Sequential State Generation by Model Neural Networks

David Kleinfeld

Proceedings of the National Academy of Sciences of the United States of America, Volume 83, Issue 24 (Dec. 15, 1986), 9469-9473.

Proc. Natl. Acad. Sci. USA
Vol. 83, pp. 9469–9473, December 1986
Biophysics

Sequential state generation by model neural networks

(associative memory/central pattern generators/feedback/synaptic connections/time delays)

DAVID KLEINFELD

Department of Molecular Biophysics, AT&T Bell Laboratories, Murray Hill, NJ 07974

Communicated by F. A. Bovey, September 3, 1986

ABSTRACT **Sequential patterns of neural output activity form the basis of many biological processes, such as the cyclic pattern of outputs that control locomotion. I show how such sequences can be generated by a class of model neural networks that make defined sets of transitions between selected memory states. Sequence-generating networks depend upon the interplay between two sets of synaptic connections. One set acts to stabilize the network in its current memory state, while the second set, whose action is delayed in time, causes the network to make specified transitions between the memories. The dynamic properties of these networks are described in terms of motion along an energy surface. The performance of the networks, both with intact connections and with noisy or missing connections, is illustrated by numerical examples. In addition, I present a scheme for the recognition of externally generated sequences by these networks.**

Cyclic patterns of motor neuron activity are involved in the production of many rhythmic movements, such as locomotion. The neural circuits that control the muscles involved in executing the movements are typically referred to as central pattern generators (CPGs) (for reviews see refs. 1–3). A common feature of CPGs is that sequence generation can occur in the absence of both sensory feedback and feedback from other neural centers. A second feature of some CPGs is that they do not contain intrinsic pacemakers. Thus the cyclic activity, and the timing of this activity, is a collective property of the CPG.

In this paper I present a formal model of sequence generation in the context of networks of model neurons, such as those discussed by Hopfield (4, 5) (see also refs. 6–13). Hopfield networks use highly interconnected model neurons to perform complex computational tasks. These tasks, such as recalling information by association (4, 9, 13) or solving optimization problems (14), involve networks that seek a single, stationary state as their output. The sequential state generators that I describe do not come to rest in a single state. Rather they make transitions between selected states and thus generate sequential patterns of bit sequences. Sequence-generating networks provide a formal setting for understanding some CPGs. They also provide a basis for the sequential recall of information in associative memories.

THEORY

Background. As an antecedent to understanding sequence generation, we consider a network consisting of N model neurons that functions as an associative memory (inner circuit, Fig. 1) (4, 5). Neurons are represented by nonlinear amplifiers whose output, $V_i(t)$, saturates when the input, $u_i(t)$, exceeds a threshold value. The detailed form of this behavior does not affect the function of these networks. We take

$$V_i(t) = \begin{cases} +1 & \text{if } u_i(t) \geqslant 1/\beta \\ \beta \cdot u_i(t) & \text{if } -1/\beta < u_i(t) < 1/\beta, \\ -1 & \text{if } u_i(t) \leqslant -1/\beta \end{cases} \quad [1]$$

where β is the gain of the neuron in the linear operating region.

A state of the network at time t, $\mathbf{V}(t)$, is defined by a sequence of N numbers that specify the value of the output of each neuron. Under saturating conditions, neurons are either fully active (+1) or quiescent (−1) and $\mathbf{V}(t)$ is defined by a binary sequence. There are 2^N possible binary sequences. Memories, $\mathbf{M}^\nu$, are represented by specific states chosen from the 2^N possibilities. For example, $\mathbf{M}^\nu = (+1\ -1\ -1\ \ldots)$ implies that neuron 1 is active, neurons 2 and 3 are quiescent, etc.

Each model neuron receives input from all other neurons via a set of pairwise synaptic connections (Fig. 1). The network functions as an associative memory when the strength of each synapse is determined from the memories according to the outer-product rule (e.g., ref. 9):

$$T_{ij} = \frac{1}{N}\sum_{\nu=1}^{n} M_i^\nu M_j^\nu \quad (i \neq j), \quad [2]$$

where n is the number of stored memories and $T_{ii} = 0$. The T matrix is essentially a sum of projection operators that map the present state of the network to the closest memory—i.e., the memory with the least number of differing bits. The output of the network will converge from an arbitrary initial state to a stationary memory according to the circuit equations (5) (Fig. 2):

$$u_i(t) + \tau_{Si}\frac{du_i(t)}{dt} = \sum_{j=1}^{N} T_{ij}V_j(t), \quad [3]$$

where $\tau_{Si} = R_{Si}C_{Si}$ is the charging time of the ith neuron and R_{Si} is related to the input resistance of the neuron, r_{Si}, and the synaptic resistances t_{ij} and d_{ij} (next section) by

$$1/R_{Si} = 1/r_{Si} + \sum_{j=1}^{N} 1/t_{ij} + \sum_{j=1}^{N} 1/d_{ij}. \quad [4]$$

The magnitudes of the synaptic strengths T_{ij} are given by $|T_{ij}| = R_{Si}/t_{ij}$, with negative values of T_{ij} obtained by using the inverted value of the output, $-V_j(t)$ (broken lines in Fig. 1).

Sequence Generation. Projection operators that map the state of the network from one memory to the next—i.e., terms of the form $M_i^{\nu+1} M_j^\nu$—should generate transitions

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

Abbreviation: CPG, central pattern generator.

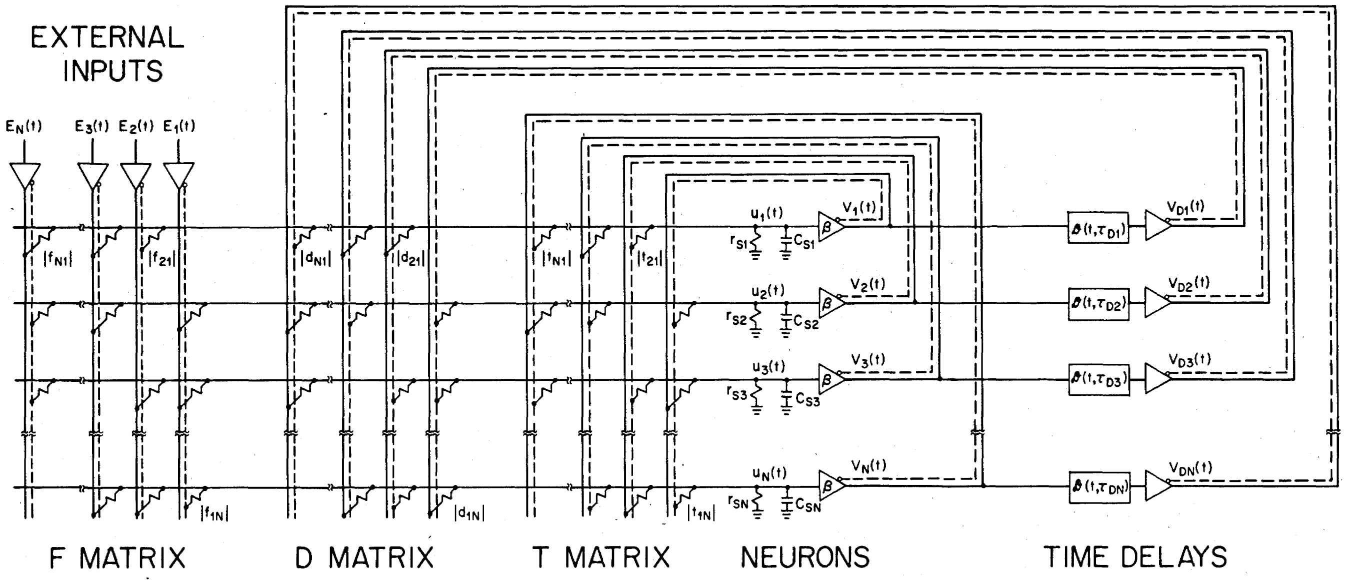

FIG. 1. Diagram of the neural network sequence generator and related circuits. The ith neuron is modeled by a nonlinear amplifier (triangle labeled β) that buffers a capacitive charging circuit; there are N neurons. Broken lines correspond to inverted output signals. The inner circuit, consisting of the neurons and the synaptic strengths $T_{ij} \propto 1/t_{ij}$, constitutes an associative memory. The addition of time delays, $\mathcal{D}(t, \tau_{Di})$, linear buffering amplifiers (open triangles), and the synaptic strengths $D_{ij} \propto 1/d_{ij}$ forms the sequence generator. Sequence recognition is performed on the external inputs, $E_i(t)$, multiplied by the synaptic strengths $F_{ij} \propto 1/f_{ij}$.

from the νth to the $(\nu + 1)$th memories (4, 15, 16). However, when terms of this form were added to the T matrix, only sequences of limited length were achieved (4). The difficulty with this construction is that the network does not become stationary in one memory before the transition to the next memory begins. This causes the state of the network to become progressively mixed among all the memories making up a sequence, which makes the generation of arbitrarily long linear sequences, or cyclic sequences, unattainable.

The essential requirement for sequence generation is to allow the network to become stationary in one memory before inducing a transition to another memory. The transitions between memories can be delayed by having the input to each neuron depend upon the history of the network via a set of delayed output states, $V_{Dj}(t)$. These states are defined as the convolution of $V_j(t)$ with a (normalized) delay function, $\mathcal{D}(t, \tau_{Dj})$—i.e.,

$$V_{Dj}(t) = \int_0^\infty V_j(t - x)\mathcal{D}(x, \tau_{Dj})dx. \quad [5]$$

The delay time, τ_{Dj}, must be long compared to the charging time, τ_{Sj}. For the propagation delays inherent in nondispersive cables or with active propagation along axons, $\mathcal{D}(t, \tau_{Dj}) = \delta(t - \tau_{Dj})$ and $V_{Dj}(t) = V_j(t - \tau_{Dj})$. For the delays associated with charging a capacitor, $\mathcal{D}(t, \tau_{Dj}) = e^{-t/\tau_{Dj}}/\tau_{Dj}$.

The delayed outputs form a second set of inputs to the neurons via the synaptic connections D_{ij} (Fig. 1). The strengths of these connections are chosen to cause transitions through a sequence of memories. For each independent sequence of length m, $m \leq n$:

$$D_{ij} = \frac{1}{N}\sum_{\nu=1}^{m} M_i^{\nu+1} M_j^{\nu} \quad (i \neq j) \quad [6]$$

with $D_{ii} = 0$. The magnitudes of the strengths D_{ij} are related to the resistances d_{ij} and R_{Si} by $|D_{ij}| = R_{Si}/d_{ij}$. To generate cyclic sequences the sum in Eq. **6** is performed modulo m. The circuit equations describing the dynamics of the sequence generator are

$$u_i(t) + \tau_{Si}\frac{du_i(t)}{dt} = \sum_{j=1}^{N} T_{ij}V_j(t) + \sum_{j=1}^{N} D_{ij}V_{Dj}(t). \quad [7]$$

Binary Limit with Delta Function Delay. The model discussed above uses neurons with a graded input-output relation (Eq. **1**). It is useful to consider the "binary" limit of this model—i.e., $\beta \to \infty$ and $\tau_{Si} \to 0$—with the additional constraint $\mathcal{D}(t, \tau_{Dj}) = \delta(t - \tau_D)$. This case simplifies the analysis of the sequence generator without changing the essential features of its dynamics. The input to the ith neuron is now

$$u_i(k) = \sum_{j=1}^{N} T_{ij}V_j(k) + \sum_{j=1}^{N} D_{ij}V_j(k - \kappa_D), \quad [8]$$

where the time step k approximates the charging time, τ_S. The delay interval, κ_D, is taken to be the same for all neurons and is equivalent to $\kappa_D \approx \tau_D/\tau_S$. The outputs $V_i(k)$ are updated to either $+1$ or -1 depending upon the sign of $u_i(k)$ (Eq. **1** with $\beta \to \infty$). The updating is performed asynchronously—i.e., the value of the index i is chosen at random to simulate variations in τ_{Si} between neurons.

Sequence Dynamics. The dynamic properties of the sequence generator can be qualitatively understood by examining the input to each neuron as a function of $\mathbf{V}(k)$ and $\mathbf{V}(k - \kappa_D)$. To place this analysis in a physical setting, we illustrate the dynamics in terms of motion on a quadratic, "energy," surface defined by

$$E = -\left(\sum_{i,j}^{N} V_i(k)\, T_{ij}V_j(k) + \sum_{i,j}^{N} V_i(k)D_{ij}V_j(k - \kappa_D)\right). \quad [9]$$

Fig. 2 shows contour plot of the energy (E) as a function of $\mathbf{V}(k)$ and $\mathbf{V}(k - \kappa_D)$ for the case of orthogonal memories—i.e.,

$$\frac{1}{N}\sum_{j=1}^{N} M_j^{\nu} M_j^{\nu'} = \delta(\nu - \nu'). \quad [10]$$

Memories appear as minima, or bowls, with $E = -2$. They are separated by relatively broad saddles with $E \approx -1$.

We consider a network soon after it has settled into the νth memory—i.e., $\mathbf{V}(k) = \mathbf{M}^\nu$ (point a, Fig. 2). For a time short compared to κ_D, the value of the delayed state is $\mathbf{V}(k - \kappa_D) = \mathbf{M}^{\nu-1}$. The energy of the network is presently at a minimum and the input to the ith neuron is (Eqs. **8** and **10**):

$$u_i(k) = \sum_{j=1}^{N} T_{ij}M_j^{\nu} + \sum_{j=1}^{N} D_{ij}M_j^{\nu-1} \approx 2 \cdot M_i^{\nu}. \quad [11]$$

Contributions from both the current state and the delayed state thus stabilize the network in the νth memory.

After the network has remained stationary in the νth memory for an interval $\sim \kappa_D$, the value of $V(k - \kappa_D)$ shifts toward M^{ν} (path *a–b*, Fig. 2). This change increases the energy of the network such that it moves towards the top of a bowl in the energy surface (point *b*, Fig. 2). When $V(k - \kappa_D)$ finally equals M^{ν}, the input to the ith neurons becomes

$$u_i(k) = \sum_{j=1}^{N} T_{ij}M_j^{\nu} + \sum_{j=1}^{N} D_{ij}M_j^{\nu} \approx M_i^{\nu} + M_i^{\nu+1}. \quad [12]$$

These new values for $u_i(k)$ place the network in a mixed, nonstationary state from which it can make the transition from M^{ν} to $M^{\nu+1}$. The transition corresponds to moving across a saddle region on the energy surface (path *b–c*, Fig. 2) and falling into the adjacent bowl (path *c–d*, Fig. 2). The network will remain in the new memory for an interval $\sim \kappa_D$, after which it will make a transition to the next memory in the sequence, etc.

Sequence Recognition. An extension of the sequence generator can be used to recognize external sequences (Fig. 1). Recognition is defined as a completed set of transitions of the sequence generator in response to a particular external input, $E(k)$ (17). We require that the sequence of states, L^{ν}, composing the external input can be mapped onto the sequence of memories used to form the **D** matrix (Eq. **6**). A set of synaptic connections that produce this mapping are

$$F_{ij} = \frac{1}{N}\sum_{\nu=1}^{m} M_i^{\nu}L_j^{\nu} \quad (i \neq j), \quad [13]$$

where F_{ij} connects the ith component of the external input to the jth neuron, and $F_{ii} = 0$. The external input is clocked with an average period of κ_E. The circuit equations for the recognition network, shown in Fig. 1, are

$$u_i(k) = \sum_{j=1}^{N} T_{ij}V_j(k) + \lambda \sum_{j=1}^{N} D_{ij}V_j(k - \kappa_D) + \varepsilon \sum_{j=1}^{N} F_{ij}E_j(k). \quad [14]$$

The coefficient λ is adjusted so that the contribution to $u_i(k)$ from the delayed states alone is insufficient to cause transitions between memories. Sequence generation is now possible only when the external input can assist in driving the transitions. Thus, a completed set of transitions occurs when the externally generated sequence of states is equivalent to the internal sequence of memories (Eq. **12**). This set is completed in a period $\Delta k = m \cdot \kappa_E$ with cyclic sequences. In this scheme, ε must be large enough to elicit transitions but small enough so that $E(k)$ does not dominate the inputs to the individual neurons. The external period must be longer than the delay time—i.e., $\kappa_E > \kappa_D$.

The recognition scheme can be interpreted in terms of the energy diagram of Fig. 2. With $\lambda << 1$, inputs from the delayed state can force the network toward the top of an energy bowl (point *b*, Fig. 2) but not out of that bowl. The additional force required to move the network up to the saddle region and over into the next bowl, or memory, must be supplied by the external input.

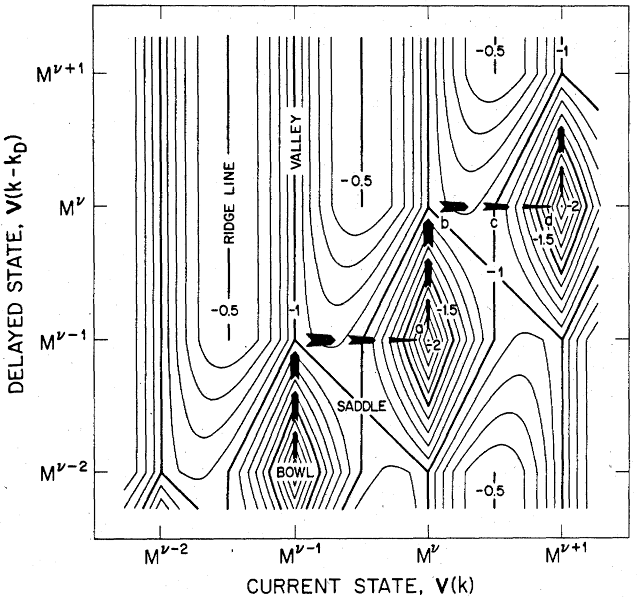

FIG. 2. Contour plot of the energy function, *E*, for the sequence generator. The plot was constructed by using Eq. **9** with orthogonal memory states. The arrows indicate a segment of the path the output of the network follows in the limit $\beta \rightarrow \infty$ (Eq. **1**) with a delta function delay.

NUMERICAL SIMULATIONS

Sequence Generation. I first demonstrate the functioning of the sequence generator (Eqs. **1**, **2**, **6**, and **8**) with a small network. Fig. 3*a* shows the individual neural outputs from a network containing 7 neurons and 3 memories with a delay interval $\kappa_D = 6$. The statistical independence of the memories used to form the **T** and **D** matrices was ensured by selecting nearly orthogonal memories (Eq. **10**). Starting from a randomly selected initial state, the network converged toward a memory and subsequently made cyclic transitions between all three memories. Relatively smooth transitions, however, did not occur until after the first few cycles; this is illustrated by the output of neuron 7 (Fig. 3*a*), which was

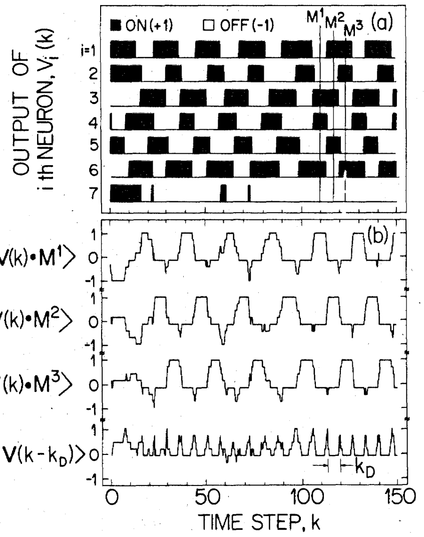

FIG. 3. Simulation of a network with 7 neurons and 3 memories, labeled M^1, M^2, and M^3, that generates cyclic sequences; see text for details. (*a*) Output of each neuron, $V_i(k)$, plotted as a function of the time step *k*. (*b*) Overlap of the current state of the network with each of the memories (Eq. **14**), $\langle V(k) \cdot M^{\nu} \rangle$ (top three traces); the overlap of the current state with the delayed state, $\langle V(k) \cdot V(k - \kappa_D) \rangle$ (bottom trace).

specified to have the value $V_7(k) = -1$ for all 3 memories.

Details of the sequential behavior can be highlighted by examining the overlap of $\mathbf{V}(k)$ with each of the memories, as opposed to examining the outputs of the individual neurons (23). We define the (normalized) overlap, $\langle \mathbf{V}(k) \cdot \mathbf{M}^{\nu} \rangle$, as

$$\langle \mathbf{V}(k) \cdot \mathbf{M}^{\nu} \rangle = \frac{1}{N} \sum_{j=1}^{N} V_j(k) M_j^{\nu}. \qquad [15]$$

A magnitude of 1 implies that the network is in the νth memory. These overlaps are plotted in Fig. 3*b* for the 7-neuron sequence generator. The width of each overlap, $\sim \kappa_D$, corresponds to the time spent in each memory. I also show in Fig. 3*b* the overlap of the current state with the delayed state, $\langle \mathbf{V}(k) \cdot \mathbf{V}(k - \kappa_D) \rangle$. This overlap peaks when the network is undergoing a transition. The maximum value $\langle \mathbf{V}(k) \cdot \mathbf{V}(k - \kappa_D) \rangle = 1$ occurs just prior to a transition. It corresponds to the network sitting on the upper edge of a bowl in the energy surface (point *b*, Fig. 2). The width of the peak, 2–3 time steps, corresponds to the transition time between memories.

We now focus on sequence generators containing 100 neurons, in which the statistical properties of large neural networks should emerge. For these studies we used memories composed of bit sequences chosen at random. Fig. 4 illustrates the cyclic output of a network with $\kappa_D = 6$ and 14 memories; 14 memories is just above the value for which retrieval errors can occur with associative memories (18). Approximately two complete cycles were required before the output reached a steady periodicity, after which the cycle period remained constant. Details of the initial transient behavior depended on both the initial state of the network, $\mathbf{V}(0)$, and the value of κ_D; with longer delays fewer cycles were required to reach steady conditions.

The cyclic output of these networks continued essentially indefinitely for sequences of length $m = n < 25$ and delay intervals $\kappa_D \geq 5$. With shorter values of κ_D the network did not become stationary in each memory and often became trapped in a spurious stable state (19). The period for each cycle under steady conditions was approximately $\Delta k = n \cdot (\kappa_D + 2)$; the time step of 2 corresponds to the transition time between memories.

Fault Tolerance. Neural networks are expected to function properly with errors in their synaptic connections. Qualitatively, this occurs because $n \cdot N$ bits of information are contained in the memories while the **D** and **T** matrices each contain $N \cdot (N - 1) \cdot \log_2 n >> n \cdot N$ bits. Errors in the values of some synaptic strengths are offset by the redundant storage of information in the synapses.

I assessed the ability of the sequence generator to function with noisy synaptic connections by randomly incrementing or decrementing the T_{ij} and D_{ij} connections ($i \neq j$) in one-bit increments. For a network of 100 neurons with 14 memories (Fig. 4), the sequential output remained essentially unaffected with root mean square (rms) noise levels up to approximately twice the rms value of the synaptic strength. At this threshold level for failure, the rms signal-to-noise of the input to each neuron is reduced from the intrinsic value of $\sim(N/n)^{1/2} = 3$ to a value of ≈ 1.5.

I examined the effect of stochastic removal of synaptic connections on sequence generation by setting randomly selected T_{ij} and D_{ij} connections to zero. For a network of 100 neurons and 14 memories (Fig. 4) the cyclic behavior was essentially unperturbed so long as 40% or fewer of the connections were removed. However, when additional connections were removed the output degraded rapidly, ceasing within one cycle with 50% of the connections missing.

A second method for removing 50% of the connections is to set one randomly chosen member of each T_{ij} and T_{ji} pair and each D_{ij} and D_{ji} pair to zero. Sequence generation was essentially unaffected by this alteration. This shows that the sequence generator functions without bidirectional connections between neurons. In contrast to the first method, the second method removes connections in a relatively uniform manner and therefore did not cause the large fluctuations in the **D** matrix necessary to halt transitions.

Capacitative Delays. An exponential weighting function was used as the time delay—i.e., $\mathfrak{D}(k, \kappa_D) = e^{-k/\kappa_D}/\kappa_D$—for

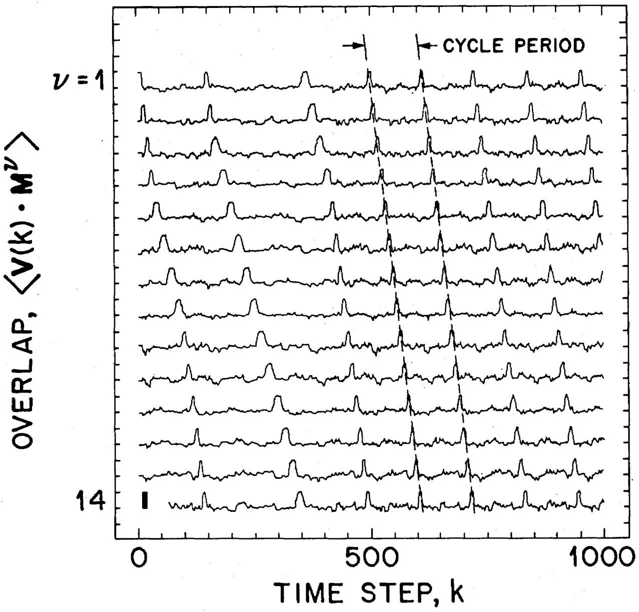

FIG. 4. Simulation of a network with 100 neurons and 14 memories that generates cyclic sequences; see text for details. The overlap of the current state of the network with each of the memories is plotted as a function of the time step, k. The bar in the lower left indicates an overlap value of 1.

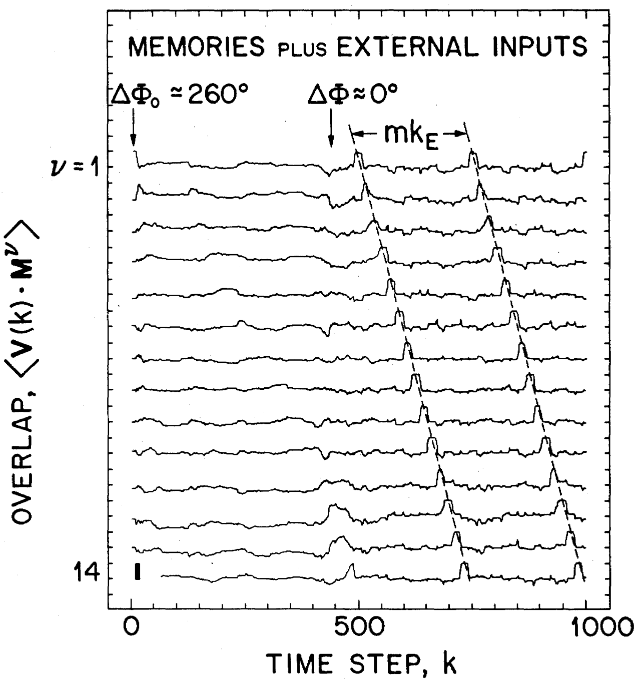

FIG. 5. Recognition of an external cyclic sequence by a network with 100 neurons and 14 memories; see text for details. The overlap of the current state with each of the memories is plotted as a function of the time step, k. The bar in the lower left indicates an overlap value of 1.

simulations with networks of 100 neurons. Sequential output was obtained with delay intervals as short as $\kappa_D = 3$, although the maximum length of a sequence was limited to $m = n \approx 7$. This result implies that sequence generation does not depend on the detailed form of the time delay. Furthermore, sequential output was maintained only if the peak value of $\langle \mathrm{V}(k)\cdot\mathrm{V_D}(k)\rangle$ exceeded $\approx$0.8 prior to a transition; for a delta function delay the peak value is $\approx$1.0. This suggests that reliable transitions occur only if the trajectory of the network along its energy surface comes close to the top of a bowl (point b, Fig. 2).

Sequence Recognition. I demonstrate the sequence recognition scheme (Eqs. **1**, **2**, **6**, **13**, and **14**) with a network of 100 neurons and 14 randomly selected memories. The threshold value of λ, below which sequence generation does not occur, was found to be $\lambda \approx 0.5$. Thus I choose $\lambda = 0.4$ and $\varepsilon = 0.2$ as parameters for the simulation to ensure that only the external input, $\mathbf{E}(k)$, could elicit sequential outputs. The external sequence was clocked with a period $\kappa_E = 3\cdot\kappa_D = 18$. The sequence consisted of 14 randomly chosen states that were mapped onto the memories via synapses with strengths F_{ij} (Eq. **13**). The initial state of the network was set to $\mathbf{V}(0) = \mathbf{M}^1$ and the initial state of the external input was $\mathbf{E}(0) = \mathbf{L}^{11}$. The difference between the states, $\Delta\nu = 10$, corresponds to an initial phase difference of $\Delta\Phi_0 = (\Delta\nu/n)\cdot 360° \approx 260°$ between the internal and external states.

The output of the sequence recognizer is shown in Fig. 5. At first the output fluctuated between memories, concomitant with the phase difference between $\mathbf{E}(k)$ and $\mathbf{V}(k)$ converging toward zero. Sequential outputs emerged when the states of the external sequence and the memories of the sequence generator varied in synchrony—i.e., $\Delta\Phi = 0°$. The period of the cyclic output ($m = n$) was $\Delta k = n\cdot\kappa_E \approx 250$, as compared with $\Delta k \approx 110$ for the free-running sequence generator (Eq. **14** with $\lambda = 1$ and $\varepsilon = 0$); cf. Figs. 4 and 5. The time required for the external and internal sequences to synchronize included contributions from the overall settling time of the network (Fig. 4) and corresponded roughly to $2\cdot n\cdot\kappa_E$, independent of the value of $\Delta\Phi_0$.

DISCUSSION

I have shown that the generation of sequential output states can be understood in the context of a model neural network This network contains no intrinsic oscillators and functions in the absence of external inputs. Sequence generation is an emergent property of the network that depends upon the interplay between two sets of synaptic connections. One set stabilizes the network in its current memory state, while a second set, whose action is delayed in time, causes the network to make transitions between memories.

The model I demonstrated contains connections between all pairs of neurons, while biological systems almost universally have a lesser degree of connectivity. Reducing the number of connections in the model network will decrease the number of memories that can be incorporated into sequences. However, sequence generation will still occur with a reduced number of connections if, on average, a T_{ij} connection is paralleled by a D_{ij} connection of the opposite sign. This corresponds in biological systems to balancing short-term inhibition by delayed excitation, and vice versa. Reciprocal connections of the same sign, which can cause oscillations between pairs of neurons (3), are not necessary for CPGs to function.

How does the behavior of the model compare with the measured properties of CPGs? Details of the networks that underlie CPGs have been studied in a number of preparations (1–3). We focus on the analysis by Getting (20, 21) of the CPG controlling the escape swimming response in the mollusk *Tritonia diomedea*. This CPG functions with 4 neural groups and 2 stable states ($\mathbf{M}$ and $-\mathbf{M}$). Despite these small numbers, it has many features in common with the model network. The neurons exhibit well-defined on (bursting) and off (quiescent) states. None of the neurons are intrinsic pacemakers. Many of the synaptic connections show both a short-term and a long-term response ($\kappa_D \approx 20$). The connections exhibit short-term excitation followed by delayed inhibition and short-term inhibition followed by delayed excitation ($\mathrm{D}_{ij} \approx -\mathrm{T}_{ij}$).

The output activity of many CPGs can be switched on and off via external inputs from command neurons (22). These neurons modulate a select fraction of the neurons in a CPG. The corresponding control of cyclic output can be achieved with schemes similar to that for sequence recognition. For example, when the D_{ij} connections are formed without including the term $\mathrm{M}_i^1\mathrm{M}_j^m$, only a linear sequence can be generated (Eq. **6**). Cyclic output will be enabled by an external input $\mathbf{E}(k) = \mathbf{E}^0$ if the connection strengths F_{ij} are chosen to cause the transition from the last memory to the first memory in the sequence—i.e., $\mathrm{F}_{ij} = \mathrm{M}_i^1\mathrm{E}_j^0$ (Eq. **12**). Different command neurons can elicit different output patterns from the same CPG (22). This corresponds to storing multiple sequences in the model network. A particular sequence is activated when the external input causes a transition to one of the memories in that sequence.

Note Added in Proof. A similar theory for generating sequences has been independently derived by Kanter and Sompolinsky (24).

I thank G. E. Blonder, H. J. Chiel, J. J. Hopfield, and H. Sompolinsky for many useful discussions. The sequence recognition scheme follows a suggestion by Hopfield. D. B. Pendergraft assisted with the simulations.

1. Roberts, A. & Roberts, B. L., eds. (1983) *Neural Origin of Rhythmic Movements* (Cambridge Univ. Press, Cambridge).
2. Selverston, A. I. & Moulins, M. (1985) *Annu. Rev. Physiol.* **47,** 29–48.
3. Cohen, A. H., Rossignol, S. & Grillner, S., eds. (1986) *Neural Control of Rhythmic Movements* (Wiley, New York).
4. Hopfield, J. J. (1982) *Proc. Natl. Acad. Sci. USA* **79,** 2554–2558.
5. Hopfield, J. J. (1984) *Proc. Natl. Acad. Sci. USA* **81,** 3088–3092.
6. Steinbuch, K. & Piske, U. A. W. (1963) *IEEE Trans. Electron. Comput.* **12,** 846–862.
7. Caianiello, E. R. (1966) *Kybernetik* **3,** 98–100.
8. Grossberg, S. (1970) *Studies Appl. Math.* **49,** 135–166.
9. Nakano, K. (1972) *IEEE Trans. Sys. Man Cybern.* **2,** 380–387.
10. Cooper, L. N. (1973) in *Proceedings of the Nobel Symposium on Collective Properties of Physical Systems*, eds. Lundqvist, B. & Lundqvist, S. (Academic, New York), pp. 252–264.
11. Little, W. A. & Shaw, G. L. (1975) *Behav. Biol.* **14,** 115–133.
12. Anderson, J. A., Silverstein, J. W., Ritz, S. A. & Jones, R. S. (1977) *Psych. Rev.* **84,** 412–451.
13. Kohonen, T. (1980) *Content Addressable Memories* (Springer, Berlin).
14. Hopfield, J. J. & Tank, D. W. (1985) *Biol. Cybern.* **52,** 141–152.
15. Amari, S.-I. (1972) *IEEE Trans. Comput.* **12,** 1197–1206.
16. Platt, J. C. (1985) *Sequential Threshold Circuits* (California Institute of Technology, Pasadena), Tech. Rep. 5197:TR:85.
17. Fukushima, K. (1973) *Kybernetik* **12,** 58–63.
18. Amit, D. J., Gutfreund, H. & Sompolinsky, H. (1985) *Phys. Rev. Lett.* **55,** 1530–1533.
19. Amit, D. J., Gutfreund, H. & Sompolinsky, H. (1985) *Phys. Rev. A* **32,** 1007–1018.
20. Getting, P. A. (1981) *J. Neurophysiol.* **46,** 65–79.
21. Getting, P. A. (1983) *J. Neurophysiol.* **49,** 1036–1050.
22. Kupfermann, I. & Weiss, K. R. (1978) *Behav. Brain Sci.* **1,** 3–39.
23. Peretto, P. & Niez, J. J. (1986) *Disordered Systems and Biological Organization*, ed. Bienenstock, E. *et al.* (Springer, Berlin), pp. 171–185.
24. Kanter, I. & Sompolinsky, H. (1986) *Phys. Rev. Lett.*, in press.