ for $t_n < t < t_{n+1}$, corresponding to the capture of the current state $Z(t_n)$.

The reset signal is triggered by a permanent source of inhibition I, which in turn is under the inhibitory control of a subsystem A (Figure 1): when A is active, source I is inhibited (inactive) and conversely, when A is inactive, source I is released (active). Figure 2b illustrates this opposition by showing that the intermittent updating signal resulting from I is generated by short breaks in the activity of A. The control device A (which was not modelled here) is assumed to be capable of modulating the frequency of the breaks, and thus to be responsible for determining the resolution at which the temporal sequence $Z(t)$ is sampled. This control device is initially at rest, so that layer F is spontaneously inhibited and thus maintained in the inactive state. When an input sequence $Z(t)$ occurs, module A needs to be activated, and the inhibitory influence of I on layer F is therefore suppressed so that the first sample $F(t_1)$ of an early state $Z(t_1)$ of the input sequence begins to take place.

2.2.2. *Other Layers*. The basic principles we have just outlined in the case of layer F also apply to the other layers. The inhibition I that is necessary for zeroing to occur has to be conveyed to all the layers in the architecture if we want them to be simultaneously updated and the information to circulate throughout the whole network. The speed at which the information circulates depends entirely on the frequency of the updating signal $I(t)$.

The role of layer C is similar to that of F: it recodes and samples the identity pattern ID, which has to be kept constant over time, $C(t_n) = C$, while the associated sequence $Z(t)$ is being processed.

Layer S is similar in structure to layer F because both layers are linked together by specific connections: these are assigned with weights that are large in comparison with those assigned to the distributed connections from G to S. Consequently, when F and G simultaneously activate S, the outcome of the competition inside the clusters in S will depend only on the influence exerted by F. The winning cells in S will therefore be homologous with those in F, and hence $S(t_n) = F(t_n)$ at each updating $I(t_n)$.

Layer B is distributively activated by the delayed output from layer S and by its own delayed activity through self-recurrent connections. The lag modules R, which have not been explicitly modelled, are assumed to transmit a change in their input to their output after an interval that can be fairly short because the state information is picked up quickly in the layers. At each updating of the system at $t = t_n$, layer B acquires a state that depends both on the previous state $S(t_{n-1})$ of layer S and on its own previous state $B(t_{n-1})$, corresponding to the previous updating that occurred at t_{n-1}. If we take ϕ to denote the layer B output function, this yields the following expression:

$$B(t_n) = \phi[S(t_{n-1}), B(t_{n-1})].$$

This equality can be successively written as:

$$B(t_n) = \phi[S(t_{n-1}), \phi[S(t_{n-2}), B(t_{n-2})]],$$

$$B(t_n) = \phi[S(t_{n-1}), \phi[S(t_{n-2}), \phi[S(t_{n-3}), B(t_{n-3})]]],$$

and so on,

$$B(t_n) = \phi[S(t_{n-1}), \phi[S(t_{n-2}),$$
$$\phi[S(t_{n-3}), \ldots \phi[S(t_{n-k}), B(t_{n-k})] \ldots]]],$$

where it can be seen that at each updating t_n, layer B contains a representation of the current history of the sequence $S(t)$, the length of which is limited because the diminishing influence exerted by a remote past state of S will disappear. This layer is a temporary memory, the state of which can remain preserved as long as the control module A continues to be active without any interruptions. The size of the time window encompassing the current sequence history (the span size of the temporary memory) depends on several factors, including the ratio between the strength of the self-activation of layer B and that of layer S, which is mainly a question of the relative sizes of these two layers. Here a parameter p weighting the self-activation of B was introduced as a means of dealing with this source of variation in the simulations. This overall parameter can be taken to account for example for variations in the density of the axonal branches that convey feedback information to the temporary memory. If p is very small

or equal to zero, for instance, the temporary memory will contain only the most recent previous state in the sequence (span size = 1), because only the influence of S is taken into account in the competitive process; when p increases, a concomitant increase in the size of the time span occurs. If p becomes too large, however, the competitive process will no longer be affected by the activation of S, and layer B will become merely a sequence generator bearing no relation to the input $Z(t)$. On the other hand, the current size of the time span also depends on the structure of the sequence. These performances, which can easily be predicted at the theoretical level, were checked by carrying out simulations.

In an early version of the model (Ans, 1990b), the temporary memory consisted of a series of K layers, each of which was specifically activated by the lagged output from the previous layer. One advantage of a structure of this kind can be that a trace of each individual item is stored in the temporary memory, but there are also disadvantages because the time span has a fixed size that depends entirely on the number K of layers (spatial buffer of length K). In the present version, we felt it would be preferable to deal with a succession of separate events that could be represented in a single layer, and with an adjustable time span: layer B is therefore now a kind of "distributed buffer." The model developed by Jordan (1986) also involved a temporary memory consisting of a single layer with recurrent connections: but this layer (and the others) is not based on a local competition principle capable of maintaining a frozen state for some time, and the processing system proposed has a set speed that cannot be regulated by an external module.

Layer G combines the current sequence history $B(t_n)$ with the constant pattern C identifying the sequence being processed, which can be written $G(t_n) = \psi[C, B(t_n)]$, where ψ denotes the output function of layer G. Here again a factor p', playing a similar role to that of p, is introduced to facilitate matters when it comes to dealing with the relative influences of layers C and B on G. As in the case of layer F, these three layers have to be large enough to be compatible with a wide range of different states.

2.2.3. *Learning Rule in Layer S.* Last, layer G is connected to layer S, through synapses of modifiable weight. Figure 3 shows one of the clusters in layer S, where the external activation applied to each unit i consists on the one hand of the output f_i from the homologous unit in the input layer F, and on the other hand, of the elementary components g_j of layer G having a positive weighting m_{ij}. The latter distributed activation is denoted $u_i = \sum_j m_{ij} g_j$. The changes in the synaptic weights m_{ij} with time are governed by the following synaptic plasticity rule (of the error correction type), or learning rule (Ans, 1990a,b):
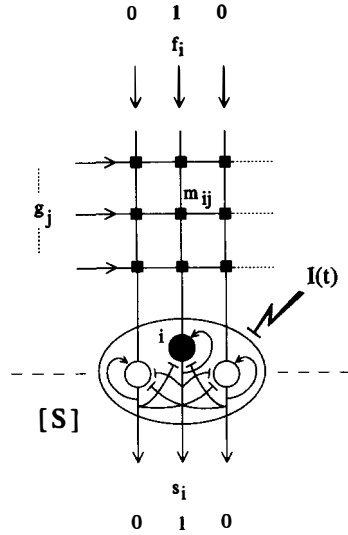
FIGURE 3. Activation of one of the clusters in layer S by a highly weighted specific input F (weighting not shown here) and by distributed associative connections with modifiable weights m_{ij}. In this case, which arises during the learning period, the outcome of the competition within a cluster depends only on the strongly weighted specific input, and therefore $S = F$.

$$\mathrm{d}m_{ij}/\mathrm{d}t = \alpha g_j (\beta + \gamma s_i - u_i) \text{ with } \alpha, \beta, \gamma \text{ Const} > 0. \quad (1)$$

In any cluster belonging to S, the value of the output s_i (0 or 1) of any unit i depends on the outcome of the competitive process. As mentioned previously, when layers F and G are activated simultaneously, the outcome of the competition depends only on the strong specific external activation exerted by F: $S(t_n) = F(t_n)$ after each updating $I(t_n)$. This means that during the learning phase of a given sequence $Z(t)$, rule (1) stores the relation between the current item in the corresponding sequence $S(t_n)$ and the pattern $G(t_n)$ accounting for its recent history $B(t_n)$ in the light of the sequence identifier C. At the retrieval phase on the contrary, when layer F is silent, the outcome of the competition inside the clusters in S depends only on the distributed activation exerted by G, which is calculated with the modified weights.

Apart from the additional parameters β and γ, expression (1) is close to the learning rule used for temporal sequence storage in Reiss and Taylor (1991), where it is shown how the rule finds (within an architectural context functionally equivalent in principle to ours) its neurobiological justification in the hippocampal structure and activity (Reiss & Taylor, 1991; Taylor & Reiss, 1992). We have introduced in rule (1) the parameters β and γ (which may be selected within a very wide range of values) in order to prevent the adjustable weights from changing their sign during the learning phase (by preventing the excitatory connections m_{ij} from becoming inhibitory), as is often the case with classical error correction rules, or generalized rules such as back propagation, where the synaptic ef-

ficacies undergo several changes of sign (which is a strong neurophysiological assumption). Even greater freedom in the choice of these parameters can be obtained by adding to the learning rule a further constraint with a null bottom threshold in regard to the adjustable weights.

2.3. Sequence Learning and Recall

2.3.1. Sequence Learning. The learning of a given sequence takes place as follows. The control module A is initially at rest, which implies that the system is in an inhibited, inactive state. When sequence $Z(t)$ occurs, accompanied by its constantly maintained identifying item ID, module A becomes active apart from an intermittent series of short breaks, the modulated frequency of which regulates the speed at which the states of the architecture are updated, particularly in regard to the fineness of the sampling in the input layer F. Once the sequence has been processed, module A enters a resting state that leads to a prolonged inhibition on all the layers, as the result of which the lag modules R become completely unloaded. The system thus reset is then ready to process (i.e., record or produce) another sequence. Each sequence has to be run several times before becoming properly memorized by the permanent memory.

2.3.2. Sequence Recall. Because input F is not activated at the sequence recall phase, it is the onset of the identity pattern ID which triggers the recall of its associated sequence at output S. Temporary memory B is initially empty, and permanent memory S reconstructs the first item of this sequence on the sole basis of the identifier. During the successive passages through the loop B, G, S (at each updating t_n), memory S progressively builds up the successive items of the recalled sequence based on the maintained identifier and the temporary memory, which itself is built up dynamically, exactly as in the learning phase. The production speed can be freely adjusted by varying the frequency of the intermittent signal emanating from the control module A. In particular, a given output item $S(t_n)$ can be sustained as long as module A maintains its activity without interruption; the normal course of events can then be resumed as required, because the history of item $S(t_n)$ is also held and preserved in temporary memory B. It is also worth mentioning that the sequences learned should theoretically all be given the same terminal symbol, which could then be used to stop the closed-loop production process. The occurrence of this particular item at the production phase would then serve to inactivate the control device A and hence to bring the activity of the whole system to a stop. This stopping device was not actually modelled here, however.

2.3.3. *Simulation.* The following simple simulation of the processing of two sequences of formal items serves to illustrate the ability of our model to deal with long repeated subsequences that may be the source of intrasequence and intersequence ambiguities. In this simulation, there is no initial recoding of the input items Z and ID: a time sequence is expressed directly in the form of a series of states $F(t_n)$, and its corresponding identifier is expressed in the form of a constant state $C(t_n) = C$. The two following sequences, identified by items $C1$ and $C2$, were processed:

$C1$: ABCDEFGHIJKLMNOP

ABCDEFGHIJKLMNOP**M**

$C2$: ABCDEFGHIJKLMNOP

RRRRRRRRRRRRRRRRRRRR**E**.

Here the letters forming sequences constitute special configurations of winning units in layer F, which were picked up at the various instants t_n at which the control module A performed the successive updatings. The same is true for $C1$ and $C2$, which are two distinct constant states of layer C. The spaces within sequences have no temporal significance but have been inserted as a means of clearly demarcating the subsequences. The letters in bold print indicate the first successor of a preceding subsequence. The first sequence $C1$ consists of two identical subsequences each consisting of 16 items, the first of which is followed by item **A** and the second by **M**. At the recall phase, the permanent memory S must be able to reconstruct a given item on the basis of this item's history, which is contained in the temporary memory B, in the light of the sequence identifier $C1$. Because this identifier is held constant during the recall of the whole sequence, it cannot have any intrasequence discriminatory power. Hence for the two items **A** and **M** to be unambiguously reconstituted on the basis of two distinct histories, the current span in the temporary memory must encompass at least the 17 previous items. If it is not the case, for example with a time window of size 16, or less, items **A** and **M** will have the same history and there will be intrasequence ambiguity. The second of the above sequences, $C2$, begins with the same subsequence as $C1$, but this first part is followed by **R** instead of **A**. Hence these two items have the same history, say $B_A = B_R = B$, within the temporary memory B, whatever the current span size. But layer G, which combines the history of an item with the sequence identifier, provides two distinct combined histories for items **A** and **R**, respectively $G_A = \psi[C1, B]$ and $G_R = \psi[C2, B]$, which will avoid an intersequence ambiguity during the recall of these two items. It should also be noted that item **R** is repeated 20 times in $C2$ before the occurrence of **E**. For **E** to be correctly recalled, it is necessary for the current memory span to encompass at least the 20 previous items.

When the above example was simulated numerically, each of the clusters constituting all the layers of the model was taken to contain five competing units, and layers F and S to contain four clusters. A state corresponding to the letters in the above two sequences is given for example by [(00001)(00010)(00001) (01000)]. Layers B, G, and C contain 15, 30, and 10 clusters, respectively. Fairly similar compositions between identity items $C1$ and $C2$ were chosen in order to demonstrate that their intersequence discriminatory power does not necessarily depend on the existence of strong differences. With the type of coding process adopted, involving 10 clusters of five units, these two identity patterns have five clusters in the same state (the same winning unit) and five in different states. In layers B and G, the fixed weights assigned to each unit are chosen at random on the basis of a uniform law in the range 0–1, and then normalized. Parameters p and p' are taken to be equal, hence $p = p' = 5$. In layer S the adjustable weights assigned to each unit are chosen initially at random on the basis of a uniform law in the range 0–0.005. In the synaptic plasticity rule (1), $\alpha = 0.03$, $\beta = 1$, and $\gamma = 0.1$.

With these parameters, sequences $C1$ and $C2$ can be learned very quickly: after being introduced only six times each, they can be *perfectly* recalled in response to their identifier. Any within, or between, sequence ambiguities can thus be resolved in the case of the present difficult example, which means that the constraints imposed on the span size of the temporary memory have been satisfied.

The above simulation was chosen because it is most representative of the model efficiency. A large number of other simulations, performed with sequences of various lengths and structures, have led to satisfactory results. This efficiency arises, in particular, from the length of the current history preserved in the temporary memory; this history can be very long hence allowing the resolution of complex ambiguity problems. Some general features of the model and some particular sequence structures have to be examined however.

The input temporal patterns to be stored, $Z(t)$, have frequently been referred to as sequences in the foregoing because, in storage modelling, these temporal signals are usually simple successions of distinct patterns of variable duration (square waves principally). Actually, our model can cope with time-varying multidimensional signals $Z(t)$, with either graded or discrete values, whose temporal form may be rather general: analog signals, continuous or not, smooth or not. This vector signal is sampled and recoded by the input layer F in the form of a sequence of patterns, the components of which are quantified signal with binary values; it is that recoded sequence $F(t)$, and not the original input $Z(t)$, that is actually stored in the model and recalled at the output $S(t)$.

In those cases where it is necessary to recall a time-

varying vector in its original form $Z(t)$, a conversion module playing the opposite role to that of layer F should be able to reconstitute the original input from the output sequence $S(t)$. This will in fact be done in Section 3 where this conversion module simply consists of an associative memory. The reconstituted signals at the output of this neuromimetic digital to analog converter will be all the more accurate because the sampling frequency, during learning, of the original time-varying vector $Z(t)$ is high. This high frequency condition is especially required when the original vector to be reconstructed has components that are smoothly changing signals; in this case the converted outputs will appear as finely quantified signals. But if we suppose that the neuron-like units composing the neural converter have time constants, then these quantified signals will consequently be smoothed by the persistent activity of the converter units. On the other hand, the converted outputs will be more or less temporally contracted or expanded in comparison with the original input $Z(t)$, and that according to the production speed of the dynamic memory controlled by module A. This temporal similarity (here used in the sense of temporally homothetical forms) will preserve, for instance, the relative durations of the patterns making up the usual sequences that are simply composed of distinct patterns of variable duration.

Another time variation type that has to be examined is that where the input time-varying vector $Z(t)$ contains gaps, that is, time intervals of variable length where all the components of the vector are null. In this case, the model will show some working difficulties unless it is supposed that an input gap is in fact represented by a non-null specific pattern $\emptyset$ in the input layer F. This is easily achieved if, during the learning of an input $Z(t)$, layer F is jointly activated by a constant activity pattern H. When an input gap occurs, only pattern H activates layer F and hence the specific state $\emptyset$ is induced. With this simple additional assumption, which was not mentioned in the foregoing for the sake of simplicity, we come back to an easily tractable situation, where gaps of different lengths within $Z(t)$ are respectively coded in layer F by subsequences of distinct sizes composed of the repeated item $\emptyset$; as previously, the original input $Z(t)$ containing gaps can be reconstituted by a converter. This additional hypothesis and the time constants of the converter units will not be taken into account in the following application.

3. APPLICATION TO PRODUCTION OF MOTOR FORMS

The general dynamic memory model was coupled with a sensorimotor system. A combined model was thus developed for learning spatial shapes, which are expressed in terms of temporal sequences of relevant exteroceptive quantities, and producing these shapes in

terms of motor activity. Hence we assumed that the format in which a given shape is memorized need not be the same as that into which it is translated in motor terms by the effector systems.

3.1. Sensorimotor System

A neuromimetic model for a sensorimotor system was developed in a previous study in the framework of research on the organization of goal-directed movements (Coiton et al., 1991; Gilhodes et al., 1991). The pointing or target tracking movements in question were either simulated numerically or performed by an artificial arm. The network drives the arm movements on the basis of proprioceptive and exteroceptive information picked up by sensory receptors. The proprioceptive information is given by sensors that code the angular values at the joints. The exteroceptive information specifies the position of any point in the arm's working space. Simulations were carried out on arms with two or three degrees of freedom moving in two- and three-dimensional space. Among the various versions of the model we have studied, one has been chosen here that is now briefly described.

A very schematic diagram of this sensorimotor module is given in Figure 4. The sensory layer K consists of cells forming a two-dimensional network. The cells are interconnected in such a way that each unit is linked via fixed excitatory connections to its nearest neighbours and via fixed inhibitory connections to the cells in the less immediate environment. In this competitive situation, a single activity focus emerges, the site of which depends on the input afferent patterns. The cells in this layer have a binary activity (the active cells in the winning focus score 1, and the others, 0). The structure of this layer, which was based on the description by Kohonen (1988), served here mainly to implement both a plurimodal signal integration and a nonlinear sensorimotor coupling. Generally speaking, it can be used as a hidden layer in the learning of highly arbitrary associations (Ans, 1989).

The sensorimotor module is organized during a learning period, during which the arm randomly explores its own working space. This exploration is induced by applying a forcing activity M on the cells constituting the motor layer by means of a random activity generator RG producing diversity. The movements performed during the learning period are not directed toward any particular goal: they are simply "blindfold" exploratory movements. It is assumed that during this phase, the exteroceptive signals will inform the network about the position of extremity D of the arm in the working space. At each new arm position, a pair of sensory signals (E, P) contributes to the changes in the weights of the input adaptive connections in layer K. Kohonen's simplified self-organization algorithm (Kohonen, 1988) was used here to compute
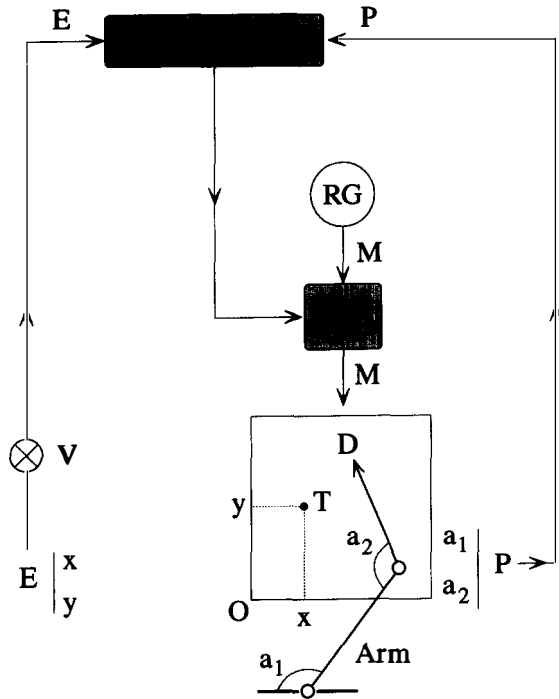
FIGURE 4. Diagram of the sensorimotor module. A simulated arm moves in a plane and its proprioceptive detectors encode its joint angles a_1 and a_2 in the form of a graded activity pattern $P = (a_1, a_2)$. The exteroceptive information is given in the form of a graded activity pattern $E = (x, y)$, the components of which are positive quantities giving the Cartesian coordinates of any point T in the arm's working space referred to the arbitrary origin O. This exteroceptive information can possibly be gated off by attentional valve V. The arm is controlled by motor layer $AM1$, consisting of two cells, each of which activates the effector responsible for moving one of the joints. The motor command M is a graded pattern of activity, the two components of which specify the pairs of joint angles (a_1, a_2) to be adopted by the arm. The proprioceptive and exteroceptive information P and E is distributed among all the cells in sensory layer K, which is itself fully connected to motor layer $AM1$. These connections are all of the adaptive type during a learning period when the arm actively explores its working space under the control of the random activity generator RG.

the activity of the units in this layer. On the other hand, sensory layer K participates in developing the adaptive links with motor layer $AM1$, which is an associative memory, and the sensorimotor coupling is thus built up. Here a simple classical error correction rule is used, where the required output is induced by the specific activation M delivered by the random generator RG. Because layer K has a localized activity, the positive sign of the synaptic weights does not change in layer $AM1$ when an error correction rule is applied. In this highly particular case, using this simple rule is not incompatible with the requirements of neurobiological realism, which is not generally true (cf. Section 2.2.3).

At some stage in the exploration of the arm's working space, the single activity focus in layer K begins to specifically encode the bimodal information consisting of the arm posture P and position E of its extremity D.

Any changes in the arm position are accompanied by changes in the site of the activity focus in the layer. This layer moreover undergoes some self-organization as the result of which it reflects the topological relations implicitly contained in the inputs it receives. At the end of the learning phase, the random generator is removed and the connections lose their previous plasticity. At the subsequent operational phase, the coordinates of any target T in the working plane can be specified on the basis of the exteroceptive information. In this case, the arm movement is such that the distal extremity D gradually approaches the target T. Because of the model's structure, movements of the arm toward the target consist of a series of successive steps. If the target begins to move, the arm then performs a tracking movement.

To make this system a little more natural, it was decided to add a switching device, simulated in the model by "valve" V, with which any irrelevant exteroceptive information can be gated off. Without this switch, the arm systematically performs the movement whenever the exteroceptive information is conveyed to any point other than extremity D of the arm. The switching device is assumed to be under the control of an attentional system that decides whether or not it is desirable to reach a given target.

3.2. Learning and Motor Production of Shapes

3.2.1. *Temporal Encoding of Spatial Shapes.* Figure 5 shows a simple example of a shape drawn on the working plane of the artificial arm. It was assumed that this shape was first studied by the exteroceptive modality, with a view to sequentially specifying the coordinates of its most noteworthy points (here, the vertices of the triangle), denoted $E1$, $E2$, and $E3$. These points correspond to position information encoded by vectors E
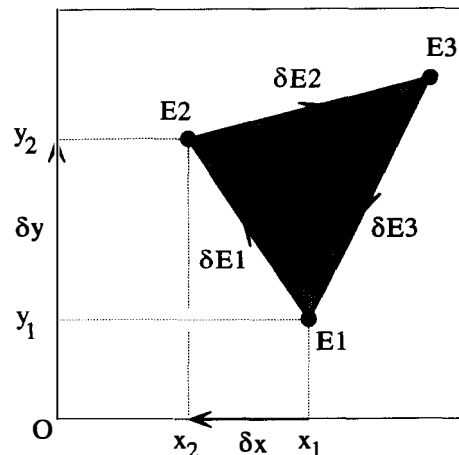

FIGURE 5. A simple example of a shape (a triangle) drawn on the working plane. The contour of the triangle is encoded in temporal terms by the exteroceptive modality E that sequentially explores the vertices $E1$, $E2$, and $E3$.

with origin 0 and (x, y) components. The shape can therefore be defined by the temporal sequence of events $E1$, $E2$, $E3$, $E1$, assuming for the sake of simplicity that the form is scanned starting at vertex $E1$, in the direction indicated by the arrows. Let us imagine that it is then proposed to store this sequence in a dynamic memory capable of delivering it as and when required via an appropriate interface to activate the sensorimotor module, which would consequently be activated by a series of virtual targets. The arm would respond by performing a tracking movement, and could thus be said to have produced the previously learned shape "from memory." Because the events constituting the sequence studied are absolute positions in a fixed Cartesian reference frame unrelated to the object, the arm should be unable to repeat the same figure consistently at various points in its working space.

What we intend to demonstrate here is precisely that one same learned shape can be reproduced as required whatever the initial arm position in space, and in a whole range of sizes. To achieve this, it is assumed that the exteroception, which up to now simply specified the absolute positions E, is now in addition capable of providing transient information δE, in terms of quantities proportional to the differences between successive positions. It is then assumed without any loss of generality that these quantities are simply equal to these differences. In the example given in Figure 5, for instance, when the exteroception successively specifies points $E1$ and $E2$, it will concomitantly also generate the transient information $\delta E1$, which can be formally expressed as the difference between vectors $E2$ and $E1$:
$$\delta E1 = E2 - E1 = (\delta x, \delta y).$$

Now the components of a vector difference can sometimes be negative, as occurs here in the case of the component $\delta x = x_2 - x_1$, and there exists no justification for assuming a priori in a neuromimetic metaphor that the sign of a signal being transmitted along a nerve fibre (which is analogous to a spike frequency having no sign) might be variable. In order to overcome this impediment to neurobiological realism, it is assumed that these elementary algebraic differences are encoded by two pairs of associated increment captors: $(\Delta x^+, \Delta x^-)$ for the x axis and $(\Delta y^+, \Delta y^-)$ for the y axis. When δx is positive, then $\Delta x^+ = \delta x$ and $\Delta x^- = 0$, and when δx is negative, then $\Delta x^+ = 0$ and $\Delta x^- = |\delta x|$, and likewise in the case of the two associated difference captors on the y axis. A shape segment is therefore defined by the quadruplet $\Delta E = (\Delta x^+, \Delta x^-, \Delta y^+, \Delta y^-)$.

Learning the contour of an object will then consist of sequentially specifying its most singular points in the exteroceptive modality E and storing in the dynamic memory the resulting temporal sequence $\Delta E(t)$. In the case of the example in Figure 5, the sequence $\delta E1$, $\delta E2$, $\delta E3$ will therefore be replaced by the corresponding sequence of patterns $\Delta E1$, $\Delta E2$, $\Delta E3$.

3.2.2. General Architecture of Combined Model.

Figure 6 gives the general architecture of the model in which the sensorimotor module is combined with the dynamic memory module DM. The two modules were linked together by means of layers $AM2$, $AM3$, and Σ, the role of which will be described in detail below. Introducing these layers had no effect on the way in which the sensorimotor module performed simple pointing and target tracking movements, because when these tasks are being carried out, unit DM is silent, that is, device A controlling it is inactive.

In the present application, the mode of activation of input layer F of the dynamic memory DM was adapted in order to simplify the processing of output sequence S by the interface module. It was seen in Section 2, in the framework of very general hypotheses, that this activation could be fully distributed and that the fixed input weights into the clusters in F could be chosen at random. Now it is assumed that each cluster in F is activated by only one of the elementary components of the pattern ΔE, which means that layer F will contain four clusters because vector ΔE has four components. The fixed input weights are now no longer random but result from previous learning, as occurred in layer K of the sensorimotor module. The outcome of this initial learning is that the units in a given cluster F respond selectively to consecutive intervals, breaking up the range of values given to the elementary graded difference signal assigned to this cluster. A cluster thus serves to classify the values of the scalar input it receives, and the larger the number of units per cluster, the finer this
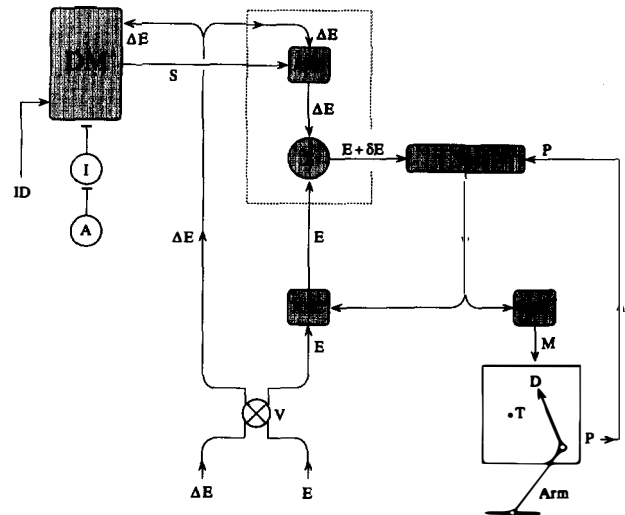

FIGURE 6. The general architecture of the model. The left-hand side mainly processes the exteroceptive difference information ΔE in the dynamic memory module DM. The right-hand side deals mainly with the exteroceptive position information E within the sensorimotor module. The interface module, which is shown here in a dotted frame, additively combines the two types of exteroceptive information, the sensory flow of which can be interrupted by attentional valve V.

classification will be. The architecture and working principles of the other *DM* layers remain unchanged.

The interface module schematized in Figure 7 contains two processing stages. The first is an associative memory *AM*3, which has previously undergone a learning period at the time when the self-organization of layer *F* of the *DM* module was being set up. At that time, layer *AM*3 learned to convert the localized format of the output *S* (which was a replica of *F*) delivered by module *DM* into a graded activity conveyed by the exteroceptive signal ΔE; here each channel is processed separately by this digital to analog converter. Taking any one channel as an example, the graded component Δx^+ imposes the output of the first cell in *AM*3, which is adaptively connected by all the units in the cluster expressing this same difference component in a localized way. Changes in the adaptive connections are governed by a classical error correction rule satisfying the constraint whereby no change in the signs of connections should occur because the associative information is localized. A similar process takes place in all the other three channels. After this learning phase, layer *AM*3 is able to reconstruct the information ΔE when it receives no specific forcing sensory input. The accuracy of this digital to analog conversion depends on the number of units in the clusters belonging to *S*.

The second stage in the interface module is an additive device denoted Σ assumed to have prewired input connections and which additively combines the two types of exteroceptive information (absolute and differential positions). The first cell Σ_x in layer Σ therefore calculates the sum $x + \Delta x^+ - \Delta x^-$, where the subtractive part is given by an inhibitory connection. Because the signals Δx^+ and Δx^- are mutually exclusive (when the one is active, the other is null), cell Σ_x actually calculates the algebraic sum $x + \delta x$; the procedure is the same with unit Σ_y. In this way layer Σ carries out the vectorial sum $E + \delta E$, and it is this sum, which
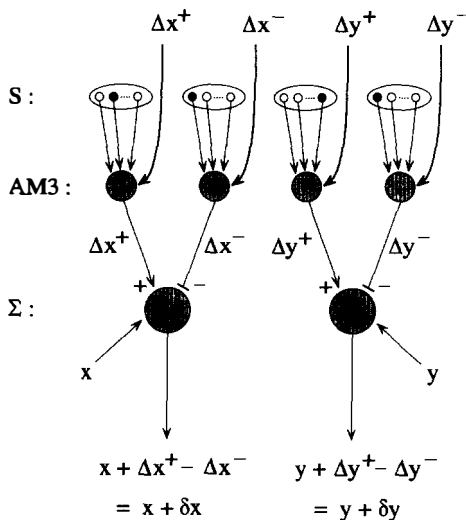

FIGURE 7. Working scheme of the interface module.

is of a purely positional nature, that will actually activate layer *K* in the sensorimotor module.

The associative memory *AM*2 implements a new relation between sensory layer *K* and the exteroceptive position information *E*. Learning in this layer is assumed to take place concomitantly with that in the whole sensorimotor module, and is set up in a very similar way to that of the symmetrical layer *AM*1 on the motor side. The specific forcing input here is the exteroceptive sensory information *E*, whereas the distributed adaptive input is a collateral of the localized activity in layer *K*. Changes in the weighting are subject to the same error correction learning rule and to the same requirement that no change in the signs of synaptic weights should occur. Introducing this layer can prove to be useful in situations where the exteroceptive sensory flow is interrupted by the attentional switch *V*: when this occurs, the layer continuously reconstructs the *virtual* exteroceptive information about the position *E* of the distal extremity *D* of the artificial arm on the basis of the currently active focus in layer *K*.

3.3. Results

3.3.1. *Shape Learning.* Once the prestructuring of the various modules has been completed, the whole system is ready to learn spatial shapes and to produce them in motor terms. Temporal sequence $\Delta E(t)$ is processed in the dynamic memory *DM* as described in Section 2, except that now we have $Z(t) = \Delta E(t)$. The learning of a given shape occurs when the temporal sequence $\Delta E(t)$ induced by exploring the contour of this shape (with attentional valve *V* open) is repetitively fed into input layer *F*. The control device *A* must be made to operate periodically at a constant frequency adapted to the fineness of the transient signal ΔE sampling. In parallel, the activity of layer *C* must be sustained by an item *ID* that encodes the identity of the shape being learned, such as triangle, square, etc.

The relation between this identity item and the sequence being learned is encoded in terms of changes in the connections in the permanent memory *S* of *DM*, which are now the *only* permanently adaptive links in the whole architecture. During the learning period, whatever arm movements are induced have no effect on the memorizing of the shape. It should be mentioned however that these arm movements generally tend to approximate the contour of the shape being learned.

3.3.2. *Motor Shape Production.* Let us take the most interesting case, where the sensory flux arising from the exteroceptive detectors is broken by the attentional switch *V*. In this case, the point in space being encoded at that moment by the exteroceptive sensors is not taken into account and has no effect on the arm posture, which is of no importance at the beginning; whereas the virtual exteroceptive information *E* reconstructed

by $AM2$ gives the current position of the arm's extremity D.

The motor production by the arm of a shape memorized in exteroceptive terms ΔE occurs as follows. The occurrence of an item ID identifying a given shape to be generated triggers the production of the corresponding temporal sequence $S(t_n)$, where t_n denotes the successive updatings of the module DM; the speed of these updatings is regulated by the control module A. The associative memory $AM3$ converts the sequence of localized activities $S(t_n)$ into a graded sequence $\Delta E(t_n)$, which is then added to the position information E in the summator Σ.

At the beginning, the sensorimotor module is in an equilibrium configuration $[P(0), E(0)]$ corresponding to any arm posture whatever. When the first element in sequence $\Delta E(t_1)$ begins to reach the summator Σ, sensory layer K receives the input activation $E(0) + \delta E(t_1)$, specifying the position of a virtual target, and an arm movement toward this target is triggered. The directional pattern $\Delta E(t_1)$, or $\delta E(t_1)$, is no longer a transient signal generated as in the learning stage because it can now be sustained for some length of time under the control of module A. As mentioned in Section 3.1, the arm movement toward a fixed target is performed step-wise, which means that several iterations of the sensorimotor loop are necessary for a target to be reached. At the first iteration τ_1 in the sensorimotor module, the distal extremity of the arm will reach position $E(\tau_1)$, midway between $E(0)$ and position $E(0) + \delta E(t_1)$ of the virtual target. The information $E(t_1)$ is reconstructed by memory $AM2$, and at the subsequent iteration τ_2, if the activity $\Delta E(t_1)$ is sustained, layer K is activated by the sum $E(\tau_1) + \delta E(t_1)$, specifying a new virtual target that has shifted in the direction $\delta E(t_1)$. The extremity of the arm continues to move in this direction and reaches position $E(\tau_2)$, midway between $E(\tau_1)$ and the new position of the virtual target $E(\tau_1) + \delta E(t_1)$, and so on. As long as the activation $\Delta E(t_1)$ is sustained, layer K is activated at the subsequent iterations τ_ν by the successive inputs $E(\tau_{\nu-1}) + \delta E(t_1)$, and the arm's extremity thus continues to track the elusive target as it continues to move in the same direction $\delta E(t_1)$. As the other directional vectors $\Delta E(t_n)$ are sequentially generated by the subsequent updatings t_n of the dynamic memory, the arm continues to track the virtual target as previously, but makes a change of direction whenever the vectors $\Delta E(t_n)$ do so.

In this way, an identified shape will be sequentially produced at the motor level in a size that depends on the frequency at which the dynamic memory is updated. This frequency must be maintained constant throughout the motor shape production period to ensure that the resulting product is not distorted. If the updating of module DM takes place at a low frequency, the final shape will tend to be large because the arm

will continue to track the moving virtual target in the same direction for long periods. Conversely, if the updating frequency is high, the final shape will tend to be small sized. Generally speaking, it is via the control exerted by module A on the dynamic memory's output speed that the size of the motor figure produced *in fine* can be adjusted as required.

The velocity of the arm movement can furthermore be controlled. First, it should be mentioned that when a given directional item $\Delta E(t_n)$ is produced by the dynamic memory, the velocity of the arm movement induced step-wise in the appropriate direction depends on the timer controlling the iterations τ_ν in the sensorimotor loop. The setting of this timer, which up to now was implicitly assumed to be constant over time, is in fact itself liable to be modulated in sensory layer K. This layer, the internal functioning of which involves a competitive process similar to that implemented in the dynamic memory clusters, is destined also to enter robust self-sustained states. Strictly speaking, it should also be updated by a system (A', I'), which is analogous to system (A, I). By introducing this further control device, it would be possible to modulate the frequency of the updating in layer K, and hence to adjust the velocity of the arm movement as required.

Several simulations have been carried out with the combined architecture. The following two very simple simulations clearly illustrate the main behavioural properties of the model. After a learning period during which a given shape was presented to the model at a single point in space and in a single-sized format, the artificial arm was able to draw this shape anywhere in its working space in whatever size was required. The first property in question, which we have called "translation invariance," is illustrated in Figure 8, and the second, which we have called "size invariance," in Figure 9.

4. CONCLUSION

The dynamic memory neuromimetic model described in Section 2 features an essential distinctive property: it is able to recall a learned sequence at a freely adaptable speed. This ability is fairly commonplace among
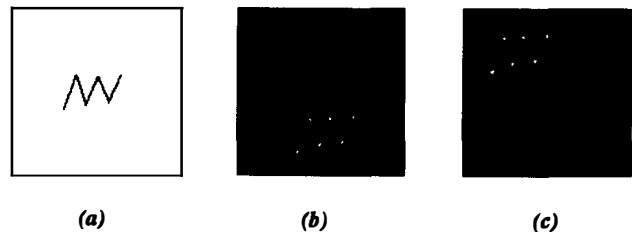

FIGURE 8. An example showing the translation invariance property. (a) The shape learned. (b) and (c) The trajectories taken by the arm's extremity while producing the learned shape starting from two different points in space.
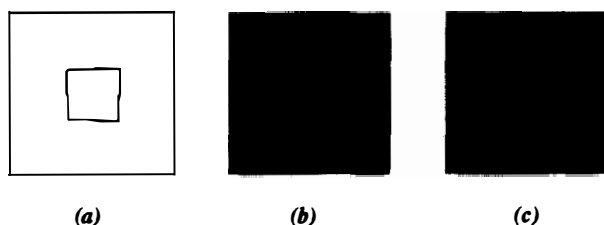
FIGURE 9. An example showing the size invariance property. (a) The shape learned. (b) The trajectory taken by the extremity of the arm while producing the learned shape. (c) The trajectory taken by the arm's extremity while producing the learned shape with a larger amplitude.

living creatures, but has rarely been dealt with in the connectionist literature. Apart from the fact that this ecological feature has to be accounted for by a general dynamic model, the flow control property can have other more far-reaching effects. We have demonstrated for example, by applying the model to the production of motor shapes, that the dynamic memory's ability to *contract* or *expand* the *time* scale when recalling a given sequence learned in exteroceptive terms resulted at the motor level in the production of forms that could be either *reduced* or *enlarged* on the *spatial* scale.

Another important property of our dynamic memory is its ability to rapidly learn sequences of temporal events. This aptitude is rare among the present day models, and yet it is essential for both living and artificial systems that constantly encounter new situations that have to be learned rapidly. Fast learning is a characteristic feature of single-layer memory models (with no hidden layer), as in Reiss and Taylor (1991) and in the present paper, but this feature is especially enhanced in our model where the memory layers are based on a competitive learning procedure. Indeed it takes much less time to approximate a set of adaptive weights resulting in only one particular unit in a cluster becoming the winner than to finely calculate, in the case of models involving no competition, the exact weights required to produce accurate, possibly graded outputs.

Generally speaking, the larger the layers constituting the model, the more efficiently they perform: the size of layer G is a particularly decisive factor in this respect. The relations stored in permanent memory S between a current item and its history, which is represented in layer G, can be highly arbitrary; and the synaptic plasticity rule (1), which was chosen mainly because of neurobiological plausibility, is unable to store in the distributed form a large number of sequences containing many items unless the dimension of the patterns G is sufficiently large. This constraint is not necessarily too severe to be compatible with the neuronal resources of a real-life brain.

Our model involved a minimal approach based on a set of synaptic weights most of which were chosen at random and fixed, and on a synaptic plasticity rule that is fairly simple and yet satisfies the most elementary natural constraints. To achieve even greater biological realism, each of these basic units with a strong autoexcitatory connection, so designed in order to simplify the formalism, could in fact be easily replaced by a micropopulation of neurons densely interconnected by excitatory synapses. Micropopulations of this kind (with two persistent states of activity) and the clusters within which they are engaged in competitive relationship (neural assemblies) have been roughly assimilated to the micro and macrocortical columns, respectively (Dehaene et al., 1987; Dehaene & Changeux, 1989, 1991; Strong & Whitehead, 1989).

The combined model described in Section 3, in which the dynamic memory module was linked up with the sensorimotor module, operates on the information of the proprioceptive and exteroceptive types. The proprioceptive sensory detectors were assumed to pick up purely postural information, although the messages conveyed by human muscle spindles are known to relate both to posture and to the velocity of the joints involved in the movement (Matthews, 1972). This additional velocity information, which is essentially transient, can in fact also be directly taken into account by the model. Another way of dealing with this problem is to assume that the purely postural information that activates sensory layer K of the model is conveyed by an intermediate structure responsible for preprocessing the composite message originating from the sensory detectors. It is worth mentioning here that we have previously described models, some with neuromimetic features (Ans & Gilhodes, 1983; Ans, Gilhodes, & Hérault, 1983; Hérault & Ans, 1984; Ans, Hérault, & Jutten, 1985), and others with a differential formalism (Gilhodes, Coiton, Roll, & Ans, 1993), including intermediate structures of this kind where composite afferent messages are decomposed into their position and velocity components.

We have used the term exteroceptive when referring to the sensory information of the second type dealt with in the model. In humans, exteroceptive information is mainly of the visual type. It is usually by means of vision that a target to be reached in extra-personal space is detected and located. In this case, the absolute position E might result from the combination of two signals: the target retinal coordinates and the extra-retinal signal, from either central or peripheral origin, informing on the eye-in-orbit position. The relative position ΔE might be obtained in two different ways: it might either result from the central processing of the retinal error (the distance and the direction of a target image relative to the fovea), or it might be processed from the extra-retinal signal changes occurring during the movement the eye performs to go from one point to another. The precise processings involved in generating the two kinds of exteroceptive information were not

modelled, this being beyond the topic of the present paper.

The basic principle adopted here, according to which the format in which a shape is memorized is not necessarily the same as that into which it is translated and produced in motor terms, gave rise to some important behavioural properties emerging from the model. The first two properties, which were mentioned above, have been called "translation invariance" and "size invariance": a shape which has been learned at a single point in space and in a given size can be drawn by the arm anywhere in its working space and in a whole range of sizes.

Another consequence of this basic principle is that the memorized shape can be repeated by other effector systems that have never participated in the shape learning because the exteroceptive temporal information produced on the output of the dynamic memory is, precisely, not related to the particular sensorimotor organisation of the simulated limb performing movements: the classic example here is the ability to write one's signature with the tip of one's foot (Bernstein, 1967). To better understand this commonplace behaviour, suppose that distinct sensorimotor modules, each composed of a set of layers $(K, AM1, AM2, \Sigma)$ similar to that depicted in Figure 6, have first separately acquired, on the basis of the common exteroceptive information giving absolute position E, the basic ability to drive the pointing movements of their corresponding specific limbs. Consider now that the output ΔE of a unique and common dynamic system, which is composed of the dynamic memory DM and its associated converter layer $AM3$, connects all the layers Σ belonging respectively to the different sensorimotor modules (as defined above). The output of $AM3$, which conveys relative position information $\Delta E(t)$, temporally defining a given shape in a common exteroceptive format, will be added in the different layers Σ together with the current absolute positions E of the respective distal extremities D of the corresponding limbs. Hence these distinct effector systems will be able to perform similar motor figures in response to the same exteroceptive shape stored in a unique dynamic memory.

The above behavioural properties, which originate from a realistic neural network model in which they are not implemented a priori, reinforce the idea that the structures responsible for planning a movement in the central nervous system might be largely independent of the motor systems performing this movement. Our model proposes that these structures (here only the dynamic memory) would store generic programs, dynamically defining "shape template," which would then be translated in various ways in the same or in several motor systems, even in the effector systems that have never been recruited during shape learning. This single entity giving rise to diversity is related to Schmidt's idea (1988) of a "generalized motor pro-gram" which would contain only one unique representation of the essential invariant features for a particular type of action and whose expression could be varied depending on the choice of certain parameters. And our contribution here was to propose *how* these abstract programs could actually be implemented at a quasi-neural level.

REFERENCES

Ans, B. (1989). Learning arbitrary associations: A neuromimetic model. In G. Tiberghien (Ed.), *Advances in cognitive science* 2 (pp. 62–85). Chichester: Ellis Horwood.

Ans, B. (1990a). Associative learning in a neuromimetic network with local competitions. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **310,** 127–132.

Ans, B. (1990b). Neuromimetic model for storage and recall of temporal sequences. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **311,** 7–12.

Ans, B., & Gilhodes, J. C. (1983). Simulation de réseaux neuronaux (SIRENE). I: Le Modèle. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **297,** 361–364.

Ans, B., Gilhodes, J. C., & Hérault, J. (1983). Simulation de réseaux neuronaux (SIRENE). II: Hypothèse de décodage du message de mouvement porté par les afférences fusoriales IA et II par un mécanisme de plasticité synaptique. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **297,** 419–422.

Ans, B., Hérault, J., & Jutten, C. (1985). Architectures neuromimétiques adaptatives: Détection de primitives. *Proceedings of "Cognitiva 85"* (pp. 593–597). Paris.

Bernstein, N. A. (1967). *The coordination and regulation of movement.* Oxford: Pergamon Press.

Coiton, Y., Gilhodes, J. C., Velay, J. L., & Roll, J. P. (1991). A neural network model for the intersensory coordination involved in directed movements. *Biological Cybernetics,* **66,** 67–176.

Crick, F. (1989). The recent excitement about neural networks. *Nature,* **337,** 129–132.

Dehaene, S., & Changeux, J. P. (1989). A simple model of prefontal cortex function in delayed-response tasks. *Journal of Cognitive Neuroscience,* **1,** 244–261.

Dehaene, S., & Changeux, J. P. (1991). The Wisconsin card sorting test: Theoretical analysis and modeling in a neuronal network. *Cerebral Cortex,* **1,** 62–79.

Dehaene, S., Changeux, J. P., & Nadal, J. P. (1987). Neural networks that learn temporal sequences by selection. *Proceedings of the National Academy of Sciences USA,* **84,** 2727–2731.

Elman, J. L. (1988). Finding structure in time. *Cognitive Science,* **14,** 179–211.

Gilhodes, J. C., Coiton, Y., Roll, J. P., & Ans, B. (1993). Propriomuscular coding sensation: Experimental approach and mathematical modelling. *Biological Cybernetics,* **68,** 509–517.

Gilhodes, J. C., Coiton, Y., & Velay, J. L. (1991). Sensorimotor space representation: A neuro-mimetic model. In J. Paillard (Ed.), *Brain and space* (pp. 433–445). Oxford: University Press.

Grossberg, G. (1973). Contour Enhancement, short-term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics,* **52,** 213–257.

Grossberg, G. (1987a). A massively parallel architecture for self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing,* **37,** 54–115.

Grossberg, G. (1987b). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science,* **11,** 23–63.

Grossberg, G. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks,* **1,** 17–61.

Hebb, D. O. (1949). *The organization of behavior.* New York: Wiley.

Heidmann, T., & Changeux, J. P. (1982). Un modèle moléculaire de

régulation d'efficacité au niveau postsynaptique d'une synapse chimique. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **295,** 665–670.

Hérault, J., & Ans, B. (1984). Réseau de neurones à synapses modifiables: Décodage de messages sensoriels composites par apprentissage non supervisé et permanent. *Comptes Rendus de l'Académie des Sciences, Paris, Série III,* **299,** 525–528.

Hinton, G. E., & Anderson, J. A. (Eds.) (1981). *Parallel models of associative memory.* Hillsdale, NJ: Lawrence Erlbaum.

Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review,* **98,** 74–95.

Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach* (Tech. Rep. ICS-8604). La Jolla: University of California at San Diego, CA.

Keele, S. W., Cohen, A., & Ivry, R. (1990). Motor programs: Concepts and issues. In M. Jeannerod (Ed.), *Attention and performance XIII: Motor representation and control.* Hillsdale, NJ: Lawrence Erlbaum.

Kohonen, T. (1977). *Associative memory: A system theoretical approach.* Berlin–Heidelberg: Springer–Verlag.

Kohonen, T. (1988). *Self-organization and associative memory.* Berlin–Heidelberg: Springer–Verlag.

Konishi, M. (1985). Birdsong: From behavior to neuron. *Annual Review of Neurosciences,* **8,** 125–170.

Le Cun, Y. (1986). Learning process in an asymetric threshold network. In E. Bienenstock, F. Fogelman-Soulié, & G. Weisbuch (Eds.), *Disorder systems and biological organization* (pp. 233–240). Berlin–Heidelberg: Springer–Verlag.

Matthews, P. B. C. (1972). *Mammalian muscle receptors and their central actions.* London: Arnold Pub.

Reiss, M., & Taylor, J. G. (1991). Storing temporal sequences. *Neural Networks,* **4,** 773–787.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 318–362). Cambridge, MA: MIT Press/Bradford Books.

Rumelhart, D. E., & McClelland, J. L. (Eds.) (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vols. 1 and 2.* Cambridge, MA: MIT Press/Bradford Books.

Schmidt, R. A. (1988). *Motor control and learning.* Champaign, IL: Human Kinetics Publishers Inc.

Shik, M. L., Severin, F. V., & Orlovskii, G. N. (1966). Control of walking and running by means of electrical stimulation of the mid-brain. *Biofizyka,* **11,** 659–666.

Strong, G. W., & Whitehead, B. A. (1989). A solution to the tag-assignement problem for neural networks. *Behavioral and Brain Sciences,* **12,** 381–433.

Taylor, J. G., & Reiss, M. (1992). Does the hippocampus store temporal patterns? *Neural Network World,* **3–4,** 365–384.

Viviani, P., & Terzuolo, C. (1980). Space-time invariance in learned motor skills. In G. E. Stelmach & J. Requin (Eds.), *Tutorials in motor behavior* (pp. 525–533). Amsterdam: North–Holland.

Willwacher, G. (1982). Storage of a temporal sequence in a network. *Biological Cybernetics,* **43,** 115–126.