

1994-06-28

Neural Models of Temporally Organized Behaviors: Handwriting Production and Working Memory

<https://hdl.handle.net/2144/2034>

"Downloaded from OpenBU. Boston University's institutional repository."

**NEURAL MODELS OF TEMPORALLY ORGANIZED BEHAVIORS:
HANDWRITING PRODUCTION AND WORKING MEMORY**

Stephen Grossberg

November 1993

Revised: June 1994

Technical Report CAS/CNS-93-057

Permission to copy without fee all or part of this material is granted provided that: 1. the copies are not made or distributed for direct commercial advantage, 2. the report title, author, document number, and release date appear, and notice is given that copying is by permission of the BOSTON UNIVERSITY CENTER FOR ADAPTIVE SYSTEMS AND DEPARTMENT OF COGNITIVE AND NEURAL SYSTEMS. To copy otherwise, or to republish, requires a fee and/or special permission.

Copyright © 1993

Boston University Center for Adaptive Systems and
Department of Cognitive and Neural Systems
111 Cummington Street
Boston, MA 02215

**NEURAL MODELS OF TEMPORALLY ORGANIZED BEHAVIORS:
HANDWRITING PRODUCTION AND WORKING MEMORY**

Stephen Grossberg§
Center for Adaptive Systems
and
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, MA 02215

Technical Report CAS/CNS-TR-93-057

November, 1993

Revised: June, 1994

To appear in: E. Covey, Ed.
Neural Representation of Temporal Patterns,
New York, NY: Plenum Press, 1994.

§ Supported in part by ARPA (ONR N00014-92-J-4015) and the Office of Naval Research (ONR N00014-91-J-4100 and ONR N00014-92-J-1309).

Acknowledgements: The authors wish to thank Diana Meyers and Robin Locke for their valuable assistance in the preparation of the manuscript.

1. Introduction: Neural Control of Movement and Memory

The brain uses several different types of mechanisms to control the temporal organization of behavior. This chapter summarizes biological neural networks which model two types of temporal control. The first model is the VITEWRITE model of handwriting production (Bullock, Grossberg, and Mannes, 1993). The second model is the STORE model for encoding sequences of events in working memory (Bradski, Carpenter, and Grossberg, 1992). Both models have arisen from a computational analysis of relevant behavioral and neural data bases. In both models, the temporal properties of the behavior are not explicitly represented in the network, but instead are emergent properties of multicellular interactions. This fact raises the issue of what organizational principles enable the networks to generate goal-oriented temporal relationships despite the fact that these relationships are not explicitly represented in model mechanisms.

2. The VITEWRITE Model of Handwriting Production

The VITEWRITE model addresses a number of key issues concerning the skilled performance of sequential actions: What is a motor program? How can a complex movement be flexibly performed at will with variable speed, size, and style without requiring new learning? How does the brain control a redundant manipulator that possesses more degrees of freedom than the space in which it moves? How can smooth curvilinear movements be organized by such a redundant manipulator? In particular, how is the timed launching of different groups, or synergies, of muscles achieved so that the desired directions, distances, and curvatures of movement are achieved? How, moreover, can “acts of will” that vary the speed and size of movements achieve their goal, thereby changing distances and curvatures of movement, without disrupting the correct directions of movement that preserve its overall form through time? The VITEWRITE model, summarized in Figure 1, introduces a new concept of how a “motor program” can control skilled sequential movements. This motor program is not explicitly represented in the model. Rather, it is an emergent property of feedback interactions between a working memory representation of desired movement directions (called a Vector Plan), and a trajectory generator for moving the limb (called a VITE circuit). The VITEWRITE model also provides a new analysis of how the use of a redundant manipulator can simplify the problem of motor planning.

The VITEWRITE model demonstrates how a working memory can control writing movements that exhibit many properties of human handwriting when it interacts reciprocally with a suitably defined trajectory generator coupled to a model hand with redundant degrees of freedom. These results extend the applicability of the VITE model from the control of reaching behaviors (Bullock and Grossberg, 1988, 1991) to the control of complex curvilinear trajectories.

Figure 1

Using a hand with redundant degrees of freedom, here taken to be three (Figure 2), simplifies the motor program, or plan, in at least three ways, that will be explained in subsequent sections. First, each of the three motor synergies, or coordinated muscle groups, of such a hand can be controlled with unimodal velocity profiles. Second, the Vector Plan working memory consists of a discrete set of difference vectors that are read into a VITE circuit at prescribed times. These difference vectors, called *planning vectors*, represent the

direction and desired amount of contraction of a motor synergy. They are denoted by DV_p below. Third, the motor program automatically launches transient directional commands to the hand synergies at only two phases in a movement—when the hand begins to move, or when a peak velocity in one of the synergies is achieved.

Figure 2

Such a motor program can be utilized with a VITE model because the VITE model contains a processing stage at which an outflow representation of intended movement velocity is represented. This stage computes the product $DV_m \cdot GO$ of difference vectors DV_m multiplied by a GO signal. The continuously changing DV_m vectors are called *movement vectors*. They are not the discrete planning vectors DV_p . The GO signals that multiply the movement vectors are “will to act”, or analog speed, signals that cause movement of a motor synergy if its DV_m is not equal to zero. The $DV_m \cdot GO$ outflow commands continuously move the synergy towards a desired target configuration until its DV_m equals zero. The maxima in time of these $DV_m \cdot GO$ outflow commands in the VITE trajectory generator are used as feedback control signals to read-out the next planning vector DV_p . Using this type of internal feedback loop, an increase in the GO signal can speed up a handwriting movement without changing its form. In a similar way, a GRO signal (defined below) can multiply the planning vectors DV_p before the net signals $DV_p \cdot GRO$ arrive at the VITE model, resulting in a handwritten movement of different size but the same form.

In summary, the VITEWRITE model converts the Vector Plan’s temporally discrete and disjoint planning vectors $DV_p \cdot GRO$ into smooth curvilinear trajectories among temporally overlapping synergetic movements. The unimodal temporal shapes of the $DV_m \cdot GO$ outflow velocity commands to the motor synergies are an emergent property of the entire VITEWRITE circuit. When a peak in one synergy’s $DV_m \cdot GO$ function is attained, it can activate read-out of a planning vector from the motor program to the VITE circuitry that controls other synergies. These properties enable the VITEWRITE model to avoid explicit storage of within-stroke time lags, to use few memory resources to store the planning vectors, to employ activity-based $DV_m \cdot GO$ decisions to automatically read-out the planning vectors, and to thereby achieve speed and size rescaling in response to scalar GRO (size) and GO (speed) acts-of-will, while effortlessly concatenating letter shapes into words.

The VITEWRITE model builds upon desirable properties of the VITE model that have been described in previous studies of VITE-controlled reaching. Indeed, a role for the VITE model in handwriting control was noticed soon after its introduction in Bullock and Grossberg (1988), since the VITE circuit generates emergent properties that mimic key properties of handwriting data. These include the isochrony principle, namely the tendency for strokes of different size to be completed with approximately equal duration; (Schomaker, Thomassen, and Teulings, 1989; Viviani and Terzuolo, 1983); skewed velocity profiles (Wann, Nimmo-Smith, and Wing, 1988), typically with faster rise and slower fall in velocity; the synthesis of continuous complex movements from unit segments (Soechting and Terzuolo, 1987); and the tendency of maximal curvatures of a trajectory to occur at locations of minimum velocity (Abend, Bizzi, and Morasso, 1982; Fetters and Todd, 1987; Viviani and Terzuolo, 1980).

The three main components of the VITEWRITE model are: (1) a geometrical model of the hand, (2) a VITE neural trajectory generator, and (3) a Vector Plan working memory. By combining these elements, precise extrinsic control of onset and offset timing is unnecessary.

Instead, the times at which subsequent movement commands are read-out from the working memory are automatically determined by events in the trajectory generator, which, in turn, are sensitive to previous working memory and volitional signals.

3. Geometry of the Hand

The number of motor segments used in handwriting is large, involving every joint from the shoulder to the fingers. The analysis here is restricted to the hand only, which still has a total of seven degrees of freedom (DOF) from the wrist to the fingertip. Most of these hand joints operate in concert during handwriting to control three main sets of motor synergists, or muscle groups. Accordingly, the hand model in Bullock, Grossberg, and Mannes (1993) has three DOFs: vertical wrist rotation (supination/pronation, called X) finger extension/retraction (called Y), and horizontal wrist rotation (called R), as in Figure 2.

The extra, third degree of freedom, R , can be used to reduce the complexity of both the motor program and the neural trajectory generator. As an example, consider the simple stroke depicted in Figure 3. In Cartesian space, this stroke can be generated by a mix of unimodal and bimodal velocity profiles with unequal component movement durations, as shown in Figure 3a. By adding a third DOF, which, at least in this example, acts in much the same way as the horizontal component, the same stroke can now be generated using only unimodal, bell-shaped velocity profiles with equal durations. Because of this simplification, there is a unique maximum outflow velocity during each synergetic movement that can be used to trigger read-out of the next working memory command. In this way, a redundant degree of freedom can be used to reduce the complexity of both trajectory generation and motor planning.

Figure 3

A further simplification is made by considering the relative amplitudes of synergetic movement that are characteristic of skilled handwriting. Both the effects of finger extension and vertical wrist rotation in handwriting are small in relation to the total range (cf. Lacquaniti *et al.*, 1987), and the radius of horizontal wrist rotation is rather large in relation to finger extension and vertical wrist rotation. The trajectories of each of these components are thus good approximations to straight lines. Therefore, we further simplify the geometrical hand model by modelling both X (vertical wrist rotation) and Y (finger extension) as an orthogonal system of spatially straight lines. However, since these axes of movement are mounted on the hand (and not fixed with respect to the drawing surface), this coordinate system can be rotated by horizontal wrist motion.

Under these assumptions, if the wrist is located at spatial location $(0,0)$, then the pen tip, or end effector location (E_x, E_y) can be found by

$$E_x = (l + y) \sin(r) + x \cos(r) \quad (1)$$

$$E_y = (l + y) \cos(r) - x \sin(r), \quad (2)$$

where x and y denote the X and Y excursions, respectively, and r stands for the horizontal angle of the hand with respect to the arm. The length of the hand from the wrist to the knuckles, denoted as l , is large relative to the X, Y and R excursions.

4. Synchronous Trajectory Formation by Vector Integration to Endpoint

The Vector Integration To Endpoint, or VITE, model of Bullock and Grossberg (1988, 1991) is a neural model of a trajectory generator whose outflow commands control multi-joint motor trajectories. The model shows how a group of muscles may be dynamically bound into a motor synergy, and once bound, how the synergy can perform *synchronous* movements at variable *speeds*. We therefore often call synergy, synchrony, and speed the “3 S’s” of trajectory formation. The VITE model outputs are the inputs to a second neural model called the FLETE model for Factorization of Length and TEension. The FLETE model suggests how outflow commands from a VITE circuit may be transformed into positionally accurate movements of an arm that is subjected to variable external forces and stiffness levels (Bullock and Grossberg, 1991; Bullock, Contreras-Vidal, and Grossberg, 1992). In other words, the VITE model forms part of the Platonic trajectory planning apparatus of a larger movement control system, whereas the FLETE model controls Newtonian force and motor plant related factors to ensure that the arm closely tracks VITE outflow movement commands. To accomplish this, the FLETE model compares VITE outflow velocity $DV \cdot GO$ and positional PPV signals with dynamic and static inflow signals from the muscles themselves to trigger either reactive responses to positional errors or adaptively timed gain changes that serve to predictively preempt errors before they can occur on future movement trials. In the original references, the VITE model is interpreted in terms of neural data about brain regions such as parietal cortex, motor cortex, and basal ganglia. The FLETE model is interpreted in terms of neural data about the spinal cord and cerebellum. These spinocerebellar interactions will not be further discussed herein.

Figure 4

The VITE circuit consists of four neural processing stages that are depicted in Figure 4. The first stage computes a Target Position Vector (TPV) that encodes desired limb positions. As in the VITEWRITE model (Figure 1), these target locations are derived from signals coded in terms of muscle lengths from higher processing stages. The Present Position Vector (PPV) stage integrates its inputs over time to generate outflow movement signals to spinal neuron pools, which in turn act on muscles capable of moving the arm. The Difference Vector (DV) stage continuously computes the difference between PPV and TPV using excitatory outflow signals from the TPV and inhibitory corollary discharge, or efference copy, signals from the PPV. This DV is denoted by DV_m in Figure 1.

Outflow from the DV to PPV is multiplied, or gated, by a nonspecific GO signal. Before any movement begins, a desired position command may be loaded into the TPV and relayed to the DV. This operation is called motor priming (Georgopoulos *et al.*, 1984). Until the GO signal grows positive, however, no change in PPV can occur. Once the GO signal becomes positive, the PPV can start integrating its input signals at the rate $GO \cdot DV$ (see Figure 4). This multiplicative interaction maintains the direction coded by DV while using the GO signal to modulate the speed of movement in this direction. The size of the GO signal is assumed to grow monotonically once a movement is initiated. Since the PPV integrates $DV \cdot GO$, the rate of change of the outflow PPV signal, namely $\frac{d}{dt}PPV$, tracks $DV \cdot GO$. Thus $DV \cdot GO$ provides an internal measure of the commanded movement velocity $\frac{d}{dt}PPV$. The DV is driven to zero by inhibitory feedback from PPV to DV as the PPV approaches the TPV. The system thus equilibrates when the PPV equals the TPV. If a single GO signal multiplies all outflow commands from the DV equally, all components of a given motor synergy tend to complete their movement synchronously, regardless of GO signal magnitude or component

movement amplitude (Bullock and Grossberg, 1988).

5. Coordination of Multiple Motor Synergies with Asynchronous Onsets and Offsets

The production of curved trajectories during handwriting requires, however, that distinct movement components have distinct but overlapping velocity profiles. These phase lags suggest that the several hand synergies (finger extension, horizontal wrist rotation, and vertical wrist rotation) cannot be grouped into one TPV with a single GO signal, since the VITE circuit would work towards making all component movements terminate at the same time. Instead, each of the three synergies of the hand model is controlled by its own VITE circuits, with separately initiated GO signals. These GO signals are reset before the onset of a new movement by each synergy. The assumption of multiple GO signal channels is consistent with data on the proposed anatomical site of GO signal generation, namely the basal ganglia (Bullock and Grossberg, 1991, Horak and Anderson, 1984a, 1984b). Recent reports indicate that pathways through the basal ganglia maintain somatotopy, or motor-channel specificity (Parent, 1990), and work summarized by Golani (1992) implicates the basal ganglia in gating the degrees of freedom that are incorporated into different movements.

6. Model Equations

The equations that govern the dynamics of the multi-channel VITE circuit are as follows. The TPV is denoted by $T = (T_1, T_2, \dots, T_n)$, the PPV by $P = (P_1, P_2, \dots, P_n)$, the movement vector DV_m by $V = (V_1, V_2, \dots, V_n)$, the planning vector DV_p by $D = (D_1, D_2, \dots, D_n)$, the GRO signal by $S = (S_1, S_2, \dots, S_n)$, and the GO signal by $G = (G_1, G_2, \dots, G_n)$, where index i denotes the i th motor synergy.

Target Position Vector

$$T_i(t_{i,j+1}) = T_i(t_{ij}) + S_i D_i(t_{ij}). \quad (3)$$

The TPV receives planning vector inputs $D_i(t_{ij})$ from higher processing stages. These inputs embody directional commands whose size, scaled by S_i , determines the distance travelled by each synergy. At launch times $t_{ij}, j = 1, \dots, n$, the j th planning vector $(D_i(t_{ij}), D_2(t_{2j}), \dots, D_n(t_{nj}))$, scaled by the GRO signals (S_1, S_2, \dots, S_n) , is added to the TPV, as in (3).

Difference Vector

$$\frac{d}{dt} V_i = \alpha(-V_i + T_i - P_i). \quad (4)$$

By equation (4), the movement vector V tracks the difference $T - P$ at rate α . Equation (4) simplifies the original VITE equations (Bullock and Grossberg, 1988), which used an opponent push-pull mechanism to avoid negative values for V_i (see Figure 4). Here, agonist and antagonist activations are lumped into one variable by allowing negative values.

GO Signal

$$G_i(t) = G_0(t - t_{ij})^n \quad t_{ij} \leq t < t_{ij}^*, \quad j = 1, \dots, n, \quad (5)$$

where G_0 is a constant and t_{ij} is the j th time at which synergy i is launched. The GO signal grows monotonically until time t_{ij}^* , when it is reset to zero. This stereotyped and

repetitive GO signal rule is capable of generating arbitrary cursive script letters shapes. In all simulations, $n = 1.4$, which produces nearly symmetrical bell-shaped velocity profiles. Equation (5) for the growth of the GO signal is one of many that could be used, and is chosen wholly for convenience. Bullock and Grossberg (1988) showed that many psychophysical properties of arm movements could be fit by a wide variety of GO signal shapes. In particular, they showed that a physically plausible GO signal could be generated by two or more neurons activated in series by a step function input. In the VITE model, using a cascade to generate a GO signal is one of two determinants of the velocity profile – the DV being the other. In the Plamondon (1989) handwriting model, a much longer cascade is used to generate the entire velocity profile.

Present Position Vector

$$\frac{d}{dt}P_i = V_i G_i \quad (6)$$

The PPV integrates its input signals at the rate $V_i G_i$. Since V_i tracks $T_i - P_i$ and P_i increases if $T_i > P_i$ or decreases if $T_i < P_i$, the process continues until all $P_i = T_i, i = 1, 2, \dots, n$.

7. Feedback Control of Sequential Movement Commands

To produce the smooth, curved trajectories of script, synergy DV_p directions and GO signal onsets need to be appropriately timed. The onset timing for the next stroke in a motor program could be determined in two ways. In one way, the time of launching the next stroke is a parameter of the motor program (cf. Schomaker *et al.*, 1989). In the second way, some event in the controller itself, or even downstream from the controller, triggers execution of the next stroke. The first possibility faces the difficulty that the motor program may not be able to compensate for changes in stroke size and speed of execution. For example, the shape of a trajectory could be very different at different execution speeds. unless the timing of successive onsets could automatically compensate for such motor variability.

If triggering a successive stroke is contingent on a characteristic event in the velocity vector of the movement, this problem is avoided, since onset lags then shift automatically with speed of execution. An outflow representation of each synergy's velocity is encoded in the VITE model by the model neurons whose activities represent the $DV_m \cdot GO$ processing stage (see Section 4). Such an outflow representation avoids the instability problems that could otherwise occur if delayed inflow signals from the muscles themselves were used. Simulation studies by Bullock, Grossberg, and Mannes (1993) have shown that two events are suitable to launch a stroke. These are: (1) times when there is a match between TPV and PPV, and consequently all velocities are close to zero. This event is called a *postural launch*; (2) times at the peak of one or more velocity traces. This event is called a *dynamic launch*. In a dynamic launch, a peak in one of the velocity profiles initiates movement of new synergies by triggering read-in of new targets and reset of their respective GO signals. The new targets may be zero for some or all components. In the postural launch, a point of zero velocity can also trigger a new movement. Thus the launch times t_{ij} in (5) and Figure 5 occur either when a synergy i is at rest or when the outflow speed command $DV \cdot GO$ of another synergy reaches a maximal size. Reset occurs at times t_{ij}^* when the PPV of the synergy equals its TPV. This control scheme is robust with respect to changes in command timing. Perturbing onset timing results in rounder shapes if a dynamic launch occurs before the peak of another velocity profile, and more angular shapes if the launch occurs after the peak.

If a new target is launched only when one of these two types of events occurs, then the phase relations between any two component velocity traces are limited to either 0 or 90 degrees. Each peak and zero in the outflow velocity trace $DV_m \cdot GO$ can activate read-out of the next planning vector DV_p from the working memory, as in Figure 5. Such a DV_p reads a new directional movement command into the TPV of the VITE circuit. Each DV_p also activates the GO signal of the corresponding synergy. These TPV commands point in the independent X, Y, and R directions. Their amplitudes equal the maximal excursion of the letter in that direction. The order, timing, and size of these synergy commands determine the curvature of the movement. All the stored commands in the vector plan that characterizes a letter in this scheme are generated at discrete times in independent directions. The VITE model automatically converts these temporally discrete commands into continuously curved trajectories of appropriate shape. Such a controller affords a huge compression of the premotor and motor commands needed to generate cursive script. Some key properties of movement generated by the VITEWRITE model are summarized below.

Figure 5

8. Simulations of Cursive Script: Flexible Control of Size, Speed, and Style

An example of a script letter “b” is shown in Figure 5. The motor program—that is the sequence of directional targets for the controller—is summarized in Table 1. Each row in Table 1 corresponds to a stroke segment shown in the small panels on the lower right side of Figure 5.

Table 1

To start with, an X motion to the right is launched (stroke segment 1 in Figure 5 and half cycle 1 in Table 1). At the time when X reaches maximum velocity, a Y motion upwards is launched (stroke 2). At the peak of this Y motion, a small X motion to the left is launched (stroke 3), and so forth. The letter “b” is a relatively simple example because the trajectory of this letter is a variation of a circle, but with different amplitudes for X and Y in every stroke. The similarity to a circular trajectory can also easily be seen by the up-down alternation of the velocity profiles. Bullock, Grossberg, and Mannes (1993) discuss how use of a consistent stylistic strategy for each letter enables letters to be effortlessly connected into word shapes, an example of which is depicted in Figure 6.

Figure 6

Some aspects of the kinematics of handwriting trajectories are invariant with respect to variations in starting point, slant, and size (Viviani and Terzuolo, 1980; Morasso, 1981). These invariances are also exhibited by the model. Figure 7 displays variations of a “b” letter trajectory achieved by rescaling the volitional GRO command. This “b” shape is created by a different combination of synergetic commands than in Figure 5, thereby illustrating the flexibility that can be achieved using a redundant manipulator. In each panel of Figure 7, all the components D_i of TVP_i are multiplied by the same GRO scalar S , but S varies across the panels. This variable GRO command modifies the size of the letters produced, but leaves the trajectory shape invariant. The simplified geometrical model defined in equations (1) and (2) produces perfect shape invariance under size scaling. If a more elaborate geometrical model of the hand is used, extreme finger angles at the border of the workspace produce

distortions; see Bullock, Grossberg, and Mannes (1993). In addition, changes in writing style can be achieved by multiplying each component D_i of TPV_i by a different scalar S_i .

Shape invariance under speed rescaling is demonstrated in Figure 8, which shows the same letter performed at a given speed and at double that speed. This is achieved by rescaling the GO signal via parameter G_0 in equation (5). This simulation assumes that new synergies are instantaneously launched at the velocity maxima of other synergies. If a small but finite reaction time is needed to launch, invariance would not be substantially influenced until speeds were attained at which the duration of each synergy was not much greater than the reaction time. Then the smooth curvature of the letter shape would begin to deteriorate, leading to straighter trajectories followed by more sudden changes of curvature.

Figure 7

The ease with which size and speed invariance are demonstrated in the VITEWRITE model derives from the model's use of DV's to control updating of the TPV in equation (3) and updating of the PPV in equation (6). Once DV directional control is available, scalar GRO and GO volitional signals can transform a stereotyped series of DV's into motor performances whose sizes and speeds can be adjusted to match variable environmental conditions. Models which utilize DV's for their spatial and trajectory control have generically been called Vector Associative Maps, or VAMs (Gaudio and Grossberg, 1991).

Figure 8

9. The Two-Thirds Power Law Relating Curvature and Velocity

Another widely observed invariant of movement is the strong coupling between velocity and curvature (Morasso, 1981; Abend, Bizzi, and Morasso, 1982). In general, peaks in the curvature profile occur at troughs in the velocity profile. Lacquaniti, Terzuolo, and Viviani (1983) formulated a "two-thirds power law" to describe the empirical relation between curvature and velocity. This law relates angular velocity $A(t)$ to curvature $C(t)$ as $A(t) = kC(t)^{2/3}$, which can be rewritten for tangential velocity $V(t)$ as $V(t) = kR(t)^{1/3}$, where $R(t) = 1/C(t)$ denotes the radius of curvature. Figure 9a plots model curvature and model tangential velocity for the letter "b"; Figure 9b plots model tangential velocity alongside the tangential velocity predicted from model curvature by the two-thirds power law. The agreement is close but not perfect. Indeed, the two-thirds power function is itself an imperfect descriptor of human performance. Wann, Nimmo-Smith, and Wing (1988) have noted that one basis for the discrepancy is that human velocity profiles are not perfectly symmetrical about the peak velocity value. VITE velocity profiles show the same duration-dependent deviation from perfect symmetry that is seen in human handwriting movements (Bullock and Grossberg, 1988, 1991; Nagasaki, 1989).

Figure 9

10. Complex Skilled Movement as an Emergent Property

The VITEWRITE model demonstrates how a multi-channel VITE trajectory generator, controlling a suitably designed hand with redundant degrees of freedom, enables a simple motor program to generate complex curvilinear movements that have many of the properties that humans exhibit when they produce cursive script. In particular, the existence of a

DV_m -GO processing stage enables the VITE model to trigger read-out of new motor commands at peak values of a synergy's outflow velocity profile. Using this trigger, the DV_m 's that update the TPV and the PPV processing stages may be modulated by volitional GO signals that rescale the speed of handwriting without changing its form. Likewise, the use of a motor program that consists of planning vectors DV_p released by velocity-sensitive events in the trajectory generator enable volitional GRO signals to rescale the size of handwriting without changing its form. From a higher computational viewpoint, the use of difference vectors such as DV_p and DV_m in Figure 1, gated by volitional commands such as GRO and GO, and integrated to yield positional commands such as TPV and PPV, provide a computational framework for analysing how many goal-oriented complex movements are made and flexibly modified under variable task conditions. Neural network architectures in which these directional, volitional, and positional commands are interactively repeated have been called VAM Cascades (Gaudio and Grossberg, 1991). Accumulating theoretical and empirical evidence points to VAM Cascades as a computational framework for the control of planned biological movements. See Bullock, Grossberg, and Guenther (1993), Grossberg, *et al.* (1993), and Guenther *et al.* (1994) for further discussion.

The second part of this chapter addresses the question of how a working memory, such as the Vector Plan in Figure 1, may be organized. What is a working memory, and what are the organizational principles that govern its design?

11. What is a Working Memory?

Working memory is the type of memory in which a novel temporally ordered sequence of events, such as a telephone number, can be temporarily stored and then performed (Baddeley, 1976). Working memory is a kind of short-term memory (STM) and, unlike long-term memory (LTM), it can be quickly erased by a distracting event. There is a large experimental literature on the topic of working memory, as well as a variety of models (Atkinson and Shiffrin, 1971; Cohen and Grossberg, 1987; Cohen, Grossberg, and Stork, 1987; Elman, 1990; Grossberg, 1970, 1978a, 1978b; Grossberg and Pepe, 1971; Grossberg and Stone, 1986; Gutfreund and Mezard, 1988; Guyon, Personnaz, Nadal, and Dreyfus, 1988; Jordan, 1986; Reeves and Sperling, 1986; Schreter and Pfeifer, 1989; Seibert, 1991; Seibert and Waxman, 1990a, 1990b; Wang and Arbib, 1990).

The present class of models, called STORE (Sustained Temporal Order REcurrent) models, exhibit properties that have not previously been available in a dynamically defined working memory (Bradski, Carpenter, and Grossberg, 1992, 1994). In particular, STORE working memories are designed to encode the invariant temporal order of sequential events, or items, that may be presented with widely differing growth rates, amplitudes, durations, and interstimulus intervals. The STORE model is also designed to enable all possible groupings of the events stored in STM to be stably learned and remembered in LTM, even when new events are perturbing the system. In other words, these working memories enable chunks (also called compressed, categorical, or unitized representations) of a stored list to be encoded in LTM in a manner that is not erased by the continuous barrage of new inputs to the working memory.

Working memories with these properties are important in many applications which require properties of behavioral self-organization. One application is the VITEWRITE model described above or, more generally, working memories for sensorimotor planning whose dis-

tributed representations can be unitized during learning and subsequently read-out on demand during performance. In the case of the VITEWRITE model, the working memory reads out planning vectors DV_p that are generated during sensory-motor imitation and learning. Then volitional commands, such as the GO and GRO signals of the VITEWRITE model, can flexibly modify the invariant contents of the working memory to generate movements that can rapidly adapt to variable task conditions. Other applications include working memories for eye movement control (Grossberg and Kuperstein, 1989), for variable-rate speech perception (Cohen and Grossberg, 1986; Cohen, Grossberg, and Stork, 1987), and for 3-D visual object recognition (Bradski, Carpenter, and Grossberg, 1992).

12. Invariance Principle and Normalization Rule

The STORE neural network working memories are based upon algebraically characterized working memories that were introduced by Grossberg (1978a, 1978b). These algebraic working memories were designed to explain psychological data concerning working memory storage and recall. In these models, individual events are stored in working memory in such a way that the pattern of STM activity across event representations encodes both the events that have occurred and the temporal order in which they have occurred. In the cognitive literature, such a working memory is often said to store both *item* information and *order* information (Healy, 1975; Lee and Estes, 1981; Ratcliff, 1978). Item information is encoded by what nodes are active. Order information is represented by their relative activation, with the most active nodes performed first. The models also include a mechanism for reading out events in the stored temporal order. A rehearsal wave, or nonspecific arousal input, causes the most active node to be read-out first, as it self-inhibits its own activation via negative feedback to enable the next-most-active node to be read-out. An event sequence can hereby be performed from STM even if it is not yet incorporated through learning into LTM, much as a new telephone number can be repeated the first time that it is heard.

The large data base on working memory shows that storage and performance of temporal order information from working memory is not always veridical (Atkinson and Shiffrin, 1971; Baddeley, 1976; Reeves and Sperling, 1986) The observed deviations from veridical temporal order in STM were shown to follow from two postulates of the algebraic working memory model that have clear adaptive value. These principles are called the Invariance Principle and the Normalization Rule (Grossberg, 1978b).

Invariance Principle: The spatial patterns of STM activation across the event representations of a working memory are stored and updated in response to sequentially presented events in such a way as to leave invariant the temporal order of all groupings of previously presented events. In particular, a temporal list of events is encoded in STM in a way that preserves the stability of previously learned LTM codes for familiar sublists of the list.

For example, suppose that the word *MY* has previously been stored in a working memory's STM and has, through learning, established a learned chunk in LTM. Suppose that the word *MYSELF* is then stored for the first time in STM. The word *MY* is a syllable of *MYSELF*. The STM encoding of *MY* as a syllable of *MYSELF* may not be the same as its STM encoding as a word in its own right. On the other hand, *MY*'s STM encoding as part of *MYSELF* should not be allowed to force forgetting of the LTM code for *MY* as a word in its own right. If it did, familiar words, such as *MY*, could not be learned as parts of larger words, such as *MYSELF*, without eliminating the smaller words from the lexicon.

More generally, new wholes could not be built from familiar parts without erasing LTM of the parts.

Figure 10

The Invariance Principle can be realized algebraically as follows, provided that no list items are repeated. Assume for simplicity that the i^{th} list item is preprocessed by a winner-take-all network. Each list item then activates a single output node of the preprocessor network. (Properties of the working memory also obtain if a finite set of output nodes is activated for each item.) In the winner-take-all case, the winning node that is activated by the i^{th} item send a binary input I_i to the first working memory level F_1 (Figure 10). Let x_i denote the activity of the i^{th} item representation of F_1 . Suppose that I_i is registered in working memory at time t_i . At time t_i , the activity pattern $(x_1(t_i), x_2(t_i), \dots, x_n(t_i))$ across F_1 stores the effects of the list I_1, I_2, \dots, I_i of previous inputs. The input I_i updates the activity values $x_k(t_{i-1})$ to new values $x_k(t_i)$ for all nodes $k = 1, 2, \dots, i$ according to the following rule:

$$x_k(t_i) = \begin{cases} 0 & \text{if } k > i \\ \mu_i & \text{if } k = i \\ \omega_i x_k(t_{i-1}) & \text{if } k < i. \end{cases} \quad (7)$$

By (7), at time t_i , the pattern $(x_1(t_{i-1}), x_2(t_{i-1}), \dots, x_{i-1}(t_{i-1}))$ of previously stored STM activities is multiplied by a common factor ω_i as the i^{th} item is stored with initial activity μ_i .

The storage rule (7) satisfies the Invariance Principle for the following reason. Suppose that F_1 is the first level of a two-level competitive learning, or self-organizing feature map, network (Grossberg, 1976). Then F_1 sends signals to the second level F_2 via an adaptive filter. The total input to the j^{th} F_2 node is $\sum_k x_k z_{kj}$, where z_{kj} denotes the LTM trace, or adaptive weight, in the path from the k^{th} F_1 node to the j^{th} F_2 node. In psychological terms, each active F_2 node represents a chunk of the F_1 activity pattern. When the j^{th} F_2 node is active, the LTM weights z_{kj} converge toward x_k ; in other words, the vector of LTM traces z_{kj} becomes parallel to the F_1 activity vector. When a new item i is added to the list, the Invariance Principle implies that the previously active items in the list will simply be multiplied by a common factor w_i , thereby maintaining a constant ratio between the STM activities of previously active items. Constant STM activity ratios imply that the STM activity vector at F_1 remains parallel to its LTM weight vectors as the magnitudes of the STM activities change under new inputs. Hence, adding new list items does not invalidate the STM and LTM codes for sublists. In particular, the temporal order of items in each sublist, encoded as relative sizes of both the STM and the LTM variables, remains invariant.

Normalization Rule: The Normalization Rule algebraically realizes the classical property of the limited capacity of STM (Atkinson and Shiffrin, 1971). A convenient statement of this property is given by the equation

$$S_i \equiv \sum_k x_k(t_i) = \mu_i \theta_i + S(1 - \theta_i), \quad (8)$$

where $\theta_1 = 1$ and θ_i decreases towards 0 as i increases. For example, let $\theta_i = \theta^{i-1}$, with $0 < \theta < 1$. Total activity S_i then increases toward an asymptote, S , as new items are presented. Parameter S characterizes the “limited capacity” of STM. In human subjects, this parameter is determined by biological constraints. In an artificial neural network, parameter S can be set at any finite value.

Using (7) and (8), it was proved in Grossberg (1978a) that the rate at which S_i approaches its asymptote S helps to determine the form of the STM activity pattern. The pattern (x_1, \dots, x_i) can exhibit primacy (all $x_{k-1} > x_k$), recency (all $x_{k-1} < x_k$), or bowing, which combines primacy for early items with recency for later items (Grossberg, 1978a). These patterns correspond to properties of STM storage by human subjects. Model parameters are typically set so that the STM activity pattern exhibits a primacy gradient in response to a short list. Since more active nodes are read-out of STM before less active nodes during performance trials, primacy storage leads to the correct order of recall in response to a short list. Using the same parameters, the STM activity pattern exhibits a bow in response to longer lists, and approaches a recency gradient in response to still longer lists. An STM bow leads to performance of items near the list beginning and end before items near the list middle. A larger STM activity at a node also leads to a higher probability of recall from that node when the network is perturbed by noise. An STM bow thus leads to earlier recall and to a higher probability of recall from items at the beginning and the end of a list.

These formal network properties are seen in experiments that test working memory in human subjects, such as free recall experiments during which subjects are asked to recall the items in a list after being exposed to them once in a prescribed order (Atkinson and Shiffrin, 1971; Healy, 1975; Lee and Estes, 1981). Effects of LTM on free recall data have also been analysed by the theory (Grossberg, 1978a, 1978b).

The multiplicative gating in equation (7) and the normalization rule in (8) are algebraic versions of the general types of properties which are found in shunting competitive feedback networks (Grossberg, 1973). A task of STORE model research was to design specialized shunting networks which realize equations (7) and (8) as emergent properties of their real-time dynamics (Bradski, Carpenter, and Grossberg, 1992). The STORE model is a real-time shunting network, defined below, which exhibits the desired emergent properties. In particular, the STORE system moves from primacy to bowing to recency as a single model parameter is increased.

13. Working Memory Invariance under Variable Input Speed, Duration, and Interstimulus Interval

Two types of real-time working memory models, *transient* models and *sustained* models, can realize the invariance and normalization properties. In a transient model, presentation of items with different durations can alter the temporal order of previously stored items. Transient memory models can still accurately represent temporal order if input durations are controlled by a preprocessing stage to have a constant duration. Sustained models allow input durations and interitem intervals to be essentially arbitrary: so long as these intervals are not too short, variations in input speed, duration, and interstimulus interval have no effect on the temporal order that is stored in STM. A sustained neural network model is defined below. This two-level STORE model codes lists of distinct items. A variant of the STORE model design can encode the invariant temporal order of lists in which each item

may occur multiple times (Bradski, Carpenter, and Grossberg, 1994). Each item may also be represented by multiple nodes.

The first level of the STORE model (Figure 10) consists of nodes with STM activity x_i . The i^{th} item is assumed to send a unit input I_i to the i^{th} node for a time interval of length α_i . After an interstimulus interval of length β_i , the next item sends an input to the $(i+1)^{\text{st}}$ node, and so on. Each STM node also receives shunting inhibition via a nonspecific feedback signal that is proportional to the total STM activity x . The second STORE level consists of excitatory interneurons whose activity y_i tracks x_i . A critical additional factor in the model is gain control that enables changes in x_i to occur only when an input is present and enables changes in y_i to occur only when no input is present.

This alternating gain control allows feedback from y_k to x_k ($k < i$) to preserve previously stored patterns even when a new input I_i is on for a long time interval. These processes are defined below in the simplest way possible to permit complete analysis and understanding of the model's emergent properties.

The STORE model is defined by the dimensionless equations

$$\frac{dx_i}{dt} = [AI_i + y_i - x_i x]I \quad (9)$$

and

$$\frac{dy_i}{dt} = [x_i - y_i]I^c, \quad (10)$$

where

$$x \equiv \sum_k x_k, \quad (11)$$

$$I \equiv \sum_k I_k, \quad (12)$$

$$I^c \equiv 1 - I, \quad (13)$$

and

$$x_i(0) = y_i(0) = 0. \quad (14)$$

Parameter A in (9) scales the relative size of bottom-up inputs I_i to top-down feedback signals y_i . In (13), the notation "c" in I^c designates that the values of I^c are complementary to those of I ; when I is on in (9), I^c is off in (10), and conversely. The input sequence I_i is given by

$$I_i(t) = \begin{cases} 1 & \text{if } t_i - \alpha_i < t < t_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

(Figure 10b). The input durations (α_i) and the interstimulus intervals ($\beta_i = t_i - \alpha_i - t_{i-1}$) are assumed to be large relative to the dimensionless relaxation times of x_i and y_i , set equal to 1 in equations (9) and (10). Thus each x_i reaches steady state when inputs are on ($I = 1$)

and each y_i reaches x_i when inputs are off ($I^c = 1$). Otherwise, t_i and α_i can be arbitrary, and their values have no effect on patterns of memory storage.

The following properties of the STORE model are proved in Bradski, Carpenter and Grossberg (1992). The relative sizes of the activities in pattern (x_1, \dots, x_{i-1}) are preserved when x_i becomes active. For large values of A , the total STM activity is approximately normalized at all times, whereas for small A , it grows rapidly as more inputs perturb the network. Since the size of parameter A in (9) reflects the degree to which the input I_i influences the STM pattern, recency for large A (present input dominates) and primacy for small A (past activities dominate) would be intuitively predicted. In fact, for large A , the pattern of STM activity (x_1, \dots, x_i) always shows a recency gradient. For small A , the STM patterns in response to short lists show a primacy gradient. In particular, if $x_1(t_1) > A$, $(x_1 \dots x_i)$ shows a primacy gradient until $x_i(t_i) \leq A$. Presenting additional inputs I_{i+1}, I_{i+2}, \dots causes the STM pattern to bow. If $x_1(t_1) \leq A$, the STM pattern always exhibits recency. Recency occurs for all list lengths whenever $A \geq 1$, while small A values allow relatively long lists to be stored by primacy gradients. The position at which the STM pattern bows can be calculated iteratively. For example, the bow occurs at position $i = 2$ if $1 > A \geq .5(3 - \sqrt{5}) \cong 0.382$.

Figure 11

These properties of the STORE model are illustrated by the computer simulations summarized in Figure 11. Each row depicts STM storage of a list at a fixed value of A . In the left column, the STM vector (x_1, x_2, \dots, x_7) is depicted at times t_1, t_2, \dots, t_7 when successive inputs I_1, I_2, \dots, I_7 are stored. Each activity x_i is represented by the height of a vertical bar. The top row depicts a recency gradient, the seventh row a primacy gradient, and intermediate columns represent bows at each successive list position. The middle column graphs the ratios x_i/x_{i+1} through time. The horizontal graphs mean that the Invariance Principle is obeyed as soon as both items in each ratio are stored. The third column graphs the growth of total activity $x(t)$ to its capacity S . The input durations α_i in equation (15) varied randomly between 10 and 40. Such variations in input parameters had no discernible effect on the stored STM patterns.

14. Concluding Remarks

The VITEWRITE and STORE models suggest several general conclusions about the temporal organization of planned behaviors. First, the temporal properties of behavior may not be explicitly encoded in their controlling neural circuits. Rather, they may be implicitly coded distributed properties that emerge as a result of interactions across multiple neurons and processing stages. For example, in the VITEWRITE model, feedback interactions between a Vector Plan working memory and a VITE trajectory generator determine the form and timing of handwriting production. The “motor program” is an emergent property of these feedback interactions. These feedback interactions enable simple GO and GRO volitional commands to flexibly alter the speed, size, and style of the handwriting representation that is invariantly coded in the Vector Plan. Likewise, the temporal order of the item representations that are remembered by a STORE working memory is implicitly coded by the relative sizes of the item representation activities. A nonspecific rehearsal wave, like the feedback signal from the $DV_m \cdot GO$ stage in Figure 1, translates these relative activities into

the order of recall, with the largest activities being read-out earliest while self-inhibiting themselves in order to prevent perseverative read-out of the same items over and over again. The relative activities, in turn, are stored in a way that enables all possible groupings of stored items to be stably learned and recalled by unitized representations, or chunks, with which the working memory reciprocally interacts.

The second general conclusion is that different organizational principles may govern the design of neural circuits that control different aspects of timed behavior. In particular, the principles governing the VITEWRITE model concern how movement synergies can be controlled in such a way that individual synergies may be synchronously performed at variable speeds, yet different synergies may be flexibly reorganized by volitional commands that govern their individual times, speeds, and sizes of production. In contrast, principles governing the STORE model concern how sequences of events may be represented in STM in such a way that they may be stably unitized in and recalled from LTM. Different brain regions are, moreover, implicated in the control of these different neural designs. The VITE model has been used to explain data about parietal cortex, motor cortex and basal ganglia, whereas the STORE model may be used to interpret data about the frontal cortex and its interactions with other brain regions.

The principles that govern the VITEWRITE and STORE models do not exhaust the model neural designs that have been used to explain behavioral and neural data about temporally organized behaviors. Some examples of neural models constructed by our research group are listed below as well as references that discuss the work of many other authors. For example, conditioning of individual motor acts during reinforcement learning is adaptively timed. This process is modelled by a Spectral Timing model that is compared with data from hippocampus and cerebellum (Bullock, Fiala, and Grossberg, 1994; Grossberg and Merrill, 1992; Grossberg and Schmajuk, 1989). Circadian rhythms that control mammalian sleep and wake cycles are regulated on a much slower time scale. This oscillatory process is modelled by a Gated Pacemaker model that is compared with data from the suprachiasmatic nuclei of the hypothalamus (Carpenter and Grossberg, 1983, 1984, 1985). Oscillatory movement gaits and gait transitions, such as the cat gaits (walk, trot, pace, gallop), and the human gaits (walk, run) operate on a much faster time scale than circadian rhythms. These oscillatory movements are modelled by a GO Gait Generator model that is compared with data about spinal cord, basal ganglia, and motor cortex, among other neural structures (Cohen, Grossberg, and Pribe, 1992, 1993a, 1993b, 1993c; Grossberg, Pribe, and Cohen, 1993). A central task of computational neuroscience is to further develop neural models of these various types of temporally organized behavior and to show how they may be integrated into a unified neural architecture for the real-time control of intelligent adaptive behavior.

References

- Abend, W., Bizzi, E., and Morasso, P. (1982). Human arm trajectory formation. *Brain*, **105**, 331–348.
- Atkinson, R.C., and Shiffrin, R.M. (1971). The control of short term memory. *Scientific American*, August, 82–90.
- Baddeley, A.D. (1976). **The psychology of memory**. New York: Basic Books.
- Bradski, G., Carpenter, G.A., and Grossberg, S. (1992). Working memory networks for learning temporal order with application to 3-D visual object recognition. *Neural Computation*, **4**, 270–286.
- Bradski, G., Carpenter, G.A., and Grossberg, S. (1994). Store working memory networks for storage and recall of arbitrary temporal sequences. *Biological Cybernetics*, in press.
- Bullock, D., Contreras-Vidal, J.L., and Grossberg, S. (1992). Equilibria and dynamics of a neural network model for opponent muscle control. In G. Bekey and K. Goldberg (Eds.), **Neural networks in robotics**. Norwell, MA: Kluwer Academic, 439–457.
- Bullock, D., Fiala, J.C., and Grossberg, S. (1994). A neural model of timed response learning in the cerebellum. *Neural Networks*, in press.
- Bullock, D. and Grossberg, S. (1988). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, **95**, 49–90.
- Bullock, D. and Grossberg, S. (1991). Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling. *Human Movement Science*, **10**, 3–53.
- Bullock, D., Grossberg, S., and Guenther, F.H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, **5**, 408–43.
- Bullock, D., Grossberg, S., and Mannes, C. (1993). A neural network model for cursive script production. *Biological Cybernetics*, in press. Technical Report CAS/CNS TR-92-029. Boston, MA: Boston University.
- Carpenter, G.A. and Grossberg, S. (1983). A neural theory of circadian rhythms: The gated pacemaker. *Biological Cybernetics*, **48**, 35–59.
- Carpenter, G.A. and Grossberg, S. (1984). A neural theory of circadian rhythms: Aschoff's rule in diurnal and nocturnal mammals. *American Journal of Physiology (Regulatory, Integrative and Comparative Physiology)*, **247**, R1067–R1082.
- Carpenter, G.A. and Grossberg, S. (1985). A neural theory of circadian rhythms: Split rhythms, after-effects, and motivational interactions. *Journal of Theoretical Biology*, **113**, 163–223.
- Cohen, M. and Grossberg, S. (1986). Neural dynamics of speech and language coding: Developmental programs, perceptual grouping, and competition for short term memory. *Human Neurobiology*, **5**, 1–22.
- Cohen, M. and Grossberg, S. (1987). Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple groupings of patterned data. *Applied Optics*, **26**, 1866–1891.

- Cohen, M.A., Grossberg, S., and Pribe, C. (1992). A neural pattern generator that exhibits frequency-dependent in-phase and anti-phase oscillations. **Proceedings of the international joint conference on neural networks, IV**, 146–151. Piscataway, NJ: IEEE Service Center.
- Cohen, M.A., Grossberg, S., and Pribe, C. (1993a). A neural pattern generator that exhibits arousal-dependent human gait transitions. **Proceedings of the world congress on neural networks, IV**, 285–288. Hillsdale, NJ: Erlbaum Associates.
- Cohen, M.A., Grossberg, S., and Pribe, C. (1993b). Frequency-dependent phase transitions in the coordination of human bimanual tasks. **Proceedings of the world congress on neural networks, IV**, 491–494. Piscataway, NJ: IEEE Service Center.
- Cohen, M.A., Grossberg, S., and Stork, D. (1987). Recent developments in a neural model of real-time speech analysis and synthesis. In M. Caudill and C. Butler (Eds.) **Proceedings of the IEEE international conference on neural networks, IV**, San Diego, 443–454.
- Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, **14**, 179–211.
- Fetters, L. and Todd, J. (1987). Quantitative assessment of infant reaching movements. *Journal of Motor Behavior*, **19**, 147–166.
- Gaudio, P. and Grossberg, S. (1991). Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks*, **4**, 147–183.
- Georgopoulos, A.P., Kalaska, J.F., Caminiti, R., and Massey, J.T. (1984). The representation of movement direction in the motor cortex: Single cell and population studies. In G.M. Edelman, W.E. Gall, and W.M. Cowan (Eds.), **Dynamic aspects of neocortical function**. New York: Wiley, 501–524.
- Golani, I. (1992). A mobility gradient in the organization of vertebrate movement: The perception of movement through symbolic language. *Behavioral and Brain Sciences*, **15**, 249–308.
- Grossberg, S. (1970). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, II. *Studies in Applied Mathematics*, **49**, 135–166.
- Grossberg, S. (1973). Contour enhancement, short-term memory and constancies in reverberating neural networks. *Studies in Applied Mathematics*, **52**, 217–257.
- Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors, *Biological Cybernetics*, **23**, 121–134.
- Grossberg, S. (1978a). Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models? *Journal of Mathematical Psychology*, **17**, 199–219.
- Grossberg, S. (1978b). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In R. Rosen and F. Snell (Eds.), **Progress in theoretical biology**, **5**, 233–374. New York: Academic Press. Reprinted in Grossberg, S. (Ed.) (1982). **Studies of mind and brain**. Boston: Reidel Press.
- Grossberg, S. and Kuperstein, M. (1989). **Neural dynamics of adaptive sensory-motor control: Expanded edition**. Elmsford, NY: Pergamon Press.
- Grossberg, S. and Merrill, J.W.L. (1992). A neural network model of adaptively timed reinforcement learning and hippocampal dynamics. *Cognitive Brain Research*, **1**, 3–38.

June 28, 1994

- Grossberg, S. and Pepe, J. (1971). Spiking threshold and overarousal effects in serial learning. *Journal of Statistical Physics*, **3**, 95–125.
- Grossberg, S., Pribe, C. and Cohen, M.A. (1993). Neural control of interlimb coordination and gait timing in bipeds and quadrupeds. Technical Report-TR-93-004, Boston, MA: Boston University.
- Grossberg, S. and Schmajuk, N.A. (1989). Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, **2**, 79–102.
- Grossberg, S. and Stone, G.O. (1986). Neural dynamics of attention switching and temporal order information in short term memory. *Memory and Cognition*, **14**, 451–468.
- Guenther, F.H., Bullock, D., Greve, D., and Grossberg, S. (1994). Neural representations for sensory-motor control, III: Learning a body-centered representation of 3-D target position. *Journal of Cognitive Neuroscience*, in press.
- Gutfreund, H. and Mezard, M. (1988). Processing of temporal sequences in neural networks. *Physiology Review Letters*, **61**, 235–238.
- Guyon, I., Personnaz, L., Nadal, J.P., and Dreyfus, G. (1988). Storage retrieval of complex sequences in neural networks. *Physiology Review A*, **38**, 6365–6372.
- Healy, A.F. (1975). Separating item from order information in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, **13**, 644–655.
- Horak, F.B. and Anderson, M.E. (1984a). Influence of globus pallidus on arm movements in monkeys. I. Effects of kainic acid-induced lesions. *Journal of Neurophysiology*, **52**, 290–304.
- Horak, F.B. and Anderson, M.E. (1984b). Influence of globus pallidus on arm movements in monkeys. II. Effects of stimulation. *Journal of Neurophysiology*, **52**, 305–322.
- Jordan, M.I. (1986). Serial order: A parallel distributed processing approach. **Institute for Cognitive Science, Report 8604**. University of California, San Diego.
- Lacquaniti, F., Ferrigno, G., Pedotti, A., Soechting, J.F., and Terzuolo, C. (1987). Changes in spatial scale in drawing and handwriting: Kinematic contributions by proximal and distal joints. *The Journal of Neuroscience*, **7**(3), 819–828.
- Lacquaniti, F., Terzuolo, C., and Viviani, P. (1983). The law relating kinematic and figural aspects of drawing movements. *Acta Psychologica*, **54**, 115–130.
- Lee, C. and Estes, W.K. (1981). Item and order information in short-term memory: Evidence for multilevel perturbation processes. *Journal of Experimental Psychology: Human Learning and Memory*, **1**, 149–169.
- Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, **42**, 223–227.
- Nagasaki, H. (1989). Asymmetric velocity profiles and acceleration profiles of human arm movements. *Experimental Brain Research*, **74**, 319–326.
- Parent, A. (1990). Extrinsic connections of the basal ganglia. *Trends in Neurosciences*, **13**, 254–258.
- Plamondon, R. (1989). Handwriting control: A functional model. In R.M.J. Cotterill (Ed.), **Models of brain function**. Cambridge: Cambridge University Press.
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, **85**, 59–108.

- Reeves, A. and Sperling, G. (1986). Attention gating in short-term visual memory. *Psychological Review*, **93**, 180–206.
- Schomaker, L., Thomassen, A., and Teulings, H.L. (1989). A computational model of cursive handwriting. In R. Plamondon, C.Y. Suen, and M.L. Simner (Eds.), **Computer recognition and human production of handwriting**. Singapore: World Scientific, 153–177.
- Schreter, Z. and Pfeifer, R. (1989). Short-term memory/long-term memory interactions in connectionist simulations of psychological experiments on list learning. In L. Personnaz and G. Dreyfus (Eds.). **Neural networks from models to applications**. Paris: I.D.S.E.T.
- Seibert, M.C. (1991). Neural networks for machine vision: Learning three-dimensional object recognition. Boston University, Ph.D. Thesis.
- Seibert, M.C. and Waxman, A.M. (1990a). Learning aspect graph representations from view sequences. In D.S. Touretzky (Ed.) **Advances in neural information processing systems 2**. San Mateo, CA: Morgan Kaufmann Publishing, 258–265.
- Seibert, M.C. and Waxman, A.M. (1990b). Learning aspect graph representations of 3D objects in a neural network. In M. Caudill (Ed.). **Proceedings of the international joint conference on neural networks (IJCNN-90)**, Washington, D.C., **2**, 233–236; Hillsdale, NJ: Erlbaum.
- Soechting, J.F. and Terzuolo, C.A. (1987). Organization of arm movements is segmented. *Neuroscience*, **23**(1), 39–51.
- Viviani, P. and Terzuolo, C.A. (1980). Space-time invariance in learned motor skills. In G.E. Stelmach and J. Requin (Eds.), **Tutorials in motor behaviour**. Amsterdam: North-Holland, 525–533.
- Viviani, P. and Terzuolo, C.A. (1983). The organization of movement in handwriting and typing. In B. Butterworth (Ed.), **Language production**, **2**. New York: Academic Press, 103–146.
- Wang, D. and Arbib, M.A. (1990). Complex temporal sequence learning based on short-term memory. *Proceedings of the IEEE*, **78**(9), 1536–1543.
- Wann, J., Nimmo-Smith, I., and Wing, A.M. (1988). Relation between velocity and curvature in movement: Equivalence and divergence between a power law and a minimum-jerk model. *Journal of Experimental Psychology: Human Perception and Performance*, **14**(4), 622–637.

Figure and Table Captions

Figure 1. Schematic of the VITEWRITE model: A vector plan functions as a motor program that stores discrete planning Difference Vectors DV_p in a working memory. A GRO signal determines the size of script and a GO signal its speed of execution. After the vector plan and these will-to-act signals are activated, the circuit generates script automatically. Size-scaled planning vectors $DV_p \cdot \text{GRO}$ are read into a Target Position Vector (TPV). An outflow representation of present position, the Present Position Vector (PPV), is subtracted from the TPV to define a movement Difference Vector (DV_m). The DV_m is multiplied by the GO signal. The net signal $DV_m \cdot \text{GO}$ is integrated by the PPV until it equals the TPV. The signal $DV_m \cdot \text{GO}$ is thus an outflow representation of movement speed. It is used to automatically trigger read-out of the next planning vector DV_p . See text for details. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 2. The geometric model of the hand to be controlled, with three degrees of freedom: finger extension/retraction, which moves the pen along the up-down (Y) axis, vertical wrist rotation (supination/pronation), which has the effect of moving the pen along the left-right (X) axis, and horizontal wrist rotation (R), which has two effects: rotating the other two axes, and moving the pen left-right. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 3. A stroke that is greatly simplified by use of three degrees of freedom. Left: With two degrees of freedom, the stroke shown in the middle can only be obtained by a mix of bimodal and unimodal velocity profiles, since the horizontal component is non-zero before and after the bend. Right: Using a third degree of freedom (R), which acts much like X, allows production of the same shape with only unimodal velocity profiles. This presumably simplifies neural control. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 4. The VITE circuit, the neural controller of each component agonist-antagonist pair of the hand. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 5. An example showing how to generate the end-effector trajectory drawn in the left panel. V_x, V_y denote X and Y velocities, respectively. GO-signal values for each of these components are plotted below the velocity profiles. The smaller panels labeled 1-10 show the end-effector trajectory during the time interval along the axis which the panels touch above. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 6. An example of connecting letters by concatenating individual motor programs. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 7. Shape invariance with two different hand geometries: Panels (a) through (c) show perfect shape invariance of the letter “b”, scaled to three different sizes by choosing three different values for the GRO parameter S . The trajectories were reduced to fit in the panels. The numbers in the right and top corners of each panel indicate the panel’s size in

mm prior to reduction. The end effector position was calculated by equations (1) and (2). [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 8. Shape invariance under speed rescaling: The same motor program is executed at two different speeds, simulated by scaling the magnitude of the GO signal. Panel (a) shows the letter “b” executed at a “normal” speed ($G_o = 1$), panel (b) at twice that speed ($G_o = 2$). [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 9. Relationship between pen tip (tangential) velocity $V(t)$ and curvature for the letter “b.” The simulated pen tip trajectory $x(t), y(t)$ was least-squares fitted to a polynomial. Velocity was computed as $V(t) = (\dot{x}^2 + \dot{y}^2)^{1/2}$, and curvature by the formula $C(t) = (\dot{x}\ddot{y} - \dot{y}\ddot{x})/V(t)^3$. Plot (a) plots curvature and velocity, which show the expected inverse relationship. Plot (b) compares the velocity $V(t)$ with the predicted curvature $kR(t)^{1/3}$, $k = 10$ according to the two-thirds power law (Wann, Nimmo-Smith, and Wing, 1988). [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

Figure 10. (a) Elementary STORE model: STM activity x_i at level 1 registers the item input I_i , nonspecific shunting inhibition x , and level 2 STM y_i . STM activity y_i at level 2 registers x_i . Complementary input-driven gain signals I and I^c control STM processing at levels 1 and 2. (b) Input $I_i(t)$ equals 1 for $t_i - \alpha_i < t \leq t_i$. When all inputs are off ($t_i < t \leq t_i + \beta_i$) level 2 variables y_k relax to level 1 values $x_k(t_i)$. [Reprinted with permission from Bradski, Carpenter, and Grossberg (1992).]

Figure 11. STORE model simulations for different values of the input parameter A . Left column: Each row codes the stored STM pattern after each of 7 items is presented; bar height codes stored STM activity of each item, with activities of later items to the right of earlier item activities. The pointer marks the list position of minimum activity in each of the 7 activity patterns. Middle column: Ratios of successive activities x_i/x_{i+1} through time, with horizontal graphs designating LTM invariance. Right column: Normalized total STM activity S_i/S through time. The STM patterns $(x_1(t_i) \dots x_i(t_i))$ show recency for large A , bowing for intermediate values of A , and primacy for small values of A . Total activity $x(t_i) \equiv S_i$ grows toward the asymptote S as i increases for each value of A . When a new input I_i is stored, the previous pattern vector $(x_1 \dots x_{i-1})$ is amplified if $S_i \equiv x(t_i) < 1$; or depressed if $S_i \equiv x(t_i) > 1$; but the pattern of relative activities is preserved. For these simulations, input durations α_i were varied randomly between 10 and 40, with the intervals $(t_i - t_{i-1})$ set equal to 50. [Reprinted with permission from Bradski, Carpenter, and Grossberg (1992).]

Table 1. Notation for a motor program, characterizing the letter shape shown in Figure 6. X is launched first, with a target of 10 length units (corresponding to about 5mm). During the next half cycle, which is launched at the velocity peak of the X motion, executes an upward (Y) motion of 110 units. At the Y velocity peak, an X motion in the other direction is triggered. This temporally overlapping succession of X and Y is continued until the last pattern of the motor program, which launches no component, and so movement comes to a halt. Numbers in round brackets denote the TPV_{*i*} during the second half-cycle, i.e. the

June 28, 1994

decreasing part of the velocity profile. [Reprinted with permission from Bullock, Grossberg, and Mannes (1993).]

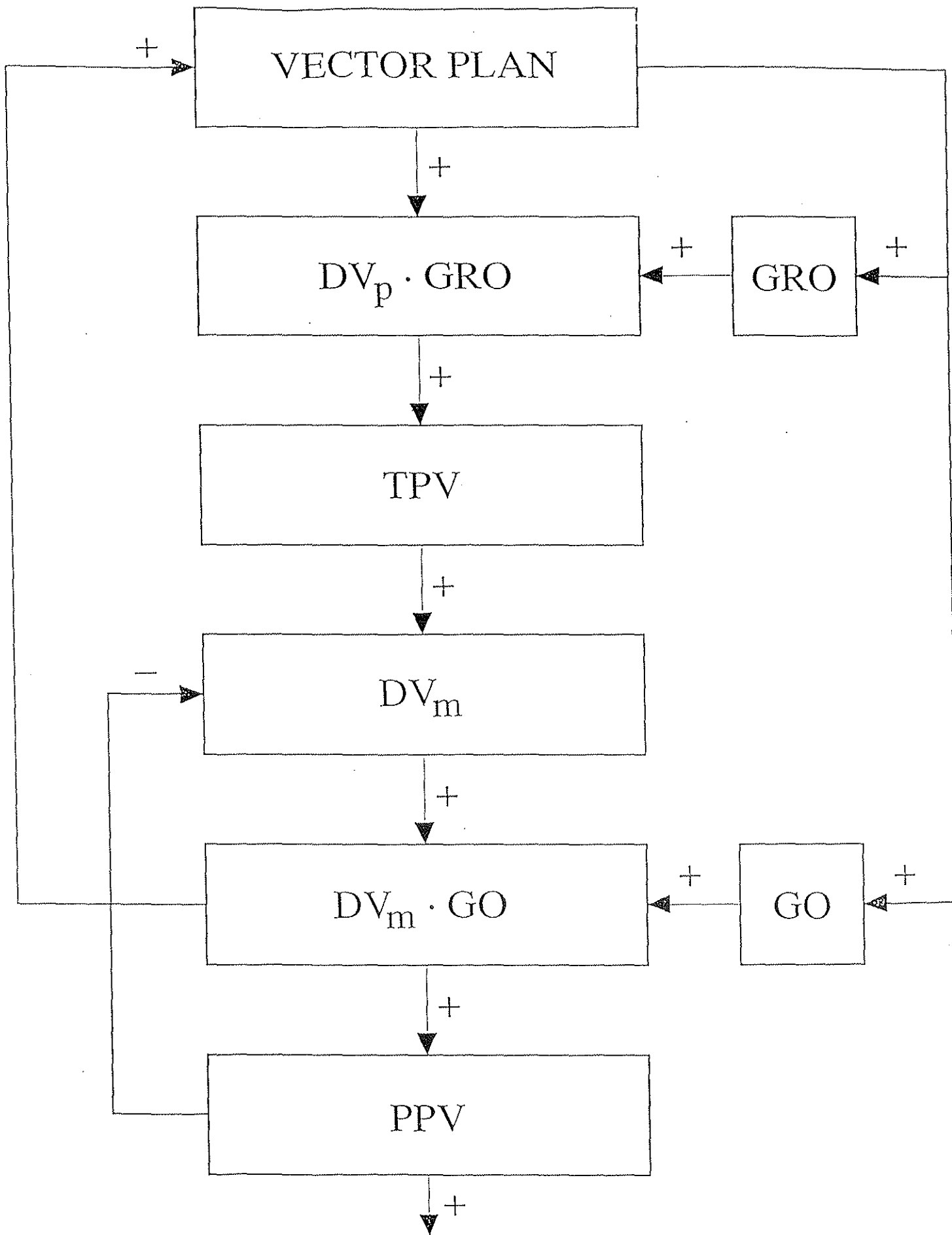


Figure 1

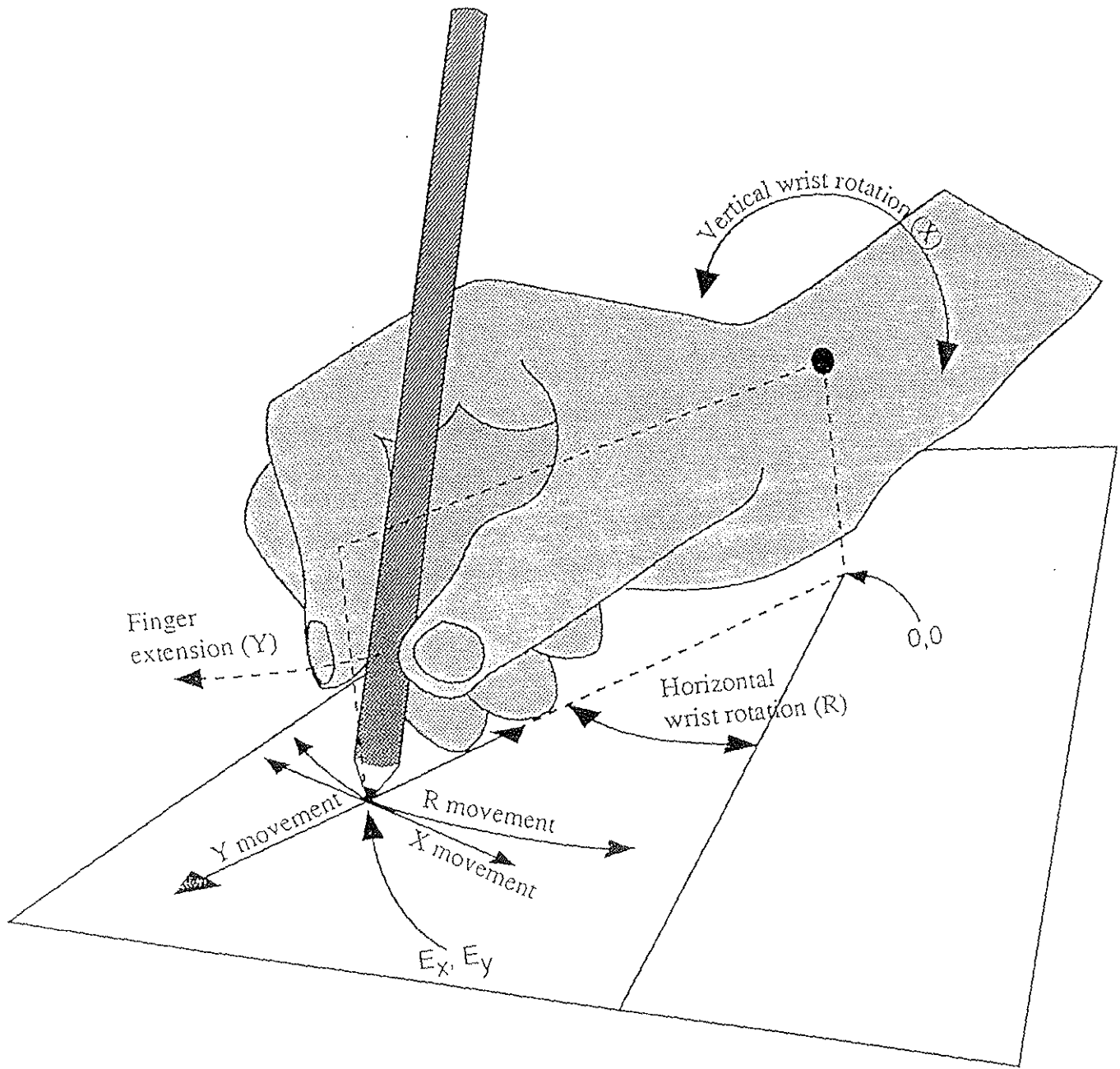


Figure 2

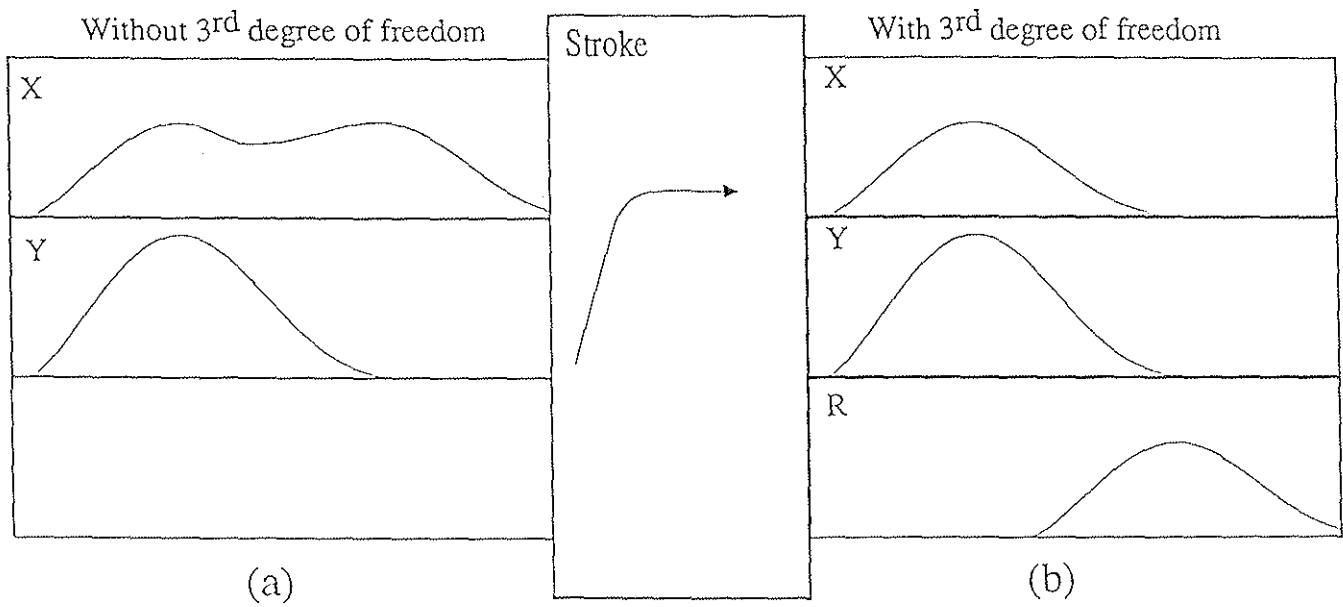


Figure 3

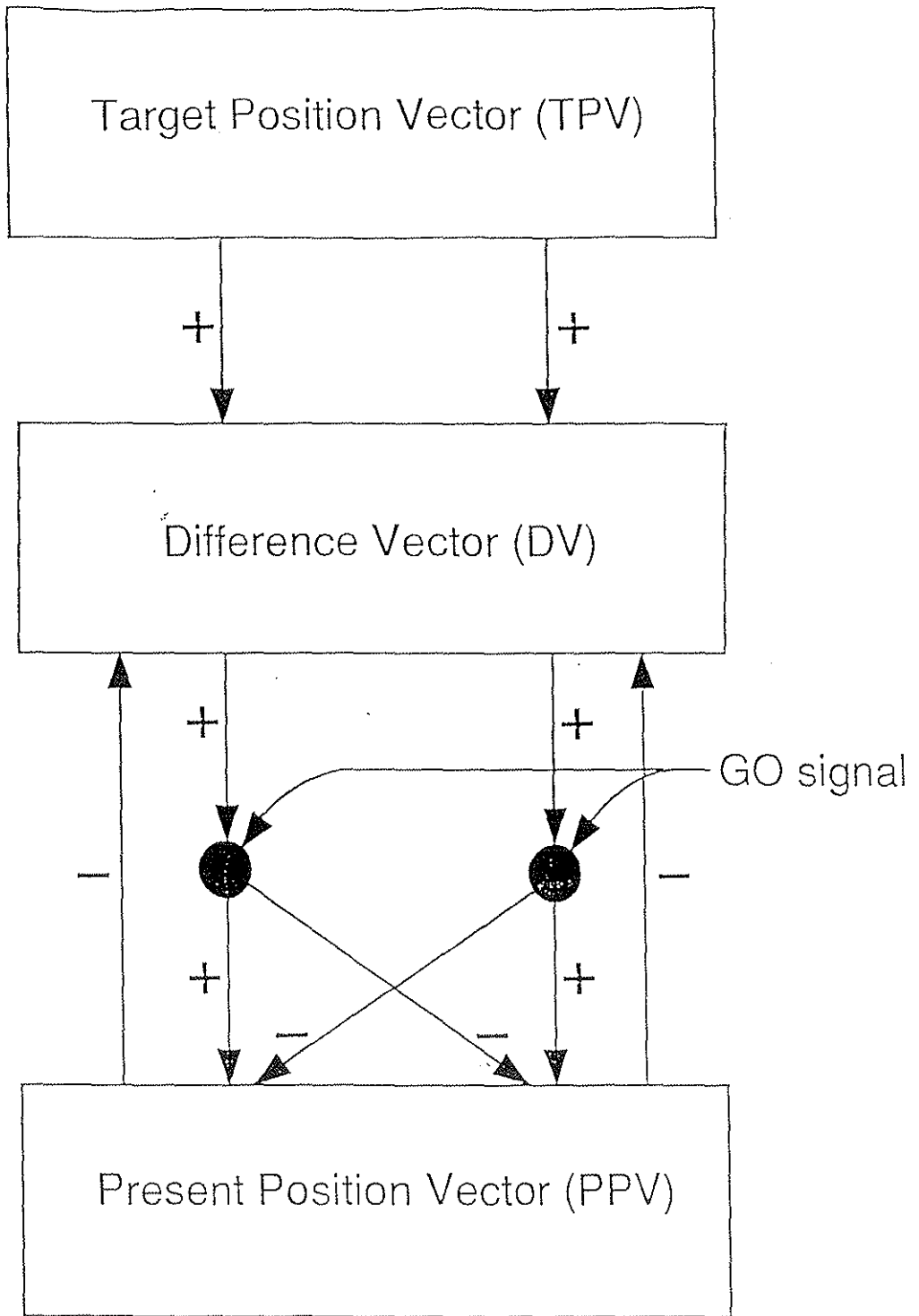


Figure 4

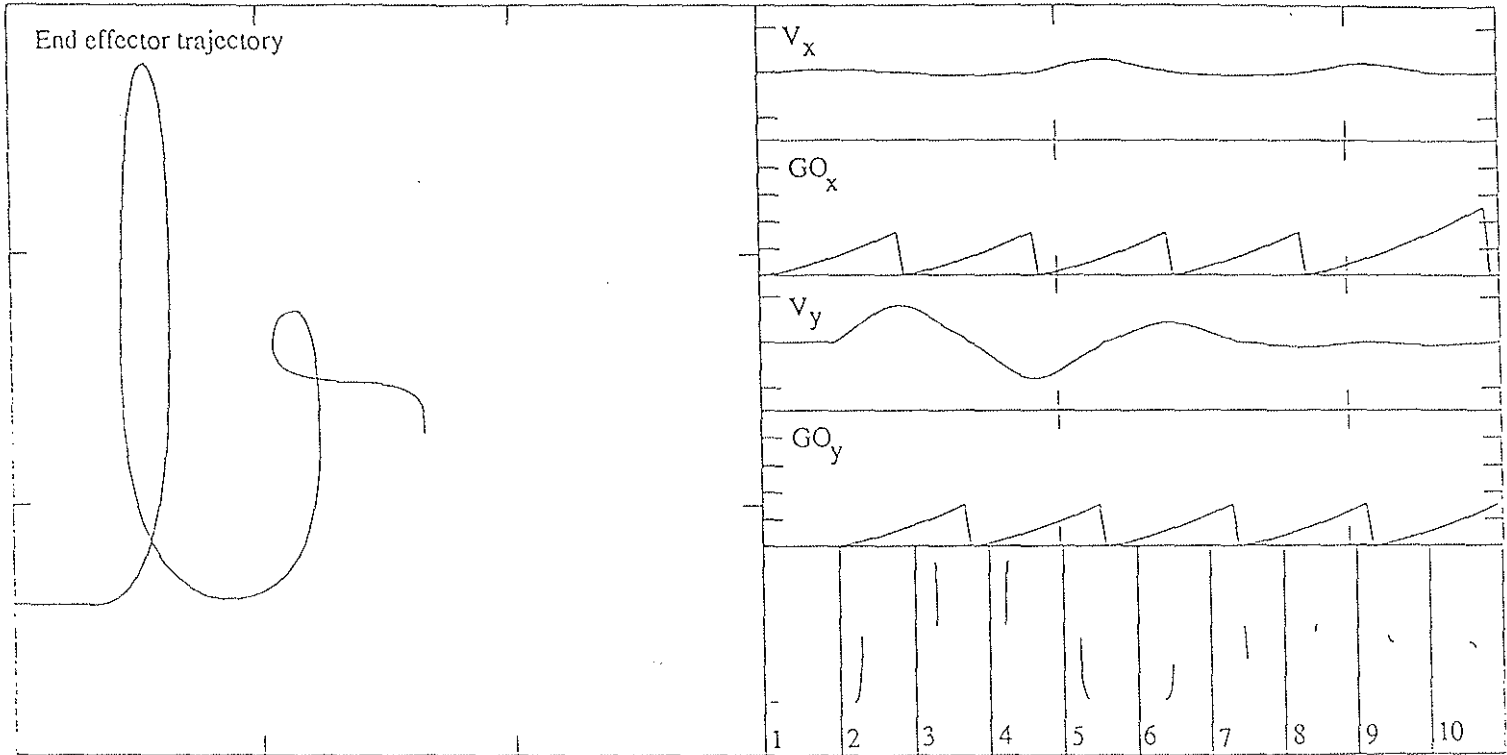


Figure 5

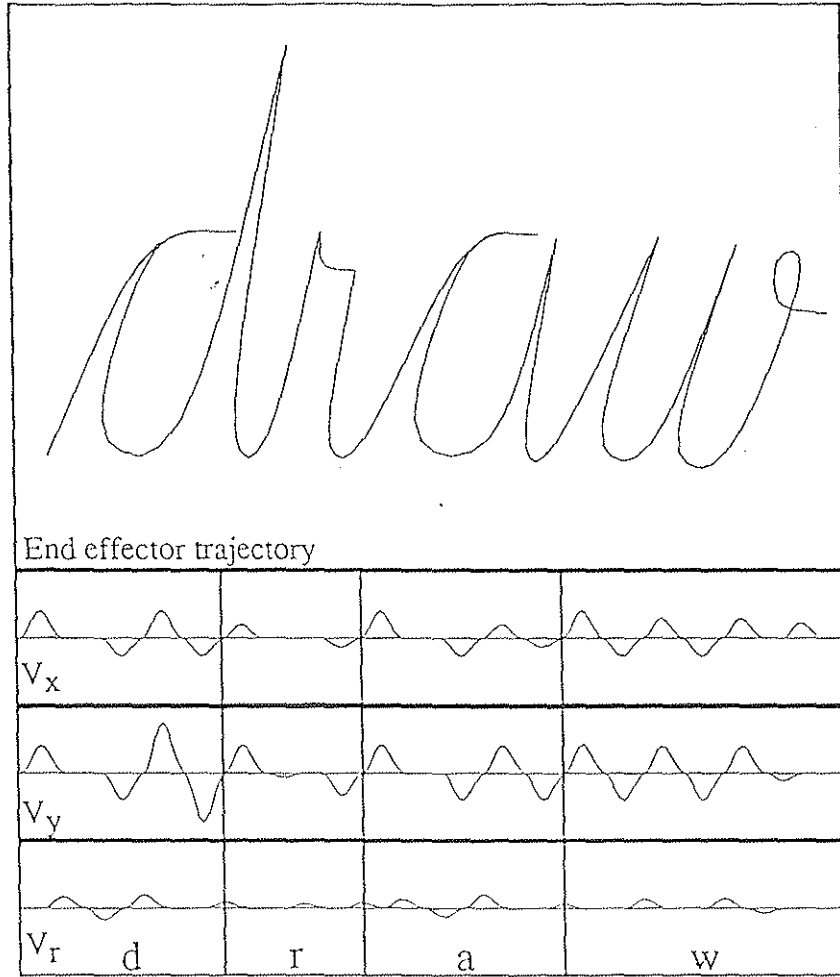


Figure 6

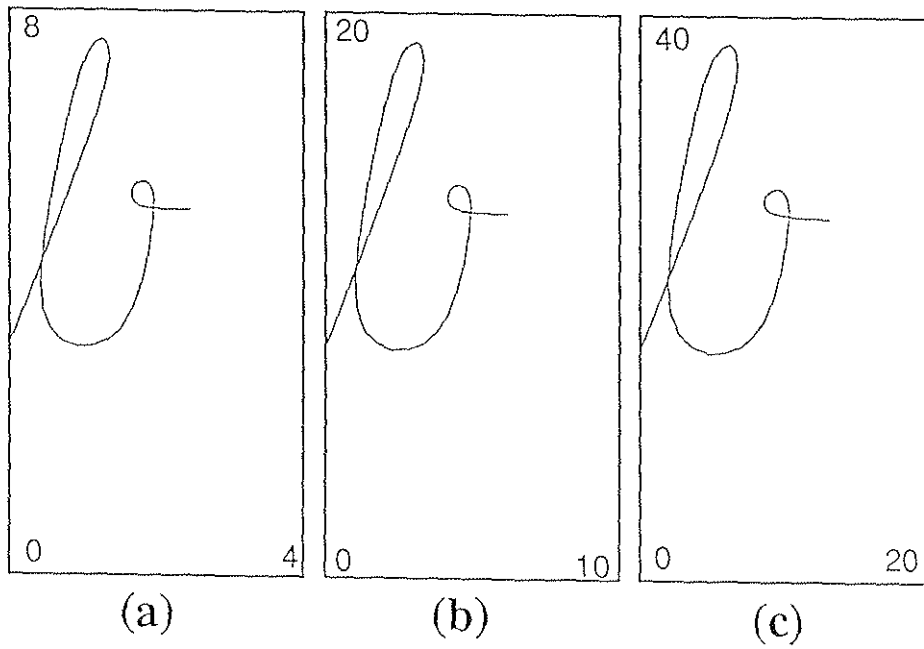


Figure 7

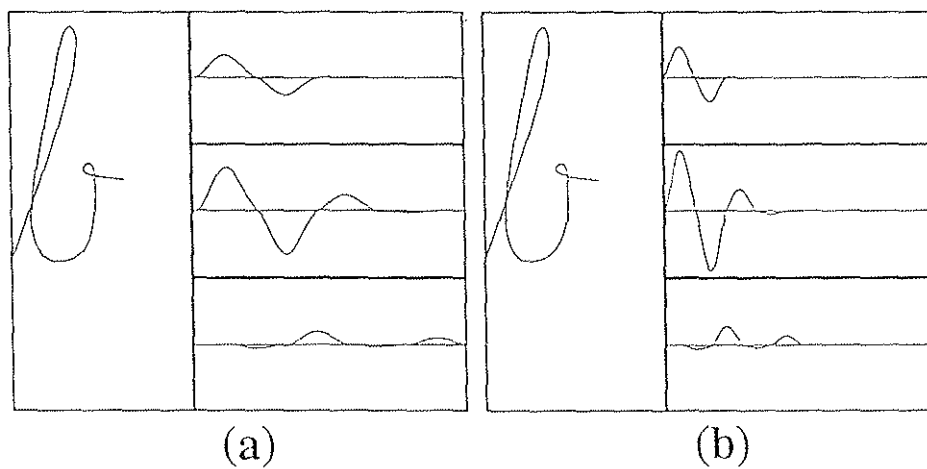


Figure 8

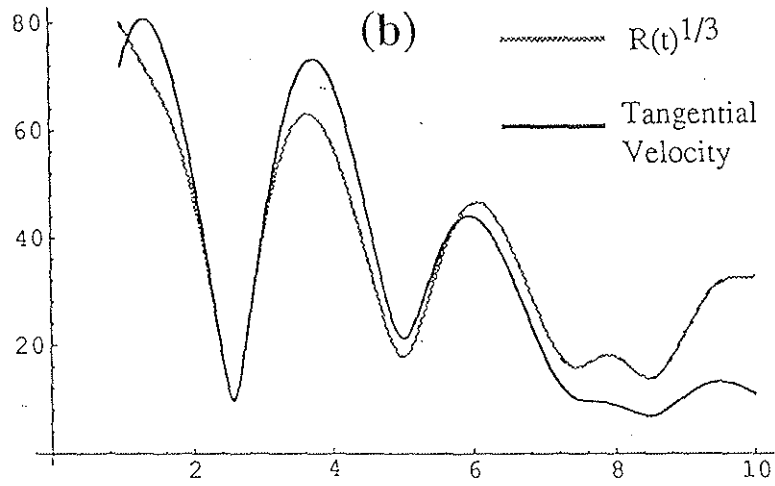
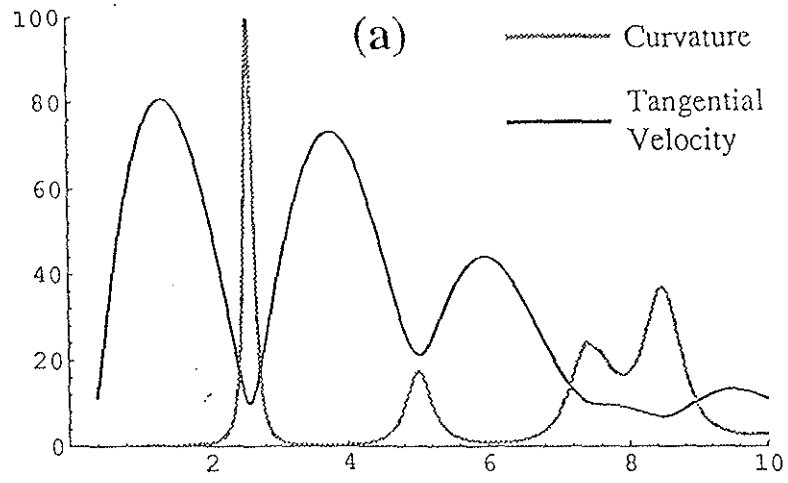
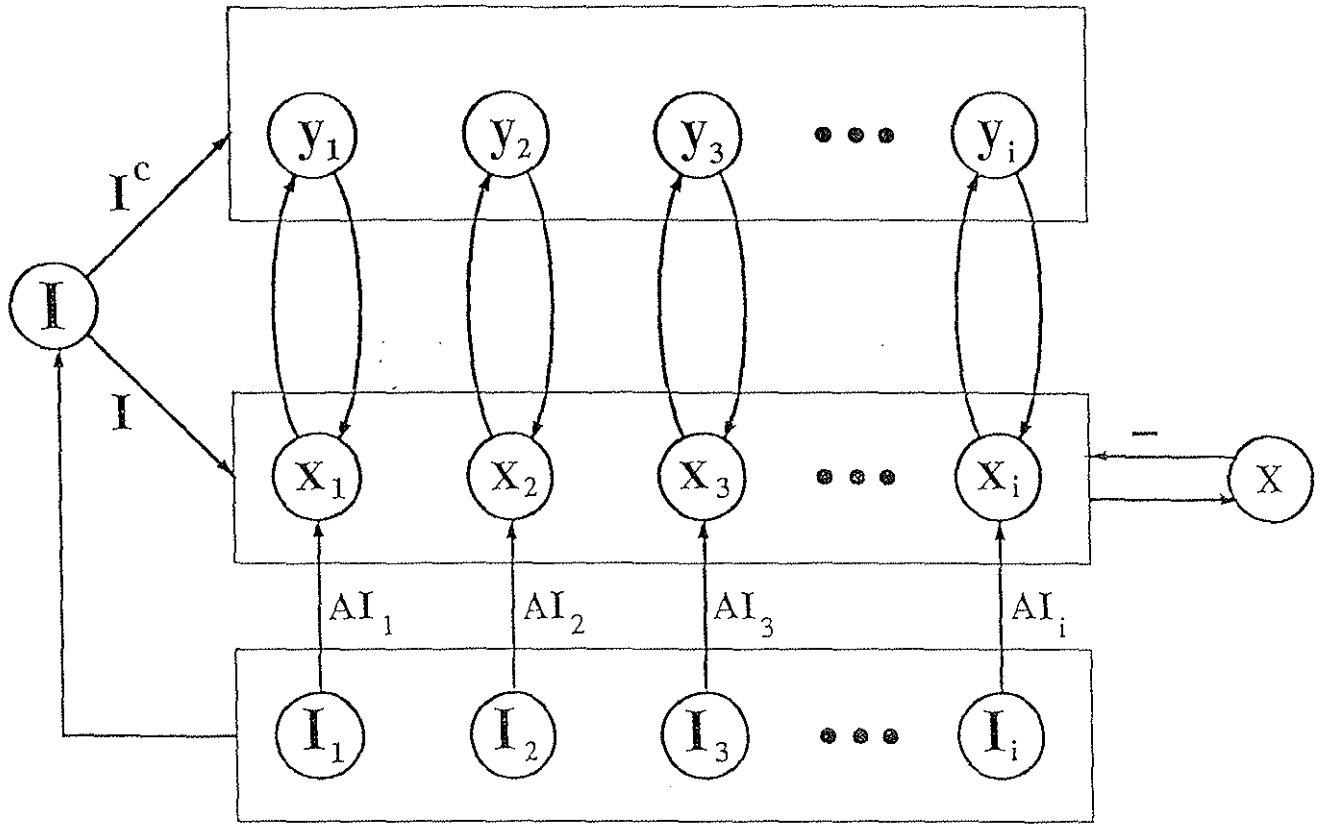


Figure 9

(a)



(b)

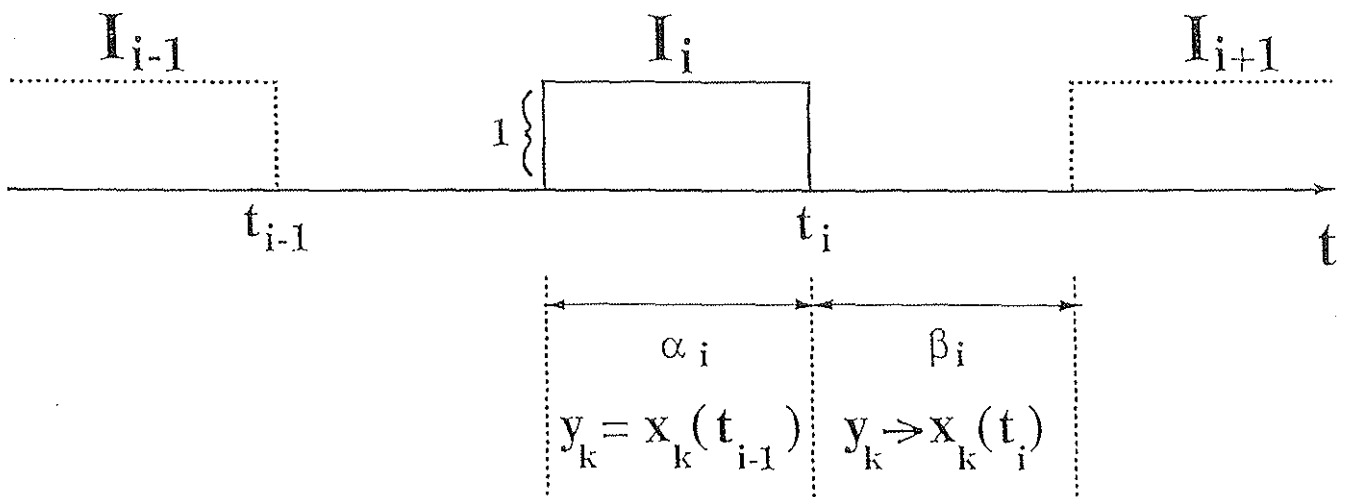


Figure 10

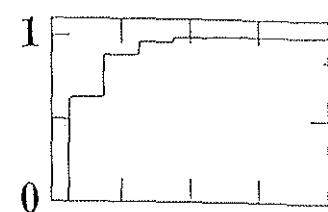
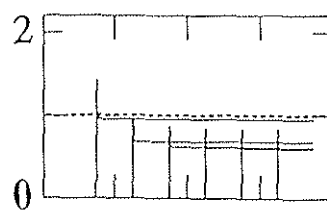
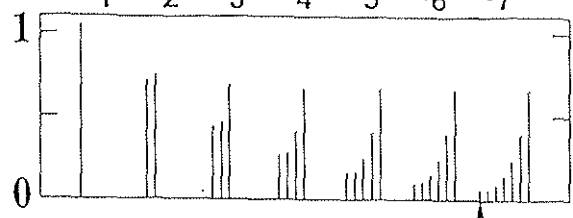
STM Pattern Invariance:

$$(X_1(t_i) \dots X_i(t_i))$$

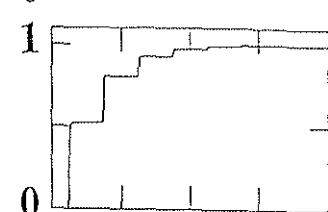
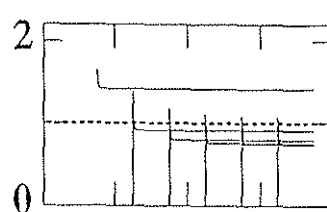
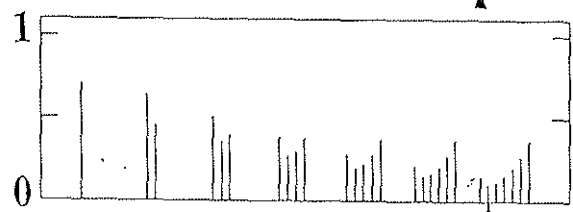
$t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7$

X_i/X_{i+1}

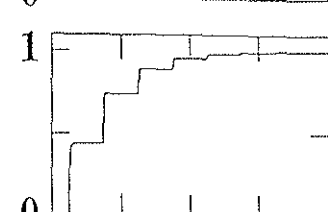
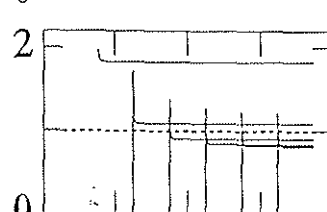
S_i/S



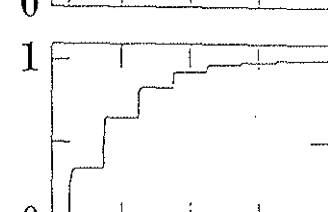
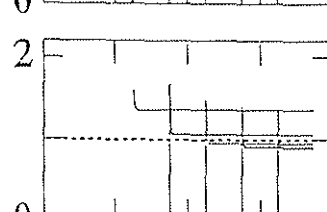
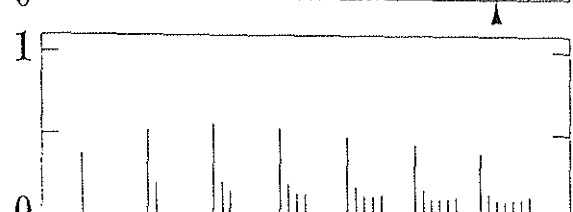
$A = 1.1$
Recency



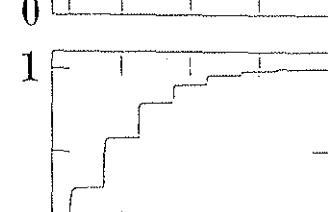
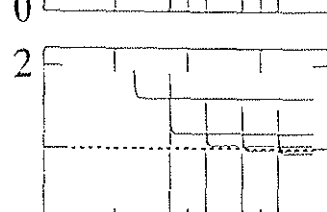
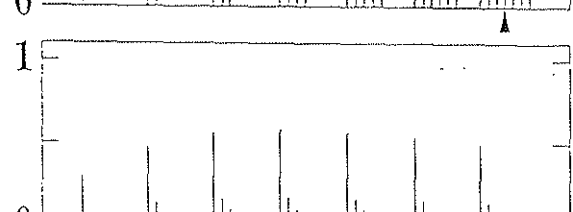
$A = 0.5$
Bow
Position 2



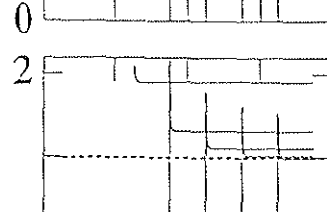
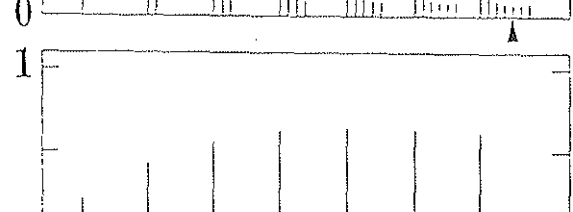
$A = 0.3$
Bow
Position 3



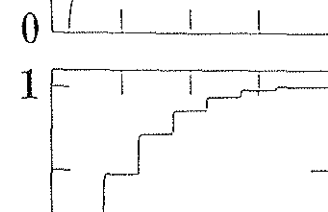
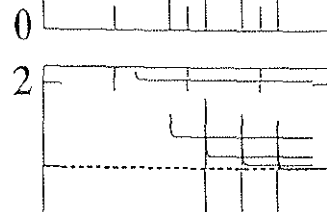
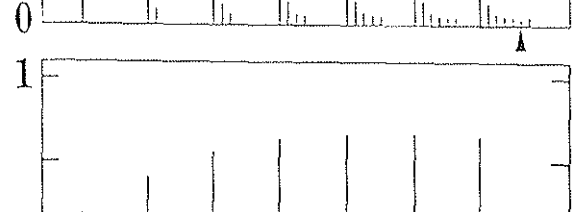
$A = 0.15$
Bow
Position 4



$A = 0.09$
Bow
Position 5



$A = 0.05$
Bow
Position 6



$A = 0.04$
Primacy

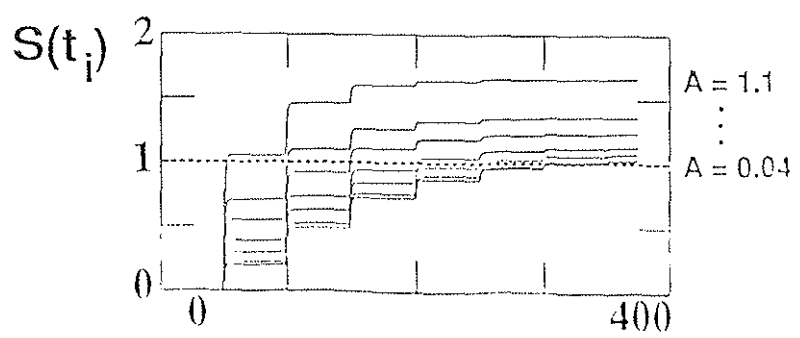


Figure 11

Half Cycle	X	Y	R
1	10	0	0
2	(10)	110	0
3	-10	(110)	0
4	(-10)	-110	0
5	40	(-110)	0
6	(40)	60	0
7	-10	(60)	0
8	(-10)	-15	0
9	30	(-15)	0
10	(30)	-10	0

Table 1