Boston University

OpenBU **http://open.bu.edu**

Cognitive & Neural Systems CAS/CNS Technical Reports

1993-03

A Neural Network Model for Cursive Script Production

https://hdl.handle.net/2144/2109

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

A neural network model for cursive script production

Daniel Bullock, Stephen Grossberg,
and Christian Mannes

September, 1992
Revised: March, 1993

Technical Report CAS/CNS-1992-029

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
677 Beacon Street
Boston, MA 02215

A NEURAL NETWORK MODEL FOR CURSIVE SCRIPT PRODUCTION

Daniel Bullock†, Stephen Grossberg‡, and Christian Mannes§
Center for Adaptive Systems
and
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, MA 02215

September, 1992
March, 1993

Boston University Technical Report CAS/CNS TR-92-029
Biological Cybernetics, in press

Reprint requests should be addressed to:
Professor Stephen Grossberg
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, MA 02215

---

† Supported in part by the National Science Foundation (NSF IRI 90-24877 and NSF IRI 87-16960) and the Office of Naval Research (ONR N00014-92-J-1309).

‡ Supported in part by Air Force Office of Scientific Research (AFOSR F49620-92-J-0499), DARPA (AFOSR 90-0083), the National Science Foundation (NSF IRI 90-24877 and NSF IRI 87-16960), and the Office of Naval Research (ONR N00014-92-J-1309).

§ Supported in part by the National Science Foundation (NSF IRI 90-24877).

Acknowledgements: The authors wish to thank Robin Locke for her valuable assistance in the preparation of the manuscript.

ABSTRACT

This article describes a neural network model, called the VITEWRITE model, for generating handwriting movements. The model consists of a sequential controller, or motor program, that interacts with a trajectory generator to move a hand with redundant degrees of freedom. The neural trajectory generator is the Vector Integration to Endpoint (VITE) model for synchronous variable-speed control of multijoint movements. VITE properties enable a simple control strategy to generate complex handwritten script if the hand model contains redundant degrees of freedom. The proposed controller launches transient directional commands to independent hand synergies at times when the hand begins to move, or when a velocity peak in a given synergy is achieved. The VITE model translates these temporally disjoint synergy commands into smooth curvilinear trajectories among temporally overlapping synergetic movements. The separate "score" of onset times used in most prior models is hereby replaced by a self-scaling activity-released "motor program" that uses few memory resources, enables each synergy to exhibit a unimodal velocity profile during any stroke, generates letters that are invariant under speed and size rescaling, and enables effortless connection of letter shapes into words. Speed and size rescaling are achieved by scalar GO and GRO signals that express computationally simple volitional commands. Psychophysical data concerning hand movements, such as the isochrony principle, asymmetric velocity profiles, and the two-thirds power law relating movement curvature and velocity arise as emergent properties of model interactions.

KEY WORDS handwriting, neural network, sensory-motor control, redundant control, VITE model.

1. Introduction

Skilled handwriting generally involves the coordinated action of a large number of joints, from the shoulder down to the joints of the fingers, each of which must be controlled by the muscle groups attached to them. This paper addresses how the kinematics of these joints may be controlled to produce the shapes of cursive script. In particular, we consider what the natural variables for the control of handwriting could be, to find out which parts of movement are explicitly planned—the motor program—and which parts are emergent properties of neural and mechanical interaction as the spatiotemporal motor trajectory unfolds.

A great deal of research has been devoted to explaining the kinematic signatures of point-to-point movements, such as the velocity and acceleration traces of joints during reaching. In particular, the Vector Integration To Endpoint (VITE) model (Bullock and Grossberg, 1988, 1991), upon which the model described in this paper is based, has been successful in explaining the generation of synchronous multi-joint reaching trajectories at variable speeds. However, handwriting goes far beyond simple point-to-point movement. The smooth, curved trajectories of a pen tip in cursive script express a motor plan that schedules and coordinates the time course of action of arm and hand synergies. Analyzing the geometry of a hand, one finds that no mere concatenation of point-to-point movements can produce the complex shapes of script. Rather, such trajectories appear to be generated by component synergies that overlap in time; that is, elementary actions have to be superimposed.

Superimposition of elementary strokes is a common assumption among modelers of handwriting (e.g. Morasso and Mussa-Ivaldi, 1982; Morasso, Mussa-Ivaldi, and Ruggiero, 1983; Edelman and Flash, 1987; Plamondon, 1989, 1992; Schomaker, Thomassen, and Teulings, 1989). Models differ in the constraints they place on stroke superimposition. Schomaker, Thomassen, and Teulings (1989), as well as Plamondon (1989, 1992), assume essentially arbitrary timing relations between onsets of overlapping movement components, whereas Morasso *et al.* (1983) constrain stoke superimposition by limiting the number of strokes that are concurrently executed to two.

Another important issue in handwriting is the choice of the most appropriate coordinate system for movement planning. Psychophysical studies of handwriting and drawing (Morasso, 1981, 1986) have shown that the spatial trajectory is more invariant than the joint rotations, or than force-time patterns (Teulings, Thomassen, and van Galen, 1986). Based on these findings, models for script generation have been proposed that assume planning in 2-D or 3-D space, with a continuous mapping from this space into the joint space that controls motor execution. Most models assume planning in a two degree of freedom system, for instance 2-D Cartesian space (Edelman and Flash, 1987; Schomaker, Thomassen, and Teulings, 1989). In particular, Schomaker *et al.* (1989) use a sinusoidal basis function. Plamondon (1989, 1992) describes pen tip trajectories in terms of differential geometry, using curvilinear and angular velocity generators. Dooijes (1983) proposes non-orthogonal "principle axes", and uses linear trend, the slow left-to-right motion that occurs during writing, as a third degree of freedom. In these models, parameters are externally chosen to adjust the onset and offset times, durations, amplitude and phase lags of component velocity profiles.

The VITEWRITE model, which is summarized in Figure 1, approaches the synergy control and degrees-of-freedom problems from a different perspective. It takes advantage of the fact that the human arm and hand have redundant degrees of freedom. The model demonstrates that these redundant degrees of freedom can be used to simplify the problem of motor planning. In particular, the VITEWRITE model demonstrates how a simple, but novel, type of motor program can control writing movements that exhibit many properties of human handwriting when it interacts with a suitably defined VITE trajectory generator coupled to a hand with redundant degrees of freedom. Our results thus extend the applicability of the VITE model from the control of reaching behaviors to the control of complex curvilinear trajectories.

Figure 1

Using a hand with redundant degrees of freedom, here taken to be three, simplifies the motor program, or plan, in at least three ways. First, each of the three motor synergies of such a hand can be controlled with unimodal velocity profiles. Second, the motor program consists of a discrete set of difference vectors that are read into a VITE circuit at prescribed times. These difference vectors represent the direction and desired amount of contraction of a motor synergy. They are called *planning vectors*, and denoted by DV_p below. Third, the motor program automatically launches transient directional commands to the 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.

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 is the DV_m·GO stage that is described below. The difference vectors DV_m that are multiplied by the GO signal are used to form continuous movement trajectories. They are not the discrete planning vector DV_p. The continuously changing DV_m vectors are called *movement vectors*. The GO signals that multiply the movement vectors are "will to act", or analog speed, signals that activate a motor synergy if its DV_m is not equal to zero. The DV_m·GO outflow commands then continuously move the synergy towards a desired target configuration until its DV_m equals zero. The maxima in time of these DV_m·GO outflow commands, in turn, can be used as control signals to read-out the next planning vector. Using this type of internal feedback loop, an increase in the GO signal can speed up a handwritten movement without changing its form. In a similar way, the GRO signal (defined below) can multiply the planning vectors DV_p before the net signals DV_p·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 motor program's temporally discrete and disjoint planning vectors DV_p·GRO into smooth curvilinear trajectories among temporally overlapping synergetic movements. The unimodal temporal shapes of the DV_m·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·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. The motor program of the VITEWRITE model thus does not require storage of within-stroke time lags, uses few memory resources to store the planning vectors, employs activity-based DV_m·GO decisions to automatically read-out the planning vectors, achieves speed and size rescaling in response to scalar GRO (size) and GO (speed) acts-of-will, and provides effortless concatenation of letter shapes into words.

The VITEWRITE model also retains desirable properties of the VITE model that were disclosed in previous studies of VITE-controlled reaching. Indeed, the plausibility of a role for the VITE model in the control of handwriting was soon noticed after its announcement in Bullock and Grossberg (1988), since VITE, by itself, generates as emergent properties several key properties of handwriting data, including the isochrony principle (Schomaker, Thomassen, and Teulings, 1989; Viviani and Terzuolo, 1983), or the tendency for strokes of different size to be completed with approximately equal duration; 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).

While many models of handwriting movement generation in the literature are aimed at reproducing the script of individual humans as exactly as possible (e.g. Plamondon, 1992; Schomaker, Thomassen, and Teulings, 1989), this paper is concerned with the psychophysical properties and neural control of handwriting as a general skill, ranging from the choice of the most appropriate coordinate system, the effects and possible benefits of motor redun-

dancy, the design of the trajectory generator, and the organization of the planning strategies whereby elementary strokes are generated and superimposed to produce the smoothly curved trajectories of handwriting. The model is defined and analysed in Sections 3–7 after various issues in the handwriting literature are described in Section 2.

2. Issues in handwriting

At the lowest level, any motor activity is expressed as the contractile state of agonist-antagonist muscle pairs over time, which are changed by neural control signals sent to these muscles. The simplest motor command, therefore, changes the angle of a joint from one value to another, and for a rotary joint the Cartesian space motion of the distal end of the segment is curved. Likewise, the Cartesian end-effector trajectory formed by straight-line trajectories planned in a multi-joint space is typically curved. For many tasks, it may be that a neural controller specifies desired trajectories in a 3-D spatial coordinate system, for example body-centered polar or Cartesian, and subsequently maps the resulting trajectory into joint angle changes (Greve *et al.*, 1992; Bullock, Grossberg, and Guenther, 1993). The resulting trajectory in this case would then be a straight line in 3-D space, but curved in joint space.

Skilled writers are able to fluently produce both straight and curved trajectories with their pen. In Cartesian space, straight strokes can be produced simply by combining two orthogonal components with a constant ratio between their velocities. But curved strokes require component velocity profiles whose onsets and offsets are shifted in time with respect to each other. Thus curved strokes are more complex. The opposite complexity ordering is true for a system that plans strokes in joint space. Which movements are simpler, and which are more complex, depends on the coordinate system chosen.

Psychophysical evidence supports the inference that arm movement planning often occurs in a spatial coordinate system. A comparison of end-effector and joint angle velocity profiles for planned arm movements has shown that the former are more invariant than the latter (Morasso, 1981; Abend, Bizzi, and Morasso, 1982). Also, it was found that the spatial characteristics of script are quite similar even across different effector systems, e.g. across handwriting and armwriting with hand-joints fixed. The spatial trajectory was also found to be more invariant than force-time patterns (Teulings, Thomassen, and van Galen, 1986).

These observations do not exclude other possibilities. In particular, studies also show that many free movements exhibit both curved end-effector trajectories and a tendency to avoid reversals of the direction of joint rotations during movement (e.g. Hollerbach *et al.*, 1986). This suggests that the system may be able to operate in various modes, and that component-wise point-to-point joint space planning, which avoids joint reversals, may be used whenever the task and limb geometry allow such planning. Whereas arm-writing may require spatial trajectory planning, hand-writing—that is writing at a scale appropriate to the hand's degrees of freedom—may only require joint space planning. We show below that given suitable degrees of freedom defined by hand muscle synergies, the resultant "elementary" movements of the hand approximate straight lines. However, because of the special nature of these joint coordinates, most of our results regarding intrinsic timing and stroke planning are directly transferable to larger-scale writing, even if the latter is planned in spatial coordinates before translation into muscle or joint coordinates. A system capable of self-organizing such spatial coordinates and a spatial-to-motor mapping has been described elsewhere (Greve, Grossberg, Guenther, and Bullock, 1992; Bullock, Grossberg, and Guenther, 1993).

Two prior models are especially relevant to our construction. Schomaker, Thomassen, and Teulings (1989) argue for a spatial Cartesian conordinate system on the grounds of spatial invariance, and consequently face the problem of how to produce curved trajectories. Their solution is based on precise control of the time lag between the respective onsets of horizontal and vertical displacements in time. The parameters they use to characterize a stroke, or movement between two zero-crossings in the velocity domain, are horizontal and

vertical displacement and stroke duration, which modulate a sinusoidal velocity function. An additional shape factor determines degree and direction of curvature by setting the relative phase of velocity zero-crossings in the horizontal and vertical components. An example of shape control by a velocity component phase shift is shown in Figure 2.

Figure 2

An alternative to the Schomaker, Thomassen, and Teulings model has been introduced by Plamondon (1989). It generates trajectories by superposing a curvilinear and an angular velocity command, rather than by combining orthogonal component velocities as in most other treatments. The velocity profile in Plamondon's model are not sinusoidal (see also Nagasaki, 1989), and are similar to VITE-generated profiles (see Section 4 below). Plamondon proposes that such profiles may arise as the output of a filter cascade perturbed by a square-wave input pulse. The VITE theory proposes a fundamentally different mechanism to explain the origin of the non-sinusoidal velocity profiles observed in voluntary movements. Otherwise, Plamondon's model is similar to the Schomaker *et al.* model in that it parameterizes the duration, amplitude, and relative phase of the two velocity components. Both models estimate their parameters from measurements of actual script.

In a model that assumes planning in 2-D, the trajectory is generated by a two degrees of freedom (DOF) system. The number of DOFs involved in handwriting, however, is much larger, involving every joint from the shoulder to the fingers. Even if we restrict our considerations to the hand, we find that the wrist has three DOFs, and each finger exhibits a total of four DOFs. The most important components are finger extension/retraction, horizontal wrist rotation and vertical wrist rotation (supination/pronation), a three DOF system.

Figure 3

We suggest that this extra, third degree of freedom 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. Thus, a redundant degree of freedom can be used to reduce the complexity of trajectory generation. In turn, a trajectory generator constrained to generate unimodal velocity profiles could help to reduce the number of solutions of the inverse kinematics problem that the nervous system faces in planning the execution of complex movement.

Bullock, Grossberg, and Guenther (1993) have addressed the issue of redundant motor control with a model of goal-oriented reaching that is called the DIRECT model. This model suggests a solution to the motor equivalence problem wherein visual information about target and end effector positions in 3-D space are transformed into spatial direction vectors. Spatial direction is adaptively mapped into joint rotations which move the effector in the desired spatial direction, given the current effector configuration. In a redundant system, the mapping from spatial direction to motor commands is one-to-many; that is, there might be many ways to move an effector like the hand towards a spatially defined target. The constraint outlined above might help to reduce the number of possible ways. In Section 5, we will demonstrate that a rich set of realistic letter shapes can be produced even if the phase relations between component movements are constrained to be either 0 or 90 degrees. Such a constraint in the timing domain might further simplify the inverse kinematics problem.

The three main aspects of the VITEWRITE model are defined below: a geometrical model of the hand, a VITE neural trajectory generator, and a vector motor plan. Our main suggestion about the read-out of planning vectors is that, by using a redundant hand,

precise extrinsic control of onset and offset timing is unnecessary, and can be replaced by an activity-released command scheme, such that the onset times of later movement components are automatically determined by events in the trajectory generator itself.

3. Geometry of the hand

As noted above, the number of motor segments used in handwriting is large, involving every joint from the shoulder to the fingers. Here, we restrict our analysis to the hand only, which still has a total of seven degrees of freedom from the wrist to the fingertip. Most of these joints operate in concert during handwriting to control three main sets of synergists. Accordingly, our hand model has three DOFs: vertical wrist rotation (supination/pronation, called X) finger extension/retraction (called Y), and horizontal wrist rotation (called R), as in Figure 4.

Figure 4

A further simplification is made by considering the relative scales of hand 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

$$E_x = (l + y)\sin(r) + x\cos(r) \tag{1}$$

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

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 (VITE) model of Bullock and Grossberg (1988, 1991) is a neural model of how the outflow commands that control multi-joint motor trajectories are formed. In particular, the model clarifies the intimate linkage that exists between movement properties of synergy, synchrony, and speed. It shows how a group of effectors may be dynamically bound into a motor synergy, and once bound, how the synergy can perform synchronous movements at variable speeds. The VITE model outputs are the input to a neural model called FLETE. The FLETE model (whose name stands for Factorization of LEngth and TEnsion) clarifies how outflow movement commands from a VITE circuit may be accurately performed at variable stiffness levels without loss of positional accuracy (Bullock and Grossberg, 1991; Bullock, Contreras-Vidal, and Grossberg, 1992). Whereas 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.

The VITE model has been used to explain many kinematic properties of synchronous multi-joint movement, such as bell-shaped velocity profiles, peak acceleration as a function of movement amplitude, Woodworth's law, Fitt's law, velocity amplification during target switching, normalized velocity profile invariance across different distances, and velocity profile asymmetry as a function of duration. These computational properties, along with the

neural and behavioral evidence supporting the assumptions of the VITE model, make it a reasonable starting point for an analysis of trajectory formation during handwriting.

Figure 5

The VITE circuit consists of four neural stages that are depicted in Figure 5: The first stage, the Target Position Vector (TPV) stage receives desired positions coded in terms of muscle lengths from higher stages. The Present Position Vector (PPV) stage, which integrates its inputs over time, generates 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 PPC can occur. Once the GO signal becomes positive, the PPV can start integrating signals at the rate GO·DV. This multiplicative interaction maintains the direction coded by DV while modulating 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·GO, the rate of change of the outflow PPV signal, namely $\frac{d}{dt}$PPV, tracks DV·GO. Thus DV·GO provides an internal measure of the commanded movement velocity. 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.

Since the 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. Even when different components are switched on at different times, their movements tend to terminate at the same time. This is called the temporal equifinality property for staggered onsets (Bullock and Grossberg, 1988). This is an important property for stably controlling a temporal series of movements during which one synergy precedes the next. For example, consider a task where an arm needs to reach in one direction before shifting to reach in another direction. The synchronous, temporally equifinal, completion of the first reach enables the second reach to be launched without causing an uncontrollable change of direction. Such a destabilizing change could occur if some, but not all, components of the first synergy were still contracting while the second synergy was activated.

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

Not all movements are controlled, however, by a serial read-out of one synergy at a time. As noted above, the production of curved trajectories during handwriting requires that distinct movement components have distinct but overlapping velocity profiles. These phase lags suggest that the synergies we have identified in the last section (finger extension, horizontal wrist rotation, and vertical wrist rotation) need to violate the equifinality property. If all synergies of the hand were grouped into one TPV with a single GO signal, the VITE circuit would work towards making all component movements terminate at the same time, despite differentially timed onsets. Therefore, we assume that the three synergies of our hand model are controlled by their own VITE circuits, with separately initiated GO signals. A mechanism is also needed to reset these GO signals before the onset of a new movement by each synergy.

Such a decomposition of hand movements into independently controllable, but temporally overlapping, synergies is analogous to the decomposition of speech articulators into coordinative structures (Fowler, 1980). In the case of hand and arm movement, various data support the idea that multiple finger, hand, and arm movement synergies can be separately

controlled during complex movements. For example, Lacquaniti *et al.* (1987) found that while arm movements are characterized by constant phase relations between shoulder and elbow motion, hand movements exhibit more variable phases (see also Jeannerod, 1988). Moreover, the proposal that multiple GO signal channels exist is consistent with data on the proposed anatomical site of GO signal generation, namely the basal ganglia (see Bullock and Grossberg, 1991). 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 the delimitation or gating of which degrees of freedom should be included in a wide variety of synergies.

6. Model Equations

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

Target Position Vector

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

The TPV receives planning inputs from higher processing stages. The planning vectors $D_i(t_{ij})$ are the components of the motor programs. They embody directional commands whose size, scaled by S_i, determines the distance travelled by a synergy. At launch times $t_{ij}, j = 1, \ldots, n$, the jth planning vector $D_i(t_{ij})$, scaled by S_i, is added to the ith channel of the TPV.

Difference Vector

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

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. Here, agonist and antagonist activations are lumped into one variable by allowing negative values.

GO Signal

$$G_i(t) = G_0(t - t_{ij})^n \qquad t_{ij} \le t < t_{ij}^*,\ j = 1, \ldots, n, \tag{5}$$

where G_0 is a constant and t_{ij} is the jth time at which component 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. In all simulations, $n = 1.4$, which produces nearly symmetrical bell-shaped velocity profiles. Equation (5) for the growth of the GO signal is used 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 a cascade of two or more neurons activated 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) model, a much longer cascade is used to generate the entire velocity profile.

Present Position Vector

$$\frac{d}{dt} P_i = V_i G_i \tag{6}$$

The PPV integrates its input signals at the rate $V_i G_i$.

7. Control of GO Signal Phase Relations

To produce the smooth, curved trajectories of script, synergy DV_p directions and GO signal onsets need to be appropriately timed. The directions and onset lags of different synergies determine script curvature. Furthermore, in order to generate a letter shape, elementary strokes need to be joined together smoothly. While in this paper we do not discuss the self-organizing process that discovers, learns, and stores representations of movement commands, we do suggest what these commands may be and how their onset times may be controlled to generate cursive letter shape trajectories as emergent properties of a multi-VITE circuit.

The onset timing for the next stroke in a motor program could be determined in two ways: Either the time of launching the next stroke is a parameter of the motor program (cf. Schomaker *et al.*, 1989), or 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, unless the timing of successive onsets could automatically compensate for such motor variability, the shape of a trajectory could be very different at different execution speeds.

If triggering a successive stroke is contingent on a characteristic event in the velocity trace of the controller, then this problem can be avoided, since onset lags then shift automatically with speed of execution. An outflow representation of each synergy's velocity is available in the VITE model in the form of the activity functions at the DV_m·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. Our simulations have shown that two events are suitable to launch a stroke: Times when all velocities are close to zero, and times at the peak of one or more velocity traces. These two types of events are called a *postural launch* (detected by a match between TPV and PPV) and a *dynamic launch* (detected by a peak in one or more velocity profiles). Figure 6 schematizes a dynamic launch: A peak in one of the velocity profiles (point B in Figure 6) can launch a new movement 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 (Figure 6, points A and C). The other type of event, a point of zero velocity, can also trigger new movement (Figure 6, point D). Thus the launch times t_{ij} in (5) occur either when synergy i is at rest or when the outflow speed command DV·GO of another synergy reaches a maximal size. Reset occurs at times t_{ij}^* when the PPV of the synergy equals its TPV. The model 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 edgier shapes if the launch occurs after the peak.

Figure 6

If a new target is launched only at the occurrence of these two types of events, then the phase relations between any two component velocity traces are limited to either 0 or 90 degrees. With this scheme, the variables characterizing the motor program are merely planning vectors, or DV_p's, that can be stored in a sequential working memory (e.g., Bradski, Carpenter, and Grossberg, 1992; Grossberg, 1982; Mannes, 1992), whose entire vector plan can be learned and read-out from a single unitized planning chunk, or set of chunks (Carpenter and Grossberg, 1991; Cohen and Grossberg, 1986, 1987; Grossberg, 1982, Chapter 12). Each peak and zero in the outflow velocity trace DV_m·GO can activate read-out of the next DV_p from the working memory, as in Figure 1. 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. In the present simulations, the 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 commands needed to generate cursive script. We now summarize simulation experiments that we performed with the VITEWRITE model.

8. Simulations of Cursive Script

An example of a script letter "b" is shown in Figure 7. 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 7.

Figure 7

Table 1

To start with, an X motion to the right is launched (stroke segment 1 in Figure 7 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.

Figure 8

A more difficult example, the letter "a", involves a richer set of maneuvers and the third degree of freedom, as shown in Figure 8. The first component to be launched in this case is R, which rotates the hand a little to the left (stroke 1), followed by an upward movement (stroke 2). Instead of launching R again, a rightward X movement (of similar effect) is launched (stroke 3). At the peak of this X movement, all targets are zero, such that the total movement comes to a full stop at the top of the letter. Stroke 4 undoes the effect of R to some extent by rotating to the left, followed by a downward movement (stroke 5). At the peak of the downward movement, a rightward movement begins (stroke 6), followed by an upward movement (stroke 7). Again, no movement is initiated at the peak of this last movement, so everything comes to a halt, which gives the system a chance to reverse direction.

Figure 9

The horizontal wrist rotation component, R, produces end effector movements very similar to X movements, at least at small scales. This redundancy makes possible some strokes that would otherwise require more complex control strategies. Examples are the strokes shown in Figure 3 and in panel 2 of Figure 8. Furthermore, redundancy allows for letters to be produced in different ways. For example, consider the beginning right-upward stroke of most letters: This type of stroke can be achieved by any of the following component movement sequences: X to the right, Y up, R right; R right, Y up, X right; or R right, X and Y in phase obliquely up, followed by R right. In the present simulations, control strategies were chosen such that the redundant DOFs X and R were not activated concurrently, in order to produce similar strokes and letters in a consistent way. Some of these choices are discussed in the next section.

9. Elements of Style in Writing Connected Words

Redundancy allows similar shapes to be realized by different motor programs; for example, compare the "b" shape in Figure 7 with the one in Figure 9. Homogeneous appearance of script and the need to connect letters into words suggests, however, that a consistent style

should be used. Especially with regard to connecting letters, consistent beginnings and endings of letter shapes are highly desirable. Also, in order to change style parameters—such as slant and width versus height— letters should be stroked in a consistent fashion. For example, if a stroke leading up and to the right were realized by X right, Y up, X right, the slant would be fixed. On the other hand, if the same stroke is realized by R right, X right/Y up in phase, R right, scaling X and Y targets by different amounts can produce the same letter with different slant. The letters of the alphabet, shown in Figure 9, were constructed with these constraints in mind. Thus, all letters shown in Figure 9 are using X and Y in phase for vertical/oblique strokes, and R for horizontal movements. Each letter consistently begins with a straight oblique X/Y stroke, and ends with a rightward R stroke. As a result of using a consistent stylistic strategy for each letter, these shapes can be effortlessly connected into word shapes, of which an example is depicted in Figure 10.

Figure 10

A further advantage of using a consistent set of strokes is the ability to scale the size and slant of letter shapes by simply scaling the elements of the motor program differentially. Some example of such variations are shown in Figure 11.

Figure 11

10. Size, Speed, Slant, and Curvature Invariance

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 11 displays variations of a trajectory achieved by differentially scaling the elements of the motor program. Here, each planning vector component D_i of TPV_i is multiplied by a different GRO scalar S_i. While the results can look quite different, the component velocity profiles are the same for all examples in Figure 11, except for their relative magnitude. Uniform size scaling of the motor program—that is multiplying each component D_i of TPV_i by the same GRO scalar S—modifies the size of the performed letters, but leaves the trajectory shape invariant. Figure 12a-c shows the letter "b" performed with uniformly scaled GRO movement commands. 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, as in Figure 12d-f, extreme finger angles at the border of the workspace produce distortions.

Figure 12

Shape invariance under speed rescaling is demonstrated in Figure 13, which shows the same letter performed at "normal" and at double speed, achieved by scaling 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. The more realistic assumption that a small but finite reaction time is needed to launch would not substantially influence the invariance result until speeds were attained at which the duration of each synergy was no longer very 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 13

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 of the PPV in equation (6). Once DV control is available, scalar GRO and GO signals can transform a stereotyped series of DV's into motor performances whose sizes and speeds can be adjusted to match variable environmental conditions (Grossberg *et al.*, 1992). Models which utilize DV's for their spatial and trajectory control have generically been called Vector Associative Maps, or VAMs (Gaudiano and Grossberg, 1991).

Figure 14

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 says that angular velocity $A(t)$ relates 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 14a plots model curvature and model tangential velocity for the letter "b"; Figure 14b 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. In fact, an adequate model of human performance must not agree perfectly with the two-thirds power function, because the latter is an imperfect descriptor, as observed by Wann, Nimmo-Smith, and Wing (1988). The latter authors also note 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 exhibited by human actors (Bullock and Grossberg, 1988, 1991; Nagasaki, 1989).

11. Concluding Remarks

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. The processing stages of the VITE model have previously been shown capable of controlling properties of movement synergy, synchrony, and speed during reaching behaviors. Here the same processing stages enable a simple type of motor program to control spatially and temporally rescalable handwriting.

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 enable volitional GRO signals to rescale the size of handwriting without changing its form. The VITEWRITE model thus provides a flexible new neural model for handwriting control while offering additional evidence that the processing stages of VITE, and more generally of Vector Associative Map, controllers may be put to multiple uses by the brain towards generating complex motor behaviors.

REFERENCES

Abend, W., Bizzi, E., and Morasso, P. (1982). Human arm trajectory formation. *Brain*, **105**, 331–348.

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.

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. 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. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, in press.

Carpenter, G.A. and Grossberg, S. (Eds.) (1991). **Pattern recognition by self-organizing neural networks**. Cambridge, MA: MIT Press.

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.

Dooijes, E.H. (1983). Analysis of handwriting movements. *Acta Psychologica*, **54**, 99–114.

Edelman, S. and Flash, T. (1987). A model of handwriting. *Biological Cybernetics*, **57**, 25–36.

Fetters, L. and Todd, J. (1987). Quantitative assessment of infant reaching movements. *Journal of Motor Behavior*, **19**, 147–166.

Fowler, C. (1980). Coarticulation and theories of extrinsic timing. *Journal of Phonetics*, **8**, 113–133.

Gaudiano, 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.

Greve, D., Grossberg, S., Guenther, F., and Bullock, D. (1992). Neural representations for sensory-motor control, I: Head-centered 3-D target positions from opponent eye commands. *Acta Psychologica*, in press.

Grossberg, S. (1982). **Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control**. Boston: Reidel Press.

Grossberg, S., Guenther, F., Bullock, D., and Greve, D. (1992). Neural representations for sensory-motor control, II: Learning a head-centered visuomotor representation of 3-D target position. *Neural Networks*, **6**, 43–67.

Hollerbach, J.M., Moore, S.P., and Atkeson, C.G. (1986). Workspace effect in arm movement kinematics derived by joint interpolation. In G. Gatchev, B. Dimitrov, and P. Gatev (Eds.), **Motor control**. New York: Plenum Press.

Jeannerod, M. (1988) **The neural and behavioral organization of goal-directed movements.** Oxford: Clarendon Press.

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.

Mannes, C. (1992). A neural network model of spatio-temporal recognition, recall, and timing. *Proceedings of the International Joint Conference on Neural Networks, Baltimore, 1992*, **IV**, 109–114.

Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, **42**, 223–227.

Morasso, P. (1986). Trajectory formation. In P. Morasso and V. Tagliasco (Eds.), **Human movement understanding.** Amsterdam: North-Holland.

Morasso, P. and Mussa-Ivaldi, F.A. (1982). Trajectory formation and handwriting: A computational model. *Biological Cybernetics*, **45**, 131–142.

Morasso, P., Mussa-Ivaldi, F.A., and Ruggiero, C. (1983). How a discontinuous mechanism can produce continuous patterns in trajectory formation and handwriting. *Acta Psychologica*, **54**, 83–98.

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: University Press.

Plamondon, R. (1992). A model-based segmentation framework for computer processing of handwriting. *Proceedings of the 11th International Conference on Pattern Recognition*, 303–307.

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.

Soechting, J.F. and Terzuolo, C.A. (1987). Organization of arm movements is segmented. *Neuroscience*, **23**(1), 39–51.

Teulings, H.L., Thomassen, A., and van Galen, G.P. (1986). Invariants in handwriting: The information contained in a motor program. In Kao, G.P. van Galen, and Hoosain (Eds.), **Graphonomics: Contemporary research in handwriting**. Amsterdam: Elsevier, 305–315.

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.

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·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·GO is integrated by the PPV until it equals the TPV. The signal DV_m·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.

Figure 2. The effect of shape parameters of Schomaker *et al.* (1989): The curvature depends on which of the orthogonal components (horizontal and vertical) leads in time. Reproduced after Schomaker, Thomassen, and Teulings (1989), Figure 2.

Figure 3. A stroke that is greatly simplified by use of a third degree of freedom. Left: With two degrees of freedom, a stroke as 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.

Figure 4. 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.

Figure 5. The VITE circuit, the neural controller of each component agonist-antagonist pair of the hand.

Figure 6. Dynamic and postural launches: Peaks in the velocity profile can launch new movement by triggering read-in of new targets and reset of the GO-signal for other components with non-zero velocity (points A and B trigger a *dynamic* launch, in case C all targets are zero, such that no movement is launched). Zero velocity can also trigger new movement (point D). This is called a *postural* launch.

Figure 7. 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.

Figure 8. Another example showing how to generate the end-effector trajectory drawn in the left panel, using the third degree of freedom. V_x, V_y, V_r denote X, Y, and R 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.

Figure 9. Some more examples of letter shapes. To the right of each letter, the three velocity profiles (X, Y, and R from top to bottom) are given. All plots on the same scale: The end effector trajectory is plotted from 0 to 10mm horizontally, and from 0 to 20mm vertically. Velocity profile plots: time, on the horizontal axis runs from 0 to 15, V_x from -50 to 50, V_y from -100 to 100, and V_r -0.05 to 0.05. The smaller excursion of r is due to the fact that r is an angle, while x and y are distances. Simulation parameters: $\alpha = 10, l = 200$. Longest motor program: q and w with 13 motor commands. Medium (10-12 commands): a,d,g,k,m,o,p,y. Short(6-9): $b,c,f,h,j,l,n,r,s,t,u,v,z$. Shortest programs: e,i (not shown), and l with 4 commands, i.e. 12 numbers characterizing a letter. The letters were modelled after the Palmer method of handwriting.

Figure 10. An example of connecting letters by simply concatenating individual motor programs.

Figure 11. Effect of scaling component targets: (a) An unscaled version of a word composed of the letter programs in Figure 9. (b) The same word written with all X targets multiplied by $S_x = 2$, Y targets by $S_y = 0.6$. (c) Another version with $S_x = 0.7, S_y = 1, S_r = 0.4$, and (d) with $S_x = 2.5, S_y = 1, S_r = 0.3$.

Figure 12. 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). Panels (d)-(f) show the result of a simulation that used a different hand model to calculate end effector position. Instead of taking the x and y axes as an orthogonal system rotated by r in the plane, a 3-D model of the hand was used. The shoulder was fixed at (0,0,0), the pen tip was constrained to touch the drawing surface ($E_z = 0$), and E_x, E_y were calculated as $E_x = c\sin(r + \gamma), E_y = c\cos(r + \gamma)$ where $c = 2l\sin(y/2)$ and $\gamma = \pi - y\sin(x)$. Using this geometry, extreme ranges (panel f) produce distortion effects.

Figure 13. 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 ($G_o = 2$).

Figure 14. 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).

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 decreasing part of the velocity profile.

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

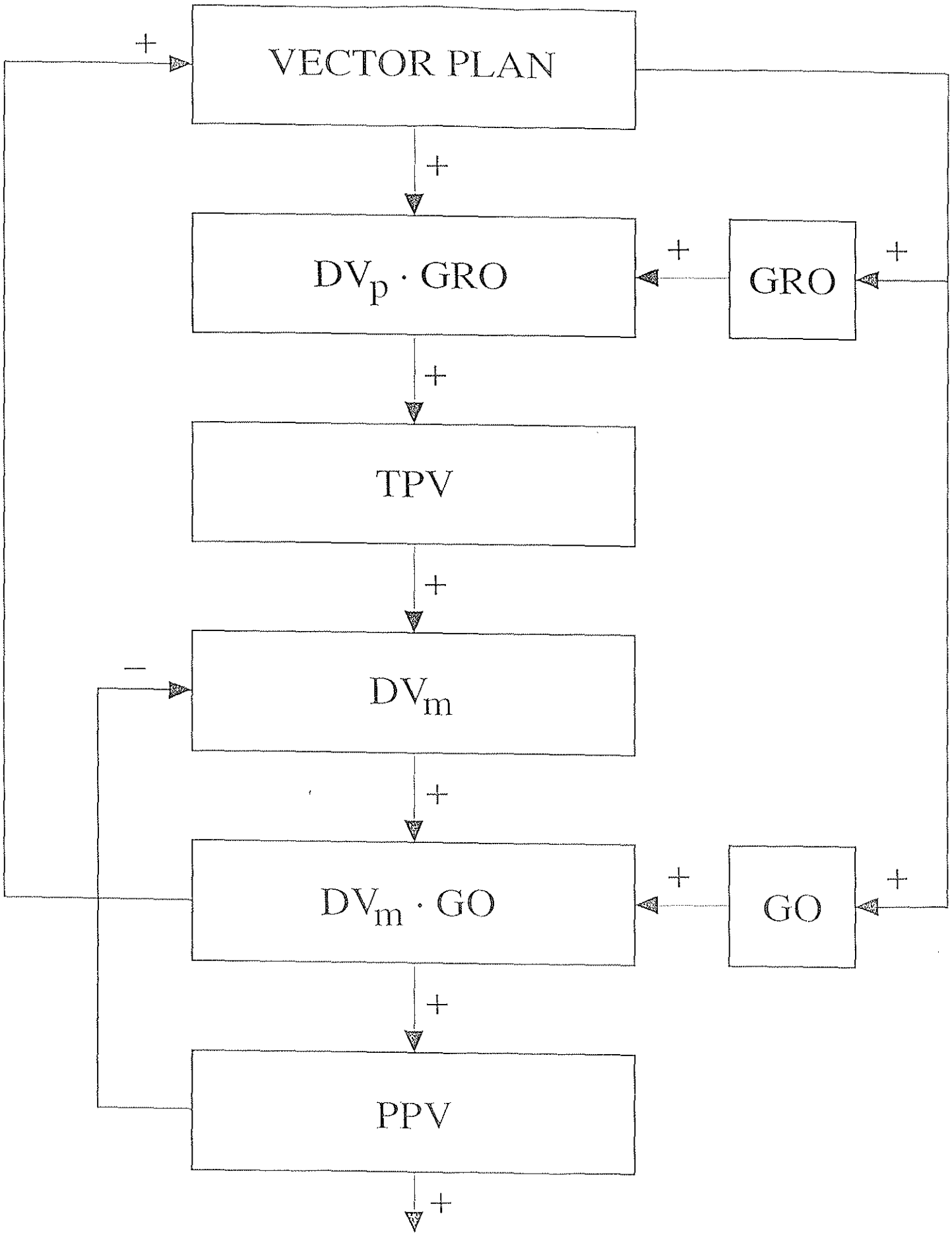


Figure 1

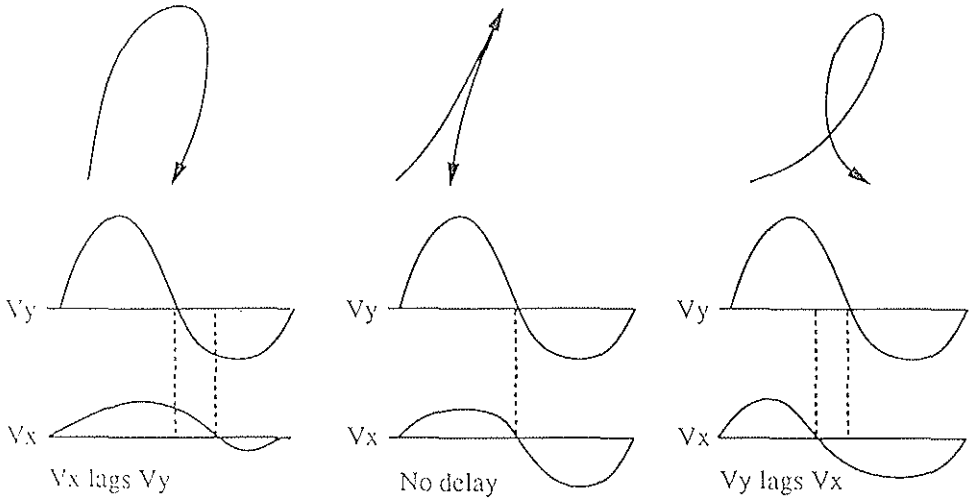


Figure 2

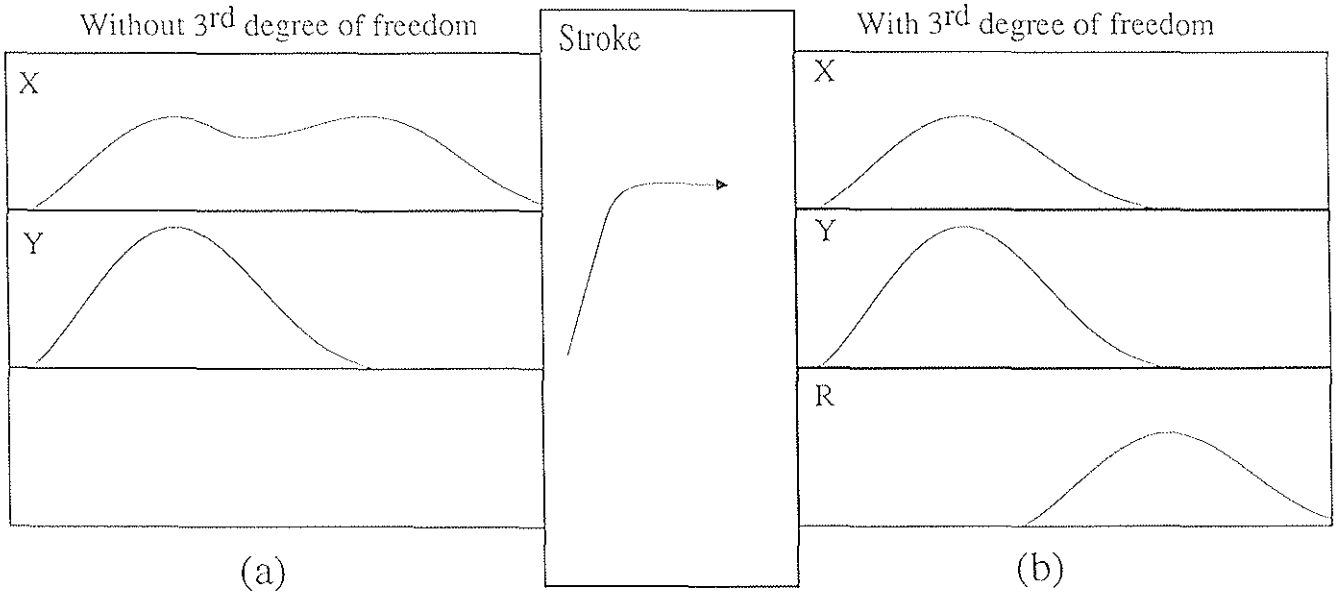

Without 3rd degree of freedom
Stroke
With 3rd degree of freedom
X
Y
X
Y
R
(a)
(b)

Figure 3

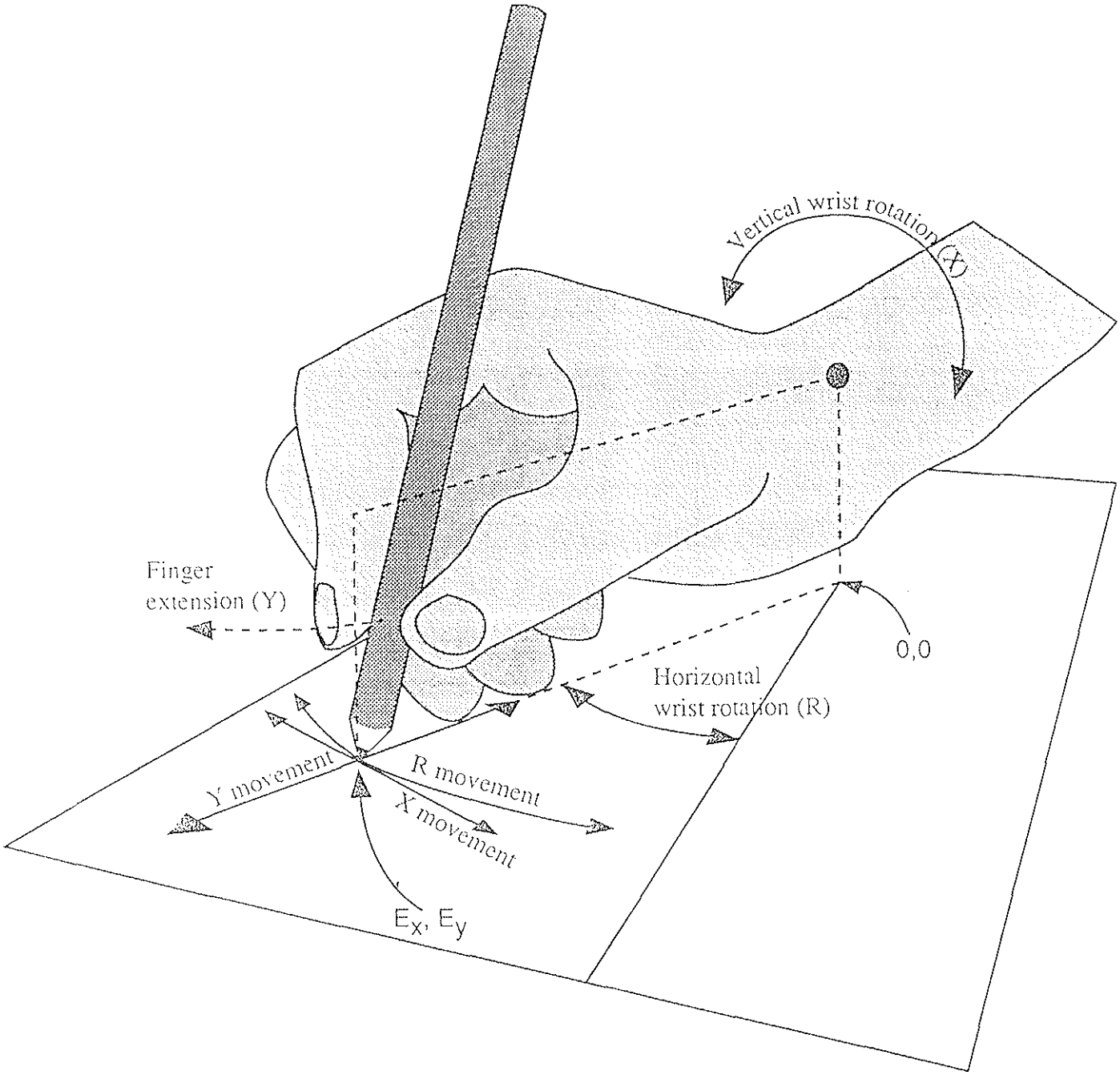

Vertical wrist rotation (X)
Finger
extension (Y)
0,0
Horizontal
wrist rotation (R)
Y movement
R movement
X movement
Ex, Ey

Figure 4

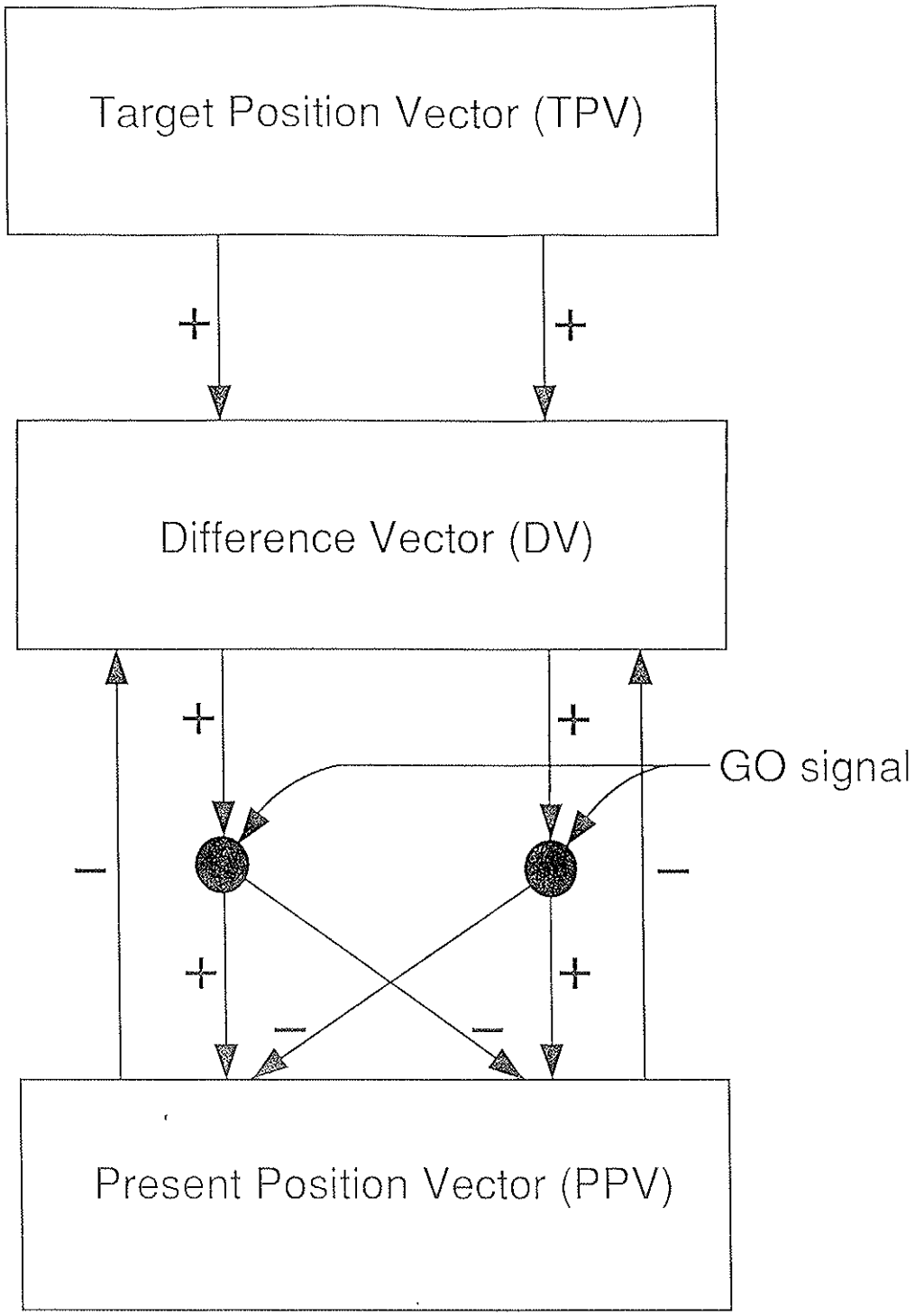

Target Position Vector (TPV)
+
+
Difference Vector (DV)
+
+
GO signal
−
−
+
+
−
−
Present Position Vector (PPV)

Figure 5

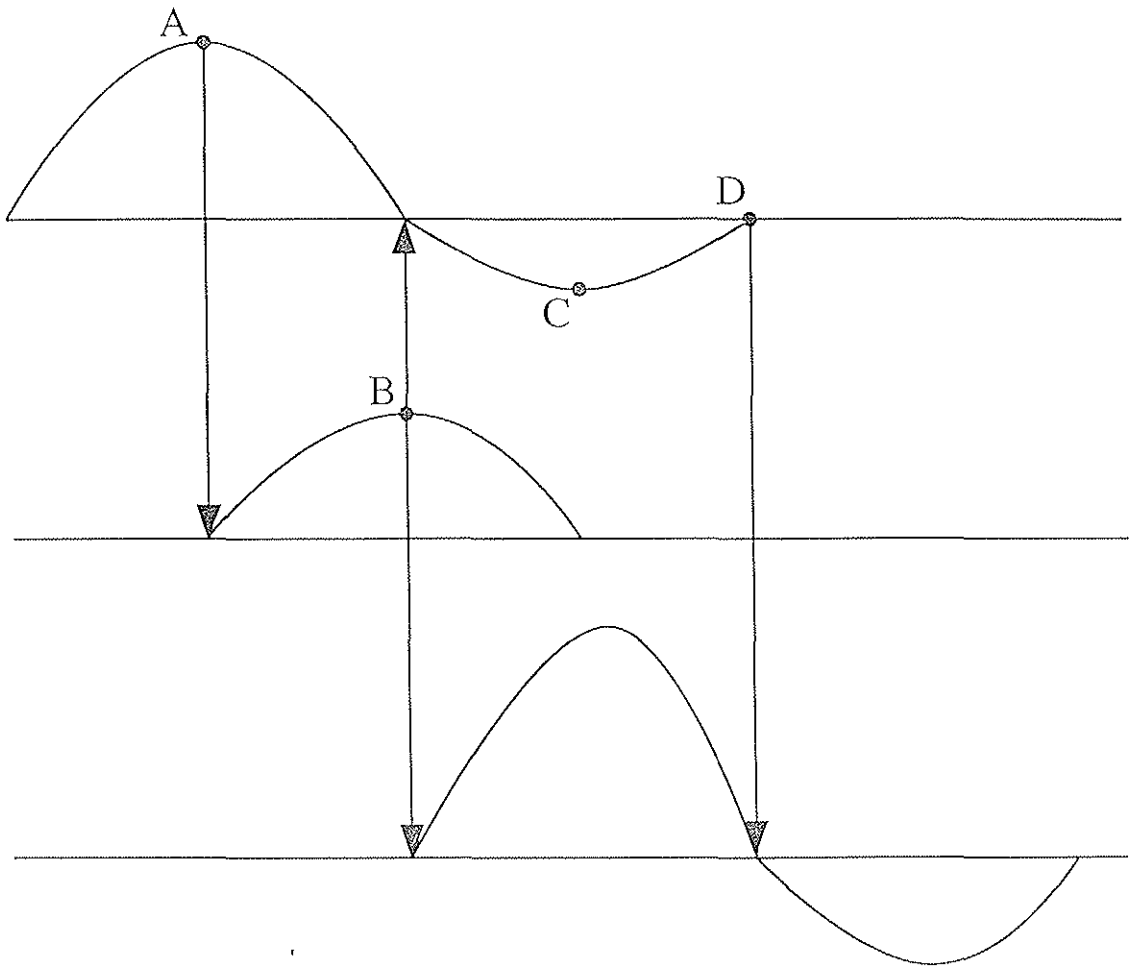


Figure 6

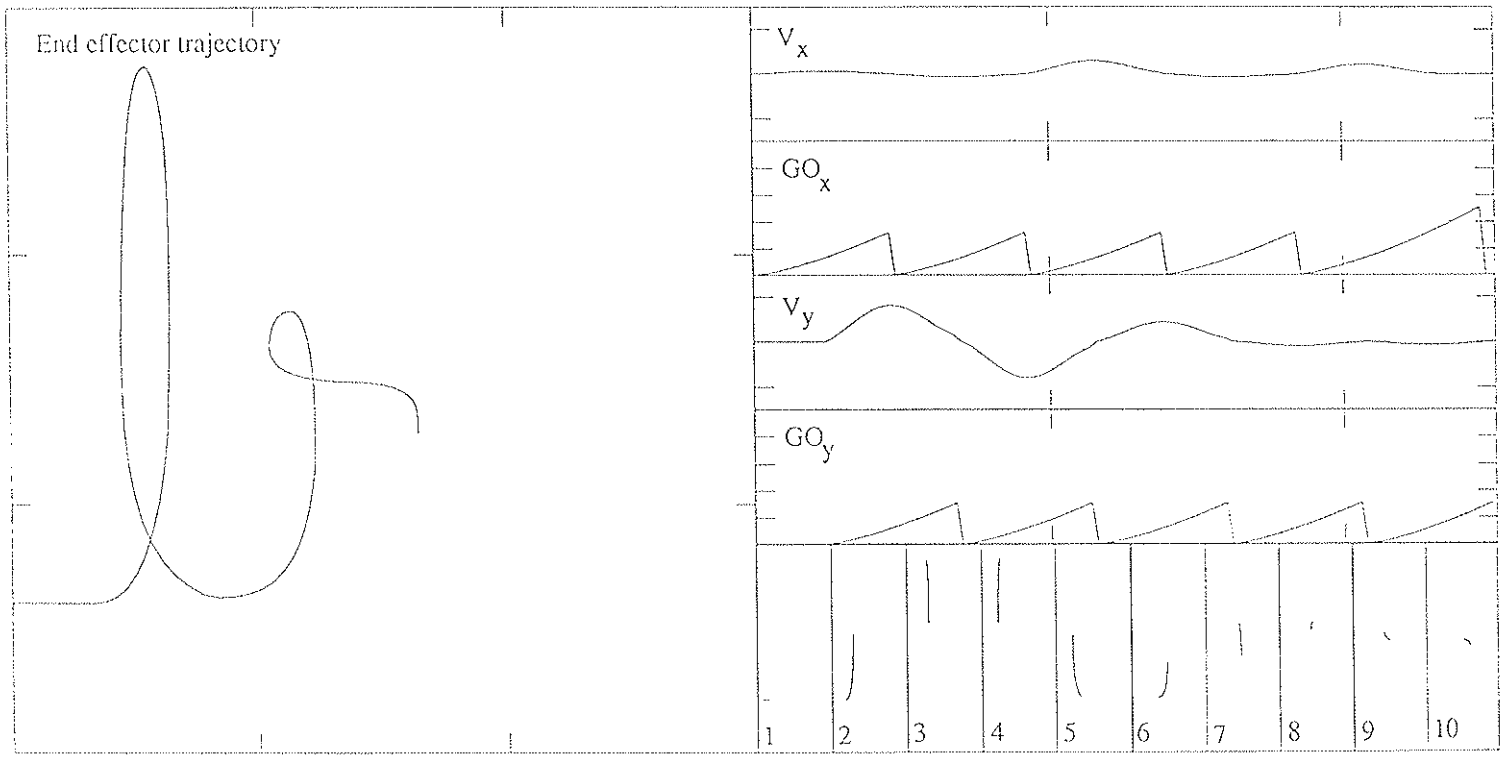


Figure 7

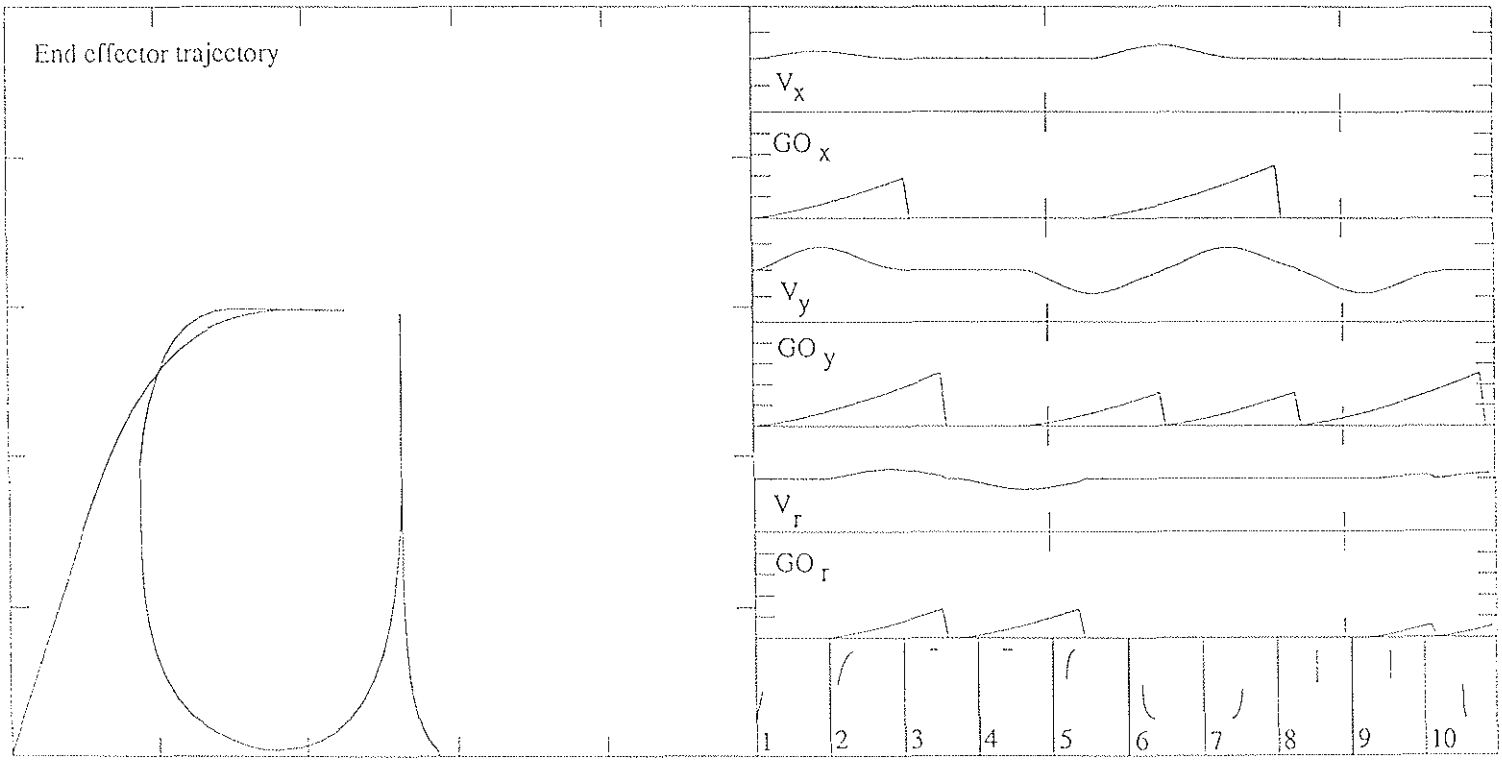


Figure 8

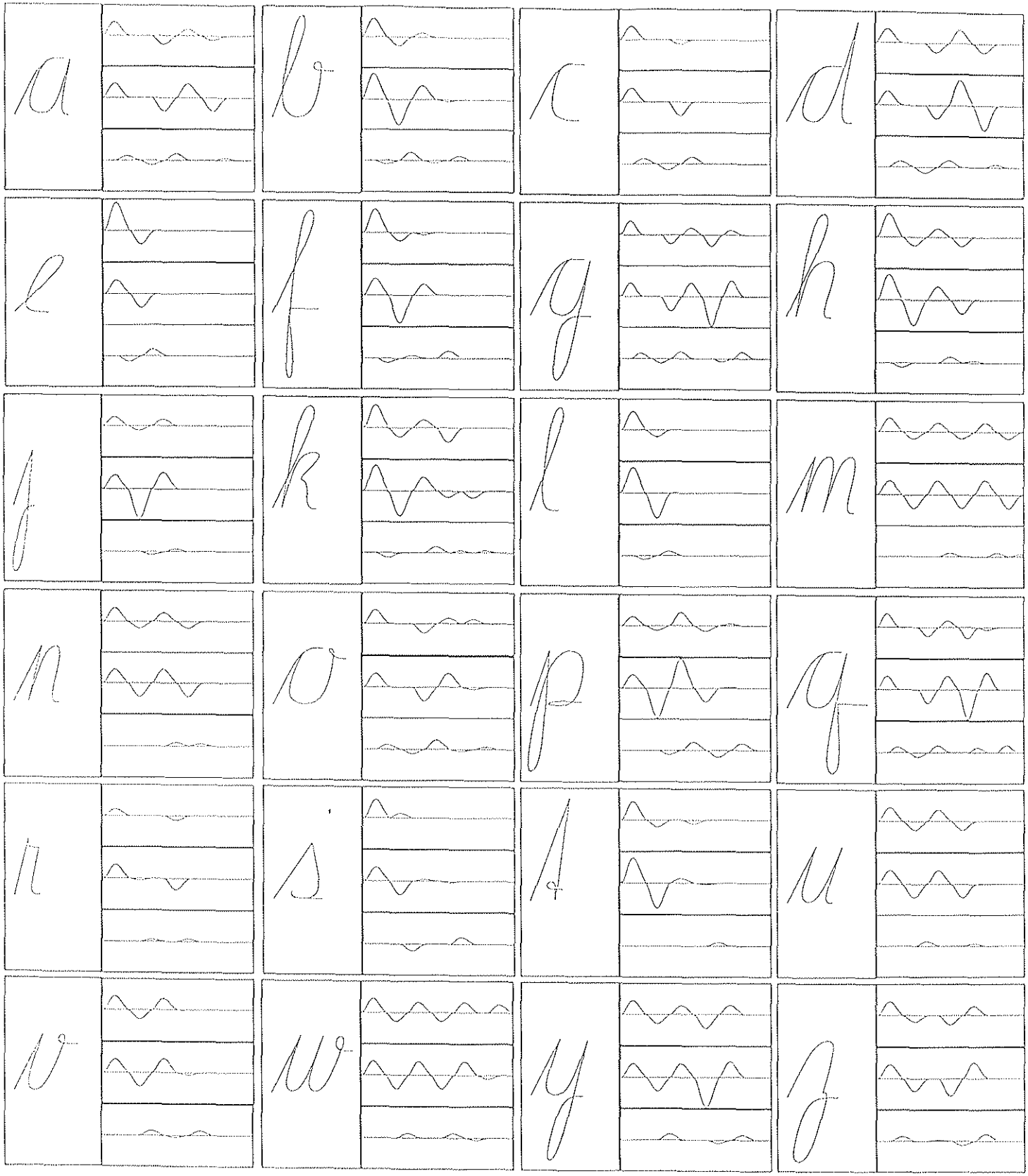


Figure 9

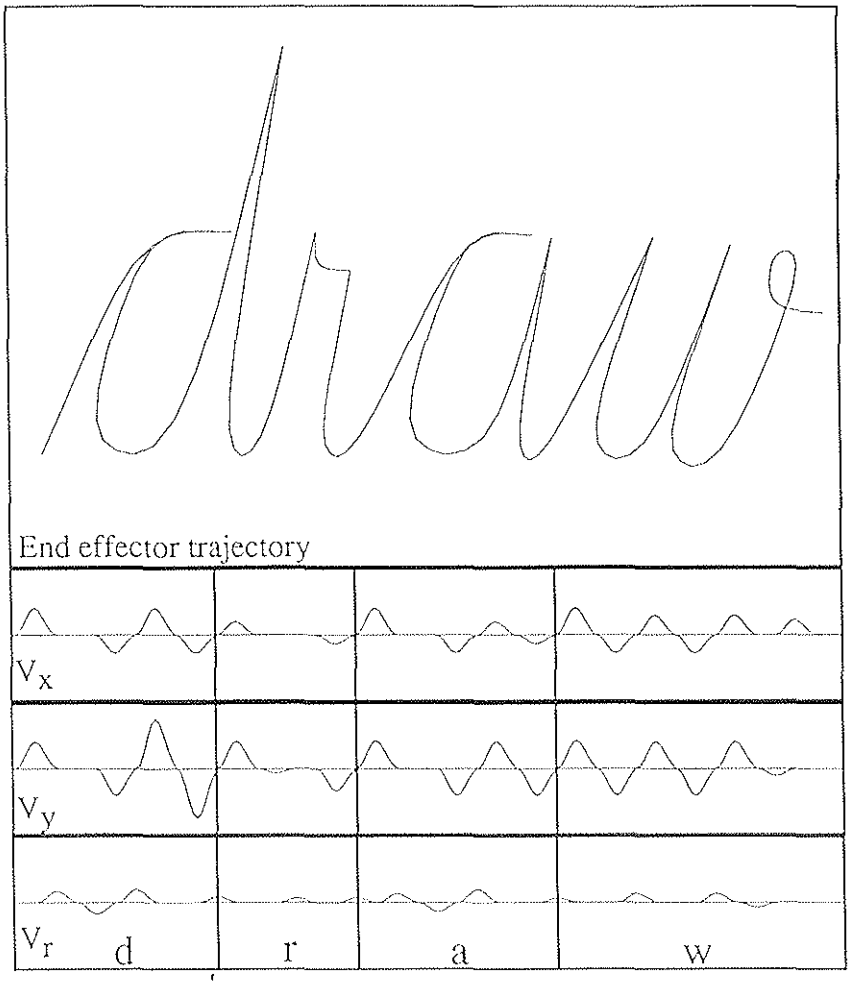


Figure 10

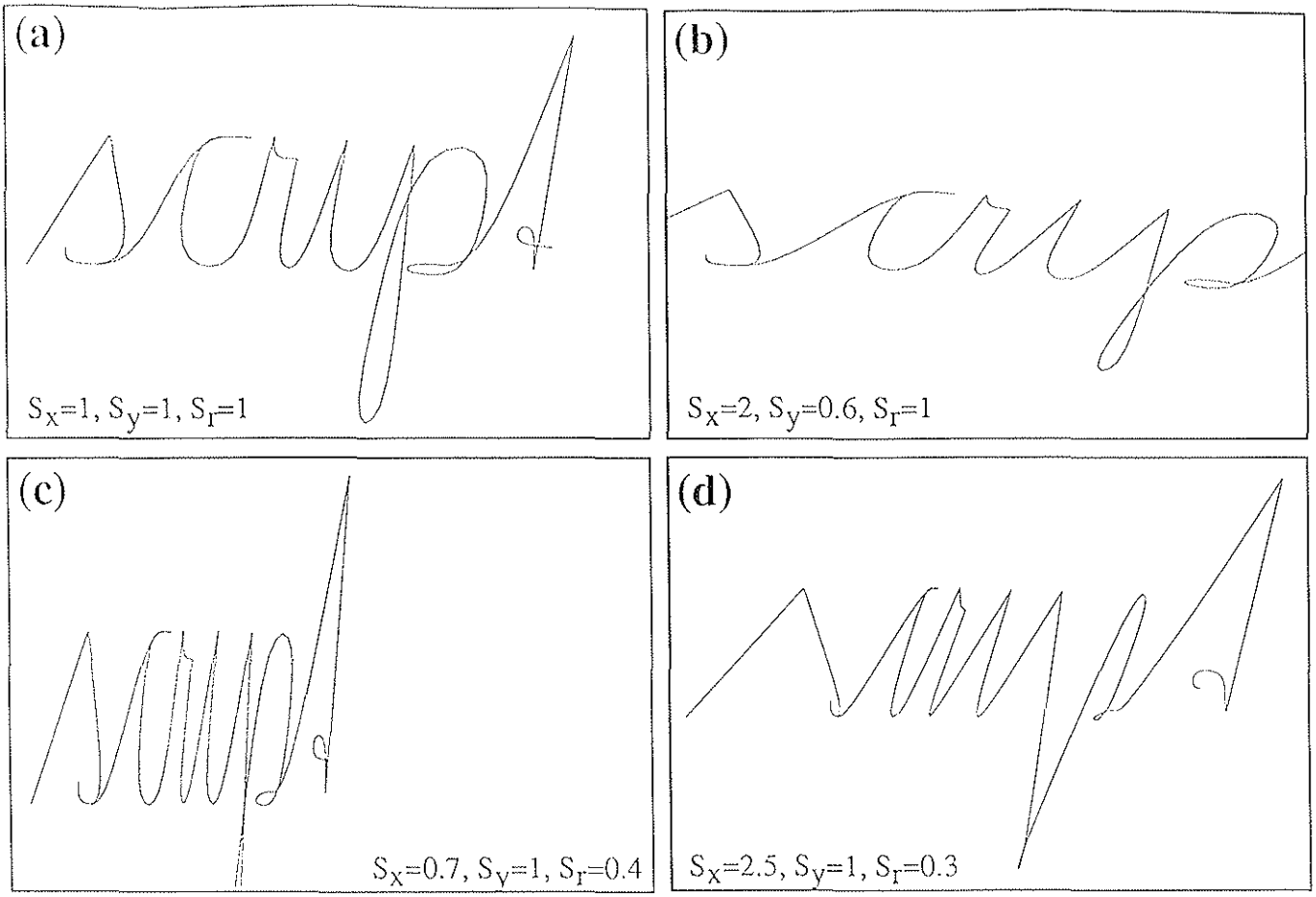
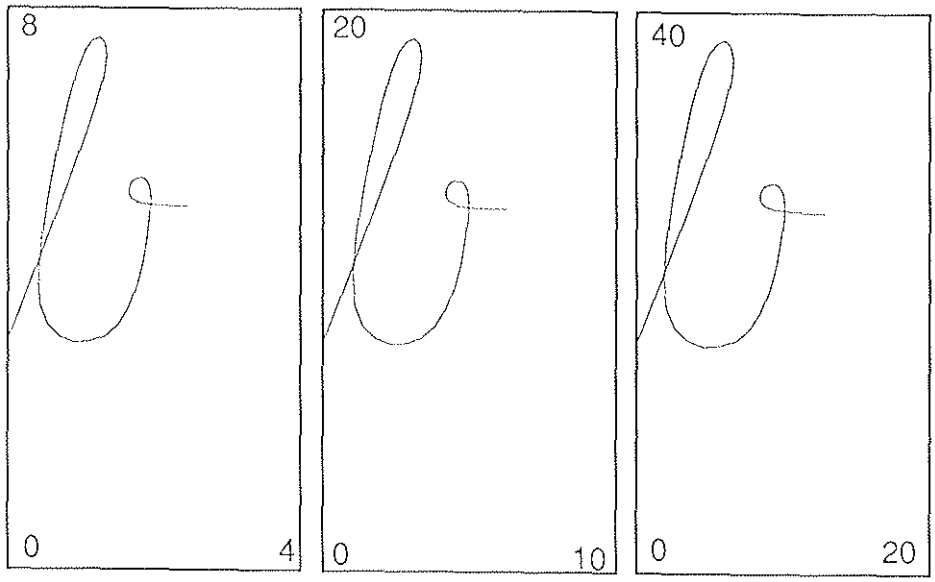
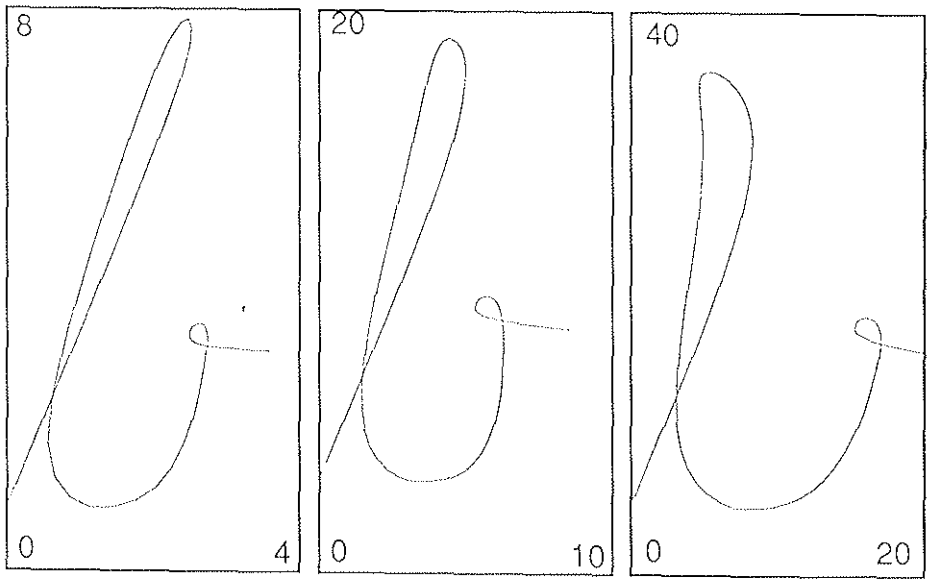


Figure 11



(a) (b) (c)



(d) (e) (f)

Figure 12

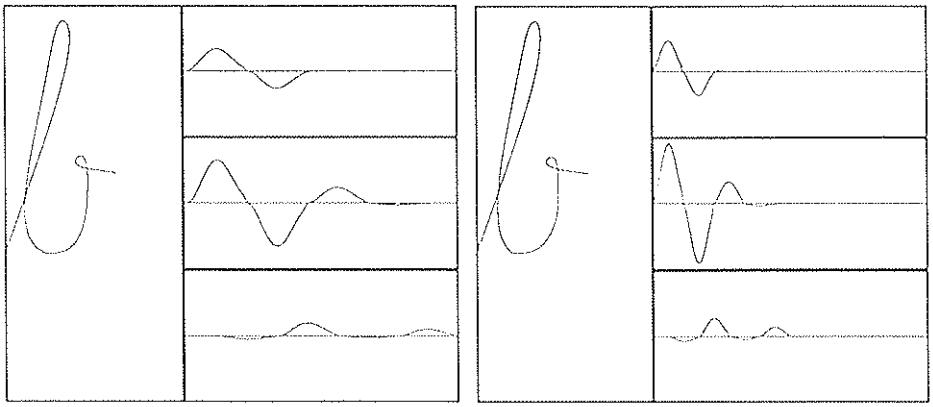


Figure 13

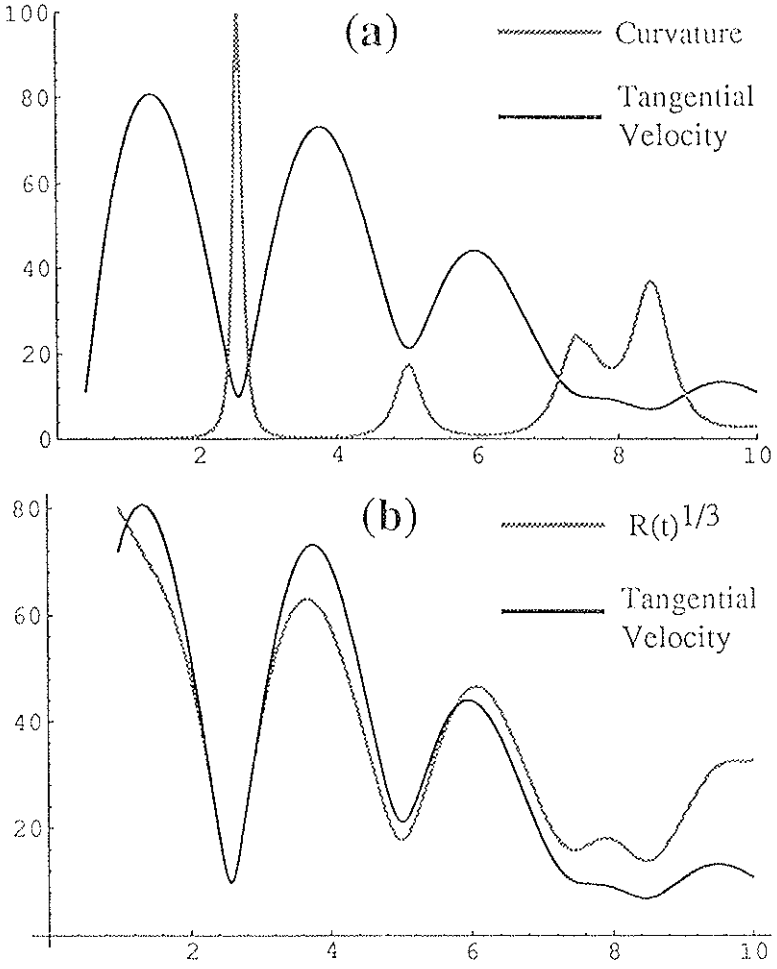

(a)
Curvature
Tangential Velocity
100
80
60
40
20
0
2
4
6
8
10
(b)
R(t)^1/3
Tangential Velocity
80
60
40
20
2
4
6
8
10

Figure 14