Neural mechanisms of sensorimotor behavior: Simulations with dynamic network models

E. E. Fetz

Department of Physiology and Biophysics and Regional Primate Research Center, University of Washington, Seattle WA 98195, USA

Introduction

The goal of systems neurophysiologists is to understand the neural mechanisms underlying behavior. Traditional approaches toward this goal have involved anatomical and physiological techniques that reveal the connectivity or activity of specific components of the system. Yet, despite innumerable studies documenting how lesions affect behavior, how stimulation evokes movement, and how single units fire in behaving animals, we still have no consensus about how neural networks actually generate movement. The basic reason that the experimental data has not yet produced a definitive causal explanation of motor behavior is clear: the traditional techniques inevitably provide only limited samples of the system. For example, recordings of single unit activity indicate their response patterns, but show nothing about their relation to the rest of the system. What is missing is what neural network models can provide: a method of generating working models of the complete system.

Two general approaches to neural network modelling should be distinguished: the "bottom-up" approach involves synthesizing experimental details about cellular mechanisms into a working network model. This approach appeals to physiologists concerned with preserving biological reality, who wish to demonstrate how physiological neurons could function when combined into circuits; however, the bottom-up approach runs into problems dealing with the whole organism, since many parameters of the nervous system are largely unknown and must be arbitrarily assigned. Moreover, modelling the whole system with "realistic" neurons soon becomes prohibitively complex. Alternatively, recent advances in neural network algorithms now make it possible to derive "top-down" simulations of sensorimotor behavior in networks of appropriately chosen connectivity. The latter techniques provide simplified neural networks that perform specified behaviors, derived entirely on the basis of examples of the behavior itself.

The first examples of such top-down simulations involved feed-forward networks trained with back-propagation to transform spatial patterns. For example, feed-forward networks have been trained to classify the shape of a surface from shaded images [1]. Another such network combined the retinal coordinates of a visual stimulus with eye displacement to provide a code for the position of the stimulus in

head coordinates [2]. Interestingly, the hidden units in such networks often have response properties resembling those seen in biological neurons recorded in animals. The shape-from-shading networks developed hidden units with center-surround receptive fields like visual cortex cells, even though the simulated task did not explicitly require such properties or present stimuli with edges. The reasons that units in network simulations of a task often have such close correspondence with units in biological systems may be related to basic properties of identification models, as discussed by Zipser [3].

Supervised training procedures have now been extended beyond feed-forward networks, which transform spatial patterns through serial layers, to dynamic recurrent networks which can deal with spatiotemporal patterns and have feedback connections [5–9]. Results obtained from top-down simulations of behavior with dynamic recurrent networks are the subject of a recent review [4].

Three key properties distinguish dynamic recurrent networks from other modelling approaches. First, the units are *dynamic*, meaning they can exhibit time-varying activity. The activation of each unit is a temporal variable, which can represent the firing rate of neurons, their synaptic potentials or some relevant time-varying sensory or motor parameters. This means that the models can incorporate physiologically recorded activity as activation patterns that the network can be required to generate. Second, the networks have a *recurrent* architecture, which allows unrestricted connectivity. In addition to feed-forward connections between successive layers of units, the networks can have feedback and cross-connections. Thus, the models can incorporate recurrent anatomical pathways. Third, the networks that simulate a particular sensorimotor behavior can be derived from examples of the behavior by *gradient descent* methods such as back-propagated error correction. The resulting network models provide complete neural network solutions of the behavior, insofar as they determine all the connections and activations of the units that simulate the behavior.

Neural networks that emulate particular sensorimotor behaviors – i.e., that transform spatiotemporal inputs to appropriate outputs – can be derived by various training strategies. These networks are usually comprised of so-called "sigmoidal" or "logistic" units, named for the fact that the unit's output is a sigmoidal function of its input. The unit's input consists of the summed activation of all other cells connected to that unit times their synaptic weight; a steady input may also be provided by a bias element. The sigmoidal input-output function mimics a biological neuron's property of saturating at maximal rates for high levels of input and decreasing to zero at low input levels.

To train the network to transform a particular input pattern to a desired output pattern, the synaptic weights are initially assigned randomly, and the output response of the network is determined; the difference between network output patterns and the desired target output activations defines the error. The back-propagation algorithm modifies the weights in such a way as to optimally reduce this error. This weight change implements a "gradient descent" of the error as a function of the weight. After each weight correction the input patterns are presented again and the new outputs

determined. This process of changing the weights in proportion to the error gradient is repeated iteratively until the network converges on a solution with minimal error. Several types of algorithms have been used to implement such gradient descent methods for dynamic recurrent networks [5–9]. A comparison of various training methods for recurrent networks is presented by Williams and Zipser [9].

The applications for these dynamic recurrent networks can be classified into three broad categories:

A. Pattern recognition applications involve identification of spatiotemporal input patterns into discrete categories. A set of input units receiving time-varying signals can be considered to represent a spatiotemporal pattern. Pattern recognition applications include speech recognition [7], and conditional delay tasks [4,8].

B. Pattern generation networks produce temporal patterns in one or more output units, either autonomously or under the control of a gating input. These include oscillating networks [3,4,8] and simulations of central pattern generators [10,11].

C. Pattern transformation networks convert spatiotemporal input patterns into spatiotemporal outputs. Examples include simulations of the leech withdrawal reflex [12], step target tracking in the primate [4,13], the vestibulo-ocular reflex (VOR) [14–16] and short-term memory tasks [4,17].

Primate target tracking

Our own interest in dynamic recurrent networks derived from a desire to understand the neural circuitry controlling forelimb muscles of the primate. In monkeys performing a step-tracking task, previous experiments have documented the physiological discharge patterns and output connections of task-related neurons [18]. The premotoneuronal (PreM) cells which affect muscle activity were identified in behaving monkeys by post-spike facilitation of target muscles in spike-triggered averages of EMG recordings. The response patterns of three groups of PreM cells – corticomotoneuronal (CM), rubromotoneuronal (RM) and dorsal root ganglion afferents – have been documented during a simple alternating flexion/extension task designed to relate their activity to changing and sustained force [18,22]. The discharge patterns observed in PreM cells, as well as in single motor units (MU) of agonist muscles [19], fall into specific classes. During a ramp-and-hold movement, all three groups include cells that show phasic-tonic discharge. The phasic component is related to the changing force and the tonic component is proportional to the amount of static force exerted. All groups also include tonic cells which show steady discharge throughout the hold period in proportion to the active force. Each group also has cells with unique firing properties. A significant proportion of motor units show decrementing discharge, which decreases gradually through the static hold period. The RM population has cells that fire during both flexion and extension, and some cells that are unmodulated with the task and provide a constant bias.

To investigate the possible functional role of these cells and to determine what other types of discharge patterns might be required to transform a step signal to the
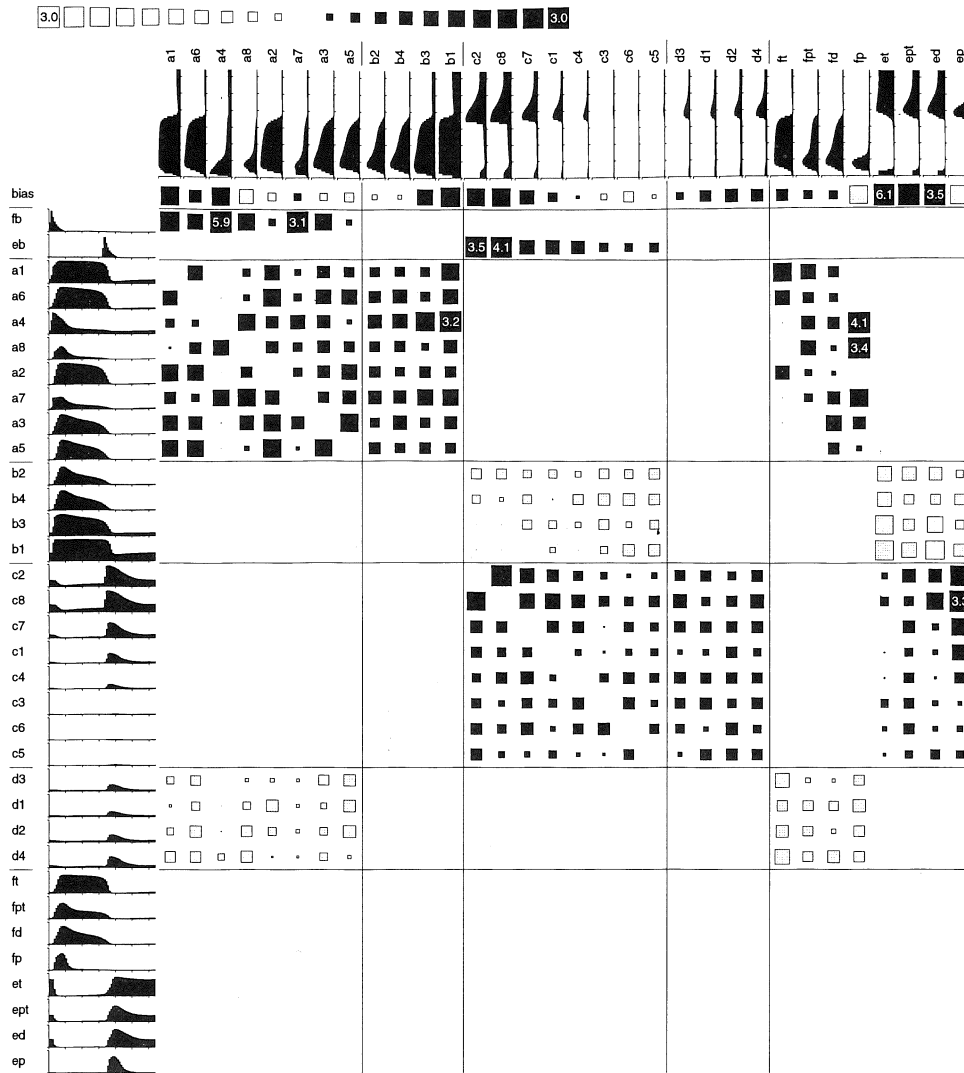
Fig. 1. Dynamic network simulating step-tracking task. The network transforms transient inputs to firing patterns of motor units at output. Unit activations during a flexion-extension cycle are shown at left and along the top. The rows represent, from top to bottom, the bias (a constant value of 1, not illustrated), the input signals (fb and eb), the excitatory hidden units (a and c), inhibitory hidden units (b and d) and the output flexor and extensor motor units (ft–ep). Squares represent strength of connection from unit at left to unit at top. Values greater than 3 (range of scale at top) are given numerically. Excitatory connections are filled, inhibitory open. Hidden units were sorted in order of contribution to flexion vs. extension outputs [unpublished simulation by Shupe, Konodi & Fetz].

observed output of motoneurons, we derived dynamic networks that generated as outputs the average firing rates of motor units recorded in monkeys performing the step-tracking task [13]. The inputs were representations of the visual targets, either

steps or transients or both. The "hidden units" mediating the transform of input to output consisted of excitatory and inhibitory units ubiquitously connected to each other and to the output. This relatively unstructured architecture was modified by training, as illustrated by previously published networks [4,13]. The post-training connectivity matrix reveals a network structure that preferentially routes the activity of phasic and tonic input signals to the appropriate output and sculpts the response patterns. We expected such networks to develop reciprocal inhibition between the flexion and extension groups, but found that many units inhibit coactivated motor units. In general, different network simulations with the same architecture and trained on the same task but starting with different initial weights, usually converged on different solutions, suggesting that additional constraints could be imposed to obtain more unique solutions.

Figure 1 illustrates a more constrained neural network which simulates the step-tracking task; it shows the activation patterns and the connection matrix of all units after the network produces the eight different output patterns in response to only transient inputs. The discharge pattern of each unit is shown along the left, next to the row of output weights of that unit, and is shown again at the top of the column of weights representing the input connections to that unit. Self-recurrent connections (corresponding to weights on the diagonal) were excluded. The connectivity of this network was constrained to separate flexor and extensor circuits and to include reciprocal inhibitory connections. The step change in target position is represented by a brief transient input at the onset of each target change (fb and eb). The network transforms these input signals to the response patterns of eight motor units at the output. The four types of motor unit patterns observed experimentally – tonic, phasic-tonic, decrementing and phasic – are generated for both flexor and extensor movements (ft–ep). The intervening hidden units consist of excitatory and inhibitory units configured with reciprocal connections. The transient flexor signal activates eight excitatory units (a1–a8) which are recurrently interconnected, and which also excite four inhibitory units (b1–b4), and the four flexor output units (ft–fp). An equivalent set of units exists for the extensor input and output. The inhibitory units of each set are reciprocally cross-connected to the excitatory units of the opposite set.

To better visualize the relationships between units, the hidden units were sorted in order of the strength of their contribution to the phasic-tonic output units. Thus, the first hidden unit (a1) makes the largest relative contribution to the flexion phasic-tonic output unit (fpt). This hidden unit developed the strongest weights to the flexion tonic output unit (ft), but was also connected to other flexor units. Such divergent connections to different motor units, as well as to synergist muscles are representative of CM cells [18].

The activation patterns of the hidden units show several interesting features. The transient flexor input is converted to a sustained tonic activity by recurrent connections between excitatory units (e.g., a1, a2 and a8). This strategy is typical of neural integrators [20]. The discharge patterns in the hidden units involve some recognizable variants of the output patterns, i.e., tonic, phasic, phasic-tonic, and decrementing patterns. Secondly, although the activation profiles of the flexor and extensor target
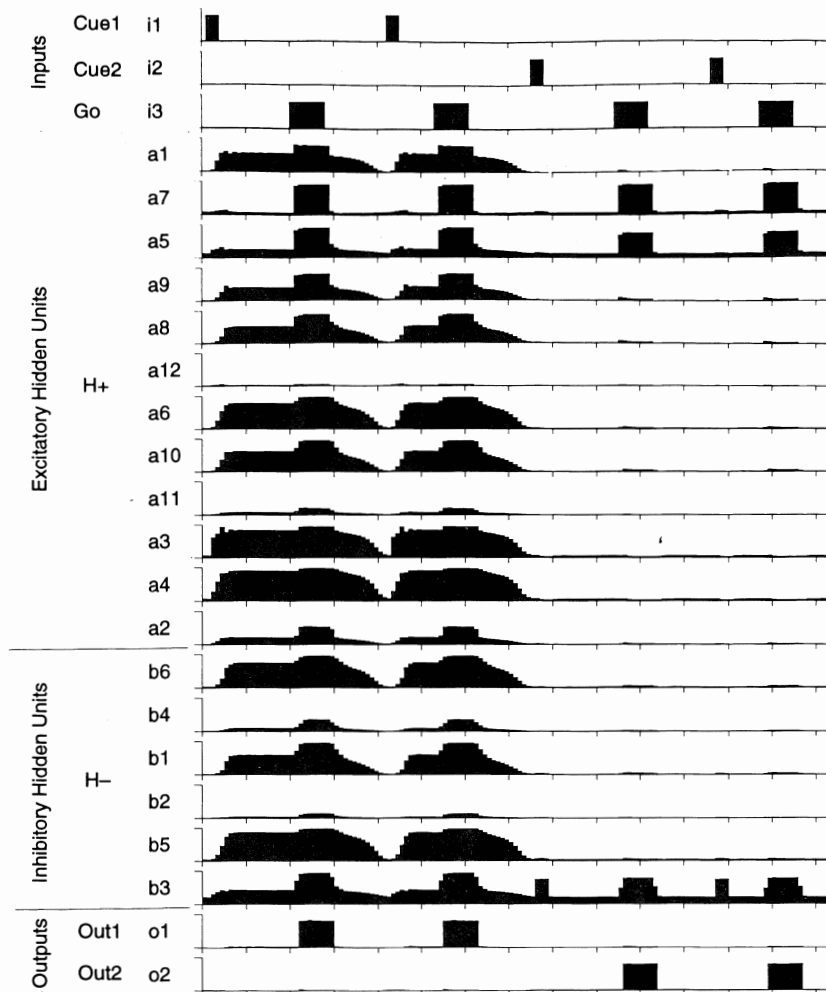
280

Fig. 2. Activity patterns of representative units in a network simulating the instructed delay task. Input pulses on inputs i1 and i2 are cues for the network to generate outputs o1 and o2, respectively when the go signal i3 is presented after a delay [Fetz & Shupe, unpublished].

motor units are the same, the network solution is different for each: the tonic flexor outputs are derived by integrating the transient input; the tonic extensor activity is derived from the bias minus input from the reciprocal inhibitory units active with flexion (b's).

These simulations are obviously too simplistic to be taken as realistic models of the primate motor system. Nevertheless, even these highly abstracted networks reflect some of the features of the biological system in the monkey. For example, many different combinations of hidden units contribute to the same output. Furthermore, some complex activity patterns seen in premotor neurons of monkeys, such as bidirectional responses of RM cells, also appear in the networks. Even certain

apparently paradoxical relations seen in monkeys, such as cortical units that covary with muscles which they inhibit, appear in networks and make contributions that are understandable in terms of other units [4,13,22]. Thus, the networks have been useful in elucidating the function of many puzzling features of biological networks.

Short-term memory tasks

Neural mechanisms of short-term memory have been investigated in many experiments by recording cortical cell activity in animals performing instructed delay tasks. One variant of these tasks begins with an instruction cue that the monkey has to remember during a delay period. At the end of the delay a go stimulus signals the time to execute the appropriate response indicated by the cue. A simulation of this paradigm in a dynamic recurrent network is illustrated in Fig. 2. The network receives three inputs, consisting of the two separate cues and the go signal; the network has two outputs, corresponding to the response appropriate for each cue. Thus, following cue 1 (or 2) the go signal will generate activation of output unit 1 (or 2). Activations of the units in a trained network for several trials involving different delays are illustrated in Fig. 2. The connectivity matrix shows that the hidden units are configured to form a flip-flop that routes the go signal to the appropriate output. This network simulates the operation of short-term memory during the delay; some hidden unit activations (e.g., b3) carry mixed representations of cue and movement signals, which have been seen in premotor and prefrontal cortex neurons.

Another type of instructed delay task involves the requirement to remember the value of a particular stimulus, which can range over a continuum of values. Zipser [17] trained recurrent networks to simulate short-term memory of an analog value during the delay; in essence the network implements a sample-and-hold function. The network has two inputs: an analog signal representing the stimulus value to be remembered and a gate signal specifying the sample and output times. The network output is the sustained value of the analog input at the time of the previous gate. During the delay the activity of many hidden units resembles the response patterns of cortical neurons recorded in monkeys performing comparable instructed delay tasks [17,21]. The activity patterns of hidden units, like those of cortical neurons, fall into three main classes: activation profiles proportional to the remembered analog value, often with a delay, activation during the gate signal, and some combination of the two. The network simulations allow the function of each of these patterns observed in the animal to be interpreted in terms of its role in the task.

We have also investigated such short-term memory networks in order to further analyze their operation. To elucidate the underlying computational algorithm we have found it useful to constrain units to have either excitatory or inhibitory output weights (not both), and to reduce the network to the minimal essential network that performs the same function. A reduced network performing the sample-and-hold function with sign constraints on the weights is illustrated in Fig. 3. It consists of three

SA M1 M2 SM O

Reduced
Sample-
and-hold
Network

bias

S | 5.5

A

SA | 5.4

M1

M2

SM | 5.5

O
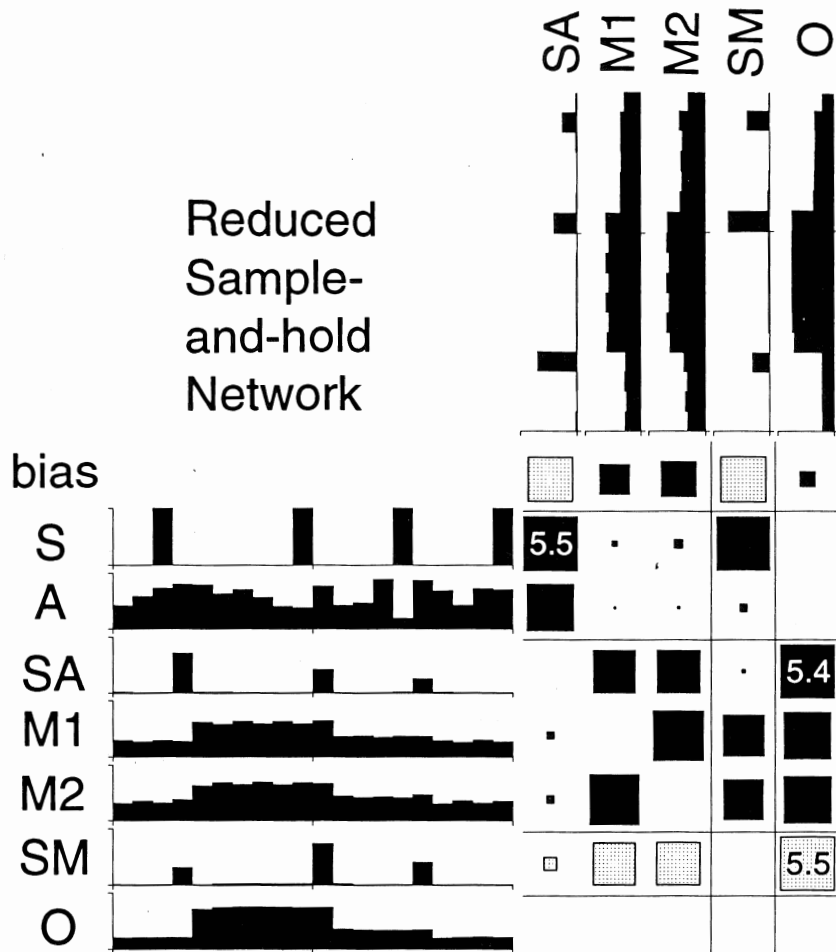
Fig. 3. Reduced network performing a sample-and-hold function, simulating short-term memory. A larger network with weight constraints was originally trained on the task, then reduced to the essential minimum by eliminating redundant and unnecessary hidden units. The two inputs are the sample signal (S) and a random analog value (A); the output (O) is the sustained value of the last sampled analog value [Fetz & Shupe, unpublished].

excitatory and one inhibitory unit. The two inputs are the gate signal to take a sample (S) and the analog variable (A); the output (O) is the value of A at the last sample gate. This reduced version reveals an elegant computational algorithm that exploits the non-linear sigmoidal input-output function of the units. The first excitatory unit (SA) carries a transient signal proportional to the value of A at the time of the gate. This signal is derived by clipping the sum of the analog and gating inputs with a negative bias, as shown by the input weights to SA in the first column. This input sample is then held in memory by two excitatory units (M1 and M2) that maintain their activity by reciprocal connections and which feed their summed activity to the

output. The inhibitory unit (SM) carries a transient signal proportional to the previous value of A. Its value is derived from a clipped sum of the gate S and the previous values held in M1 and M2. The function of SM is to subtract the previously held value from the integrating hidden units and from the output. Thus, the network reveals an elegant use of non-linearity and integration to yield the appropriate remembered value. Networks with more units may well implement a comparable algorithm in a distributed manner.

Models of motor learning

Lisberger and Sejnowski [16] used dynamic networks to investigate mechanisms of motor learning in the vestibulo-ocular system. The network was constructed to include many anatomical and physiological constraints, including pathways through the cerebellar flocculus. The two inputs to the network, head velocity and target velocity, were converted to one output: eye velocity. The network was initially trained to simulate three behaviors: smooth pursuit of a moving visual target, the VOR to head movement, and suppression of the VOR (when head and target move together). Then the network was required to change the gain of the VOR, as occurs after wearing magnifying or minifying goggles, and also to maintain accurate smooth pursuit visual tracking. These requirements led to changes in the weights of connections at two specific sites: the vestibular input to the flocculus and to the brainstem neurons controlling oculomotoneurons. A change in the balance of input from phasic-tonic and tonic vestibular afferents contributed to a change in the gain of the VOR, because the transient signal was integrated by a positive feedback loop [16]. This study exemplifies the usefulness of a biologically constrained dynamic model that can incorporate the time course of neural activity observed under different behavioral conditions, and shows the power of such simulations to reveal novel network mechanisms.

Conclusion

The unique insights provided by neural network simulations assures their continued use in elucidating the operations of neural systems. The basic limitation of conventional physiological and anatomical data is that they provide a highly selective sample of a complex system, leaving a wide gap between particular glimpses of neural activity or anatomical structure, and the behavior of the overall system. This gap is usually bridged by intuitive inferences, often based on selective interpretation of the data [22]. A more objective method of bridging the gap is with models that provide complete neural networks that simulate the behavior. Dynamic models can incorporate the observed response patterns of neurons and can help explain their functional meaning. Thus, systems neurophysiologists can profitably use a combination of unit recording techniques and neural modeling to elucidate the network

mechanisms generating sensorimotor behavior. Unit recordings can provide important information on the activity of related neurons, but the network models can provide working examples of complete solutions simulating the behavior. To the extent that models can incorporate anatomical and physiological constraints, they can provide plausible explanations of the neural mechanisms underlying behavior.

Acknowledgements

References

1. Lehky SR and Sejnowski TJ. Proc R Soc Lond 1990; B240:251–278.
2. Zipser D and Andersen RA. Nature 1988; 331:679–684.
3. Zipser D. Neurosci 1992; 47:853–862.
4. Fetz EE. In: Gardner D. (ed.) The Neurobiology of Neural Networks, Cambridge, MA: MIT Press; in press.
5. Pearlmutter BA. Neural Comp 1989; 1:263–269.
6. Pineda FJ. Neural Comp 1989; 161–172.
7. Watrous RL and Shastri L. Tech Report MS-CIS-86-78, Linc Lab 44, U Penn, Phila, PA. 1986.
8. Williams RJ and Zipser D. Neural Comp 1989; 1:270–280.
9. Williams RJ and Zipser D. Tech Report NU-CCS-90-9, College of Computer Science, Northeastern University, Boston, MA. 1990.
10. Rowat PF and Selverston AI. Network 1991; 2:17–41.
11. Tsung F-S, Cottrell GW and Selverston AI. Proc IJCNN-90, 1990; I:169–174.
12. Lockery SR and Sejnowski TJ. J Neurosci 1992; 12:3877–3895.
13. Fetz EE and Shupe LE. In: Eckmiller R (ed.) Advanced Neural Computers Amsterdam: Elsevier, 1990; 43–50.
14. Anastasio TJ. Biol Cybernetics 1991; 64:187–196.
15. Arnold DB and Robinson DA. Biol Cybernetics 1991; 64:447–454.
16. Lisberger SG and Sejnowski TJ. Tech Report INC-9201, Inst for Neural Comp, U.C. San Diego. 1992.
17. Zipser D. Neural Comp 1991; 3:179–193.
18. Fetz EE, Cheney PD, Mewes K and Palmer S. Prog Brain Res 1989; 80:437–449.
19. Palmer SS and Fetz EE. J Neurophysiol 1985; 54:1178–1193.
20. Munro E, Shupe L and Fetz E. Neural Comp (In press).
21. Fuster J. Trends Neurosci 1984; 7:408-414.
22. Fetz EE. Behavioral and Brain Sciences (In press).