IJDAR (2007) 10:69–84
DOI 10.1007/s10032-007-0046-0

ORIGINAL PAPER

An oscillatory neuromotor model of handwriting generation

Garipelli Gangadhar · Denny Joseph · V. Srinivasa Chakravarthy

Received: 15 January 2007 / Accepted: 3 April 2007 / Published online: 22 June 2007

Abstract A neuromotor model of handwritten stroke generation, in which stroke velocities are expressed as a Fourier-style decomposition of oscillatory neural activities, is presented. The neural network architecture consists of an input or stroke-selection layer, an oscillatory layer, and the output layer where stroke velocities are estimated. A separate timing network prepares the network's initial state, which is crucial for accurate stroke generation. Neurobiological significance of this preparation, and a possible mapping of our architecture onto human motor system is suggested. Interaction between timing network and oscillatory layer closely resembles interaction between Basal Ganglia and Supplementary Motor Area in the brain.

Keywords Handwriting models · Oscillatory neural networks

1 Introduction

Handwriting is a learned, highly practiced human motor skill that involves control and coordination of several subsystems in our motor system. Production of handwriting requires a hierarchically organized flow of information undergoing a series of transformations [1,2]. The writer starts with the intention to write a message (semantic level), which is transformed into words (lexical and syntactical level). When the individual letters (graphemes) are known, the writer selects specific letter shape variants (allographs). The selection is done according to the formal allograph selection syntax, individual preference or just random choice [3]. Below this level, the allographs are transformed into movement patterns, which is the object of focus of the present work.

Models of handwriting We now briefly review some of the key models of handwriting, particularly emphasizing those that inspired development of the proposed handwriting model. Two general methodologies of handwriting modeling seem to have been adopted by modelers in the past. The first one, dubbed the "bottom-up" approach, refers to computational models which attempt to empirically reproduce features of human writing such as velocity and acceleration profiles etc; they do not claim any fidelity to neuromotor processes underlying handwriting processes [4,6,7]. The second methodology of handwriting modeling focuses on psychologically descriptive models [8,9]. These "top-down" models usually summarize many issues such as, motor learning, movement memory, planning and sequencing, co-articulation and task complexity of strokes, etc. The present work is closer to the "top-down" category.

Hollerbach's oscillation theory of handwriting An important class of handwriting models is centered on the philosophy that stroke data can be resolved into certain oscillatory components by a Fourier-style decomposition. The approach was pioneered by Hollerbach [6] who proposed an insightful model of handwriting generation where the hand-pen system is represented by two orthogonal pairs of opposing springs acting on an inertial load. It was pointed out that the oscillatory natural motions of this system resemble real handwriting segments. Anatomical justification of such a simple system has also been explored [6].

G. Gangadhar
Machine learning group, IDIAP Research Institute,
Martigny, Switzerland
e-mail: Gangadhar.Garipelli@idiap.ch

D. Joseph · V. Srinivasa Chakravarthy (✉)
Department of Biotechnology, Indian Institute of Technology,
Madras, India
e-mail: schakra@ee.iitm.ac.in

D. Joseph
e-mail: denny_cns@yahoo.com

Schomaker's model Schomaker [3] proposed a neural network model in which a network of oscillators outputs horizontal and vertical pen motion. Network training, performed using a variation of delta-rule, led to uncertain results: performance depended critically on network parameters. In spite of the shortcomings of the performance of the model, Schomaker's work clearly elucidates certain issues related to any possible handwriting model. Accordingly, the handwriting process—and hence its model—must have four basic events or phases both in chaining and shaping of handwriting [3]:

1. *System configuration* This stage is variously known as motor programming, coordinative structure gearing, preparation, planning, schema build-up etc.
2. *Start of pattern* After configuring the system for the task at hand, there must be a signal releasing the pattern at the right time.
3. *Execution of Pattern* The duration of this phase and actions that are performed depend on parameters such as amount of time elapsed, the distance from a spatial target position or force target value, or even the number of motor segments produced.
4. *End of pattern* This stage deals with the termination of the movement.

Kalveram's model More recently, Kalveram [7] proposed a model in which stroke data is resolved to its Fourier components. This simple mathematical operation is described using the metaphor of 'central target pattern generator'. The model, in our view, has several drawbacks. Since a handwritten stroke — for that matter any real motor sequence — lives for a finite duration, the dynamics of a system that generates it must be appropriately initiated and terminated. Fourier decomposition assumes a set of oscillators whose initial state is accurately prepared with precise phase-relationships among the oscillators. In a large network of oscillators this preparation of the initial state can be a challenge in itself, in addition to accurate stroke learning/acquisition and production. These issues are not addressed in [7] which assumes a prepared initial state. Another drawback is that in [7] a separate network has to be trained for every stroke.

Plamondon's model Plamondon [4] presented a bottom-up model using "delta-lognormal synergies". The name refers to authors' estimation of the velocity of a muscle synergy as a Gaussian function of the movement parameters that varies logarithmically with time. The model therefore produces bell-shaped velocity profiles similar to those seen in simple hand movements. They also demonstrated the "Two-Thirds Power Law" relation between angular velocity and curvature for a limited range of elliptical movements for which the law accurately describes human writing.

AVITE WRITE model Adaptive VITEWRITE (AVITE) model [5] is a neural network handwriting learning and generation system that brings together the mechanisms from Bullock's [9] cortical VITE (Vector Integration to Endpoint) and VITEWRITE trajectory generation models, and the cerebellar spectral timing model of Fiala et al [10]. This synthesis creates a single system capable of both reactive movements as well as memory based movements based on previous cerebellar movement learning and subsequent read out from long-term memory. AVITEWRITE model successfully explains the psychophysical and neurobiological data about how synchronous multi-joint reaching trajectories could be generated at variable speeds. The AVITEWRITE model is used to simulate key psychophysical and neural data about learning to make curved movements, including reduction in writing time as learning progresses; generation of unimodal, bell shaped velocity profiles for each movement synergy; size and scaling with preservation of the letter shape and shapes of velocity profiles; an inverse relation between curvature and tangential velocity; and two-thirds power law relation between angular velocity and curvature. Though the model successfully explains several features of handwriting, it may be noted that it does not belong to the family of "oscillatory" models of handwriting. We will argue in this paper that investigating handwriting in terms of its oscillatory components throws up certain important aspects of handwriting—or perhaps all voluntary control—like preparation, motor delay etc. The present model addresses these issues.

A key contribution of the present model of handwriting is its interpretation of motor preparation as the preparation of the initial state of the oscillatory layer. The model presents a network realization of oscillation theory of handwriting [6], and, unlike Schomaker's model [3], the proposed model exhibits reliable training performance.

The outline of the paper is as follows. In Sect. 2, features of the present model are discussed. A mechanism for preparing the initial state of the network, training, and validation procedures are described. In Sect. 3, simulation results of learning lower-case, single-stroke English characters are presented. The need for appropriately preparing the network state raises new issues, which are discussed from a biological perspective in the final section. Proposed extensions of the model are also discussed in the same section.

2 The model

The essence of our approach is to produce a stable rhythm in a network of oscillators, and resolve the stroke output in a Fourier-like fashion, in terms of the oscillatory activities of the network of oscillators. The architecture of our network that learns to generate strokes has 3 layers: 1. input layer, 2. oscillatory layer, and 3. output layer (Fig. 1). Each node in the input layer represents a separate stroke. Under resting conditions, all inputs are in a 'low' (0) state. To generate a stroke,

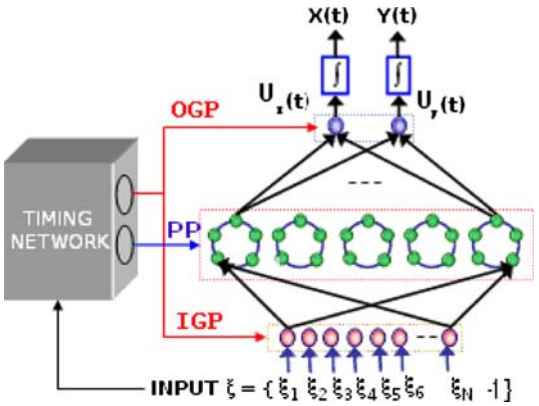

Fig. 1 Architecture of oscillatory network. The stroke selection vector, ξ, is presented at the input layer. Output of the network are pen velocities, $U_x(t)$ and $U_y(t)$. The timing network coordinates events in the network. It gates the input and output layers using IGP and OGP signals, respectively. It prepares the oscillatory layer using PP signal. See text for details

the corresponding input line is taken to a 'high' (1) state, and held in that state throughout execution of the stroke. The oscillatory layer has several sublayers. All the neurons in a sublayer have the same oscillation frequency. In each sublayer, neurons are connected in a ring topology. Our model differs from the model of Schomaker [3] in this respect: lateral connections are absent in Schomaker's model.[1] Output layer has two outputs representing horizontal and vertical velocities (U_x and U_y) of the pen tip. Each of the outputs is connected to all the oscillators in the oscillator layer. The timing network controls the events in the above 3-layered network (Fig. 1).

Single oscillator model Dynamics of a single neural oscillator used in the oscillatory layer of our network are given as:

$$\tau_x \frac{dx}{dt} = -x + V - s + I \tag{1}$$

$$V = \tanh(\lambda x) \tag{2}$$

$$\tau_s \frac{ds}{dt} = -s + V \tag{3}$$

where 'V' denotes the oscillatory output, and 'x' and 's' are auxiliary, internal variables of the neuron, respectively. Note that while 'x' has excitatory influence on 's', 's' in turn inhibits 'x'. Such excitatory-inhibitory pair is a standard recipe for producing oscillations.

Analysis of Eqs. (1), (2) and (3) shows that, for $I = 0$, $s = 0$, and $\lambda >> 1$, V in Eqs. (1) and (2) has two stable states, $V \approx \pm 1$. Correspondingly x also has two stable states, a positive and a negative value. Moreover, if x is at negative (positive) stable state, a sufficiently large negative (positive) 's' in Eq. (1) flips 'x' to its positive (negative) stable state. In Eq. (3), 's' simply follows 'x' with a delay. Therefore, a persistent value of x induces a change in 's' such that 'x' is toggled periodically. Oscillations are produced by the above system (see appendix for a formal proof), but only within certain limits of the external input I (Fig. 2). Beyond those limits the neuron has fixed point behavior. Note also Eqs. (1), (2) and (3) have an unstable fixed point at origin ($x = 0, s = 0$). The average output of the neuron as a function of 'I' has a sigmoidal form (Fig. 2). τ_x and τ_s are the time constants of Eqs. (1) and (3). The period of oscillation of the oscillator can be varied trivially by scaling these two time constants appropriately. This method is used to vary the frequency of various sublayers in the oscillatory layer in Sect. 3, experiment no. 1.

Sublayer model Each sublayer consists of a network of oscillators [see Eqs. (6), (7) and (8) below] connected in a ring topology. By a proper choice of parameters, such a network of oscillators can produce a limit cycle, with specific phase relationships among individual oscillators. Odd number of oscillators in ring (sublayer) is preferred for mode locking as even number of oscillators may lead to loss of rhythm stability i.e., "oscillator death."[12]. Choice of such special architectures is imperative since it is known that a general network of nonlinear oscillators is intrinsically chaotic [11]. A sublayer with ring topology, odd number of oscillators, where each oscillator is coupled with one (right or left only) neighbor with sufficiently strong (negative) coupling strength exhibits mode locking, where each oscillator produces a periodic output and adjacent oscillators differ by a phase difference of $\Delta\phi = \pi + 2\pi/\mathrm{m}$ (m is the number of oscillators) [12].

Preparing the network state This important stage is described variously in literature as system configuration, motor programming, coordinative structure gearing, preparation, planning, schema build-up etc [3]. We will use the term *preparation* in this paper. Although the problem of motor preparation has several dimensions, in context of our network model we give it a specific meaning. Since the network is a dynamic system, it must be brought to a "standard state", V^s, if possible, from a random, unspecified state, before it can produce a stroke. This preparation is achieved by giving a preparatory pulse (PP) to a specific neuron (chosen to be the first neuron in every sublayer without loss of generality) and waiting for a specific delay interval. The delay must be long enough to allow the oscillatory layer to approach the limit cycle sufficiently closely; beyond this minimum value the delay must be precisely chosen such that the oscillatory layer state is at a predetermined phase in the limit cycle. We refer to this state as the "standard state", V^s, henceforth. Since the oscillatory layer has a limit cycle attractor, once the steady state is reached, the oscillatory layer, in free-running conditions (no external input), periodically visits every point on the limit cycle. The standard state is chosen to be a point

[1] This might be a reason behind uncertain results of this model, since lateral connections are essential to stabilize the rhythm in the oscillatory layer.

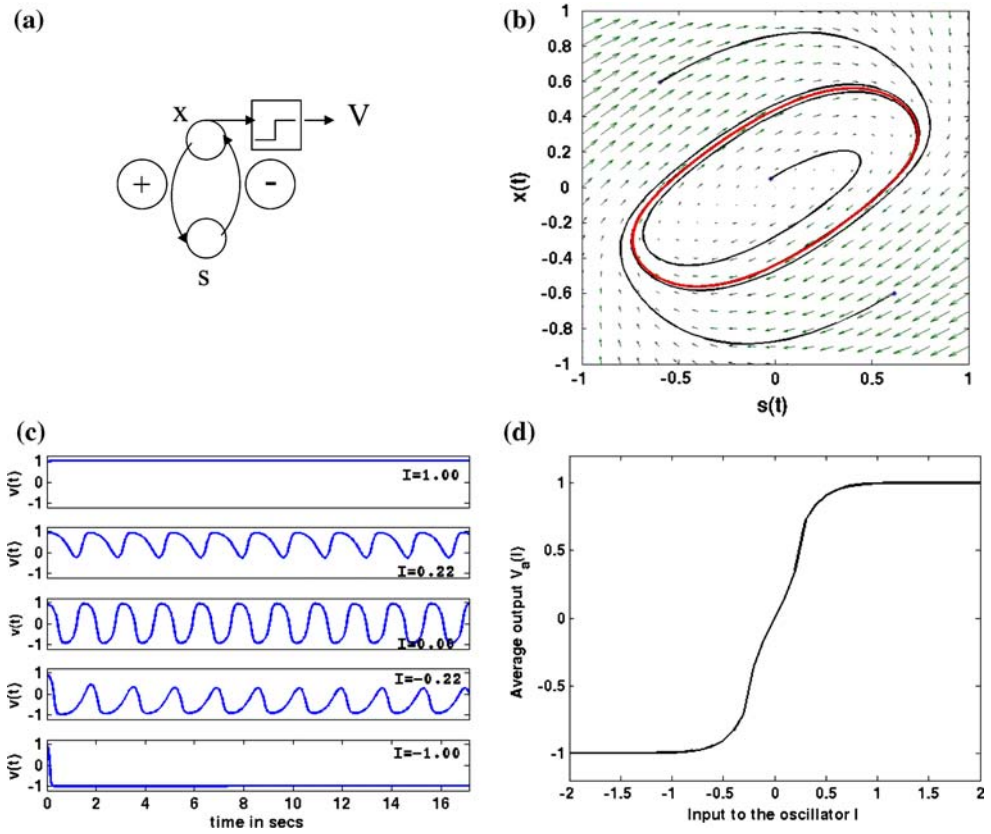

Fig. 2 **a** Schematic depicting the dynamics of a single neural oscillator. Variable 'x' excites 's', which in turn inhibits 'x'. 'x' passed through a sigmoid yields 'V'. **b** Dynamic flow of the system described by Eqs. (1), (2), and (3) indicating presence of a limit cycle attractor. **c** Response of single neuron for various values of external input I(= 1.00, 0.22, 0.00, −0.22, −1.00, respectively from *top to bottom*). **d** Average output response (V_a) of a single neuron as a function of I. The average is computed with simulations of oscillator with 100 cycles (= 100 × 120/70s) and 100 trials for each I with random initial values for 'x' and 's' in the range [−1 1]. ($\tau_x = \tau_y = 0.24$ s, $\lambda = 3$)

on the limit cycle. We *define* the standard state, V^s, as the state reached by the oscillatory layer at the end of a specific preparatory delay (time elapsed after the PP), Δ (=600 time units), and with a specific PP of duration, τ, (=20 time units) and amplitude, A (=20).

The timing network The timing network controls the timing of various events in the network (Fig. 3). The command to execute a stroke corresponding to the jth neuron in the input layer, is received by the timing network at $t = 0$. At the same time the jth input line in the input layer is set to a 'high' value. Immediately (at $t = 0+$) the timing network sends PPs (of duration τ) to all the sublayers in the oscillatory layer. After a delay, Δ, (i.e., $t = \tau + \Delta$) the timing network sends an enabling signal, the input gating pulse (IGP) to the input layer so that the input signal, transformed by a weight stage, reaches the oscillatory layer. An output gate pulse (OGP) is also sent to the output layer enabling the output. That is, during this interval ($t = [0, \tau+\Delta]$) the oscillatory layer does not know about the change in the state of the input lines. Immediately after ($t > \tau + \Delta$), the OGP is given to the output layer, and the stroke velocity information begins streaming out of the output layer. The output gating duration, T_i, generally speaking, must be specific to the stroke that is being produced. However we consider a simpler situation where all strokes are of equal duration, which is presently equal to the time period, T_f, of the slowest oscillators (those of first sublayer) in the oscillatory layer.

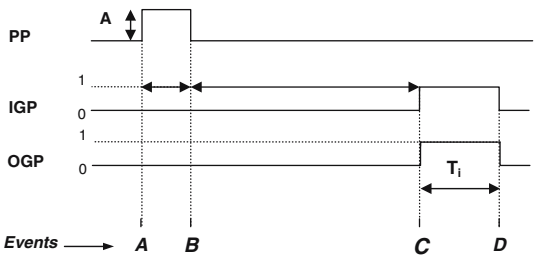

Fig. 3 The timing signals (*PP* preparatory pulse, *IGP* input gate pulse, *OGP* output gate pulse), '*A*' is the amplitude of preparatory pulse, τ is the duration of PP, T_i is the duration of OGP for ith stroke, which is presently the same, T_f, for all strokes. See text for a summary of events A, B, C and D

Summary of events A, B, C, and D in Fig. 3:

A The input is fed to the network (also to timing network). The timing network injects PP for the duration ($\boldsymbol{\tau}$), to the First oscillator in every sublayer. Input to the oscillatory network is disabled during this interval since the IGP is low.
B This event is the end of PP and start of preparatory delay for duration $\boldsymbol{\Delta}$. IGP and OGP continue to be low.
C Start of IGP and OGP with duration T_i, which enable the input and output. The network starts generating velocity information.
D The end of IGP and OGP, network output is again disabled, velocities become 0, and the pen tip stops.

Network response Pen-tip velocities (U_x and U_y) estimated by the network are expressed as weighted sum of the outputs of neurons in the oscillatory layer:

$$U_x(t) = \sum_{k=1}^{Ns} \sum_{i=1}^{Nk} W_{ik}^x V_{ik}(t) \tag{4}$$

$$U_y(t) = \sum_{k=1}^{Ns} \sum_{i=1}^{Nk} W_{ik}^y V_{ik}(t) \tag{5}$$

where, N_s is the number of sublayers in the oscillatory layer and N_k is the number of oscillators in kth sublayer, W_{ik}^x and W_{ik}^y are connections from ith oscillator in kth sublayer to output nodes U_x and U_y, respectively. Output, V_{ik}, of the ith oscillator in the kth sublayer is given by:

$$\tau_x \frac{\mathrm{d}x_{ik}}{\mathrm{d}t} = -x_{ik} + \sum W_{irk}^{\mathrm{lat}} V_{rk} - s_{ik} + I_{ik}^{\mathrm{net}} \tag{6}$$

$$V_{ik} = \tanh(\lambda x_{ik}) \tag{7}$$

$$\tau_s \frac{\mathrm{d}s_{ik}}{\mathrm{d}t} = -s_{ik} + V_{ik} \tag{8}$$

where, x_{ik} is the state of ith neuron in kth sublayer, s_{ik} is the auxiliary internal variable of the ith oscillator in the kth sublayer, W_{irk}^{lat} is the lateral connection from rth oscillator to ith oscillator in kth sublayer. As described earlier each sublayer is a ring in which oscillators are connected in a unidirectional fashion with negative coupling strengths as follows:

$$\begin{aligned} W_{irk}^{\mathrm{lat}} &= \nu; \quad \text{if } (r = i+1), \quad \text{or } (r = N_k \text{ and } i = 1) \\ &= 0, \quad \text{otherwise.} \end{aligned} \tag{9}$$

In the simulations of the following section we take $\nu = -0.5$.

I_{ik}^{net} is the net input to ith oscillator in the kth sublayer is given by

$$I_{ik}^{\mathrm{net}} = \sum_l W_{lik}^1 \xi_l \tag{10}$$

where, ξ_l is lth input in input vector $\xi = \{\xi_1, \xi_2, \xi_3 \ldots \xi_l \ldots \xi n, -1\}$ and W_{lik}^1 is the weight connecting lth input node and ith oscillator in kth sublayer. The last component of ξ, -1, is the bias input to the oscillatory layer.

Preparatory pulse (PP) is given as external input, I_{ik}^{net}, in Eq. (6). PP is a rectangular pulse of amplitude, A, and duration, τ, given to the first neuron in each sublayer.

Training Since the time-averaged output of the oscillatory neuron varies in a sigmoid form as a function of external input (see Fig. 2) backpropagation (BP) algorithm may be used for training [13]. Backpropagation with (BP momentum) and without momentum (plain BP) are applied [13]. BP algorithm is normally used to train a multi-layered perceptron model to map static input/output vector pairs. In the present case, the oscillatory network is trained to produce stroke velocities as follows. To train the network on lth stroke, ξ_l, the lth input component in input vector $\xi = \{\xi_1, \xi_2, \xi_3 \ldots \xi_l \ldots \xi n, -1\}$, is set to 1, and all other input components are set to 0. The corresponding target output is a set of stroke velocities, $V_x(t)$ and $V_y(t)$. Note that the oscillatory layer is prepared, as described earlier, and brought close to the standard state, before training every stroke. Since time is discretized, when the network is trained to produce a stroke, it is actually trained to map the following sequence of input/output pairs:

$$\xi(t_m) \rightarrow (V_x(t_m), V_y(t_m))$$

where $\xi(t_m) = \xi$, (input is constant throughout the stroke) and t_m is the mth instant.

Only the first and second stage weights are trained; the lateral weights in the oscillatory layer are held constant. Weight update equations are as in [13]. Comparison of training error corresponding to learning algorithms Plain BP and BP with momentum is shown in Fig. 4.

Calculation of mean error The mean error shown in the Fig. 4 is calculated using the formula,

$$E = \sum_p^{N_S} \sum_q^{N_L} \left\{ (V_x^{pq} - U_x^{pq})^2 + (V_y^{pq} - U_y^{pq})^2 \right\} \tag{11}$$

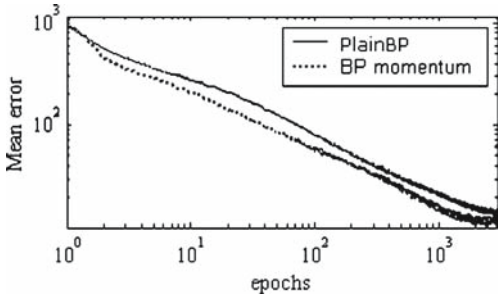

Fig. 4 Comparison of training error corresponding to learning algorithms Plain BP and BP with momentum. The mean error for "BP with momentum" converges faster than plain BP learning mechanism. An epoch means a single presentation of all strokes

where, V_x^{pq} and U_x^{pq} are the qth points in the desired and actual x-velocities of the pth stroke respectively. Similarly subscript 'y' indicates y-velocity. E is the average reconstruction error in stroke velocity, N_S is the number of strokes and N_L is the number of points in velocity profile of a stroke, which is the same for all strokes.

3 Results

Lower case English alphabets are collected using a stylus (electronic pen) connected to a computer. These strokes are represented by pen tip coordinates, $x(t)$ and $y(t)$, along x-direction, and y-direction, respectively. The sampling frequency of the device is approximately 70 Hz and hence the sampling time (referred as 'time unit' in the paper) is equal to 1/70 s. The collected strokes are nearly of the same length; points towards the end are dropped to make them all equally long (120 points per stroke). The duration (T) of each stroke is therefore $120 \times (1/70) = 1.7143$ s. These strokes are used to train the oscillator neural network model of handwriting generation. The frequency of sublayer with lowest frequency of oscillations is set to $f = 1/T$. In the following simulation experiments, the effect of various network parameters on training performance is studied. Training is performed using BP with momentum. Learning rates for the first and second stage weights are 0.000005, and 0.0001, and the momentum factor is 0.7.

3.1 Experiment no. 1

(i) In this experiment, the oscillatory layer is designed with $n = 10$ sublayers with intrinsic frequencies of the oscillators taken as $\{n \times f : n = 1\text{–}10\}$, where $f = 1/T$ and T is the (common) duration of strokes. The number of oscillators per sublayer is kept constant and is equal to 25.

The results of training of the above network show that the contribution of weights corresponding to oscillators with higher frequencies is not significant (see Fig. 5b); and the reconstructed strokes have a high-frequency 'tremor-like' distortion (see strokes in **B** of Fig. 5a). In order to eliminate this high-frequency distortion, we consider the following network modification.

(ii) In this simulation the oscillatory layer has five sublayers, and each sublayer has 25 oscillators. The oscillators are assigned frequencies limited to the band Δ f = [f, 3f], where the frequency of any oscillator in the kth sublayer is given as $f^k = f + \Delta f \times (k-1)/(N_s - 1)$, and N_s is the number of sublayers in the oscillatory layer. On testing, it is observed that reconstructed strokes have substantially lesser high-frequency distortion (see strokes of row B in Fig. 6a) than those of row **C** in Fig. 5b.

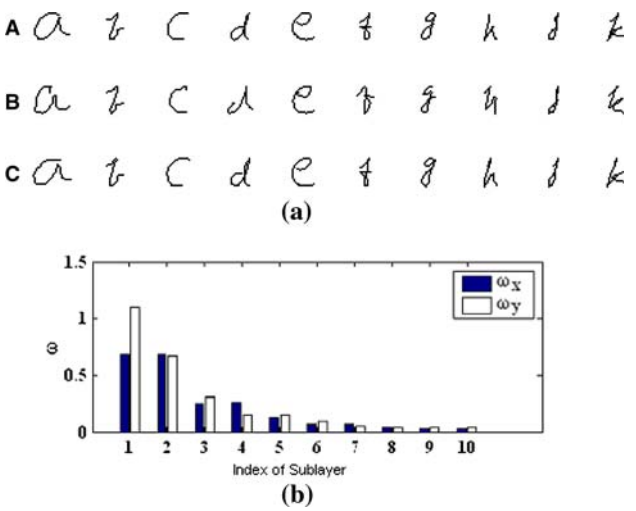

Fig. 5 **a** Original and reconstructed strokes illustrating effect of harmonics in the oscillator layer *A* Original strokes, *B* Reproduced with harmonics, *C* Reproduced without harmonics **b** Average magnitude of weights at the end of training; Number of sublayers = 10; Frequencies = $\{f, 2f, 3f, 4f, 5f, 6f, 7f, 8f, 9f, 10f\}$; Number of units per subnet = 25. ω_k^x and ω_k^y are mean of absolute values of weights connecting each sublayer to output nodes, U_x, U_y (i.e., $\omega_k^x = \frac{\sum_i^{N_k} |w_{ik}^x|}{N_k}$ and $\omega_k^y = \frac{\sum_i^{N_k} |w_{ik}^y|}{N_k}$), where w_{ik}^x and w_{ik}^y are the weights connected from ith oscillator in kth sublayer to U_x and U_y node in the output layer, respectively

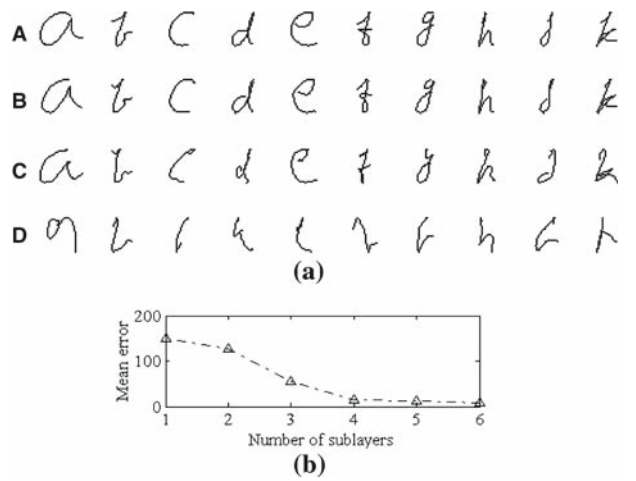

Fig. 6 **a** Reconstructed strokes with varying number of sublayers *A* Original strokes, *B* five sublayers, *C* three sublayers, *D* one sublayer **b** Mean of reconstruction error versus Number of sublayers

3.2 Experiment no. 2

In this experiment, the number of sublayers is varied from 6 to 1 and the number of oscillators per sublayer is kept constant (=25). The frequencies of the oscillators are limited to band Δ f (as discussed in experiment no. 1). The network is trained on 10 strokes (see Fig. 6a). It is observed that as

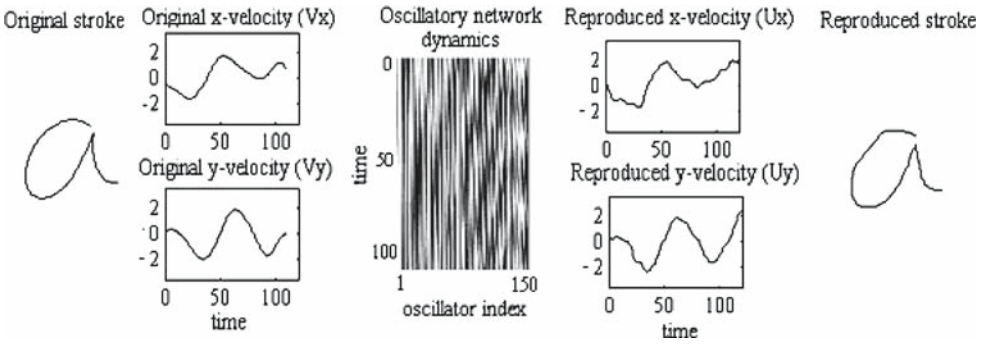

Fig. 7 Reconstruction of the stroke '*a*' by the oscillatory network. A network with six sublayers and 25 oscillators per sublayer is trained with strokes shown in Fig. 5a. Stroke '*a*' on the left side corresponds to the original stroke right side corresponds to reconstruction. The velocities profiles of original stroke and reconstructed stroke are shown along with the profile of dynamics of oscillator network

the number of sublayers increases, the mean reconstruction error decreases (see Fig. 6b).

The reconstruction of the stroke '*a*' by the oscillatory network with six sublayers, with 25 oscillators in each sublayer is shown in Fig. 7. Frequencies of various sublayers are confined to the band $[f, 3f]$ as described above.

3.3 Experiment no. 3

Significance of post-preparatory delay (PPD, Δ): After the preparatory pulse (PP) is given to the oscillatory layer, the layer is allowed to run freely for a post-preparatory delay (PPD) period (Δ) before the input is presented (see Fig. 8). How does the performance of the network depend on Δ? Does the performance error decrease gradually with increasing PPD since the network gets more time to settle? Simulations conducted to answer this question show that the quality of stroke generation depends on PPD in non-intuitive ways. For effective preparation, the timing network should allow enough delay to allow the oscillator layer to settle to a 'standard state'. The following studies illustrate the implications of such a delay.

From Fig. 8 it is clear that error in reproduction does not vary monotonically with PPD, but acquires locally minimal values if the stroke onset occurs at discrete, and nearly periodic intervals after the preparation. This is because the state of the oscillatory layer (almost) periodically approaches the 'standard state' in its circling approach to its final limit cycle attractor (see Fig. 9). The network is originally trained using a PPD of 600. Reproduction of strokes is legible at PPD values of 240, 360, 480, etc., even though 600 is the PPD value used during training. Therefore, for faster stroke execution, the network may not really need to wait for long periods: what matters is the precise delay after the preparatory pulse. This experiment inspires some clear predictions on human handwriting, or more generally, perhaps on all voluntary movements. The onset of handwriting probably always occurs only at characteristic, discrete intervals after the command for execution is given. Further, one might surmise that

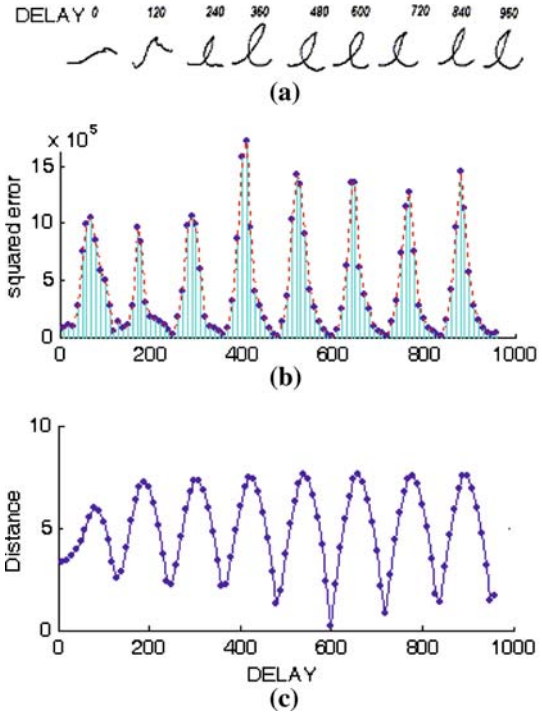

Fig. 8 Reconstruction error and reconstructed strokes with various PPDs. **a** Strokes reconstructed with delays mentioned at the top of the strokes, **b** Reconstruction error as a function of PPD. **c** The deviation from the standard state with delay. Note that the error is not a monotonic function of PPD. Error reaches small values at periodic values of delay (PPD)

handwriting movements forcibly constrained to commence at other instants should manifest larger errors.

3.4 Experiment no. 4

Origins of motor variability: One of the most commonly seen features in human movement is motor variability. Motor control researchers view it as a window into the central organization of the system that produces voluntary movements [14]. One of the obvious origins of motor variability is motor redundancy. Motor variability naturally emerges in the

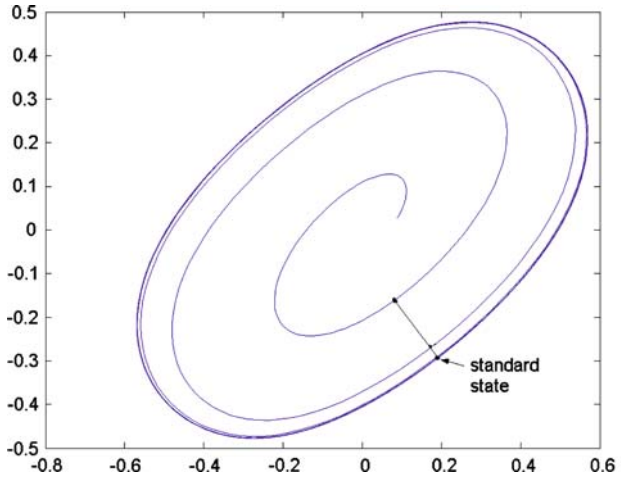

Fig. 9 Schematic depicting the manner in which oscillatory layer approaches the limit cycle. The state of the oscillatory layer, $V(t)$, in its circling approach to the limit cycle, (nearly) periodically passes by the standard state. Stroke onset at those points produces locally optimal reconstruction

present model. We now describe the various sources of motor variability in the model.

(i) Variability due to variation of PPD in motor preparation. Two cases arise:
   (A) PPD is an integral multiple of fundamental period, T (=120): locally optimal reconstruction is obtained for such values of PPD, which correspond to the minima of the error curve in Fig. 8, though the exact form of reconstruction varies with the precise value of PPD (Fig. 8).
   (B) PPD is not an integral multiple of fundamental period: reconstruction error increases drastically with increased deviation from the precise discrete values of PPD (Fig. 10).

(ii) Variability due to random initial state also introduces variability in the stroke produced (Fig. 11).

The network starts from a random state before preparation; preparation ensures that the initial variability in the network state, just before stroke onset, is suppressed. Therefore variability in stroke output due to variability in initial state is not as significant as variability due to PPD (Fig. 11).

3.5 Experiment no. 5

Generating a stroke sequence: Natural handwriting involves a smooth, flowing execution of multiple strokes in a desired sequence. So far we have only discussed the execution of single strokes. Can the model be extended to execute a stroke sequence? How are timing events coordinated when multiple strokes are executed in a sequence? Does the network

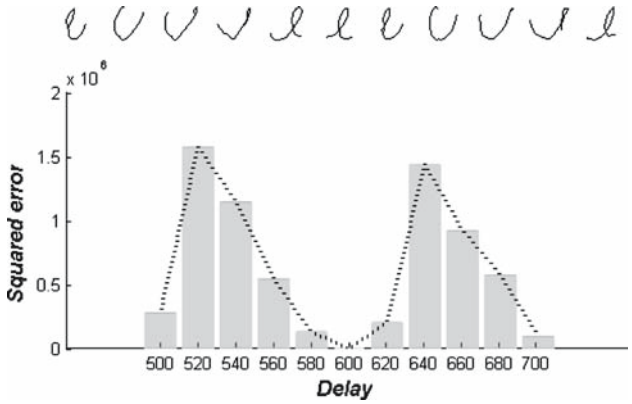

Fig. 10 Variation of reconstruction error with PPD in the neighborhood of a delay at which minimum error is obtained

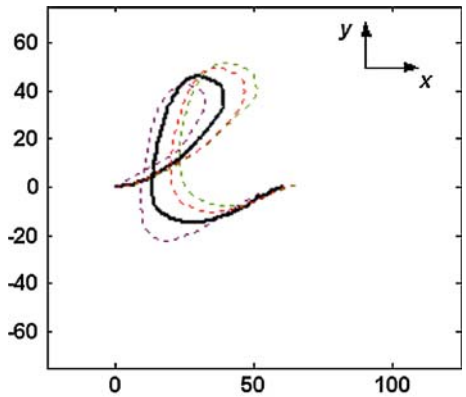

Fig. 11 Variability due to random initial state

have to be prepared afresh after every stroke? Two options immediately suggest themselves: (1) to prepare the network after every stroke, or, (2) generate multiple strokes with a single preparation before the first stroke. The main issues in the generation of a stroke sequence are: (1) the total time of execution should be small (inter-stroke delays should be minimal), and (2) the generated stroke sequence should be robust and accurate.

Before describing the methods evolved to address the above issues we describe a notation—we name it the *Event Chain* notation— that simplifies description of the following experiments. Any handwriting sample consists of a sequence of events, some visible (e.g., a stroke), and some invisible (preparatory processes before stroke execution). The Event Chain notation describes the sequence of events in a convenient fashion. An *event chain* is a sequence of *events*, like, for example:

My_event_chain = [<event 1>, <event 2>, ... <event n>]

Description of each *event* has multiple fields, as, for example,

My_event = <*event_descriptor*, *duration*, *param1*, *param2*, ...>

In each event, the first field is a textual description of the event, the second denotes event duration, and the subsequent fields are optional, representing other parameters that characterize the event. Thus Event Chain notation for a sample sequence is shown below:

My_event_chain = [<*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'e', 120>, <*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'l', 120>].

Explanation of events:

<*Preparatory Pulse*, 20, 20>: A preparatory pulse of duration =20, and amplitude =20

<*Preparatory Delay*, 600>: Preparatory Delay of duration 600

<S*troke*'e', 120>: Execution of stroke 'e' of duration 120 (The two subsequent events are as described above.)

<S*troke*'l', 120>: Execution of stroke 'l' of duration 120

Thus the above event chain describes an 'el' stroke sequence executed with full preparation before every stroke. With this notation in place, we now proceed to describe development of methods for executing a sequence of strokes.

(a) *Stroke sequence production with multiple preparations* According to the procedure for stroke generation described in Sect. 2, the network has to be prepared before the execution of every stroke. Thus production of a stroke sequence by this method incurs a long preparation before every stroke. The Event Chain description for the stroke sequence 'e-l-l-e' produced by this method is as follows:

Stroke_Sequence_elle = [<*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'e', 120>, <*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'l', 120>, <*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'l', 120>, <*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'e', 120>].

The inter–stroke delay (duration of preparatory pulse + post-preparatory delay i.e., $= \tau_p + \Delta = 620$) for stroke production by this strategy is much longer than the duration of stroke production ($T = 120$) itself. The quality of the stroke sequence 'elle' generated with this method (Fig. 12a) is robust because the network is fully prepared before every stroke. This is because the network approaches the standard state before execution of every stroke (Fig. 12c). However it takes 2,960 ($= (620 + 120) \times 4$) time units to execute these strokes instead of the ideal 480 ($= 120 \times 4$) time units. By way of reducing total time for stroke production, let us consider stroke production without preparation between strokes.

In this method there is no preparation between strokes. Event chain description of the 'elle' sequence produced by this method is:

Stroke_Sequence_elle = [<*Preparatory Pulse*, 20, 20>, <*Preparatory Delay*, 600>, <S*troke*'e', 120>, <S*troke*'l', 120>, <S*troke*'l', 120>, <S*troke*'e', 120>].

Although the sequence now takes only 1,100 time units, production quality after the first stroke is poor (Fig. 12b). In this case the network approached the standard state before the first stroke. However, its distance from the standard state at the onset of subsequent strokes continued to increase (Fig. 12d).

Since the network has always been trained such that the oscillatory layer is in the standard state at the onset of every stroke, it is only natural that performance is impaired when the initial state is different from the standard state. How do we ensure that the system returns to the standard state before every stroke with a short inter-stroke delay (preferably much less than stroke duration)? Our present method of preparation involves giving an initial pulse and *waiting* for the system to arrive at the standard state. Can this waiting be cut short by *driving* the system to the standard state within a short interval?

Active preparation We now present an alternate method of preparation, viz., active preparation (AP), by which the oscillatory layer is driven to the standard state from an arbitrary initial state. Eqations (6), (7) and (8) are modified as follows for incorporating AP:

$$\tau_x \frac{\mathrm{d}x_{ik}}{\mathrm{d}t} = -x_{ik} + \sum W^{\mathrm{lat}}_{irk} V_{rk} - s_{ik} + I^{\mathrm{net}}_{ik} + \gamma (V^S_{ik} - V_{ik}) \tag{12}$$

$$V_{ik} = \tanh(\lambda x_{ik}) \tag{13}$$

$$\tau_s \frac{\mathrm{d}s_{ik}}{\mathrm{d}t} = -s_{ik} + V_{ik} \tag{14}$$

Note the extra term, $\gamma (V^S_{ik} - V_{ik})$, the drive term, which is the only difference between Eqs. (6), (7), (8) and Eqs. (12), (13) and (14) above. Equations (13) and (14) which are identical to Eqs. (7) and (8) are simply reproduced here for clarity. In this method, the state of the oscillatory layer is driven towards the standard state, (V^s), through some sort of proportional control. The drive factor, γ, is reset to zero (no drive) during stroke generation, but set to a finite value in the inter-stroke interval, during which the network is actively prepared for the subsequent stroke. AP can be performed before the first stroke also, thereby reducing the prolonged preparatory delay. The Event Chain notation for a single event corresponding to AP is:

$$< \text{ Active Preparation}, d, \gamma >$$

where d is AP duration (APD), and γ is *drive factor* of Eq. (12).

Network Training using AP Using AP mechanism, we now train a network on six letters (strokes) 'a', 'c', 'd', 'h', 'r', 's'. The oscillatory layer in this case has four sublayers,

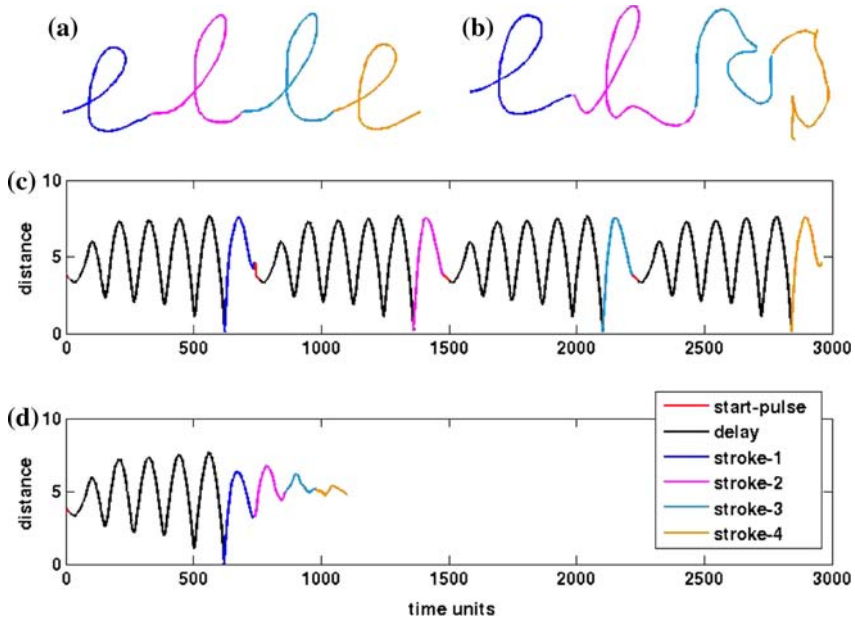

Fig. 12 **a** The 'e-l-l-e' sequence produced with full preparation before every stroke. **b** The 'e-l-l-e' sequence produced with full preparation before the first stroke only. Note that the stroke quality quickly degenerates over subsequent strokes. **c**, **d** Variation of Euclidean distance between the state of the oscillatory layer and the standard state , i.e., $||V - V^s||$. In **c**, since full preparation is made before every stroke, the system reaches standard state before every stroke. In **d**, since the system is fully prepared only once, before the first stroke, first stroke is robustly generated; quality of reconstruction of subsequent strokes quickly degenerates

each sublayer having 25 oscillators. Strokes are presented as sequences of 3. The following sequences are used for training: d–a–c, c–a–d, h–a–d, a–r–c, h–a–s, c–a–r. The network is trained for 5,000 epochs. Learning rates for the first and second stage weights are 0.000005 and 0.0001, and momentum factor is 0.7. Before presentation of each stroke, the oscillatory layer is prepared by AP. For concreteness, we present event chain notation for training of the network for a single two-stroke sequence 'h–a.'

`Stroke_sequence_ha = [<Preparatory Pulse, 20, 20>, <Preparatory Delay, 100>, <Active Preparation, 30, γ = 7 >, <stroke 'h', 120>, <Active Preparation, 30, γ = 7 >, <stroke 'a', 120>].`

A production of stroke sequence 'ha' with the above event chain description is shown in Fig. 13a. Note that due to AP, the state of the oscillatory layer is pushed very close to the standard state by the time of stroke onset. Note also that the preparatory delay is significantly lesser than what was used in passive preparation (100 as opposed to 600). Total time taken in this case is 420 (= 20+100+30+120+30+120), which is a significant savings from the previous case of passive preparation. Although the second stroke '*a*' is constructed reliably, note that the ligature between strokes is not smooth. In natural handwriting, ligature between two successive strokes has a smooth flowing quality. This is because often the pen tip velocity does not drop to zero at the border of two strokes and merely goes to a minimum. After extensive experimentation, we found that effective ligature handling consists of terminating the preceding stroke at an early stage, and initiating the succeeding stroke at a late stage. These results are explained below.

Ligature handling Early termination of the preceding stroke and late beginning of the succeeding stroke is the recipe used for achieving smooth ligatures. Event chain for executing the stroke sequence 'ha' with ligature handling introduced is given below:

`Stroke_sequence_ha = [<Preparatory Pulse, 20, 20>, <Preparatory Delay, 100>, <Active Preparation, 30, γ = 7 >, <stroke 'h', 100, 0, 100>, <Active Preparation, 30, γ = 7 >, < stroke 'a', 100, 10, 120>].`

Event description is as follows:

<Preparatory Pulse, 20>: preparatory pulse of duration 20 and amplitude 20.

<Preparatory Delay, 100>: preparatory delay of duration 100 (note the reduction from 600)

<Active Preparation, 30, $\gamma = 7$ >: AP of duration 30 and drive factor, γ, of 7.

<stroke 'h', 100, 0, 100>: stroke 'h' is executed.

The execution, which is of duration 100, is begun at $t_1 = 0$, and terminated at $t_2 = 100$, without completing the full 120 time units. This decremented interval at the termination end is called early stopping interval (ESI), which in this case is 120–100 = 20.

<Active Preparation, 30, $\gamma = 7$ >: AP of duration 30 and drive factor, γ, of 7.

<stroke 'a', 110, 10, 120>: stroke 'a' is executed. The execution is begun at $t_1 = 10$, and terminated at $t_2 = 120$,

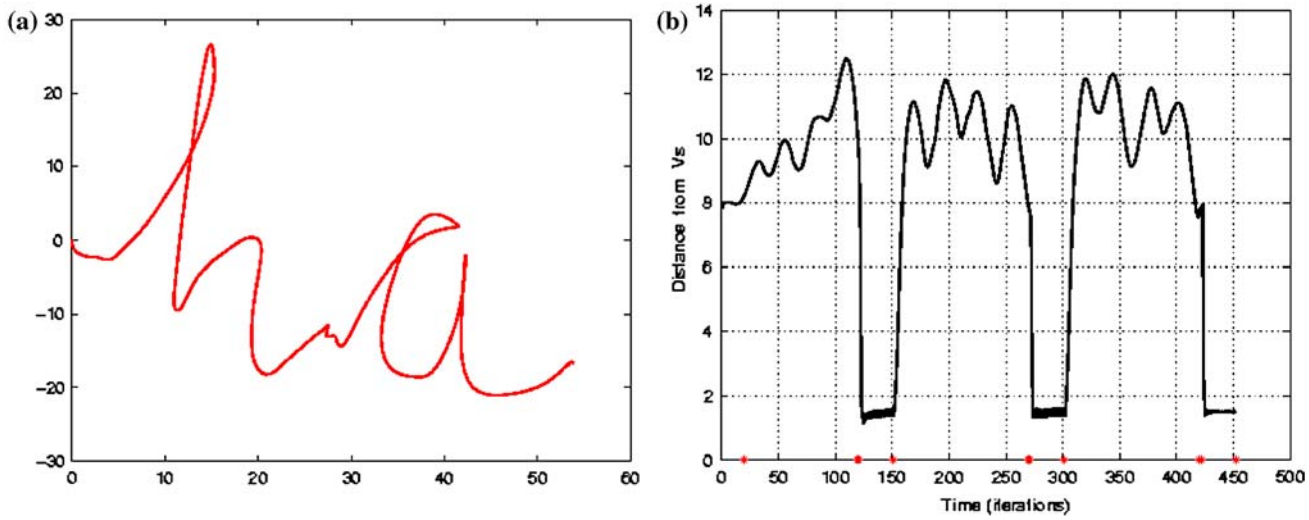

Fig. 13 **a** Stroke sequence 'h–a' produced by event chain description given above. Though individual strokes 'h' and 'a' are reproduced robustly, ligature is poor. Also the line supporting the stroke sequence is descending in left–right direction. **b** Euclidean distance between the state of the oscillatory layer and the standard state, V^s. Stretches of low value of distance represent AP events

making the stroke duration only 110 time units. This decrement in the stroke duration in the initiation end is called Late Beginning Interval (LBI).

Production of 'ha' obtained by the above method is shown in Fig. 14. Note the smoother ligature obtained by this method. However, in Fig. 14 we may also note an undesirable descent of the strokes from left to right.

Correcting line orientation The stroke sequence presented in Fig. 14 forms not a horizontal but an oriented line. The cause of this orientation is the discrepancy between the vertical positions of the two terminals of individual strokes. Since in the present method, the network estimates stroke velocities only and the actual stroke is constructed by velocity integration, vertical discrepancy between the two ends of the first stroke carries over to the second stroke; the second stroke therefore is drawn at a higher/lower level than the first. This error in orientation can be corrected by adding an appropriate offset in vertical velocity: positive vertical velocity correction must be applied to correct a dip in orientation. This offset is called Vertical Velocity Offset (VYOFF). A reconstruction of the stroke sequence 'h–a' of Fig. 14, drawn with a VYOFF=0.1 is shown in Fig. 15.

So far in this section, we have introduced three mechanisms for improving the quality of multiple stroke execution: (1) active preparation, (2) ligature handling, and (3)

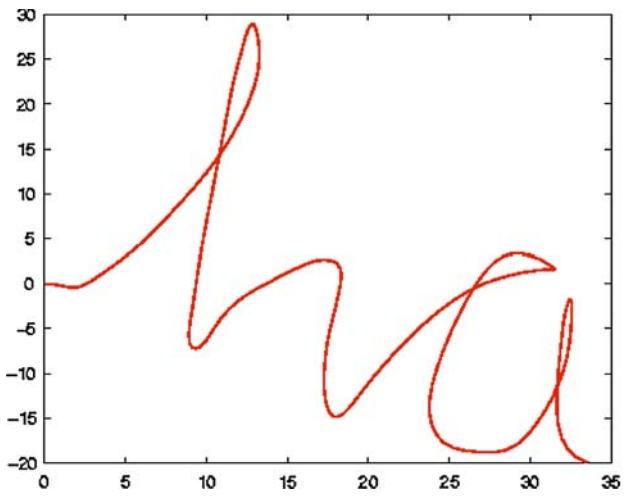

Fig. 14 Stroke sequence 'h–a' produced with ligature handling. Although ligature is smoother than the sequence shown in Fig. 13, there is an undesirable descent of strokes from *left to right*

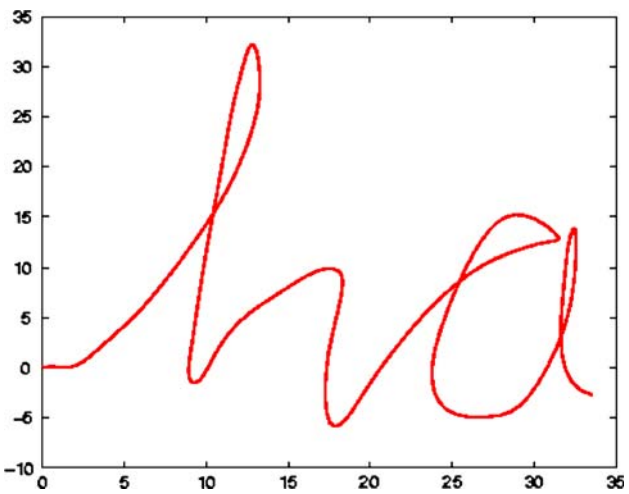

Fig. 15 Stroke sequence 'h–a' drawn with active preparation, ligature handling, and orientation correction

Correcting line orientation—to facilitate reliable multiple stroke generation. Each of these processes involves several parameters. AP involves its duration, APD, and drive factor, γ; ligature handling involves ESI and LBI; correcting line orientation involves VYOFF. Resetting the magnitude of any of these parameters to zero amounts to withdrawal of the corresponding corrective mechanism. We now demonstrate the significance of presence/absence of each of these mechanisms for effective generation of a sequence of multiple strokes (Fig. 16).

The parameter values used for execution of sequence 'h–a' in Fig. 16a are as follows: AP duration = 30, drive factor = 7, ESI = 20, LBI = 10, VYOFF = 0.1. Withdrawal of any of the three corrective mechanisms described above produced characteristic distortions in the stroke sequence generated (Fig. 16b–e). With these parameters we now present results related to execution of words — 2-letter, 3-letter, 4-letter and 5-letter words — produced by the network described above. The network is trained on six letters/strokes: 'a', 'c', 'd', 'h', 'r', and 's'. Ideally, the above parameters ought to vary depending on the stroke combinations that are produced. But such a characterization requires a much more detailed study and is deferred to future efforts in this direction (Figs. 17, 18).

4 Discussion

We present a neural network model of handwritten stroke generation in which stroke velocities are expressed as a Fourier-style decomposition of oscillatory neural activities. Though oscillatory neural models are typically used to model generation of rhythmic behavior like walking, swimming, etc. [15], they have proved to be useful in non-rhythmic motor function also [16]. Since Hollerbach's insightful observation on the oscillatory elements in handwriting, neural oscillators have also figured in models of handwriting. An oscillatory neural model of handwriting, for it to be biologically viable, has to address certain fundamental issues.

A key issue addressed in this paper is one of preparing the initial state of the oscillatory network. This question does not seem have received adequate attention in modeling literature [3–7]. Primarily the oscillatory layer must generate a stable rhythm appropriately registered with respect to the time of onset of the stroke. Further there must be a mechanism to switch the network to a different, stable rhythm to produce a different stroke. Even if the network dynamics are stable enough to flow into a stable trajectory on random initialization, the phase of the network's rhythmic state may not be specific enough to produce a desired movement. There must be some level of forgetting of initial conditions, and therefore linear oscillator models are disallowed. Networks of nonlinear oscillators, with their proneness to chaos [11], must be handled with extreme delicacy to produce stable, specific rhythms.

In the present work, we believe that a reasonable solution that addresses the above issues is provided. Two forms of preparation are described: (1) Passive preparation, (2) Active preparation. In passive preparation, the oscillatory layer is initialized with small random noise, but is immediately given a large PP to specific neurons (the first neuron in each sublayer). This pushes the evolution of oscillatory layer in a specific direction, which, after a specific delay, Δ, assumes a nearly standard rhythm (in spite of the low-amplitude

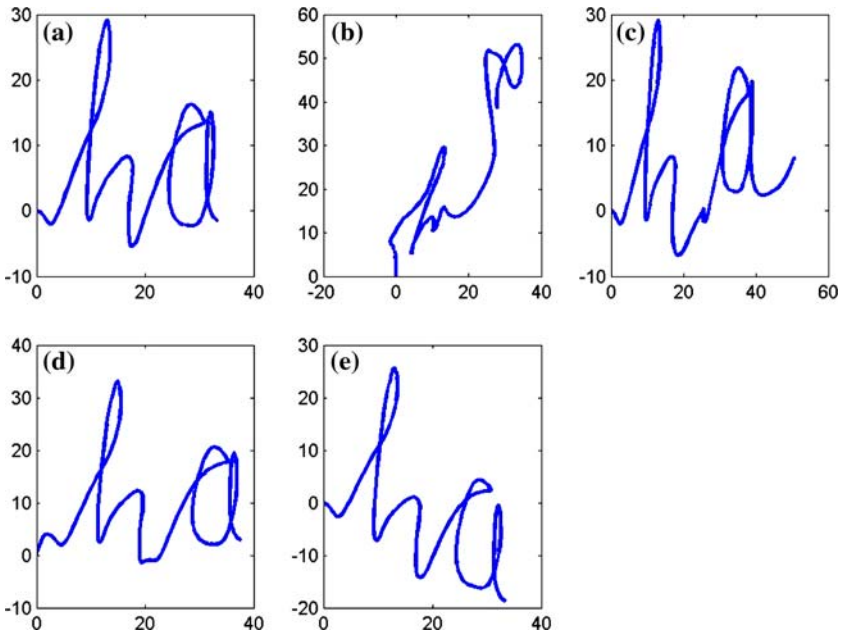

Fig. 16 **a** Stroke sequence 'h–a' executed with AP duration = 30, drive factor = 7, ESI = 20, LBI = 10, VYOFF = 0.1. **b** Sequence with no AP, all else being equal. **c** Sequence with ESI = 0, all else being equal. **d** Sequence with LBI = 0, all else being equal. **e** Sequence with VYOFF = 0, all else being equal

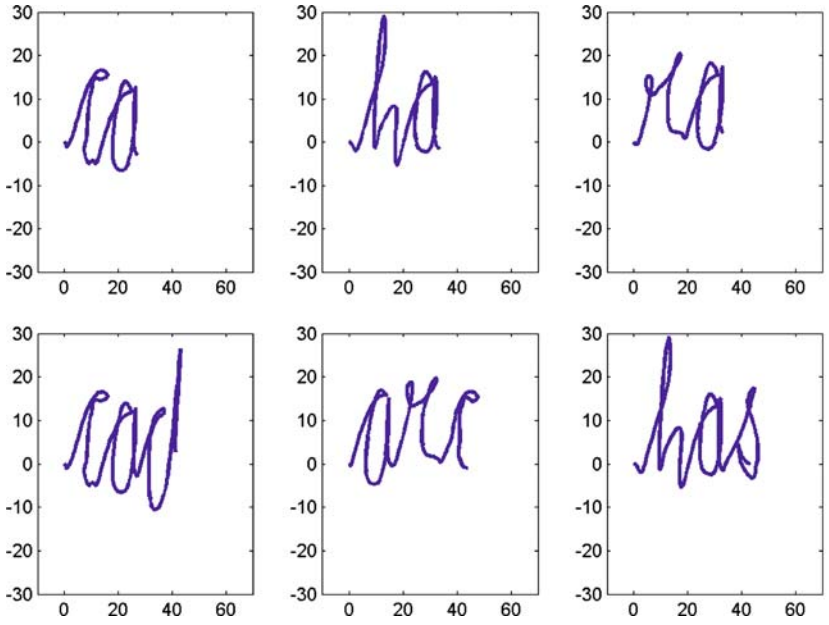

Fig. 17 Two-and three-letter words generated by the handwriting network

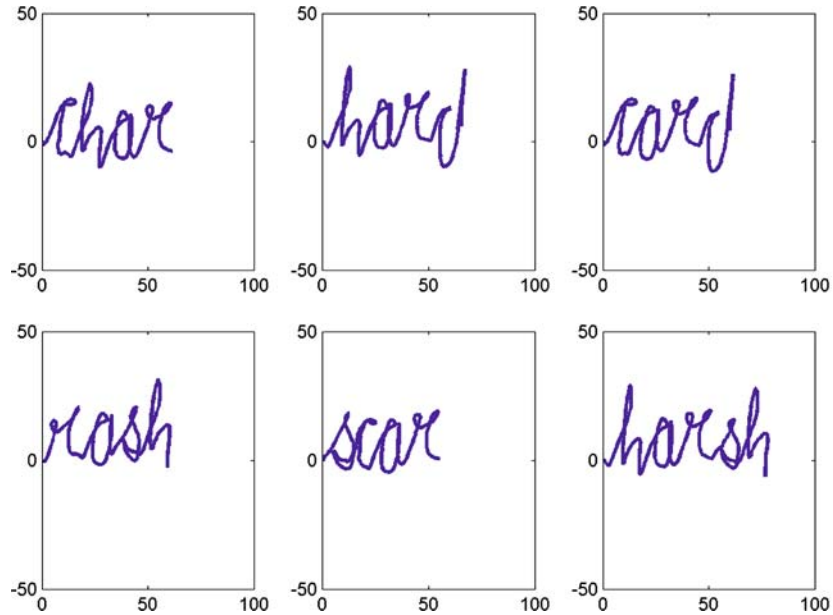

Fig. 18 Four and five-letter words produced by the handwriting network

fluctuation in the initial conditions). This is the form of preparation used in Sect. 3, experiments no. 1–4. Passive preparation involves unduly long preparation times, and also yields suboptimal results when a sequence of strokes is executed. Efficient reconstruction of a stroke sequence is achieved by introducing active preparation (in Sect. 3, experiment no. 5), in which the state of the oscillatory layer is actively driven towards the standard state. With AP, we show that it is possible to robustly produce longer stroke sequences. From purely algorithmic point of view, AP is more efficient (shorter preparation times, robust production of stroke sequences) than passive preparation. However, at this point it is not clear which of these faithfully describes the preparatory processes underlying biological motor function.

In the example shown in Sect. 3, Experiment no. 5, the network is trained only on a small number of strokes. To train on larger numbers of strokes, it is desirable to train subsets of strokes on multiple networks and use a mechanism for gating the outputs of the networks in appropriate sequence. For example, if Network-1 is trained on {a,b,c,d,e} and Network-2 is trained on {f,g,h,k,l}, then, to execute the stroke sequence 'h-e-a-l-e-d', the two networks have to gated in the following sequence {2,1,1,2,1,1}. Such an ensemble of networks can be trained on a large public online handwriting database and performance can be evaluated.

Mechanisms for ligature handling and orientation correction have been described in the paper. The parameters involved in these mechanisms are optimized by trial and error

for a small number of stroke sequences. However, more systematic methods for dynamically controlling these parameters during execution of arbitrary stroke sequences have to be investigated.

We now go deeper into the neuromotor significance of the proposed model of handwriting generation.

Motor preparation In their classic EEG studies of voluntary motor action, Kornhuber and Deecke [17] found slow negative shifts in cortical potential much before the initiation of movement. This potential, termed the Bereitschafts potential (BP), is believed to signify pre-movement preparation of motor cortical areas. Careful current dipole source analysis of BP has identified supplementary motor area (SMA) as a key player [18]. However, preparatory activity corresponding to movement direction has been found in many other brain areas including M1 [19], premotor cortex [20], prefrontal cortex [21], the parietal cortex [22], and basal ganglia [23]. An interesting functional definition of motor preparation emerges out of primate experiments by Churchland et al. (2006). This group hypothesizes that preparation is a process by which activity of the motor cortical neurons, random and variable in early stages of preparation, is progressively pushed into a limited region of the state space that is specific to a given movement. Data from premotor cortical neurons from primates appears to confirm their hypothesis [24]. Our model essentially conceives an idealization of this process in which preparation pushes the oscillatory state close to the standard state.

SMA and motor preparation From the above account SMA seems to compete with several other motor areas as a primary source of motor preparatory signals. Single cell recordings in primates revealed more marked preparation-related changes in SMA neurons than in neurons of M1 [25]. The question can be resolved if it can be shown that preparatory activity in SMA neurons precedes similar activity in M1. It has been shown that SMA neurons exhibiting preparatory activity can be identified to project to M1 [26]. Contrarily, it was also established that M1 neurons that exhibit preparatory activity receive inputs from SMA and not from thalamus or parietal cortex [27]. Such studies strongly implicate a role to SMA in motor preparation. However, perhaps SMA may not be solely responsible for motor preparation. Its preparatory action might involve interactions among subcortical structures like basal ganglia, which are often implicated in motor timing functions.

Basal ganglia and motor timing Coordinating the relative timing of multiple streams of processing is crucial in both motor performance and sensory perception. Temporal processing in biological systems occurs over a range of time scales and is broadly classified into three categories: (1) circadian timing, which corresponds to durations of the order of days, and handled by brain structures like suprachiasmatic nuclei, (2) interval timing, which corresponds to durations in the range of seconds to minutes, and coordinated primarily by corticostriatal interactions, and (3) millisecond timing, which obviously corresponds to millisecond durations, controlled by the cerebellum [28].

The role of basal ganglia in 'interval timing' appears to emerge from the dynamics of thalamo-cortico-striatal loops. In a model— the striatal beat frequency (SBF) model [29]— that highlights the timing function of basal ganglila, the cortical oscillators are assumed to increase synchrony just before movement onset and maintain the rhythm throughout the performance. The dopaminergic burst at trial onset could trigger synchronization of cortical oscillators according to SBF model [28]. Striatal neurons are tuned to respond to specific patterns of cortical oscillations [29].

SMA and basal ganglia in sequence generation Interaction between SMA and basal ganglia is believed to play a crucial role in learnt motor sequences [30]. It has been suggested that phasic activity of basal ganglia may act as a "reset" signal to the SMA clearing the traces after one movement and preparing it for the consecutive movement [31].

The above description of cortico-striatal interaction in event timing and sequence generation is much in line with the treatment of these temporal processing mechanisms in our model of handwriting generation. The timing network (basal ganglia) sends a preparatory signal to the oscillatory layer (SMA) so as to induce a stable rhythm in the latter. Once a stroke is executed, the timing network waits for a specific state in the oscillatory layer and initiates execution of the next stroke. We have seen that other ways of determining stroke onset moment yielded suboptimal results. This intricate two-way interaction between the timing network and the oscillatory layer is strongly analogous to the above description of the role of basal ganglia in timing and sequence generation.

Handwriting variability Intrinsic variability in handwriting —and in fact all motor function—is a source of difficulty in robust handwritten character recognition. Handwriting variability might seem to be a source of irritation if the goal is handwriting recognition, but one must remember that motor variability is most probably the enabling mechanism by which organisms acquire motor skills [32]. In the present work, we show that the time of stroke initiation is an important source of variability. Stroke onset must be precisely timed with respect to the evolving rhythm of the oscillatory layer. One might envisage that a lot of variability in real handwriting originates in the variability in the duration between the time of termination of one stroke *command*, and the time of initiation of the next. However, such assertions stand to be confirmed or rejected by analysis of real handwriting supported by data from underlying neural processes.

We are aware that our model has several simplifying assumptions. The input layer in our model which represents inputs from source areas of handwriting information (probably language areas in parietal cortex or dorsolateral prefrontal

cortex if the writing is driven by the contents of working memory) and the model's output layer which represents all motor areas in motor hierarchy below SMA are obviously given a summary treatment. This is because one of the key motivations of the work is to highlight the role of SMA and basal ganglia in sequential behavior, specifically handwriting. The timing network, which represents basal ganglia, is at the moment defined in terms of its inputs and outputs and not implemented as a neural network model. Further, the most important element of basal ganglia is perhaps the dopamine signal, which is thought to contain reward information, is also missing in the model. These necessities provide direction to future extensions of the biological aspect of the present model.

In the applied domain, the potential of the present model to generate synthetic handwriting can probably be exploited as a generator of "handwritten CAPTCHAs" [33]. To this end, the present model has to be trained on a large database of online cursive data as described above. The model can also be trained on data from a specific individual. However, more efficient ways of deciding stroke onset and preparing the oscillator layer have yet to be investigated.

Appendix

The proof for the system governed by the equations (i), (ii) and (iii) has a "*limit cycle*".

$$\frac{dx}{dt} = -x + v - s + I \tag{i}$$

$$v = tanh(\lambda x) \tag{ii}$$

$$\frac{ds}{dt} = -s + v \tag{iii}$$

We can use Lienard's theorem for existence of limit cycle. Follow the steps given below to convert (i), (ii) and (iii) to Lienard's system [34].

Differentiating (i)

$$\ddot{x} = -\dot{x} + \lambda \sec h^2(\lambda x)\dot{x} - \dot{s} \tag{iv}$$

Substituting (ii) and (iii) in (iv)

$$\ddot{x} = -\dot{x} + \lambda \sec h^2(\lambda x)\dot{x} - (-s + \tanh(\lambda x)) \tag{v}$$

Using (i) and (v)

$$\ddot{x} = -\dot{x} + \lambda \sec h^2(\lambda x)\dot{x} - (-(-\dot{x} - x + \tanh(\lambda x) + I) + \tanh(\lambda x))$$

On rearranging

$$\ddot{x} + \dot{x}(2 - \lambda \sec h^2(\lambda x)) + (x - I) = 0 \tag{vi}$$

is similar to Lienard's equation $\ddot{x} + \dot{x} f(x) + g(x) = 0$ where $f(x) = 2 - \lambda \sec h^2(\lambda x)$, and $g(x) = x - I$.

Checking for the Lienard's conditions: Let us assume I = 0.

Both $f(x)$ and $g(x)$ are continuously differentiable for all x;

$g(-x) = -g(x)$ for all x (i.e., $g(x)$ is an odd function);

$g(x) > 0 \quad \text{for } x > 0;$

$f(-x) = f(x)$ for all x (i.e., $f(x)$ is an even function);

The odd function $F(x) = \int_0^x f(u)du = 2x - \tanh(\lambda x)$ has exactly one positive zero at $x = x_o$, is negative for $0 < x < x_o$, is positive and non decreasing for $x > x_o$, and $F(x) \to \infty$ as $x \to \infty$. (one can estimate xo from graph of $F(x)$).

So the system has a unique stable limit cycle surrounding the origin in the phase plane.

References

1. Ellis, A.W., Modeling the writing process. In: Denes, G., Semenza, C., Bisiacchi, P., Andreewsky, E. (eds.) Perspectives in Cognitive Neuropsychology. Erlbaum, London (1986)
2. Teulings, H.L., Thomassen, A.J.W.M., Schomaker, L.R.B., Morasso, P.: Experimental protocol for cursive script acquisition: the use of motor information for the automatic recognition of cursive script. In: Report 3.1.2., ESPRIT Project, 419 (1986)
3. Schomaker, L.R.B.: Simulation and recognition of handwriting movements: a vertical approach to modeling human motor behavior. Ph.D. Thesis, Nijmegen University, Netherlands (1991)
4. Plamondon, R.: An evaluation of motor models of handwriting. IEEE Trans. Syst. Man Cybern. **19**(5), 1060–1072 (1989)
5. Grossberg, S., Paine, R.W.: A neural model of corticocerebellar interactions during attentive imitation and predictive learning of sequential handwriting movements. Neural Netw. **13**, 999–1046 (2000)
6. Hollerbach, J.M.: An oscillation theory of handwriting. Biol. Cybern. **156**(39), 139 (1981)
7. Kalveram, K.Th.: A neural oscillator model learning given trajectories, or how an allo-imitation algorithm can be implemented into a motor controller. In: Piek, J. (ed) Motor Control and Human Skill: A Multi-disciplinary Perspective. Human Kinetics, Champaign (1998)
8. Galen, G.V., Weber, J.: On-line size control in handwriting demonstrates the continuous nature of motor programs. Acta Psychol. **100**, 195–216 (1998)
9. Bullock, D., Grossberg, S.: The VITE: a neural command circuit for generating arm and articulator trajectories. In : Kelso, A., Shlesinger, M.M. (eds) Dynamic Patterns in Complex Systems. World Scientific, Singapore (1988)
10. Fiala, J., Grossberg, S., Bullock, D.: Metabotropic glutamate receptor activation in cerebellar Purkinje cells as substrate for adaptive timing of the classically conditioned eye-blink response. J. Neurosci. **16**, 3760–3774 (1996)
11. Chirikov, B.: A universal instability of many—dimensional oscillator systems. Phy. Rev. **52**, 263–379 (1979)
12. Bressloff, P.C., Coombes, S., Souza, B.: Dynamics of a ring of pulse-coupled oscillators: group theoretic approach. Phys. Rev. E Stat Nonlin Soft Matter Phys 66, (2002)
13. Haykin, S., Neural Networks: A Comprehensive Foundation, Prentice Hall PTR, Englewood Cliffs (1998)

14. Latash, M.L., Scholz, J.F., Danion, F., Schöner, G.: Structure of motor variability in marginally redundant multi-finger force production tasks. Exp. Brain Res. **141**, 153–165 (2001)
15. Marder, E., Calabrese, R.L.: Principles of rhythmic motor pattern production. Physiol. Rev. **76**, 687–717 (1996)
16. Chakravarthy, V.S., Thomas, S.T., Nair, N.: A model for scheduling motor unit recruitment in skeletal muscle. In: Proceedings of International Conference Theoretical Neurobio, Gurgoan (2003)
17. Kornhuber, H.H., Deecke, L.: Readiness for movement—the Bereitschafts potential-story. Curr. Contents Life Sci. **33**, 22 (1990)
18. Lang, W., Cheyne, R., Kristeva, R., Beistener, R., Lindinger, G., Deeke, L.: Three dimensional localization of SMA activity preceding voluntary movement: a study of electric and magnetic fields in a patient with inflation of the right supplementary motor area. Exp. Brain Res. **87**, 688–695 (1991)
19. Georgopoulos, A.P., Lurito, J.T., Petrides, M., Schwartz, A.B., Massey, J.T.: Mental rotation of the neuronal population vector. Science **243**, 234–236 (1989)
20. Kubota, K., Hamada, I.: Visual tracking and neuron activity in the post—arcurate area in monkeys. J. Physiol. Paris. **74**, 297–312 (1978)
21. Kubota, K., Funahashi, S.: Neuron activities of monkey prefrontal cortex during the learning of visual discriminations tasks with go/no-go performances. Neurosci. Res. **3**, 106–129 (1982)
22. Crammond, D.J., Kalaska, J.F.: Neuronal activity in primate parietal cortex area 5 varies with intended movement direction during an instructed—delay period. Exp. Brain Res. **76**, 458–462 (1989)
23. Alexander, G.E.: Selective neuronal discharge in monkey putamen reflects intended direction of planned limb movements. Exp. Brain Res. **67**, 623–634 (1987)
24. Churchland, M.M., Yu, B.M., Ryu, S.I., Santhanam, G., Shenoy, K.: Neural variability in premotor cortex provides a signature of motor preparation. J. Neurosci. **26**(14), 3697–3712 (2006)
25. Tanji, J.: The supplementary motor area in the cerebral cortex. Neurosci. Res. **19**(3), 251–268 (1994)
26. Tanji, J., Taniguchi, K., Saga, T.: Supplementary motor area: neuronal response to motor instructions. J. Neurophysiol. **43**, 60–68 (1980)
27. Aizawa, H., Tanji, J.: Cortico-cortical and thalamo-cortical responses of neurons in the monkey primary motor cortex and their relation to a trained motor task. J. Neurophysiol. **71**, 550–560 (1994)
28. Buhusi, C.V., Meck, W.H.: What makes us tick? Functional and neural mechanisms of interval timing. Nat. Rev. Neurosci. **6**, 755 (2005)
29. Matell, M.S., Meck, W.H.: Cortico-striatal circuits and interval timing: coincidence detection of oscillatory processes. Cogn. Brain Res. **21**, 139–170 (2004)
30. Cunnington, R., Iansek, R., Bradshaw, J.A., Phillips, J.G.: Movement-related potentials in Parkinson's disease. Presence and predictability of temporal and spatial cues. Brain **118**, 935–950 (1995)
31. Georgiou, N., Iansek, R., Bradshaw, J.L., Phillips, J.G., Mattingley, J.B.: An evaluation of the role of internal cues in the pathogenesis of Parkinsonian hypokinesia. Brain **116**, 1575–1587 (1993)
32. Kao, M.H., Doupe, A.J., Brainard, M.S.: Contributions of an avian basal ganglia-forebrain circuit to real-time modulation of song. Nature **433**(7026), 638–643 (2005)
33. Rusu, A., Govindaraju, V.: Handwritten CAPTCHA: using the difference in the abilities of humans and machines in reading handwritten words, IWFHR, pp. 226–231. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR'04) (2004)
34. Perko, L.: Differential Equations and Dynamical Systems. Springer, Heidelberg (2006)