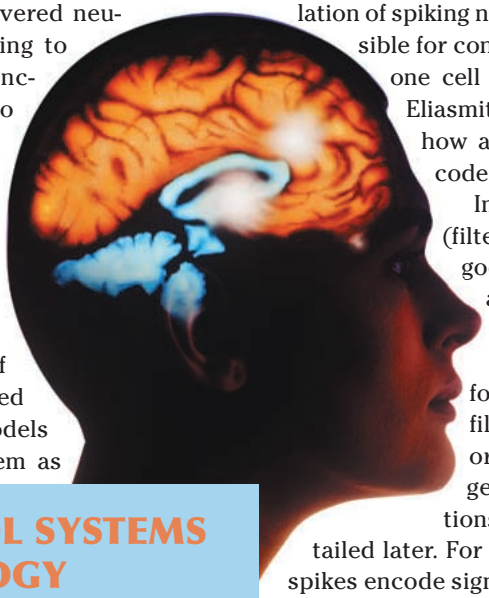
FEATURE

Signal Processing and Control Problems in the Brain

By Zoran Nenadic and Bijoy K. Ghosh



CONTROL SYSTEMS IN BIOLOGY

Ever since scientists discovered neurons, they have been trying to understand how they function and communicate to perform control tasks. Although some mathematical description of nerve and muscle membrane properties preceded the monumental work of Hodgkin and Huxley [1], it was their equations for ionic channels that created the natural starting point for the discipline of computational neuroscience. This led to various types of mathematical models of neurons, enabling us to treat them as nonlinear dynamical systems driven by sets of differential equations. One such dynamical system and an associated simplified model, introduced by Wilson [2], [3], is discussed in this article. An analysis of the transient response of the dynamical system shows the existence of a limit cycle in its phase plane, which corresponds to membrane potential oscillations (spikes) in the temporal domain.

As an important element of this article, we focus on the problem of representing an analog signal with a *spike train*, a representation commonly associated with a population of neurons. In the process of perception and control, the associated analog signals are believed to be encoded by a population of spiking neurons wherein the spikes are responsible for conveying the encoded information from one cell to another. Following prior work of Eliasmith and Anderson [4], [5], we describe how a collection of spike trains might encode analog signals.

Implementation of a suitable decoder (filter) is shown to produce a reasonably good reconstruction of the encoded analog signal. The decoding filter, as described later, could be a simple counter, or it could be designed via a formal optimization, assuming that the filter is causal and linear shift invariant or noncausal with impulse response generated by a family of basis functions. Each of the three cases will be detailed later. For a comprehensive discussion on how spikes encode signals and how signals can be decoded, the reader is referred to [6].

In the section on encoding and decoding using a population of neurons, we generalize the basic idea of an on/off pair of neurons to a population of neurons and show how the activity functions of these neurons can be used to encode and decode an analog signal. The term "activity" refers to the number of spikes in a given fixed window of time. Initially, the considered signals are constant. Subsequently, we consider signals that are changing in time and show that a neural population can be organized so that it solves ordinary differential equations (ODEs). Specifically, we show that an ODE can be equivalently

Ghosh (ghosh@zach.wustl.edu) and Nenadic are with the Department of Systems Science and Mathematics, Washington University, Campus Box 1040, St. Louis, MO 63130-4899, U.S.A.

0272-1708/01/$10.00©2001IEEE

written as a system of ODEs with state variables that are the activities of a given population of neurons. Assuming a piecewise linear function as the activity curve of the associated neuron, we analyze the problem of controlling a two-link robot arm with controls generated by using the above-described population of neurons. The control signals are obtained as a linear combination of the activity functions of the neural population.

Neural populations arise naturally in the brain, and in many instances it is unclear what computation, if any, they perform. To illustrate a practical example that we are currently studying [7], we describe the neuronal structure of the visual cortex of freshwater turtles. The role of the neural population in the cortex can only be conjectured, and it is possible that the neurons provide a precise representation of the position and velocity of a moving source of light. We show via simulations that the appearance of a novel stimulus in the visual space produces a wave of activity that propagates across the visual cortex. "Activity" here refers to the level of the membrane potential depolarization of individual cortical cells. Using voltage-sensitive dyes, these activities have been recorded by Senseman [8] and his collaborators. Using a simulation model of the visual cortex, we are able to reproduce these waves of activity for different input stimuli. The problem we are interested in studying is to what extent the inputs can be discriminated by observing the waves in the cortex. We show that the cortical waves can be analyzed via the Karhunen-Loeve (KL) decomposition [9], where the cortical waves are represented in a low-dimensional subspace. A three-dimensional phase space provides a good representation of the cortical wave and can be used to predict the location of a source of light in the visual space. We have also shown (see [10]) that the cortical waves can predict the velocity of a moving source of light in the visual space.

To summarize, this article emphasizes the role of a population of neurons in representing analog signals for computation and control, as well as its role in representing spatiotemporal waves of activity such as those observed in the visual cortex of a turtle.

A population of interacting neurons can sustain a wavelike activity that can discriminate the geometric position of the incoming stimulus in the visual space.

Structure of Neurons

A neuron consists of a cell body and one or more slender branches that grow out from it [11]. The cell body is called the soma, and the branches are termed neurites. In most neurons, there is one long neurite called the axon and several shorter neurites called dendrites. An axon is functionally defined as a neurite that conveys information away from the cell body. A dendrite is a neurite that conveys information toward the cell body. A specialized site at which communication between one neuron and another takes place is called a synapse (see Fig. 1).

Ion Channels

Like all other cells, neurons are enclosed by a cell membrane, which is composed mainly of lipids and proteins. The lipid structure of the membrane is double layered and impermeable to ions. The proteins are usually embedded in the membrane, and some of them have hollow openings in the center that allow ions to move passively across the membrane by diffusion. These proteins are known as ion channels. On the other hand, there are membrane proteins that use the energy stored in the cell to transfer ions across the membrane in an active fashion. These proteins are commonly known as ion pumps.

Membrane Potential

Across the membrane of each cell is a small difference in electrical potential, with the inside of the cell being electrically negative with respect to the outside. This difference, called the membrane potential, results from a separation of positive and negative charges across the cell membrane and is generated by the movement of ions (see Fig. 2). Three factors can induce an ion to cross a membrane: a difference in concentration of the ion on the two sides of the membrane, an electrical potential difference across the membrane, and the action of an ion pump. Except when electrically active, neurons are in a steady state, neither gaining nor losing ions of any particular type. In living neurons, the membrane potential is established by the diffusion of potassium ions out of the cell. The potential is maintained by the action of the sodium-potassium exchange pump, which expels the small number of sodium ions that leak into the neuron while taking up the potassium ions that have leaked out. If we change this poten-

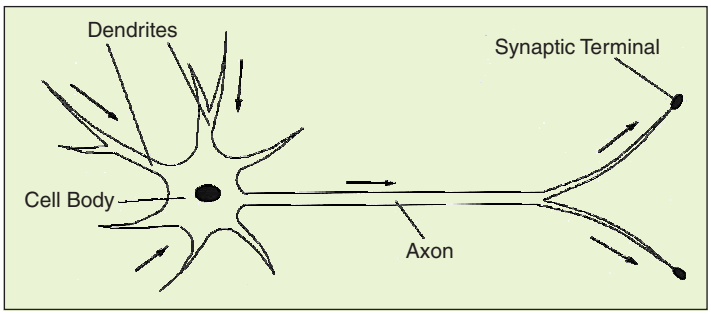


Figure 1. *Structure of a neuron showing the cell body, dendrites, axon, and synaptic terminals. The arrows indicate the direction of the flow of information.*

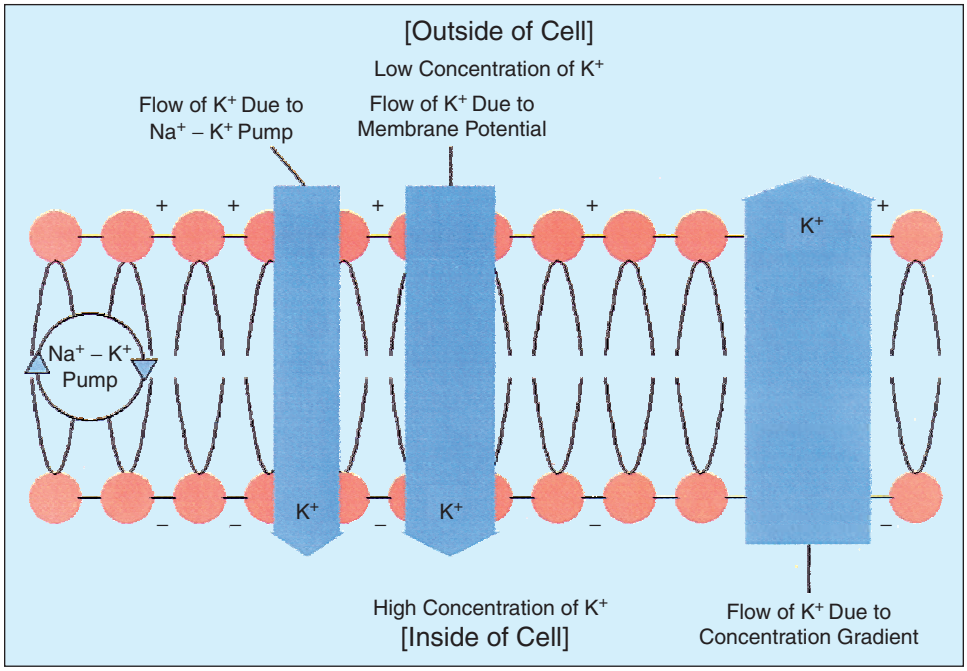


Figure 2. *A membrane showing the inside and outside of a cell. The arrows indicate the flow of potassium ions.*

tial difference between the inside and outside, the change can propagate in much the same passive way that heat is conducted down a metal rod.

The flow of information through a concentration gradient may be adequate for short cells (such as rods or cones in the retina), but if the axon is long, this mechanism is completely inadequate. To overcome this problem, most nerve cells have developed an efficient mechanism called the action potential.

Action Potential

A nerve impulse, technically called an action potential (spike), is a brief, transient reversal of the membrane potential that sweeps along the membrane of a neuron (see Fig. 3).

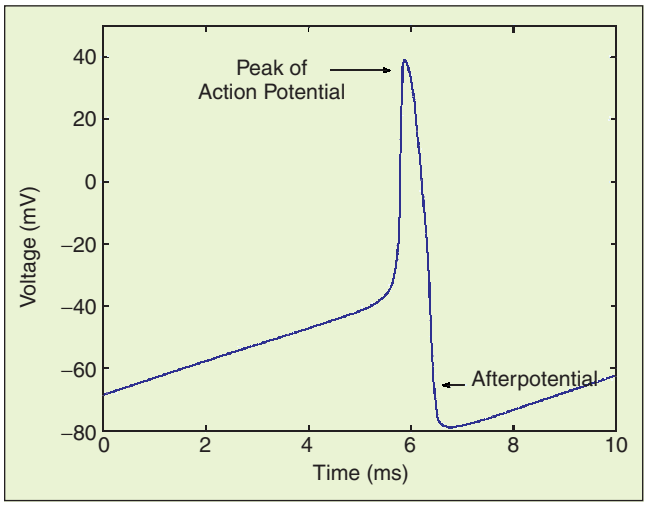


Figure 3. *Time response of a point on a membrane undergoing an action potential.*

An action potential has several important features. It is temporary, lasting only about a millisecond. Its appearance at one location on a membrane may induce an action potential in the adjacent membrane as well; hence it propagates along the length of the neurite at full amplitude without fading passively. It is all or none (i.e., it either reaches its full amplitude or does not occur at all). And its amplitude does not depend on the magnitude of the stimulus that elicited it.

To generate an action potential, the neuron must be stimulated sufficiently so that the membrane potential is depolarized to reach or exceed some minimal value, called the threshold. The channels that are involved in the generation of an action potential are voltage sensitive (voltage gated); their gates open or close in response to the magnitude and polarity of the electrical potential across the membrane.

Models of Neural Dynamics

To understand how to formulate a mathematical model of a neuron, we need to explain the electrical properties of a cellular membrane. Hodgkin and Huxley [1] recognized three different components of a membrane current, which they called sodium, potassium, and leakage. Today, the names *Na-channel* and *K-channel* are universally accepted for the corresponding ionic channels in axons. The name *leakage channel* is also used, although there is no experimental evidence regarding the ions or transport mechanism involved. These channels make a membrane permeable to ions, rendering its resistance finite (a membrane entirely lacking in channels would have an infinite resistance). While sodium and potassium channels have voltage-dependent (gated) resistances (conductances), leakage channels have conductances that are constant. To a first approximation, the current-voltage law is linear for every channel in question (i.e., Ohm's law applies):

$$I_j = g_j\left(V - E_j\right), \tag{1}$$

where E_j is the equilibrium potential of the ion in the jth channel, g_j is the conductance of the channel, I_j is the current in the channel, and V is the membrane potential (see Fig. 4). In addition to containing many conducting channels, the lipid bilayer of a biological membrane separates internal and external conducting solutions by an extremely thin in-

sulating layer. Such a narrow gap between the two conductors forms, of necessity, a significant electrical capacitor C. Having explained the basic behavior of the membrane, we can view one such neuron as a resistor-capacitor (RC) circuit. In modeling complex neurons, a compartmental model approach is frequently used. The neuron is divided into several compartments, each modeled as an RC circuit. Here we assume that the neurons are one-compartment models (Fig. 4).

The circuit equation corresponding to Fig. 4 is obtained as follows:

$$C\frac{dV}{dt} = -I_{\mathrm{Na}} - I_{\mathrm{K}} - I_l + I_{\mathrm{in}}, \tag{2}$$

where I_{in} is the injected current to the cell and I_{Na}, I_{K}, and I_l are the sodium, potassium, and leakage currents, respectively. Combining (1) and (2), we obtain

$$C\frac{dV}{dt} = -g_{\mathrm{Na}}(V - E_{\mathrm{Na}}) - g_{\mathrm{K}}(V - E_{\mathrm{K}}) - g_l(V - E_l) + I_{\mathrm{in}}.$$

The original work of Hodgkin and Huxley [1] proposes that the voltage-dependent sodium and potassium conductances are given as $g_{\mathrm{Na}} = \bar{g}_{\mathrm{Na}} m^3 h$ and $g_{\mathrm{K}} = \bar{g}_{\mathrm{K}} n^4$, where $\bar{g}_{\mathrm{Na}}$ and $\bar{g}_{\mathrm{K}}$ are constants, m,h,n are driven by the following differential equations:

$$\begin{aligned}
\frac{dm}{dt} &= \frac{m_\infty - m}{\tau_m} & m_\infty &= \frac{\alpha_m}{\alpha_m + \beta_m} & \tau_m &= \frac{1}{\alpha_m + \beta_m} \\
\frac{dh}{dt} &= \frac{h_\infty - h}{\tau_h} & h_\infty &= \frac{\alpha_h}{\alpha_h + \beta_h} & \tau_h &= \frac{1}{\alpha_h + \beta_h} \\
\frac{dn}{dt} &= \frac{n_\infty - n}{\tau_n} & n_\infty &= \frac{\alpha_n}{\alpha_n + \beta_n} & \tau_n &= \frac{1}{\alpha_n + \beta_n},
\end{aligned}$$

and $\alpha_m, \beta_m, \alpha_h, \beta_h, \alpha_n, \beta_n$ are nonlinear functions of the voltage V, often having discontinuities. Rather than dealing with such a complicated system, it is a common practice to simplify the underlying dynamics while retaining some of the basic features of Hodgkin-Huxley-type kinetics. These simplifications often have a form of second-order dynamical systems and are frequently referred to as *two variable models* (see [12]). For example, the dynamics corresponding to the variable m is relatively fast compared to those of h and n; therefore, it is reasonable to approximate the value of m by its steady state m_∞. According to [3], the simplified membrane equation of a mammalian neocortical neuron can be written as

$$C\frac{dV}{dt} = -m_\infty(V - E_{\mathrm{Na}}) - 26R(V - E_K) + I_{\mathrm{in}} \tag{3}$$

$$\frac{dR}{dt} = \frac{R_\infty - R}{\tau_R},$$

where m_∞ and R_∞ are second-order polynomials

$$m_\infty = 17.8 + 47.6V + 33.8V^2$$
$$R_\infty = 1.24 + 3.7V + 3.2V^2$$

and where the leakage current in (3) has been eliminated via a change of variables. To keep the constants within reasonable bounds, the membrane potential has been scaled down by 100. Given this convention, $E_{\mathrm{Na}} = 0.5$ and $E_{\mathrm{K}} = -0.95$ correspond to 50 mV and –95 mV, respectively. For an injected current input $I_{\mathrm{in}} = 0.2$ nA, we observe that the membrane potential V is spiky, as shown in Fig. 5. The frequency of the spikes depends on the amplitude of I_{in}. Thus, the spike frequency of V encodes the analog signal I_{in}. This idea is known as *rate coding*. The oscillations in Fig. 5 correspond to a limit cycle in the (V,R) phase plane. As the parameter I_{in} increases, the number and character of equilibria change, evolving from a stable node (negative values of I_{in}) to an un-

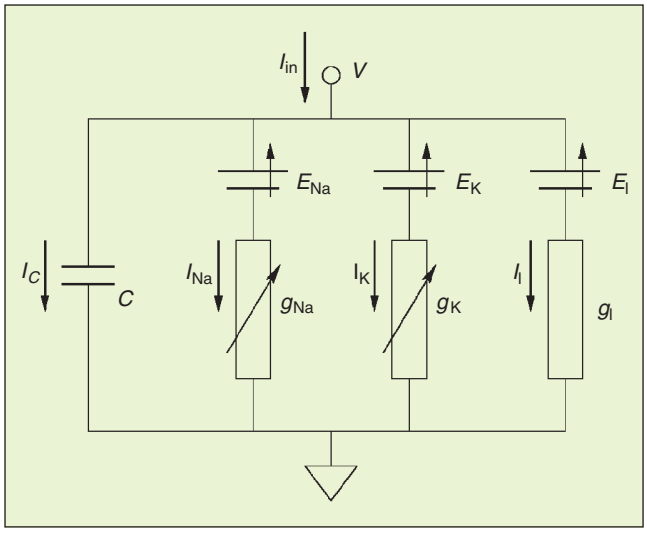


Figure 4. *The RC circuit of a cell showing the sodium, potassium, and leakage channels. Note that the conductances are functions of the voltage V.*

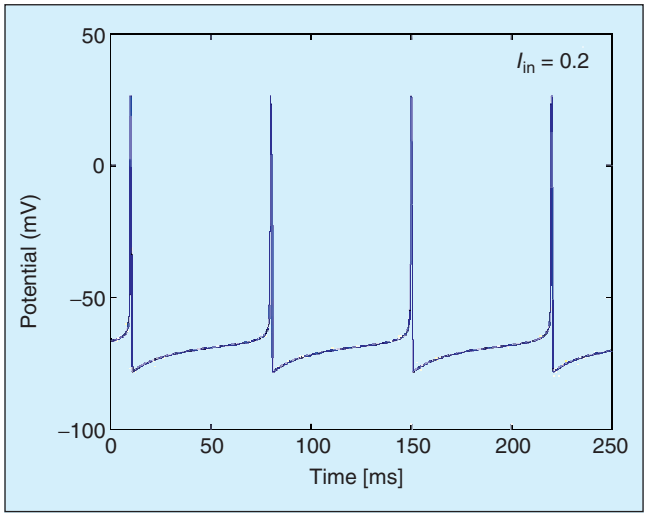


Figure 5. *Spiky membrane potential as a response to injected current.*

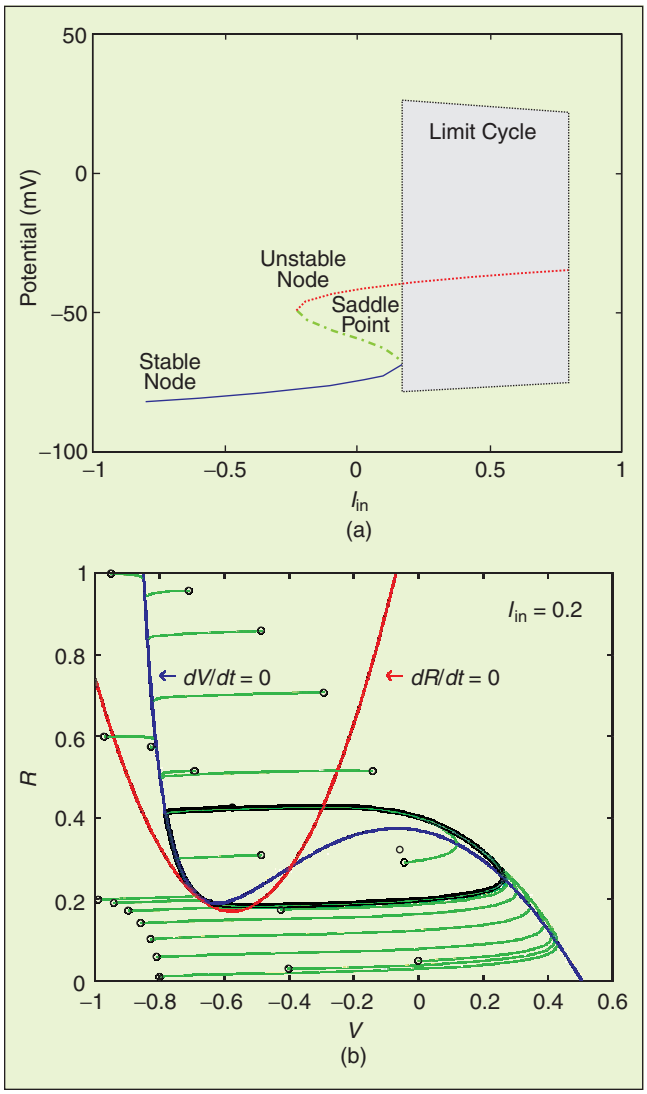


Figure 6. *Bifurcation diagram showing the changes in the equilibrium from stable to unstable to a limit cycle (a); the limit cycle shown explicitly (b).*

stable node ($I_{in} \geq 0.18$), the latter being followed by a limit cycle, as shown in Fig. 6 (see [2] for details). Note that the trajectories are almost horizontal, except where $dV/dt = 0$, indicating that the variable V is changing more rapidly than the variable R. Consequently, the variable V is called the fast variable and the curve $dV/dt = 0$ is called the *slow manifold*.

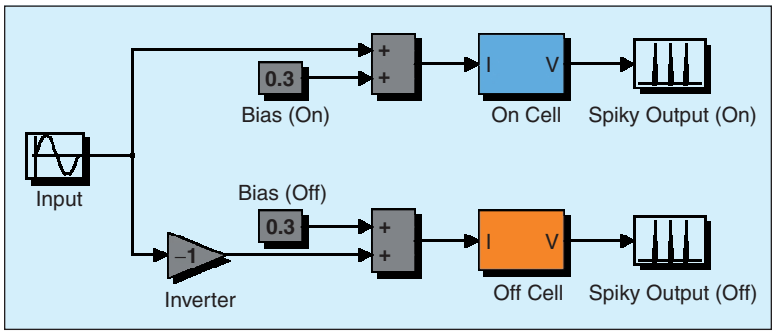


Figure 7. *Illustration of how a sinusoidal input can be encoded by an on/off cell.*

Using the Poincaré criterion, one can easily verify the stability of the limit cycle, which has been generated by applying the current stimulus $I_{in} = 0.2$.

Encoding and Decoding Signals Using On/Off Cells

Encoding Analog Signals

A neuron has the capability of representing an analog time signal as a spike train. The firing rate of the neuron encodes the amplitude of the analog signal. The encoding scheme is successful as long as the signal remains positive and above a certain threshold (i.e., a neuron is unable to encode signals below this threshold). To represent signals that might vary between a positive and a negative value, we use the idea of on and off cells.

In the vertebrate retina, a ganglion that produces an *on response* (i.e., increases its rate of firing when the center of its receptive field is stimulated by light) is called an on-center ganglion cell. Conversely, an off-center ganglion is a cell that decreases its firing rate when the center of its receptive field is stimulated by light. One such on/off cell circuit is shown in Fig. 7. The basic unit of the circuit is the model of the neuron described by (3), although it may be any other biologically valid model of a neuron or its approximation, such as the *integrate and fire model* or the *leaky integrate and fire model* (see [13]). There is no actual difference between the on and off cells, except that the input has been inverted prior to entering the off part of the network. Many neurons are capable of producing a spiky output even if no stimulus is applied. This introduces the notion of a so-called *background firing rate*, represented by a bias in both on and off units. The spiky outputs of both cells can be decoded using different types of decoding schemes, which will be discussed next.

Various Methods for Signal Reconstruction

We have seen that analog signals can be encoded into sequences of spikes and that the firing rate of neurons is proportional to the amplitudes of the corresponding stimuli. The nervous system faces the reverse problem, determining the meaning of a spiky pattern in the real world. Therefore, a legitimate question naturally arises as to what would be a good decoding scheme: How would one extract the analog signal (stimulus) that is represented by a sequence of spikes? Even though there is no scientific evidence that stimulus reconstruction takes place in an actual neural system, the problems that the nervous system solves on a routine basis suggest such a calculation. For example, the fly can initiate a turn based on visual motion signals alone, which means that it translates the spike output of its motion-sensitive visual neurons into a torque, and this torque has a component roughly pro-

portional to the time-dependent angular velocity (see [6]). The torque signal is a continuous waveform that the fly synthesizes based on discrete spike sequences in its sensory neurons. We will illustrate three different decoding procedures—the sliding window, the optimal linear causal filter, and the optimal noncausal filter—regardless of whether the computations involved seem biologically plausible. It is evident from the examples that the three procedures provide a successful reconstruction, even though there are significant differences among them.

Before we outline the details of the three reconstruction schemes, we shall discuss the underlying idea, which is based on the fact that in response to an increase in the amplitude of a static stimulus, neurons increase their rate of spiking. More precisely, the number of spikes in a fixed time window following the onset of a static stimulus represents the intensity of that stimulus. Therefore, the basic ingredient of the scheme is a frequency-amplitude curve (see Fig. 8) of a chosen spike-generating device (cell). Since the curve was generated by stimulating the cell with constant signals, it has a static character. We explore how good the reconstruction is in the case of an arbitrary nonconstant signal (e.g., sine wave). The on/off cell system has been stimulated by a sine wave of a certain amplitude and frequency. A time window of a suitably chosen width is then continuously slid over the two spiky signals (on and off), and the number of spikes per window is counted, yielding two time signals that represent the frequency code of the sine wave. A simple inversion of the frequency-amplitude curve is then used to get two amplitude time signals. The reconstruction waveform is found by subtracting the off signal from the on signal and is shown in Fig. 9 for a window width of 150 ms.

Let us now suppose that we want to reconstruct the same signal as above using a linear filter. Moreover, we wish to design an optimal filter in the sense that it gives a minimal reconstruction error—i.e., we minimize the following cost functional:

$$E = \int_0^T \left[s(t) - \hat{s}(t)\right]^2 dt, \tag{4}$$

where $\hat{s}(t)$ is the reconstruction of the signal $s(t)$. The signal $s(t)$ generates a sequence of spikes in the on and off channels. The reconstruction $\hat{s}(t)$ is obtained by low-pass filtering the on and off responses and then taking the difference between the two filtered signals. Restricting ourselves to a second-order linear filter with impulse response given by $h(t) = kte^{\lambda t}$, where k is the gain of the filter and λ is its pole, and assuming that the spiky signals can be represented as a sequence of Dirac functions, the reconstruction of signal $s(t)$ is given by

$$\hat{s}(t) = k\left(\sum_i (t - t_i)e^{\lambda(t-t_i)} - \sum_j (t - t_j)e^{\lambda(t-t_j)}\right),$$

where i and j refer to spike indexes in the on and off units, respectively. Minimizing the error E with respect to the parameters k and λ yields the reconstruction shown in Fig. 9. Note that the linear filter possesses a causality property (i.e., it does not anticipate the input signal).

When the spike trains are to be filtered using an optimum filter, rather than a linear one, we represent the kernel of the filter using a linear combination of a certain number of basis functions

$$h(t) = \sum_n c_n \phi_n(t),$$

where $\phi_n(t)$ are from a set of nonorthogonal basis functions (chosen as Gaussian functions in this analysis). We generate a set of 20 basis functions of the form $\phi_n(t) = \phi_0(t - ndt)$ for

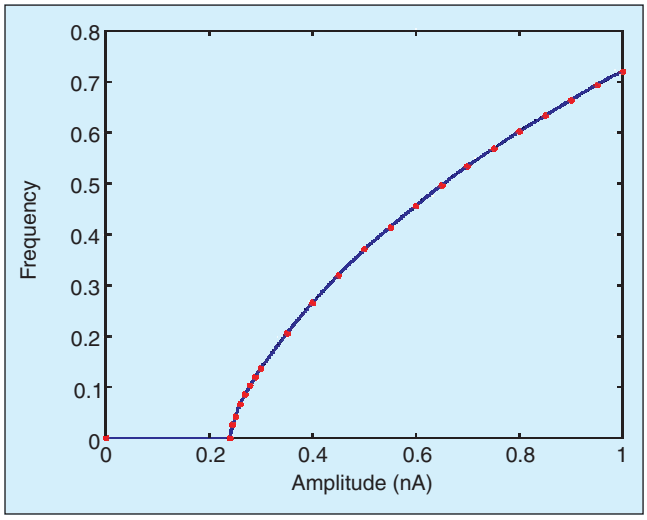


Figure 8. *Frequency of spikes versus amplitude of the injected current for a typical neuron.*

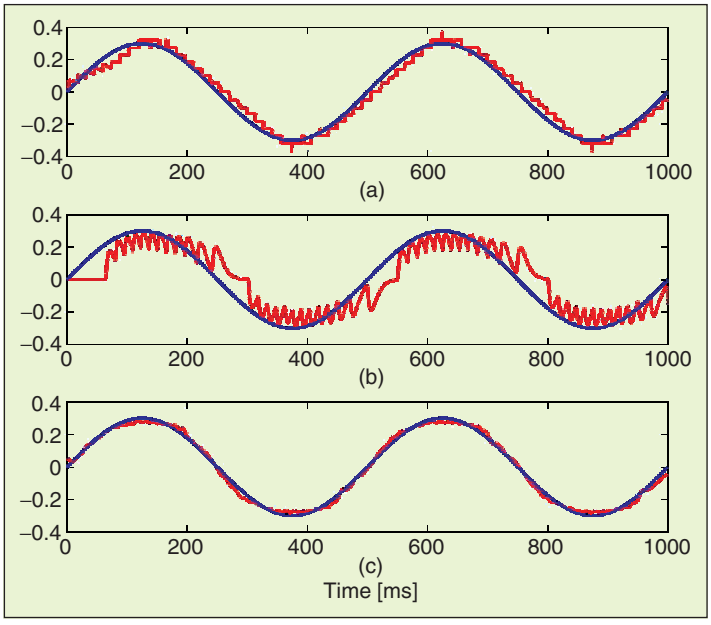


Figure 9. *Decoding of a sine wave using a sliding window (a); optimal linear causal filter (b); optimal filter (c).*

the filter kernel expansion and obtain the optimum set of coefficients c_n that minimizes the error E defined by (4). The reconstructed signal is shown in Fig. 9.

To summarize, all three decoding procedures have been successfully applied and tested. The last two are optimal in the sense that the parameters are obtained by minimizing a certain cost functional (reconstruction error, in this case). The second method assumes linearity and causality, whereas the first and third methods do not.

Encoding and Decoding Using a Population of Neurons

In the previous section, we have seen that an analog signal can be decoded from the *spike rate* or *activity* of an on/off pair of neurons. However, there is one important drawback of the decoding procedure viewed in the context of an on/off circuit. Namely, the decoding filter is good only for the input it has been designed for (e.g., sine wave). There is no guarantee that the decoding procedure would succeed for an arbitrary choice of input. To overcome this drawback, we introduce the notion of a population of neurons, which can be thought of as the generalization of the on and off cells concept. A population of neurons could be created by having an equal number of on and off cells with randomly chosen biases. Each neuron within a population has its own activity curve. These activities can be thought of as instantaneous firing rates of individual neurons and are similar in nature to the frequency-amplitude curve shown in Fig. 8. They can be computed from the spike signal, possibly by low-pass filtering, as illustrated earlier. For an extensive discussion on encoding and decoding of analog signals using a population of neurons, the reader is referred to [14]. The activity curves for a population of neurons are shown in Fig. 10. For computational efficiency, these curves are approximated by piecewise linear functions.

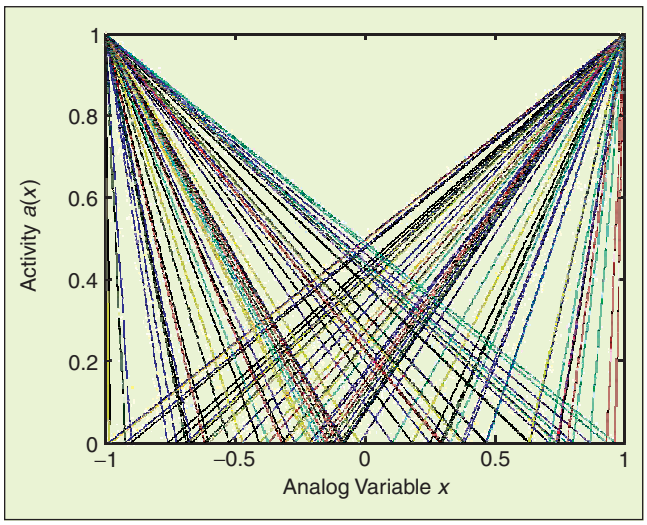


Figure 10. *Activity curves of a set of 200 neurons. The positive slopes correspond to the on cells and the negative slopes correspond to the off cells.*

To illustrate how a population of N cells is used to encode an analog signal $u(t)$, we assume that the activity function of the ith cell is given by

$$a_i(u) = [\alpha_i u + \beta_i]_+, \tag{5}$$

where $[\ ]_+$ stands for a rectification operation. The main point of this discussion is that the activity functions $a_i(u(t))$ as an ensemble can be used to estimate $u(t)$ as follows:

$$\hat{u}(t) = \sum_{i=1}^{N} X_i a_i(u(t)). \tag{6}$$

The weights X_i are calculated by assuming that the function $u(t)$ is constant and takes values in the interval $[u_{\min}, u_{\max}]$ and by minimizing the error function defined as

$$E = \left\langle \int_{u_{\min}}^{u_{\max}} [u - \hat{u}(u)]^2 du \right\rangle_\eta,$$

where η is an additive zero mean noise with variance σ^2. The symbol $\langle\cdot\rangle_\eta$ stands for an average over noise η, which is incorporated in the model via

$$\hat{u}(u) = \sum_{i=1}^{N} X_i [a_i(u) + \eta_i].$$

The presence of noise in the activities of individual neurons implies that there is an inherent randomness in the population of neurons. For sources of noise within a single neuron, the reader is referred to [6]. Once the weights are learned, they are to be kept fixed, and (5) and (6) provide a mechanism for encoding and decoding $u(t)$ in terms of a population of activity variables $a_i(t)$ or $a_i(u(t))$.

The procedure described above is illustrated by considering an example where we choose a set of 20 neurons with a random set of activity profiles. By considering a sequence of constant input signals, we compute the optimal weights X_i. The network is tested for a sinusoidal input signal, and the reconstruction is obtained using the decoding formula (6). The results are shown in Fig. 11.

Note that the procedure outlined above can be generalized. Namely, it is possible to extend the concept of decoding from a scalar variable $u(t)$ to an arbitrary function of that variable $f(u(t))$. Consistent with (6), the decoding rule can be written as

$$\hat{f}(u(t)) = \sum_{i=1}^{N} X_i a_i(u(t)), \tag{7}$$

where the only difference between (6) and (7) is in the weights X_i. They are now obtained by minimizing the error function defined as

$$E = \left\langle \int_{u_{\min}}^{u_{\max}} \left[f(u) - \hat{f}(u) \right]^2 du \right\rangle_\eta.$$

Another generalization of the encoding/decoding process would base the formulation on vectors rather than scalars. There is abundant evidence that a population of neurons can encode vectors. Roughly speaking, a neuron in a population will fire if a vector has specific orientation, a concept known as *preferred direction* (see [4]). The firing rate of a neuron will decrease if the vector is rotated with respect to the neuron's preferred direction. For a vector $X \in IR^n$, the generalized encoding rule becomes

$$a_i(X) = \left[\langle \alpha_i, X\rangle + \beta_i\right]_+,$$

where $\alpha_i = V_i/(1-\varepsilon_i)$, $\beta_i = 1/(1-\varepsilon_i)$, and the notation $<\cdot,\cdot>$ stands for inner product. The vector $V_i \in IR^n$ is the preferred direction of the ith neuron, and the variable ε_i provides a bias. Likewise, the decoding rule is given by

$$\hat{X} = \sum_{i=1}^{N} X_i a_i(X),$$

and the weight vectors X_i are found by minimizing the error functional described by

$$E = \left\langle \int \cdots \int \left\| X - \sum_{i=1}^{N} X_i [a_i(X) + \eta_i] \right\|^2 dX_1 \cdots dX_n \right\rangle_{\eta}.$$

As before, this procedure can be extended from vectors to both scalar and vector functions of vectors.

To conclude this section, let us make several important remarks. First, note that only the activities of individual neurons have been used, rather than the corresponding spike trains. The transformation from a continuous input signal to the associated activity has been implemented by a static curve, viz. the activity curve. It follows that the two time-dependent dynamical actions, the spike generation and the filtering of the spike train, have been approximated as a static process. As a consequence, the population of neurons used in this section has an unrealistically large bandwidth. Clearly, adding the dynamics of individual cells and obtaining the activities by filtering in real time (see [14]) would impose some constraints in the frequency response of the network (e.g., a sine wave of high frequency would not be successfully reconstructed). Second, the procedure of representing functions with a population of basis functions, outlined above, may seem similar to using a layered artificial neural network to approximate an arbitrary signal or function. An important difference is that an artificial neural network usually operates with a double-precision representation of the neural activities, while we emphasize that real neurons are at best 3-bit devices with dynamics of their own. However, real neurons have not been used in this expository article, and we refer readers to [14] for a more rigorous treatment of this problem with realistic (spiking) neurons. Finally, the error function being minimized can be decomposed into two parts, one due to the finite number of basis functions coming out of a finite number of cells in the neural population and the other due to the presence of noise. Both of these errors decrease monotonically as the number of basis functions (the number of neurons) is increased.

Two-Link Robot Arm Control Using a Neural Population

In this section, we show that the activity functions of a population of neurons can not only encode an analog signal, as shown earlier, but can also be used to provide an internal representation of an ordinary differential equation governing the dynamics of the signal. Thus, the activity variables, introduced in (5) and (6), provide a set of internal state variables with respect to which all dynamic equations are to be encoded as internal *activity dynamics*. This point of view is il-

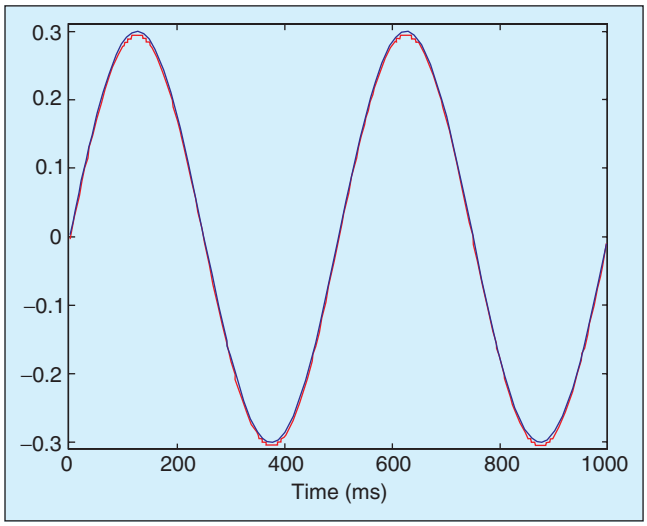


Figure 11. *Reconstruction of a sinusoid using a population of 20 neurons.*

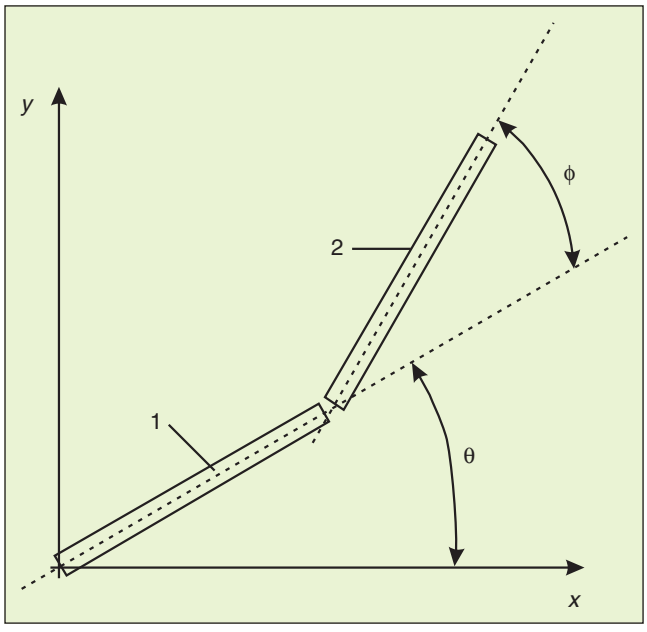


Figure 12. *Schematic of a two-link arm showing the elbow and the shoulder angle.*

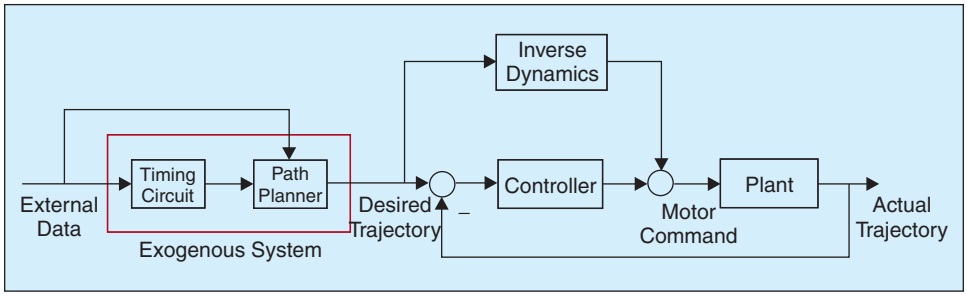


Figure 13. *Block diagram showing planning and control of the arm movement. Note that the path planner is equipped with a timing circuit.*

lustrated by considering the problem of controlling a two-link robot arm using a population of neurons.

To give some biological motivation, we remark that movements of a human arm in a horizontal plane are very stereotyped in the sense that the corresponding paths are mainly straight lines and the velocity profiles are bell-shaped functions. Using standard tools from analytical mechanics, the dynamic model of a two-link rigid body (see Fig. 12) is described as follows:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}x_3\\ x_4\end{bmatrix}$$
$$\begin{bmatrix}\dot{x}_3\\ \dot{x}_4\end{bmatrix}=-T^{-1}(x)C(x)\begin{bmatrix}x_3\\ x_4\end{bmatrix}+T^{-1}(x)\begin{bmatrix}\tau_s\\ \tau_e\end{bmatrix}, \tag{8}$$

where $T(x)$ and $C(x)$ are matrices of coriolis and centripetal terms and where x is given by

$$x=\begin{bmatrix}\theta & \phi & \dfrac{d\theta}{dt} & \dfrac{d\phi}{dt}\end{bmatrix}.$$

The vector of external inputs $u=\begin{bmatrix}\tau_s & \tau_e\end{bmatrix}^T$ contains the torque pair associated with the shoulder and elbow. The torque pair is found analytically using feedback linearization (see [15]). The torque pair calculation that would cause the system to follow the desired trajectory is given by

$$\begin{bmatrix}\tau_s\\ \tau_e\end{bmatrix}=C\left(\theta,\phi,\dot{\theta},\dot{\phi}\right)\begin{bmatrix}\dot{\theta}\\ \dot{\phi}\end{bmatrix}+T\left(\theta,\phi,\dot{\theta},\dot{\phi}\right)\begin{bmatrix}\ddot{\theta}_d+f_2\left(\dot{\theta}_d-\dot{\theta}\right)+f_1(\theta_d-\theta)\\ \ddot{\phi}_d+f_4\left(\dot{\phi}_d-\dot{\phi}\right)+f_3(\phi_d-\phi)\end{bmatrix}. \tag{9}$$

To compute the pair of torques described by (9), it is necessary to compute two sets of variables. The first consists of desired angles, velocities, and accelerations θ_d , $\dot{\theta}_d$, $\ddot{\theta}_d$, ϕ_d, $\dot{\phi}_d$, $\ddot{\phi}_d$, and the second consists of actual angles and angular velocities $\theta, \dot{\theta}, \phi, \dot{\phi}$. The first set of variables is provided by a path planner and the second set is provided by feedback, as shown in Fig. 13. It is interesting to note that the desired variables could be generated as the response of an exogenous dynamical system consisting of a timing circuit and a path planner.

The role of the timing circuit is to provide the bell-shaped velocity profile, which is the driving signal for the path planner. The timing circuit has a specific start time and end time and is driven by a set of external data (initial and final position of the arm). The fact that the bell-shaped function can be modeled as a periodic function allows the timing circuit to be designed as a second-order linear system, described by

$$\ddot{u}+\omega^2 u=\omega^2. \tag{10}$$

Note that the system (10) represents a linear oscillator with the frequency $\omega=2\pi/T$, where T is the total time elapsed in reaching the final position from the initial position. Both u and $\dot{u}$ are to be encoded using different populations of neurons with their encoding rules

$$a_i(u)=[\alpha_i u+\beta_i]_+ \qquad b_j(\dot{u})=[\gamma_j\dot{u}+\delta_j]_+$$

and their corresponding reconstructions are given by

$$\hat{u}(u)=\sum_{i=1}^{N}X_i a_i(u) \qquad \hat{u}(\dot{u})=\sum_{j=1}^{M}Y_j b_j(\dot{u}). \tag{11}$$

The time derivative of a particular activity function at time t can be approximated by

$$\frac{da_n(t)}{dt}=-\frac{1}{\tau}\left[a_n(t)-a_n(t+\tau)\right], \tag{12}$$

where the activity of the nth neuron at time $t+\tau$ using the encoding rule (5) simply becomes

$$a_n(t+\tau)=\left[\alpha_n u(t+\tau)+\beta_n\right]_+. \tag{13}$$

Note that since

$$\frac{du}{dt}=\frac{u(t+\tau)-u(t)}{\tau}, \tag{14}$$

it follows from (11) that

$$u(t+\tau)=\tau\sum_{j=1}^{M}Y_j b_j(t)+\sum_{i=1}^{N}X_i a_i(t). \tag{15}$$

It readily follows from (12), (13), and (15) that for the activities $a_n(t)$ corresponding to u

$$\frac{da_n(t)}{dt}=-\frac{1}{\tau}\left\{a_n(t)-\left[\tau\sum_{j=1}^{M}\omega_{nj}^{(\mathrm{ext})}b_j(t)+\sum_{i=1}^{N}\omega_{ni}^{(\mathrm{int})}a_i(t)+\beta_n\right]_+\right\}, \tag{16}$$

and likewise, for the activities $b_m(t)$ corresponding to $\dot{u}$

$$\frac{db_m(t)}{dt} = -\frac{1}{\tau}\left\{ b_m(t) - \left[\tau\left(\omega^2 - \omega^2 \sum_{i=1}^{N} \omega_{mi}^{(\text{ext})} a_i(t) \right) + \sum_{j=1}^{M} \omega_{mj}^{(\text{int})} b_j(t) + \delta_n \right]_+ \right\}, \tag{17}$$

where the coupling weights are given by

$$\omega_{ni}^{(\text{int})} = \alpha_n X_i \quad \omega_{nj}^{(\text{ext})} = \alpha_n Y_j \quad \omega_{mj}^{(\text{int})} = \gamma_m Y_j \quad \omega_{mi}^{(\text{ext})} = \gamma_m X_i.$$

Note that (16), (17) form a coupled pair of differential equations defined using only the activity variables and essentially solve the second-order differential equation (10). A sketch of the solution to the activity dynamics for $N = M = 40$ is shown in Fig. 14.

The reconstructions of $u(t)$ and $\dot{u}(t)$ can be obtained using the decoding rule (11). However, it is not relevant to decode these variables at this point. We already know that they represent analog signals that are solutions of (10). What is relevant is to transmit the signals $u(t)$ further in the brain and to combine them with other signals. This would lead to path planning and eventual computation of the torque pairs described by (9). Note that the decoding process will eventually take place at the level of generating motor commands, at which point it is necessary to synthesize a continuous signal (torque) from the corresponding quantities.

Once the activity function of u is known, it serves as an input to the path planner, which is described by a nonlinear differential equation

$$\begin{bmatrix} \dot{\theta}_d \\ \dot{\phi}_d \end{bmatrix} = M^{-1}(\theta_d, \phi_d) \frac{V_m}{2} u(t) \frac{r_f - r_i}{\delta}, \tag{18}$$

where V_m is the peak velocity of the arm, r_i and r_f are its initial and final positions, δ is the total distance traveled by the end point of the two-link arm in time T, and M is the transformation matrix from Cartesian to generalized coordinates. It is clear that we should seek a solution to the differential equation (18) in terms of the activity functions of $\dot{\theta}_d$ and $\dot{\phi}_d$. For the purposes of this article, we skip this procedure and solve (18) using a numerical package. The reason for doing this is not a matter of computational convenience, but rather a consequence of the lack of generality of the discretization outlined in (12) and (14). For the ordinary differential equation (18), such a discretization leads to instability, a fact that requires further investigation. Once the solution of (18) is computed, the activities of the corresponding variables are obtained by the encoding procedure already described. The activity functions of $\dot{\theta}_d, \ddot{\theta}_d, \dot{\phi}_d, \ddot{\phi}_d$ are obtained by implementing a neural differentiation, details of which have been described in [16].

Finally, the activity functions corresponding to the shoulder and elbow torques are calculated from the algebraic equation (9) using the activity profiles of the desired and actual variables. While the desired variables have been provided by the path planner in the framework described above, the activity functions of the actual position and velocity are provided directly by encoding the sensory data. An alternative approach to synthesizing the activity functions of the torque components would be to use the procedure of a vector encoding. Generally, there is no unique way of solving the encoding problem at this stage, as various methods that include encoding scalars, vectors, and scalar and vector functions of scalars and vectors could be combined. For example, both the desired and actual variables could be treated as a single vector, viz. $[\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \phi_d, \dot{\phi}_d, \ddot{\phi}_d]$ and $[\theta, \dot{\theta}, \phi, \dot{\phi}]$, and the activities of the torques can be represented as a nonlinear function of these two vectors. However, encoding a vector is computationally expensive in general, and the required number of neurons grows with the dimension of the vector. This fact needs further careful investigation.

Regardless of the method used to compute the activity functions of τ_s and τ_e, the torques are obtained from these activities using the optimal decoding rule, as described earlier. The synthesized torques are then used as an input to the dy-

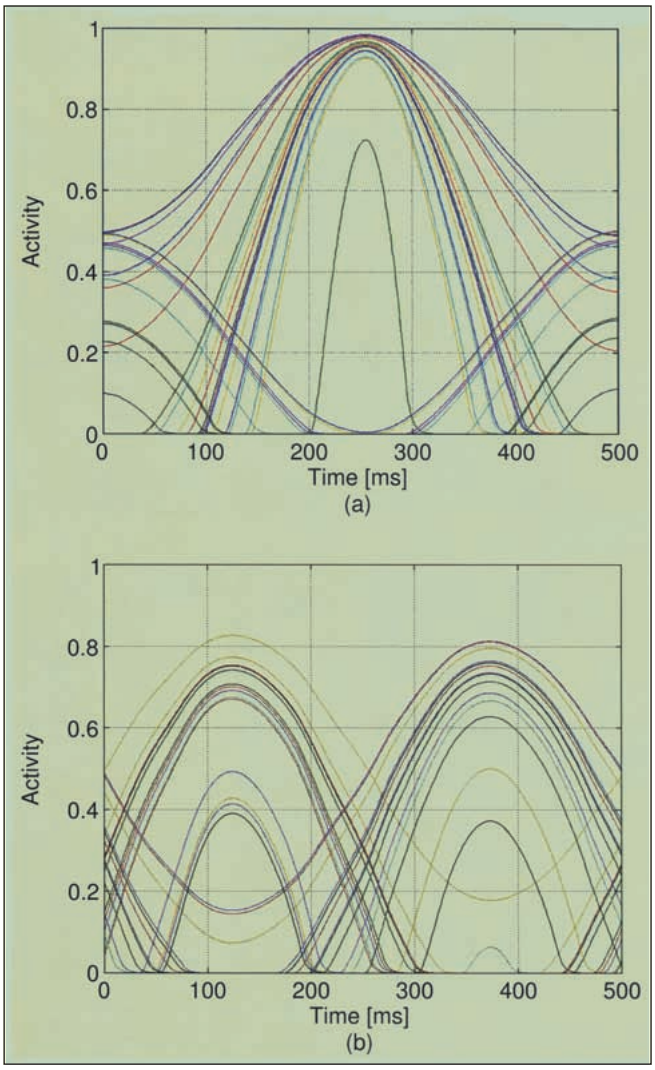


Figure 14. *Solution to the activity dynamics (16), (17). (a) represents the activities of $u(t)$, and (b) represents the activities of $\dot{u}(t)$.*

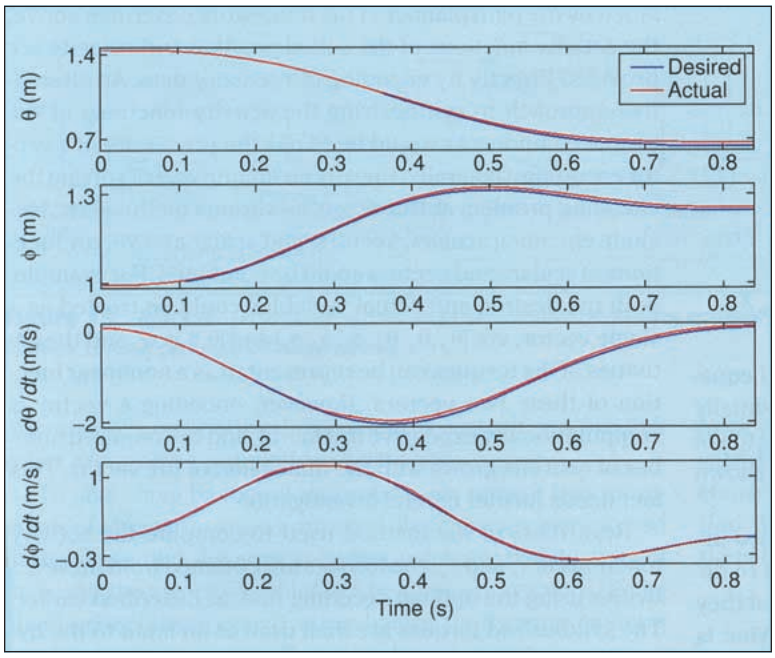


Figure 15. *The desired and actual trajectories of the two-link system in a horizontal plane.*

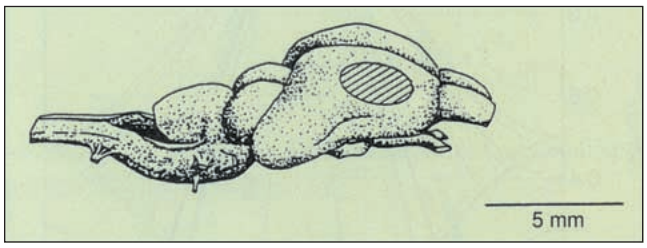


Figure 16. *Isolated turtle brain preparation. The shaded region corresponds to the visual cortex (courtesy of the Ulinski lab at the University of Chicago).*

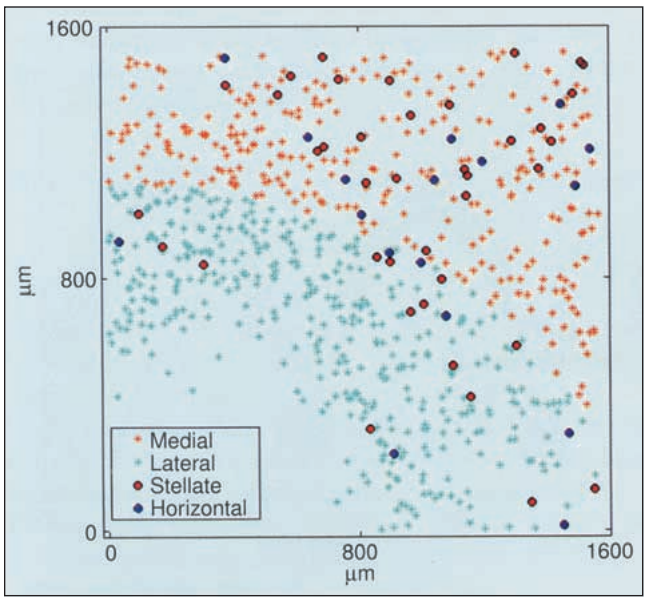


Figure 17. *The distribution of particular types of cells in a large-scale model of turtle visual cortex.*

namical system (8), and the simulation results showing both desired and actual trajectories are shown in Fig. 15.

To sum up, this section has illustrated how a population of neurons can be used to solve a differential equation and can be used to generate the required control signals, particularly for two-link robot arm manipulation. Drawbacks outlined in this section include the stability of the discretization procedure and a need for a careful analysis connecting the dimension of the variable to be encoded and the nature of the nonlinear function. As in the previous section, the dynamics of the spike generator and the low-pass filter have been ignored.

Propagating Waves in Visual Cortex of Freshwater Turtles

Having discussed the role of a population of neurons in an encoding and decoding circuit specifically designed for a control problem, we now turn our attention to a population of neurons naturally occurring in biology. A standard objection in the neuroscience community is that not all cells fire sufficiently many action potentials, suggesting that there are ways the neurons encode analog signals other than rate coding. To respond to this objection, we consider the visual cortex of a freshwater turtle (see Fig. 16). It has been experimentally shown that the cells of the turtle visual cortex fire very few action potentials in response to natural stimuli. Furthermore, studies using both multiple-electrode recordings and voltage-sensitive dye techniques demonstrate that visual stimuli produce waves of depolarizing activity that propagate across the cortex. The first evidence of such waves was reported by Mazurskaya [17], [18] using extracellular recording methods to study responses evoked in the visual cortex by presenting localized spots of light in the visual field of paralyzed turtles. The existence of a wave has also been demonstrated explicitly by Senseman [8], [19], who used voltage-sensitive dyes to record the spatiotemporal pattern of activity evoked by whole-field retinal flashes in the cortex of an isolated eye-brain preparation. The voltage-sensitive dye signal apparently corresponds to the spatially averaged membrane potentials of local populations of cortical pyramidal cells. It was demonstrated that different stimuli in the visual space produced different waves in the visual cortex, raising the possibility that the information about the stimulus is encoded in the spatiotemporal dynamics of the cortical response. Along the lines of our previous discussion, we can ask the following question: How do we decode or extract the information that is encoded in the propagating wave of activity? To answer this question, we constructed a large-scale model of turtle visual cortex and studied the response of such a network to both stationary and moving stimuli. We shall proceed by giving a brief description of the

turtle visual cortex as well as the large-scale model that we synthesized. For a detailed description of the cortex and model, see [10] and [20].

The visual cortex of freshwater turtles contains three layers. The intermediate layer 2 contains principally pyramidal (excitatory) cells with dendrites that extend into layers 1 and 3. Layer 1 (outer) and layer 3 (inner) contain mainly inhibitory interneurons. Lateral geniculate afferents (LGNs) make excitatory synapses upon the dendrites and somata of pyramidal cells in layer 2 and the dendrites and somata of stellate cells in layer 1. Our model consists of 700 pyramidal neurons (in layer 2), 40 inhibitory stellate neurons (in layer 1), and 20 inhibitory horizontal neurons (in layer 3), as shown in Fig. 17. Each neuron is represented by a multicompartment model. Each compartment is modeled by a standard membrane equation, and the parameters of individual compartments are constrained by experimental data. The visual input to the model enters through an array of 800 geniculate fibers. Even though the total number of cells in the model is much smaller than the number of cells in the real cortex, the ratio of particular types of cells is preserved. The coordinates of individual neurons are drawn randomly, obeying the densities of the cells that were found experimentally. The cells make different types of synaptic connections to their neighbors. The synaptic delays are incorporated in the model and are based on the conduction velocities of depolarizing waves and the distance between presynaptic and postsynaptic neurons.

The localized stationary stimulus is simulated by presentation of a square current pulse to a set of adjacent geniculate neurons. The family of stationary inputs is parametrized by the position of the center of the stimulus with respect to the geniculate complex. For our simulation, 20 equidistant positions of the stimulus were chosen across the LGN. Since the model has inherent randomness (the exact positions of the neurons in the three layers are drawn at random from a distribution), the repeated presentations of a stimulus produce different responses. For each of the 20 parameter values a set of 50 cortex samples has been generated. The response of the model to different inputs has been recorded and saved. The term *response* is to be understood as the membrane potential of the central compartment (soma) of the individual pyramidal (layer 2) neurons. Therefore, the response of the network to a specific stimulus can be viewed as a spatiotemporal signal $I(x,y,t)$ (see Fig. 18). Given that each frame is 64×64 pixels, and each movie has 201 frames, it is clear that the dimension of $I(x,y,t)$ is rather high. Furthermore, we are interested in

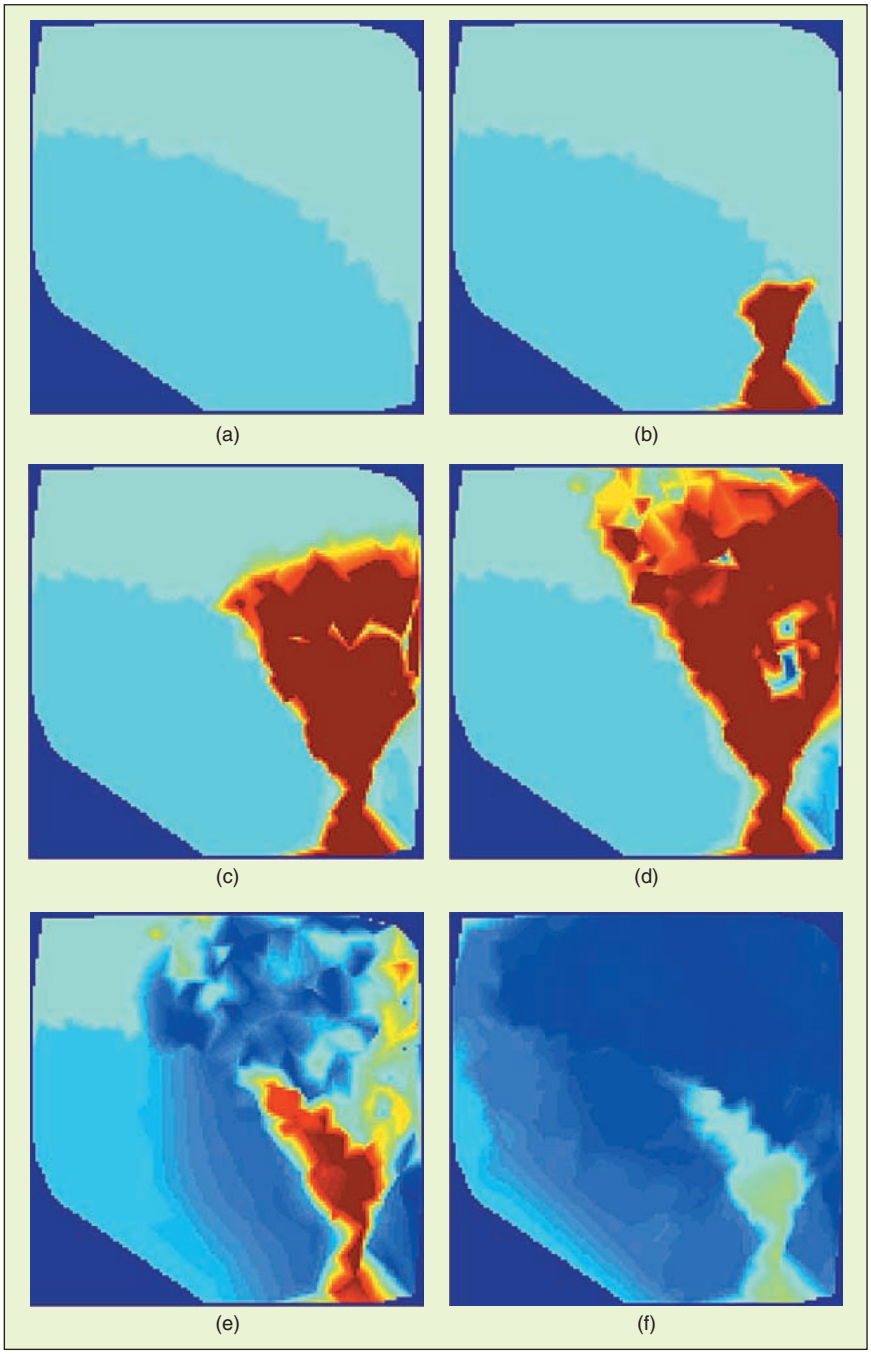


Figure 18. *Selected snapshots showing the origin and propagation of the wave of activity. The activity is color coded; light blue indicates the network at the resting potential, red indicates a high level of depolarization, and blue indicates a level of hyperpolarization. The dark blue region does not contain any cells.*

the comparison of two movies where the differences and similarities between them have to be quantified. Thus, an efficient way for movie comparison has to be developed.

Principal components analysis was introduced independently by many authors at different times. The method is widely used in various disciplines such as image and signal processing, data compression, fluid dynamics, partial differential equations, and weather prediction. Depending on the context, the method is also known as KL decomposition, proper orthogonal decomposition, Hotelling transformation, and singular-value decomposition. The transformation itself is linear and represents a rotation of a coordinate system, so that the data in the new coordinate system is less correlated. Therefore, the main action of the method is removing the redundancy from a data set, so that the data can be represented in a low-dimensional subspace. Instead of going deeper into the theory of principal components analysis, we shall briefly outline the main results. For a more rigorous mathematical treatment of the method, the reader is referred to [21] and [22]. In the new coordinate system, the movies can be viewed as an expansion

$$I(x,y,t)=\sum_{j=1}^{m}\alpha_j(t)M_j(x,y), \tag{19}$$

where $M_j(x,y)$ represent so-called principal eigenvectors or principal modes, and $\alpha_j(t)$ are the time coefficients, obtained by projecting the data set onto the jth principal mode.

The main advantage of the procedure is that the summation given by (19) can be significantly truncated (e.g., $m=3$), while the qualitative features of the movies are still preserved. Moreover, the eigenvectors $M_j(x,y)$ represent a so-called global basis and are common to all of the movies.

Thus, for $m=3$, the third-order approximation of the kth movie is given by

$$\hat{I}^k(x,y,t)=\alpha_1^k(t)M_1(x,y)+\alpha_2^k(t)M_2(x,y)+\alpha_3^k(t)M_3(x,y),$$

and the difference between two movies is captured in the time coefficients. A typical phase plot of those coefficients for three different movies is given by Fig. 19.

Because of the inherent randomness in the system, the vector function $[\alpha_1(t),\alpha_2(t),\alpha_3(t)]$ can be viewed as a random process. Statistical analysis of a random process can be facilitated if the process is parametrized using a second KL decomposition. Analogous to the procedure outlined above, the random process $[\alpha_1(t),\alpha_2(t),\alpha_3(t)]$ can be approximated as

$$\begin{bmatrix}\alpha_1(t)\\ \alpha_2(t)\\ \alpha_3(t)\end{bmatrix}=\beta_1\varphi_1(t)+\beta_2\varphi_2(t)+\beta_3\varphi_3(t),$$

where $\varphi_j(t)$ are the basis vectors and β_j are the coefficients obtained by the orthogonal projection of the random vector $[\alpha_1(t),\alpha_2(t),\alpha_3(t)]$ on the basis vectors $\varphi_j(t)$. Thus, the entire spatiotemporal signal $I(x,y,t)$ can be represented as a point $[\beta_1,\beta_2,\beta_3]$ in the three-dimensional Euclidean space. In Fig. 20, we show clusters of points indicating the representation of the cortical response to stimuli presented at different locations in the visual space and for different samples of the cortex. The points appear in a natural cluster, indicating that the turtle visual cortex can discriminate among various positions of the input stimuli.

The clusters can be used in building a statistical description of the cortical response given the position of a localized stationary stimulus. In particular, conditional probability density functions have been fit through the cluster data. We then used standard Bayesian and maximum likelihood methods for designing a detection algorithm that can predict the position (unknown parameter) of an input based on the cortical response elicited by that input. The results indicate that the position of an unknown light source can successfully be detected based on a third-order KL expansion

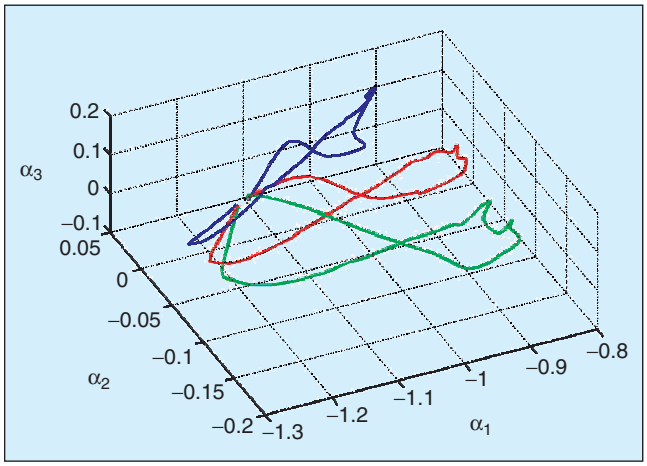


Figure 19. *Spatiotemporal signal in the cortex can be represented accurately using three eigenvectors. The phase plot of the response of the cortex to three different stimuli.*

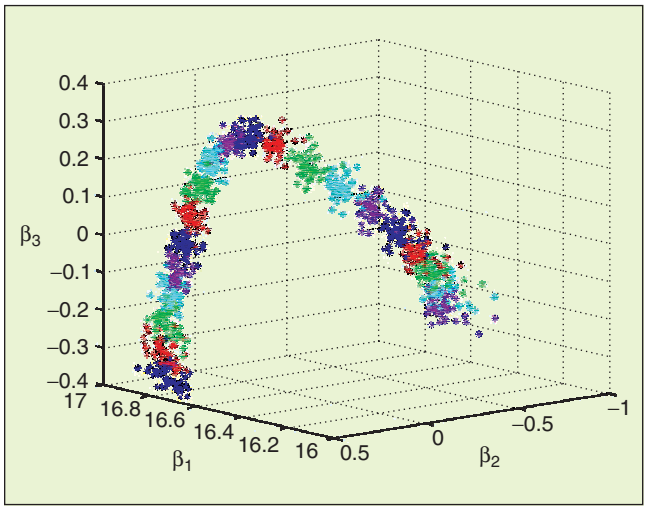


Figure 20. *Clusters of points in the space of coefficients indicate that the model of the visual cortex is able to discriminate among 20 locations of the input stimulus in the visual space.*

of the cortical response. A similar conclusion holds for the estimation of an unknown velocity parameter of a moving stimulus (see [10] for details).

Summary and Conclusion

The main topic of this article is the role of a population of neurons as opposed to a single neuron. Concrete results were presented as to how neurons, as an ensemble, can perform computations and can therefore be used to generate control signals. Likewise, motivated by concrete observations in the turtle cortex, we show that a population of interacting neurons can sustain a wavelike activity that can discriminate the geometric position of the incoming stimulus in the visual space. The examples chosen in this article are merely for illustration and reflect the research interests of the authors. A neural population plays many other nontrivial roles, many details of which are only now being discovered.

Acknowledgment

We gratefully acknowledge Prof. C.H. Anderson, from whom we learned about the role of a population of neurons in encoding analog signals. We also gratefully acknowledge Prof. P.S. Ulinski, from whom we learned many important details about the turtle visual cortex. Finally, we would like to acknowledge the anonymous reviewers and the editorial staff of this *Magazine*.

References

[1] A.L. Hodgkin and A.F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, pp. 500-544, 1952.

[2] H.R. Wilson, *Spikes, Decisions, and Actions: The Dynamical Foundation of Neuroscience*. New York: Oxford Univ. Press Inc., 1999.

[3] H.R. Wilson, "Simplified dynamics of human and mammalian neocortical neurons," *J. Theor. Biol.*, vol. 200, pp. 375-388, 1999.

[4] C. Eliasmith and C.H. Anderson, "Developing and applying a toolkit from a general neurocomputational framework," *Neurocomputing*, vol. 26-27, pp. 1013-1018, 1999.

[5] C. Eliasmith and C.H. Anderson, "Rethinking central pattern generators: A general approach," *Neurocomputing*, vol. 32-33, pp. 735-740, 2000.

[6] F. Rieke, D. Warland, R. de Ruyter, V. Steveninck, and W. Bialek, *Spikes, Exploring the Neural Code*. Cambridge: MIT Press, 1997.

[7] Z. Nenadic, B.K. Ghosh, and P.S. Ulinski, "Spatiotemporal dynamics in a model of turtle visual cortex," *Neurocomputing*, vol. 32-33, pp. 479-486, 2000.

[8] D.M. Senseman, "Correspondence between visually evoked voltage-sensitive dye signals and synaptic activity recorded in cortical pyramidal cells with intracellular microelectrodes," *Visual Neurosci.*, vol. 13, pp. 963-977, 1996.

[9] H.L. Van Trees, *Detection, Estimation and Modulation Theory, Part I*. New York: Wiley, 1968.

[10] Z. Nenadic and B.K. Ghosh, "Modeling and estimation problems in the turtle visual cortex," *IEEE Trans. Biomed. Eng.*, 2001.

[11] F. Delcomyn, *Foundations of Neurobiology*. New York: Freeman, 1998.

[12] J. Keener and J. Sneyd, *Mathematical Physiology*. New York: Springer-Verlag, 1998.

[13] M.A. Arbib, P. Erdi, and J. Szentagothai, *Neural Organization: Structure, Function, and Dynamics*. Cambridge: MIT Press, 1998.

[14] Z. Nenadic and B.K. Ghosh, "Computation with biological neurons," in *Proc. 20th Amer. Control Conf.*, 2001.

[15] A. Isidori, *Nonlinear Control Systems: An Introduction*. Berlin: Springer-Verlag, 1989.

[16] Z. Nenadic and B.K. Ghosh, "Tracking of a two-link arm with a population of neurons," in *Proc. 39th IEEE Conf. Decision and Control*, Sydney, Australia, 2000, pp. 1776-1781.

[17] P.Z. Mazurskaya, T.V. Davydova, and G.D. Smirnov, "Functional organization of exteroceptive projections in the forebrain of the turtle," *Neurosci. Trans.*, vol. 1, pp. 109-117, 1967.

[18] P.Z. Mazurskaya, "Organization of receptive fields in the forebrain of *Emys orbicularis*," *Neurosci.*, vol. 7, pp. 311-318, 1974.

[19] D.M. Senseman, "Spatiotemporal structure of depolarization spread in cortical pyramidal cell populations evoked by diffuse retinal flashes," *Visual Neurosci.*, vol. 16, pp. 65-79, 1999.

[20] Z. Nenadic, P.S. Ulinski, and B.K. Ghosh, "Propagating waves in visual cortex: A large scale model of turtle visual cortex," *Neurocomputing*, to be published.

[21] L. Sirovich, Ed., *Trends and Perspectives in Applied Mathematics*. New York: Springer-Verlag Inc., 1994.

[22] P. Holmes, J.L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

Zoran Nenadic received his B.S. degree in control engineering from the University of Belgrade, Yugoslavia, in 1995 and the M.Sc. degree in systems science and mathematics from Washington University in 1998. His research interests are in the area of computational neuroscience, image processing, and applications of control theory in biological systems. He is currently pursuing a D.Sc. degree in the Department of Systems Science and Mathematics at Washington University.

Bijoy K. Ghosh received his M.Tech. degree in electrical engineering from IIT Kanpur, India, in 1979 and the Ph.D. degree in engineering from Harvard University in 1983. He is currently a Professor in the Department of Systems Science and Mathematics and Director of the Center for Biocybernetics and Intelligent Systems at Washington University. His research interests are in the areas of multisensor fusion and automation, biosystems and control, and perspective problems in machine vision. He is a recipient of the Donald Eckman award (1998) in recognition of his outstanding contributions in the field of automatic control. He is a Fellow of the IEEE.