Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE @ DIRECT®

Mathematical and Computer Modelling 39 (2004) 181–196

MATHEMATICAL
AND
COMPUTER
MODELLING

www.elsevier.com/locate/mcm

# Encoding and Decoding of Analog Signals with a Population of Neurons

Z. NENADIĆ AND B. K. GHOSH
Department of Systems Science and Mathematics
Washington University, Campus Box 1040
St. Louis, MO 63130-4899, U.S.A.
ghosh@netra.wustl.edu

(*Received August 2002; accepted July 2003*)

**Abstract**—In this paper, we discuss how an analog signal can be encoded using biophysically realistic neural networks. Using the activity curve of a single neuron, we argue that the activities can be pooled over a population so that the weighted sum of the activities approximate a given function. Since the activities of neurons are not available as a variable, we propose to generate them in real time by a suitable low-pass filter. Using the proposed scheme, we demonstrate how simple ordinary differential equations can be solved. In effect, the ordinary differential equations are solved by dynamically updating the activities of the neurons. In an actual *biological neural network*, the activities of the cells are not obtained by a low-pass filter. They are integrated in the network by a suitable synaptic input. A new optimization algorithm for finding a set of optimal synaptic weights has been proposed and successfully implemented using a software package GENESIS. The difference between biological neural networks and artificial neural networks is discussed in somewhat greater details. The important concepts are illustrated by implementing a *memory* and by solving a periodic ordinary differential equation, the *Van der Pol oscillator*. © 2004 Elsevier Ltd. All rights reserved.

**Keywords**—Neural networks, Neuroscience, Rate coding, Limit cycles, Oscillations.

## 1. INTRODUCTION

The field of computational neuroscience has experienced a tremendous expansion over the last several decades. Ever since the fundamental work of Hodgkin and Huxley [1] in the early 1950s, that established computational neuroscience as a discipline, scientists have been trying to attach a meaning to the process of neural "computation". Even though, the "exact" mechanism of information processing at the level of neurons is still arguable, there have emerged several viable theories. One of them proposes that the information is encoded in a spike train, where in response to an excitatory synaptic stimulus, the neuron fires a series of action potentials [2]. Others claim that not all neurons are equipped with a spike generating mechanism, and even if they are, they fire very few action potentials, suggesting, therefore, that such a sparse representation cannot be

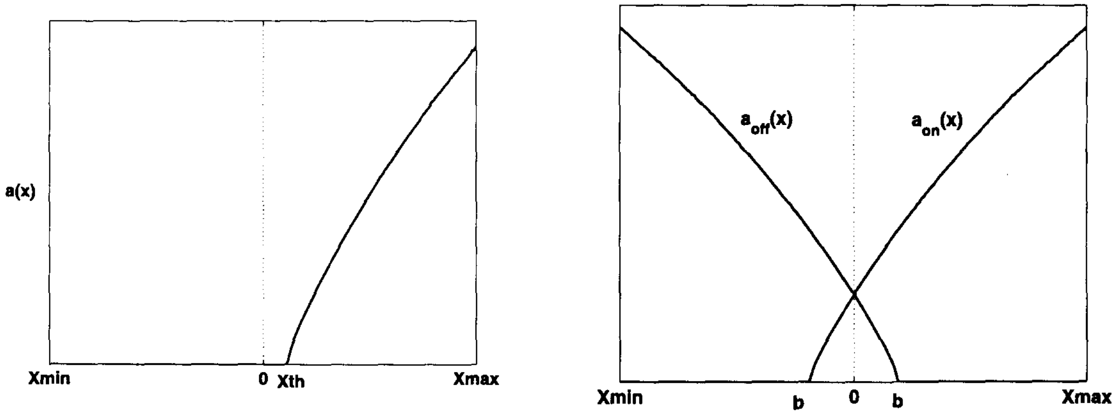
Supported by NSF, Learning and Intelligent Systems Program, under Grant Number 9720357.
The main idea of using *on/off* cells for solving a computational problems was introduced to us by C. H. Anderson of Washington University, St. Louis and we would like to thank him for his help. The idea behind modeling a biophysically realistic neuron was introduced to us by P. Ulinski who also encouraged us to use GENESIS as a simulation package.

doi: 10.1016/S0895-7177(04)00008-1

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

viewed as a frequency code. This theory proposes that the timing of spike occurrence, together with a synchronization (simultaneous or correlated firing of a group of neurons), is responsible for encoding and information processing [3]. Finally, a recently developed theory suggests that it is a nonlinear action at the level of a neuron dendritic tree that is responsible for a rich class of information processing [4]. Even though these theories do not necessarily contradict each other, this article will focus only on the first one.

The fact that neurons can encode analog, time-dependent signals into spike trains has been known for several decades [5]. Roughly speaking, a current input into the membrane of a neuron produces a "spiky" signal of the membrane potential. Moreover, the frequency of the spikes increases monotonically with the amplitude of the injected current. Therefore, an analog input signal gets encoded (modulated) into a spike train, and this is often referred to as *rate coding* [2]. For a constant current input $x$ to a neuron, the membrane potential has a constant frequency of spiking, which we shall denote by $a(x)$. This quantity we shall refer to as the activity of a neuron, although other names are also commonly used, e.g., response tuning curve [6]. A typical activity curve is shown in Figure 1.



(a) Activity curve of an *on* cell.

(b) Activity curve of an *on/off* pair.

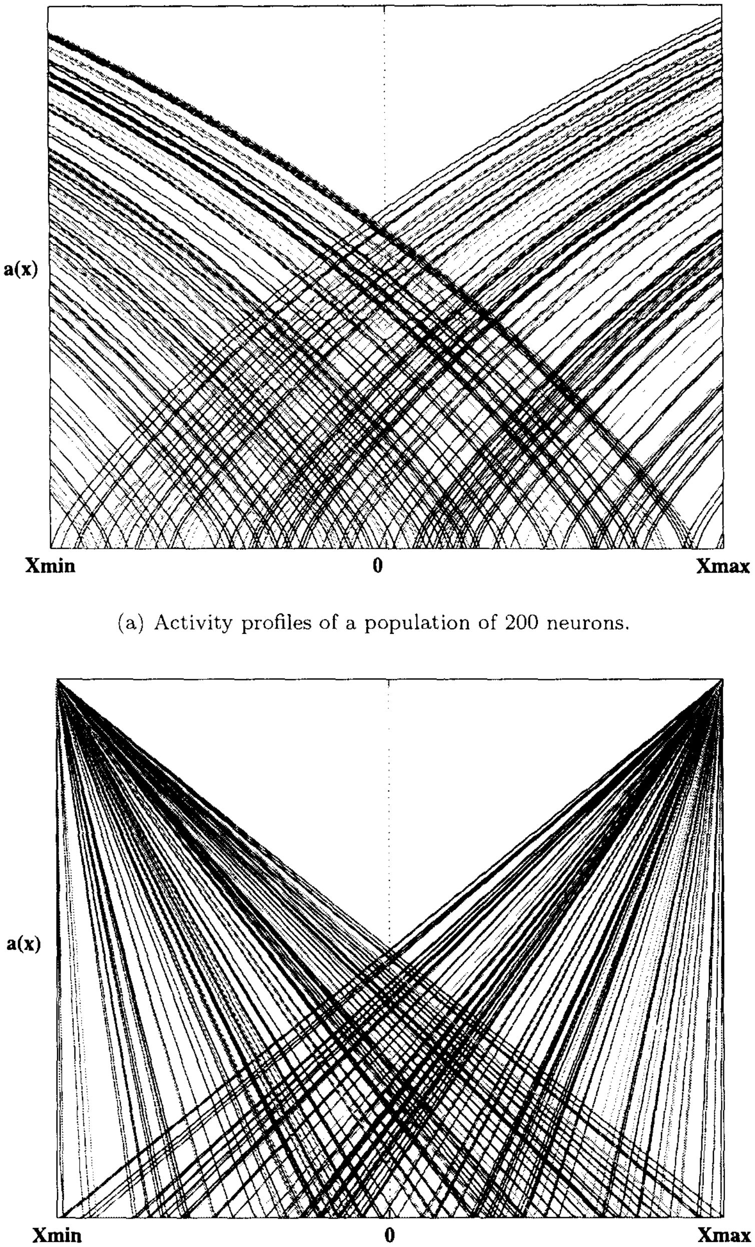
Figure 1.

Note that a neuron produces spikes if the amplitude of the current exceeds the threshold $x_{\text{th}}$. Since a single neuron can only encode signals whose amplitudes exceed the threshold, for encoding signals taking both positive and negative values, we use the concept of *on/off* cells [7], where *on* cells are primarily responsible for encoding the positive part of the signal and *off* cells are encoding the negative part of the signal. The activity curve of an *on/off* pair is shown in Figure 1. Note that the two activity curves are shifted by a bias, indicating that the neurons can fire even if no external stimulus is applied. This common biological concept is called *tonic activity* or *background firing rate*. To conclude the section, we shall point out that the signal can be decoded (recovered) once it is encoded in a single *on/off* pair. For a more elaborate discussion on this subject, we refer to [8].

## 2. COMPUTATION WITH A POPULATION OF CELLS

The scheme of representing a signal, as described above, can be generalized in several important ways. Instead of considering a single *on/off* pair, one can introduce a population of on and *off* cells. The population can be regarded as a larger number of neurons, whose activity curves have a random bias ranging from $x_{\min}$ to $x_{\max}$ and have the character of both on and *off* cells. The activity profiles of a population of 200 neurons are shown in Figure 2. For computational efficiency, a piecewise linear approximation of the activity profiles is often used (Figure 2).

Generally, a population of $N$ neurons can be represented by a set of activity curves $a_j(x)$, $j = 1, 2, \ldots, N$. Suppose that one such population encodes an analog variable $x$. Also assume



(a) Activity profiles of a population of 200 neurons.

(b) Piecewise linear approximation of the activity profiles.

Figure 2.

that we are interested in generating an arbitrary function $f(x)$ based on the population response. Then, a question as to what would be a good decoding scheme naturally arises. Despite the encoding process which is nonlinear, we propose to obtain such a function as a linear (weighted) sum of the corresponding activities.

PROBLEM 1. Given a function $f(x)$ and a positive integer $N$, find a set of $N$ real numbers $w_j$, such that the error function defined by

$$E\left(w_1, w_2, \ldots, w_N\right) = \left\langle \int_{x_{\min}}^{x_{\max}} \left[f\left(x\right) - \hat{f}\left(x\right)\right]^2 dx \right\rangle_{\eta} \tag{1}$$

is minimized, where

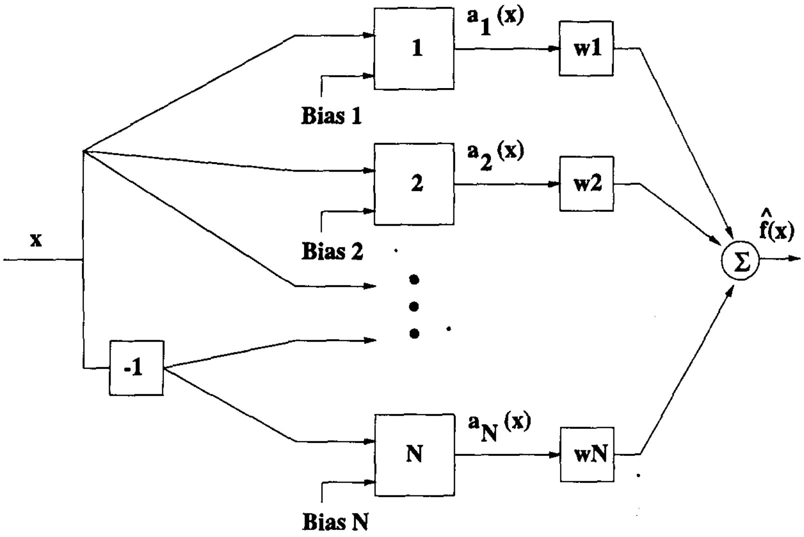
$$\hat{f}\left(x\right) = \sum_{j=1}^{N} w_j \left[a_j\left(x\right) + \eta_j\right].$$

The symbol $\langle\cdot\rangle_\eta$ stands for the average over noise and $\eta_1, \eta_2, \ldots, \eta_N$ is the sequence of independent, identically distributed random variables with zero mean and known variance $\sigma^2$, and represent additive uncertainties imposed on individual neurons. For the source of noise within a single neuron, the reader is referred to [2]. The problem of minimizing the error in (1) and finding the optimal weights reduces to solving a normal set of equations. Once the weights are found, the best approximation of the function $f(x)$ is found by

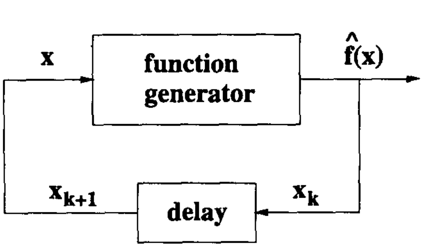
$$\hat{f}(x) = \mathrm{EX}\left\{\sum_{j=1}^{N} w_j \left[a_j(x) + \eta_j\right]\right\} = \sum_{j=1}^{N} w_j a_j(x), \tag{2}$$

where the symbol EX stands for the expectation operator. Equation (2) will frequently be referred to as the *linear decoding rule*. The block diagram corresponding to this scheme is shown in Figure 3. It should be noted that there is no actual difference between the *on* and *off* cells, except that the input has been inverted prior to entering the *off* cells. There are many different applications that result out of Problem 1. In particular, we show how this network can be used to solve an important class of ordinary differential equation, *viz.*, the Van der Pol oscillator. Note that using the function generator in Figure 3, one can easily obtain a scheme that would solve a difference equation of the form

$$x_{k+1} = f(x_k).$$



(a) The structure of the network to estimate $f(x)$.



(b) The schematic diagram of a difference equation solver.

Figure 3.

The corresponding scheme is shown in Figure 3. The Van der Pol equation (3) provides an example of an oscillator with a nonlinear damping, wherein the energy is dissipated at large amplitudes and generated at low amplitudes. Such systems typically possess limit cycles; sustained oscillations around a state at which energy generation and dissipation balance [9]. We now propose to solve a two-dimensional Van der Pol equation of the form

$$\begin{gathered}\dot{x}_1 = x_2, \\ \dot{x}_2 = \mu\left(1 - x_1^2\right) x_2 - x_1, \\ x_1(0) = x_{10}, \qquad x_2(0) = x_{20}, \qquad \mu > 0.\end{gathered} \tag{3}$$

The basic idea behind solving (3) is to have three populations of neurons represented by the corresponding activities $a_i(x_1)$, $b_j(x_2)$, and $c_k(x_1)$. Using the procedure outlined above, the three functions $f(x_1) = x_1$, $g(x_2) = x_2$, and $h(x_1) = 1 - x_1^2$, can be represented as

$$\begin{aligned}f(x_1) &= \sum_{i=1}^{N} w_i^f a_i(x_1), \\ g(x_2) &= \sum_{j=1}^{M} w_j^g b_j(x_2), \\ h(x_1) &= \sum_{k=1}^{P} w_k^h c_k(x_1),\end{aligned} \tag{4}$$

where the weights $w_i^f$, $w_j^g$, and $w_k^h$ are chosen as outlined in Problem 1, and $N$, $M$, and $P$ represent the total number of neurons in each of the three populations. Given the piecewise activity profiles of the three populations, the activities can be expressed analytically

$$\begin{aligned}a_i(x) &= [\alpha_i x + \beta_i]_+, \\ b_j(x) &= [\gamma_j x + \delta_j]_+, \\ c_k(x) &= [\xi_k x + \eta_k]_+,\end{aligned} \tag{5}$$

where

$$[x]_+ = \begin{cases} x, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0, \end{cases}$$

and $\alpha_i$, $\gamma_j$, and $\xi_k$ are the slopes of the piecewise linear activity profiles. Similarly, the variables $\beta_i$, $\delta_j$, and $\eta_k$ are connected to value of bias (intercept). Discretizing two differential equations of system (3) yields

$$\begin{aligned}x_1(t+\tau) &= x_1(t) + \tau x_2(t), \\ x_2(t+\tau) &= x_2(t) + \tau\left[\mu\left(1 - x_1^2(t)\right) x_2(t) - x_1(t)\right].\end{aligned} \tag{6}$$

Using equations (4)–(6), system (3) can be equivalently written as

$$a_n(t+\tau) = \left[\alpha_n\left(\sum_{i=1}^{N} w_i^f a_i(t) + \tau \sum_{j=1}^{M} w_j^g b_j(t)\right) + \beta_n\right]_+, \tag{7}$$

for all $n = 1, 2, \ldots, N$, and

$$b_m(t+\tau)=\left\{\gamma_m\left[\sum_{j=1}^{M} w_j^g b_j\,(t) + \tau\mu\left(\sum_{k=1}^{P} w_k^h c_k\,(t)\right)\left(\sum_{j=1}^{M} w_j^g b_j\,(t)\right) - \tau\sum_{i=1}^{N} w_i^f a_i\,(t)\right]+\delta_m\right\}_+ , \qquad (8)$$

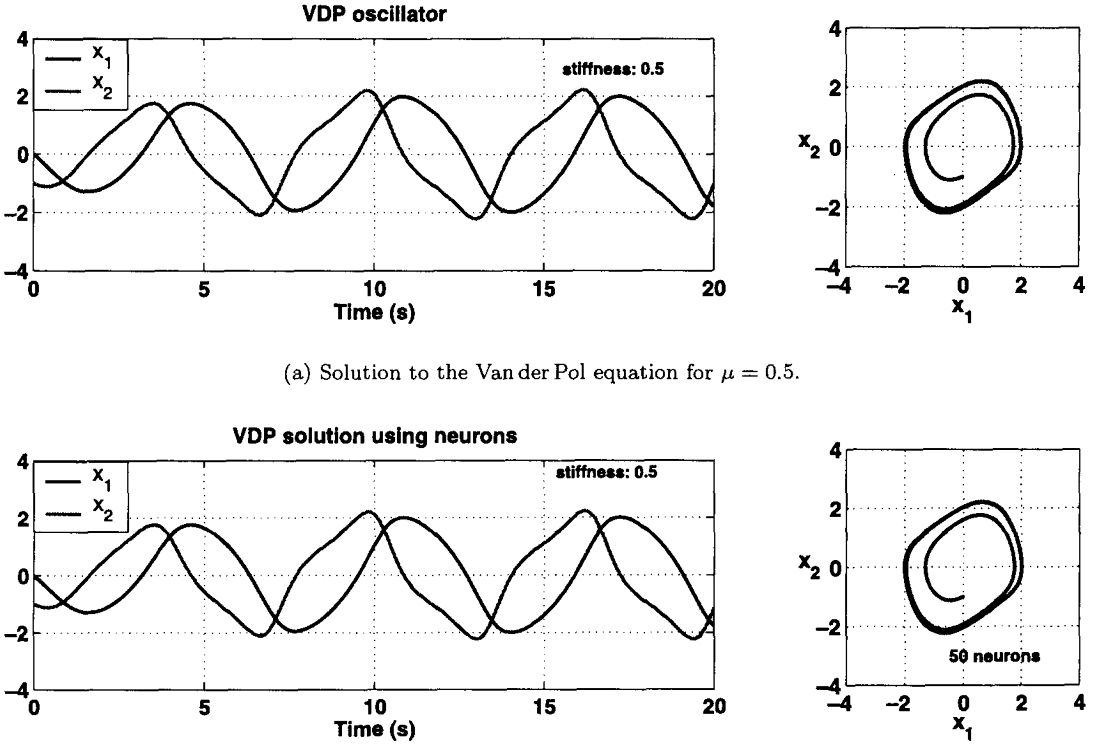
for all $m = 1, 2, \ldots, M$. Finally, from (4)–(6), it follows that the activity $c_p(t)$ of the function $h(x_1)$ satisfies the following:

$$c_p\,(t+\tau) = \left\{\xi_p\left[\sum_{k=1}^{P} w_k^h c_k\,(t) - 2\tau\xi_p\left(\sum_{i=1}^{N} w_i^f a_i\,(t)\right)\left(\sum_{j=1}^{M} w_j^g b_j\,(t)\right)\right]+\eta_p\right\}_+ , \qquad (9)$$

for all $p = 1, 2, \ldots, P$. We remark that the triplet of equations (7)–(9), called the *activity dynamics*, is an internal representation of the Van der Pol equation (3). These dynamics have been originally introduced by Anderson [10,11]. In Figure 4, we show the solution to equation (3) for $\mu = 0.5$. The same figure also shows the solution to the Van der Pol equation using the activity dynamics (7)–(9) and the solutions $x_1(t)$ and $x_2(t)$ are synthesized from the activities using the linear decoding rule

$$x_1\,(t) = \sum_{i=1}^{N} w_i^f a_i\,(t), \qquad x_2 = \sum_{j=1}^{M} w_j^g b_j\,(t).$$

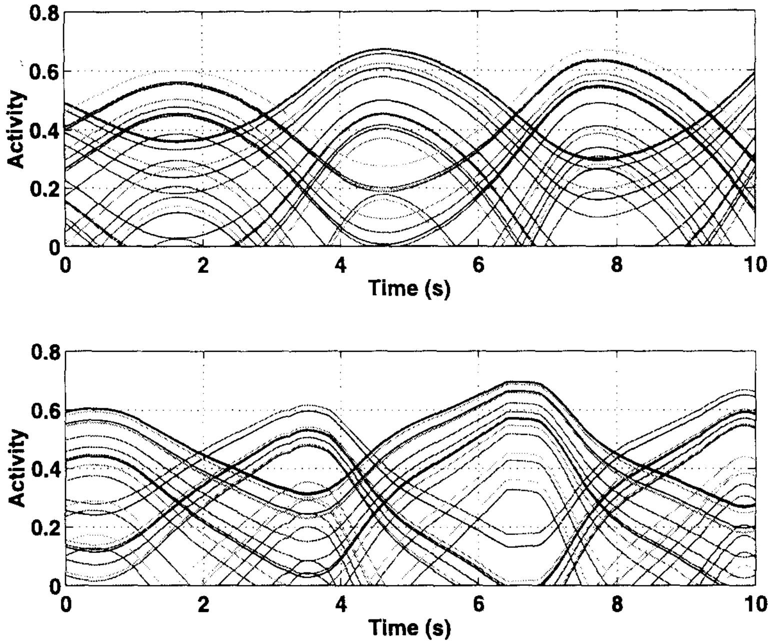
The number of neurons in the populations is chosen to be $N = M = P = 50$. The activity functions $a_n(t)$, $b_m(t)$ are also shown for $n, m = 1, 2, \ldots, 50$.



(a) Solution to the Van der Pol equation for $\mu = 0.5$.

(b) Solution using a population of neurons.

Figure 4.



(c) The activities of $x_1$ and $x_2$ as a function of time.

Figure 4. (cont.)

# 3. COMPUTATION WITH A POPULATION OF SPIKING NEURONS

We have already remarked on the role of spiking cells in a neural population. In fact, the activity curves of a population of cells, as shown in Figure 2, are not assumed to be *a priori* available during any computation. What is readily available are the spike trains generated by the cells as a result of a certain input signal. Therefore, the activities as defined previously, have to be synthesized from the spike trains. Given that the spike train is a discontinuous signal (a sequence of Dirac functions), and that the activities are continuous, it is clear that the action from the spike train to the activity has to be of integrative nature. Such an action can be obtained by low-pass filtering of the spike train

$$a(x,t) = \int_0^t s(x,\tau)\, h(t-\tau)\, d\tau, \tag{10}$$

where $s(x,t)$ is the spike train of the cell as a response to a constant input signal $x$, and $h(t)$ is the impulse response of a suitably chosen low-pass filter. For a constant input signal $x$, the activity $a(x,t)$ saturates to a constant value as $t \to \infty$, and that constant value is proportional to the frequency of the spike train. Thus, $a(x,t)$ can be viewed as the time varying version of the activity $a(x)$, hence, the notation. Further generalization of the *on/off* cell concept leads to the following problem.

PROBLEM 2. Given a function $f(x)$, a low-pass filter with impulse response $h(t)$ and a positive integer $N$, find a set of $N$ real numbers $w_j$, such that the error function defined by

$$E(w_1, w_2, \dots, w_N) = \left\langle \int_0^T \int_{x_{\min}}^{x_{\max}} \left[ f(x) - \hat{f}(x,t) \right]^2 dx\, dt \right\rangle_\eta \tag{11}$$

is minimized, where analogously to Problem 1,

$$\hat{f}(x,t) = \sum_{j=1}^{N} w_j \left[ h(t) * s_j(x,t) + \eta_j \right], \tag{12}$$

where the symbol $*$ stands for the convolution operator as defined by (10).

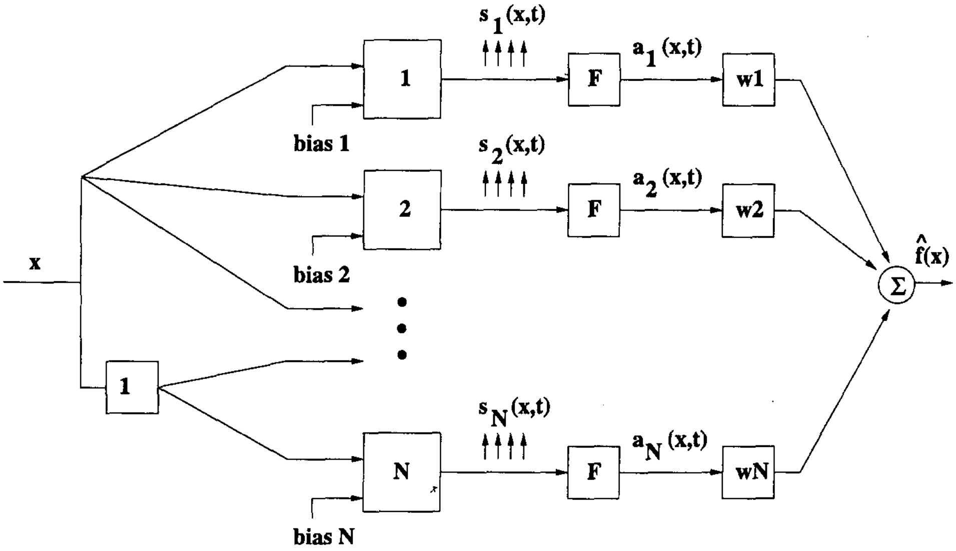


Figure 5. The structure of the network to estimate $f(x)$ in Problem 2.

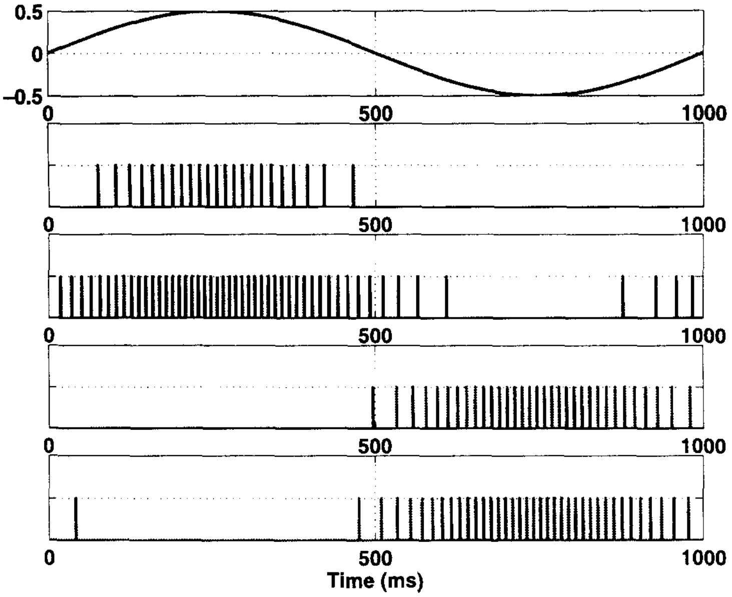
The schematic diagram of the corresponding circuit is shown in Figure 5. We also remark that even though the impulse response $h(t)$ has been fixed, one can minimize the cost function (11) with respect to the filter parameters [8,12]. Since Problem 2 is a generalization of Problem 1, their applications are identical, e.g., the same types of differential equations can be solved [13]. Note that even though the weights found in (11) are declared optimal, they actually pertain to a specific class of input signals. Namely, in the minimization process, we use the spike trains $s_j(x,t)$ elicited by a family of constant inputs, discretized properly in the interval $[x_{\min}, x_{\max}]$. Given that the network has been *trained* to the specific class of inputs, we may question the performance of such a network for an input that does not belong to the training class. As a concrete example of an application of Problem 2, we consider how a time varying signal, such as a sine wave, can be reconstructed from the spike trains of a population of spiking neurons. We consider a population of $N$ neurons. The spike generating mechanism of individual neurons is implemented as *integrate and fire* model [14]. We consider a *bona fide* first-order linear filter $F$ with the transfer function given by

$$H(s) = \frac{k}{Ts+1}.$$

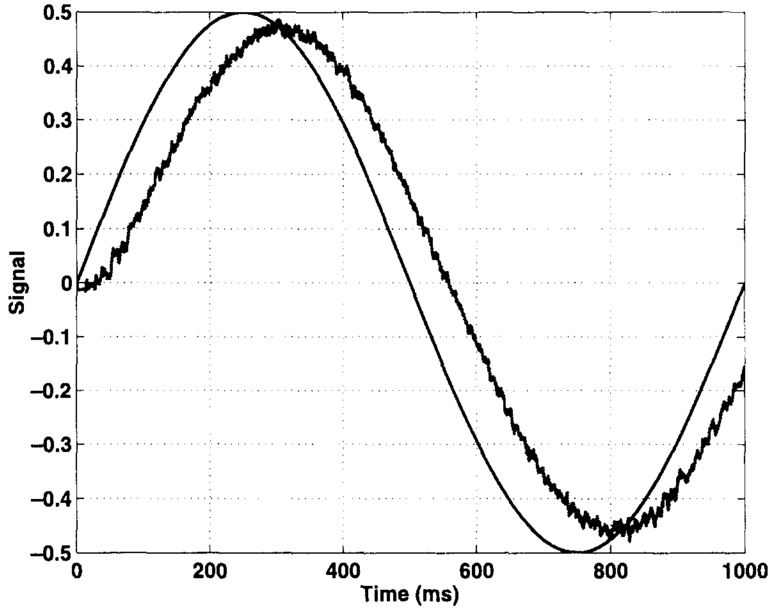
The weights $w_1, \ldots, w_N$ are calculated by minimizing (11), assuming $f(x) = x$. Let $x(t) = A\sin(\omega t)$ be a sine wave of interest, then its reconstruction follows from (12)

$$\hat{x}(t) = \sum_{j=1}^{N} w_j \left[h(t) * s_j(x(t), t)\right].$$

The above procedure has been tested for $N = 50$. The original signal with $A = 0.5$ and $\omega = 2\pi\,\text{rad/sec}$ and selected spike trains are shown in Figure 6. Note that the first two of them correspond to the *on* cells and the second two are the spike trains of the *off* cells. The spike patterns are different because of the random bias that was imposed on each neuron. The filter parameters are chosen as $k = 25$ and $T = 60\,\text{ms}$. The original wave and its reconstruction are also shown in Figure 6. Note the characteristic phase shift and amplitude attenuation, which indicates that the circuit (see Figure 5) might possess the linearity property, despite the nonlinear transformations that the input signal undergoes.



(a) Spike trains of four cells in relation to the sine wave, the top two are *on* cells and the bottom two are *off* cells.

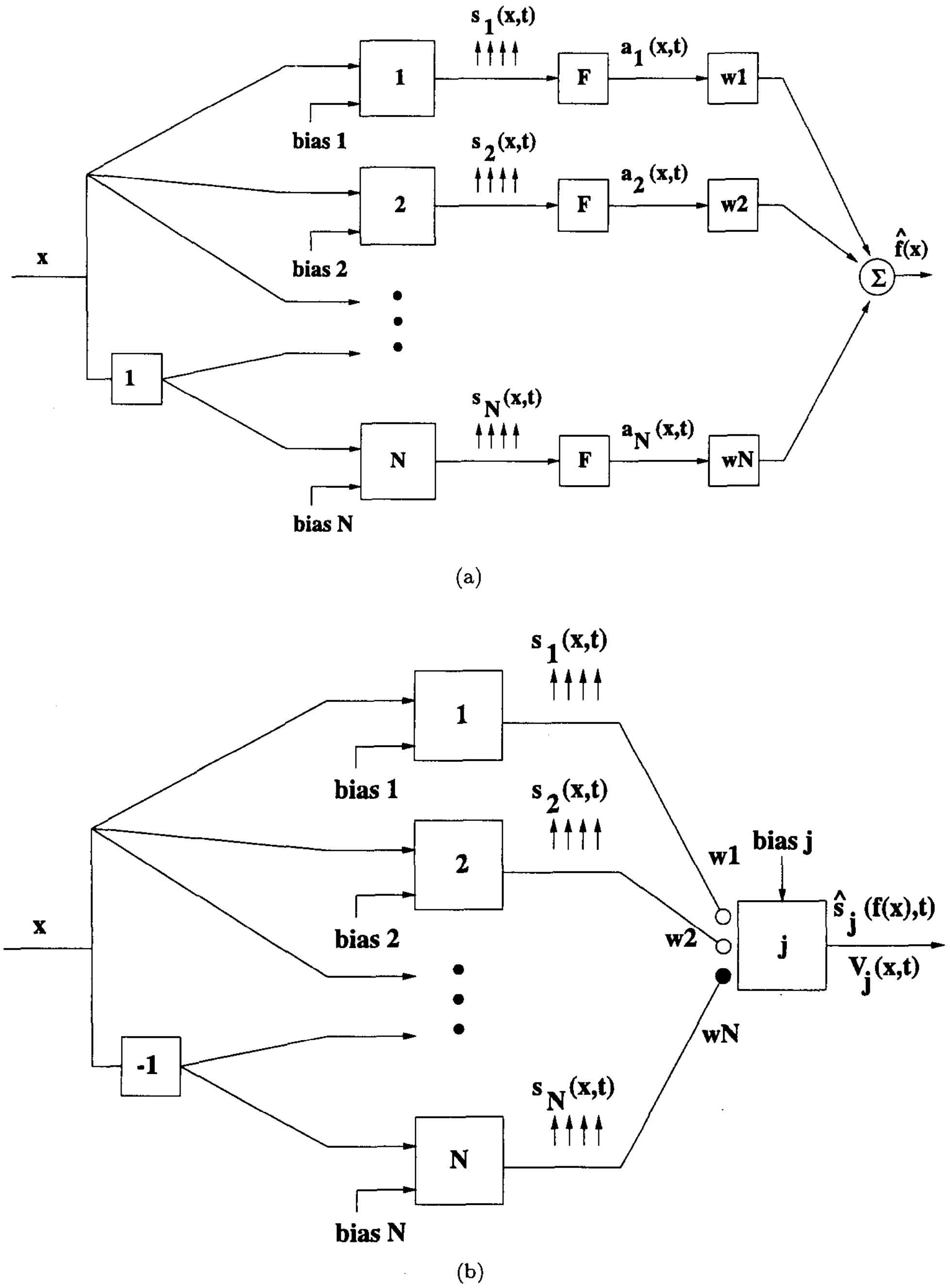


(b)

Figure 6. Decoding a sine wave from a population of 50 spiking neurons, the smooth function is the original waveform and the nonsmooth function is the reconstructed signal.

## 4. COMPUTATION WITH A POPULATION OF BIOLOGICAL NEURONS

Even though Problem 2 utilizes a spike generating cell and linear filter (see Figure 5) and seems general enough, it still lacks some of the properties of biologically realistic neural networks, which in the interest of simplicity we will refer to as biological neural networks, BNN. In a biological ensemble of neurons, such as the visual cortex, there are no natural candidates for a linear filter. Likewise, there does not exist an explicit weight variable either. Generally, in the theory of neural networks and neuroscience, it is very common to associate the concept of weights with synaptic weights. Moreover, the synaptic conductance is very often modeled through a linear filter. Up to a first approximation, synapses indeed have an integrative (filtering) action on a spike train [15]. Thus, in the formulation of Problem 2, we see some rudimentary properties of BNN. However, we shall point out some important differences. While in Problem 2 we have no constraints on the weights $w_j$, the concept of negative synaptic weight is meaningless, as the synaptic weight represents the coupling strength between two cells. In order to compensate for eventual negative weights, we use a natural biological tool which is known as *inhibitory synapse*. Furthermore, the filtered spike train represents only the synaptic conductance, but not the synaptic input (current) itself. A BNN counterpart of the network associated with Problem 2 is shown in Figure 7. The white dots indicate excitatory synaptic connections and the black dots represent inhibition.



(a)

(b)

Figure 7. A BNN with $N$ presynaptic cells and one postsynaptic cell (b).

The primary purpose of such a network is to be able to encode an arbitrary function $f(x)$ in its postsynaptic cell, i.e., one should be able to recover the function $f(x)$, based on the spike train $\hat{s}_j(f(x),t)$, possibly by low-pass filtering. The symbol ^ indicates that this spike train is not obtained by a direct encoding of $f(x)$, but through a synaptic action and merely suggests that the concept of weighted sum is to be used again. Our goal is to match the activity of the spike train $\hat{s}_j(f(x),t)$ to the activity of the spike train $s_j(f(x),t)$, obtained by stimulating the $j^{\text{th}}$ postsynaptic cell directly by the analog signal $f(x)$. Therefore, we can write

$$\hat{a}_j\left(f\left(x\right),t\right) = a_j\left(f\left(x\right),t\right). \tag{13}$$

In the interest of clarity, let us restrict our attention to a simple function $f(x) = x$. Then, equation (13) can be rewritten as

$$\hat{a}_j\left(x,t\right) = a_j\left(x,t\right). \tag{14}$$

The net synaptic current entering the $j^{\text{th}}$ postsynaptic cell is given as the weighted sum of the synaptic currents coming from each presynaptic neuron

$$I_j^{\text{syn}}\left(x,t\right) = \sum_{i=1}^{N} w_{ji} I_i\left(x,t\right).$$

The individual synaptic currents exhibit the basic form of Ohm's law

$$I_i\left(x,t\right) = g_i\left(x,t\right)\left(E_i - V_j\left(x,t\right)\right),$$

where $E_i$ is the so-called synaptic reversal potential [15] and $V_j(x,t)$ is the membrane potential of the postsynaptic neuron [16]. As indicated earlier, the synaptic conductance can simply be modeled as

$$g_i\left(x,t\right) = h\left(t\right) * s_i\left(x,t\right),$$

where $h(t)$ is the impulse response of a linear filter, usually first or second order. Therefore, the net synaptic current entering the $j^{\text{th}}$ cell is given by

$$I^{\text{syn}}(x,t) = \sum_{i=1}^{N} w_i\left[h\left(t\right) * s_i\left(x,t\right)\right]\left(E_i - V\left(x,t\right)\right), \tag{15}$$

where the index $j$ in (15) has been dropped out because of the single postsynaptic neuron. The membrane potential $V(x,t)$ contains the information about the analog signal $x$ in its spike sequence, which we shall denote by $\hat{s}_j(x,t)$. The spike sequence $\hat{s}_j(x,t)$ can once again be filtered to define a postsynaptic activity $\hat{a}_j(x,t)$. A specific choice of $\hat{a}_j(x,t)$ would dictate what the membrane potential $V^*(x,t)$ and the synaptic current $I^*(x,t)$ should be in the $j^{\text{th}}$ postsynaptic cell. Once we have chosen a $(I^*(x,t), V^*(x,t))$ pair (details of which to be shown later), the weights $w_i$ are found by minimizing the error function

$$E = \left\langle \int_0^T \int_{x_{\min}}^{x_{\max}} \left[I^*\left(x,t\right) - I^{\text{syn}}\left(x,t\right)\right]^2 dx\, dt \right\rangle_{\eta}, \tag{16}$$

where according to (15),

$$I^{\text{syn}}\left(x,t\right) = \sum_{i=1}^{N} w_i\left[h\left(t\right) * s_i\left(x,t\right) + \eta_i\right]\left(E_i - V^*\left(x,t\right)\right).$$

An important restriction in the choice of $w_i$ is that it must be nonnegative. We remark that $E_i$ is a binary variable and takes two fixed values depending on the character of the synaptic connection

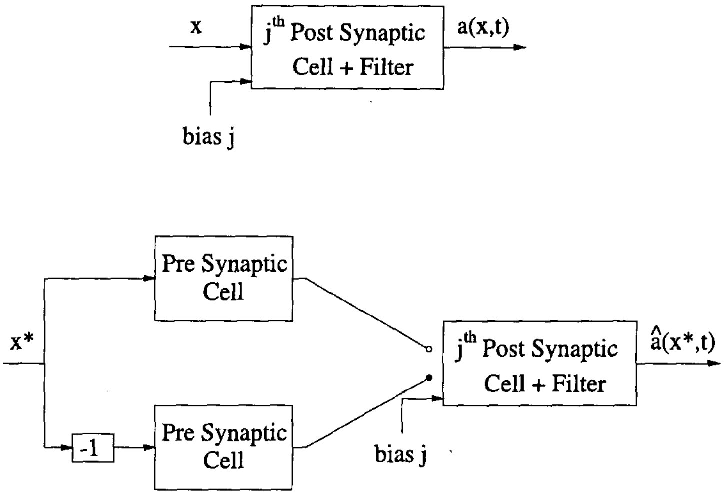

x
j^th Post Synaptic Cell + Filter
a(x,t)
bias j
x*
Pre Synaptic Cell
-1
Pre Synaptic Cell
j^th Post Synaptic Cell + Filter
â(x*,t)
bias j

Figure 8. Matching the activity functions of the $j^{\text{th}}$ postsynaptic neuron.

between the $i^{\text{th}}$ presynaptic cell and $j^{\text{th}}$ postsynaptic cell. Therefore, the choice of the synaptic connections has to be made prior to the minimization of (16). In the process of minimizing the error in (16), the pair $(I^*(x,t), V^*(x,t))$ has to be determined. Because of the nonlinear nature of the signal processing at the level of the postsynaptic neuron, these pairs have to be found experimentally. The corresponding scheme is shown in Figure 8. The $j^{\text{th}}$ postsynaptic neuron is stimulated by a series of constant inputs $x$ within a certain range $[x_{\max}, x_{\min}]$. These, in turn, produce a series of spike trains, whose activities are denoted by $a(x,t)$. The steady-state values of these signals correspond to the activity function $a(x)$ of the $j^{\text{th}}$ postsynaptic cell, and $x$ can be recovered from $a(x)$ as long as this function is invertible. The bottom plot shows the same cell, receiving only two synaptic inputs, with arbitrarily chosen synaptic weights. The purpose of these synapses is only to provide a meaningful synaptic input to the $j^{\text{th}}$ postsynaptic neuron. By running a set of constant inputs $x^*$, the corresponding postsynaptic activity is given by $\hat{a}(x^*,t)$ with the steady-state value $\hat{a}(x^*)$. Given any $x \in [x_{\max}, x_{\min}]$, we search for a particular $x^*$ such that

$$\hat{a}\,(x^*) = a\,(x)\,.$$

Such $x^*$ is found numerically, and for a continuum of values of $x$ it generates a function $x^* = \phi(x)$. By simulating the circuit from Figure 8 with the input $\phi(x)$ in GENESIS, we obtain a pair $(I^*(x,t), V^*(x,t))$ that will make the postsynaptic cell possess the activity $a(x)$, since $\hat{a}(\phi(x)) = a(x)$ by construction. Once these pairs are known for every $x \in [x_{\max}, x_{\min}]$ up to a discretization resolution, the minimization problem (16) can easily be solved, and the set of weights $w_1, w_2, \dots, w_N$ can be found. Having explained the basic procedure pertaining to Figure 7, we can now pose a more general question. Before we state the problem, let us observe that once the set of synaptic weights in (16) is found, the activity of the $j^{\text{th}}$ postsynaptic cell will preserve the information about $x$ for any particular constant input $x$, i.e., the frequency of spikes in the postsynaptic neuron will be close to the frequency of spikes in the same neuron stimulated directly by $x$. Therefore, the postsynaptic neuron will have a sustained activity corresponding to the value of $x$, as long as that input is present at the input channel of the BNN circuit. Assume now that instead of having a single postsynaptic neuron, we have an identical set of neurons in the presynaptic and postsynaptic layers. Let $N$ be the number of neurons in each layer. One such network where $N = 40$ is shown in Figure 9. Our goal is to find synaptic weights $w_{ji}$ between the $i^{\text{th}}$ presynaptic neuron and $j^{\text{th}}$ postsynaptic neuron, such that the activity $\hat{s}_j(x,t)$ matches the activity of the $j^{\text{th}}$ neuron with input $x$, for $j = 1, 2, \dots, N$. The neurons within the population differ by a bias. The choice of the bias is constrained by the physical limitations of

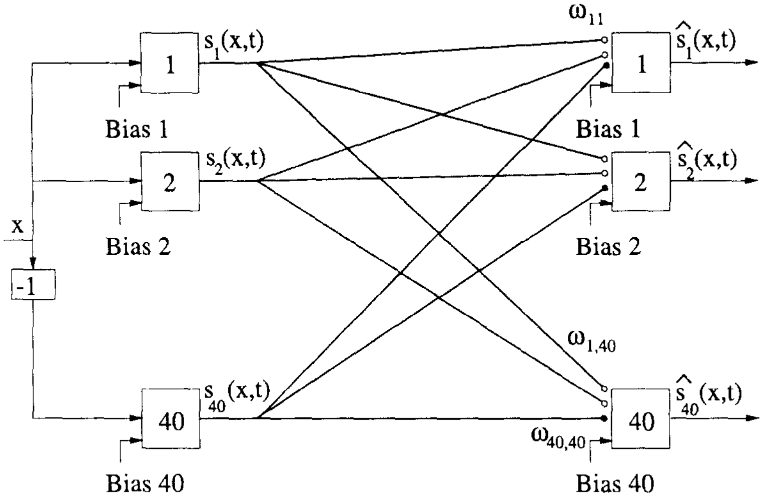


Figure 9. A BNN with 40 presynaptic and 40 postsynaptic neurons.

the cells. Adding a constant bias to a single spiking neuron changes its firing rate; a positive bias increases the firing rate, whereas a negative bias hyperpolarizes the neuron, rendering its membrane potential below the resting equilibrium [15]. Therefore, the choice of a bias is constrained by the maximum firing rate and the maximum level of hyperpolarization. In our consideration, we assume that the neurons cannot fire more than 160 spikes/sec. On the other hand, we set the maximum hyperpolarization level to be $-100\,\mathrm{mV}$. The first $N/2$ cells in each layer are of on type, and the second half is of *off* type. The synaptic connections from the presynaptic to the postsynaptic cells are given in Table 1.

Table 1.

| Presynaptic | Postsynaptic | Type of Synapse |
|---|---|---|
| on | on | excitatory |
| on | off | inhibitory |
| off | on | inhibitory |
| off | off | excitatory |

It was mentioned earlier that the character of the synaptic connections between different cells has to be known, otherwise, we are facing a combinatorial problem. The pairs $(I_j^*(x,t), V_j^*(x,t))$ are obtained experimentally for each of the $N$ postsynaptic neurons and the weights are found using the following constrained optimization procedure:

$$\min \mathrm{Error}_j \qquad \text{subject to } \omega_{ji} \geq 0,$$

where

$$\mathrm{Error}_j = \left\langle \int_0^T \int_{x_{\min}}^{x_{\max}} \left[I_j^*(x,t) - I_j^{\mathrm{syn}}(x,t)\right]^2 dx\,dt \right\rangle_\eta , \tag{17}$$

for every $j = 1, 2, \dots, N$, and where

$$I_j^{\mathrm{syn}}(x,t) = \sum_{i=1}^{N} w_{ji}\left[h(t) * s_i(x,t) + \eta_i\right]\left(E_i - V_j^*(x,t)\right).$$

This procedure yields $N$ sets of $N$ synaptic weights. An analog signal $x$ at the input channel of the network (see Figure 9), gets encoded into $N$ spike trains of the cells from the presynaptic layer. Because of the proper synaptic weight choice, each of the postsynaptic neurons will have the spike train that preserves the information about the analog variable $x$. The simulations have been performed using GENESIS. Using the linear decoding rule, the information about $x$ can eventually be recovered. A graph that shows the relationship between the original (input) signal and the decoded signal from the postsynaptic layer is given in Figure 10, for a collection of constant signals $x \in [-0.3, 0.3]$ and for $N = 40$. Consistent with the theory of memories and attractors, the points of intersection are so-called fixed points, some of which are stable, indicating that the network would stabilize at these points if the signals from the postsynaptic layer were fed back to the presynaptic layer. Since the error defined by (17) can never be zero, the number of fixed points is necessarily finite. Also note that the performance of the network is particularly good in the interval $[-0.15, 0.15]$. Selected membrane potentials corresponding to two points $x_1 = 0.15$ and $x_2 = -0.15$ are shown in Figure 11, where the traces are coming from the cells in the postsynaptic layer.

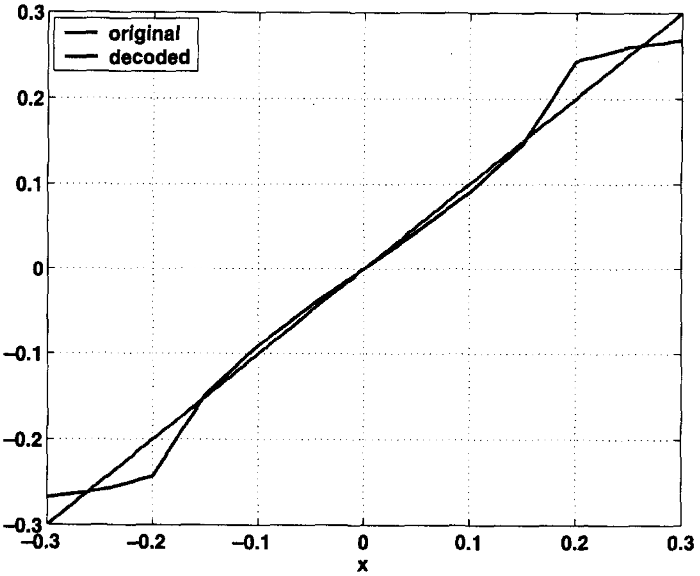
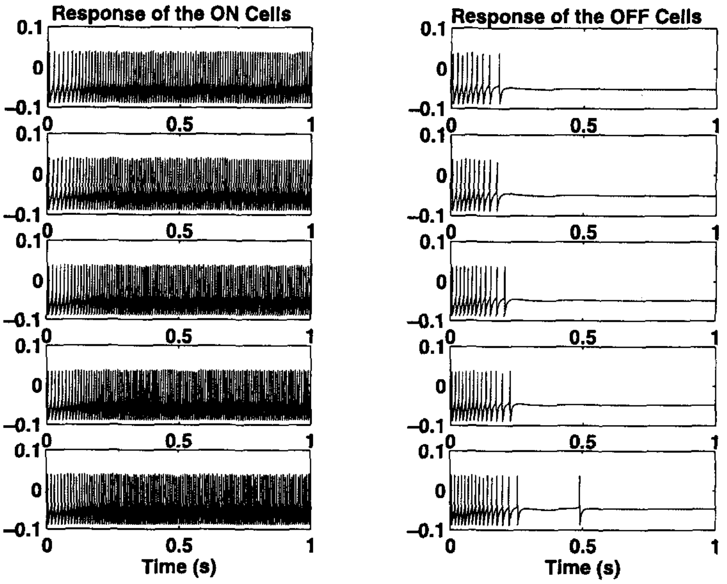
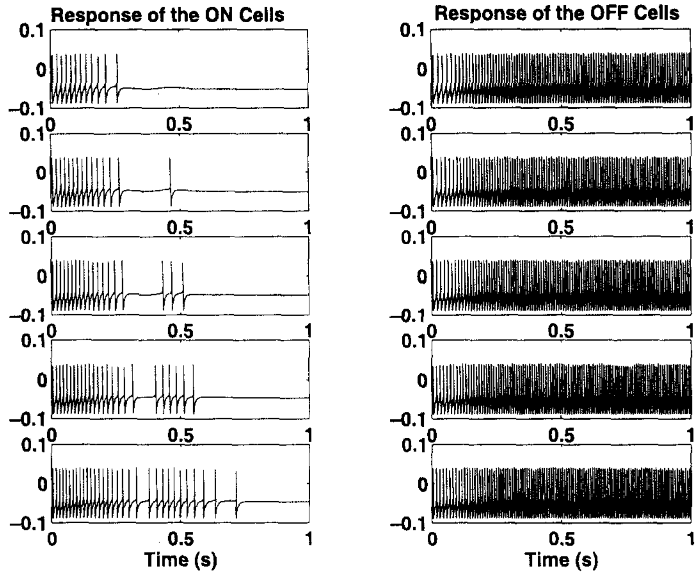


Figure 10. The input and steady-state value of the decoded signal are shown.



(a) Membrane potentials of selected postsynpatic neurons as a response to a constant input $x = 0.15$.

Figure 11.



(b) Membrane potentials of selected postsynaptic neurons as a response to a constant input $x = -0.15$.

Figure 11. (cont.)

# 5. A BNN SYNTHESIS OF A MEMORY

The synaptic weights from the presynaptic layer (Layer 1) to the postsynaptic layer (Layer 2) have been chosen such that the input to Layer 1 can be preserved and decoded at the level of Layer 2. Since the two layers are identical, feeding signal back from Layer 2 to Layer 1 with the same set of weights will form a network in which the value of an input signal is reccurently maintained. This idea is used to construct a memory which was implemented in GENESIS. The memory is initialized by a constant signal for 300 ms and the signal is switched off afterwards. The memory also has a 50 ms synaptic delay. Observing the spike trains of the individual cells merely suggests that their firing rates converge to a constant value. To verify this, we decode the signal from the postsynaptic layer using linear decoding scheme. The decoding algorithm is similar to that explained in the second section. The response of the memory network to different constant signals is observed and decoded. We have identified seven distinct fixed points, and some of the traces are shown in Figure 12. The fixed points are different from those predicted

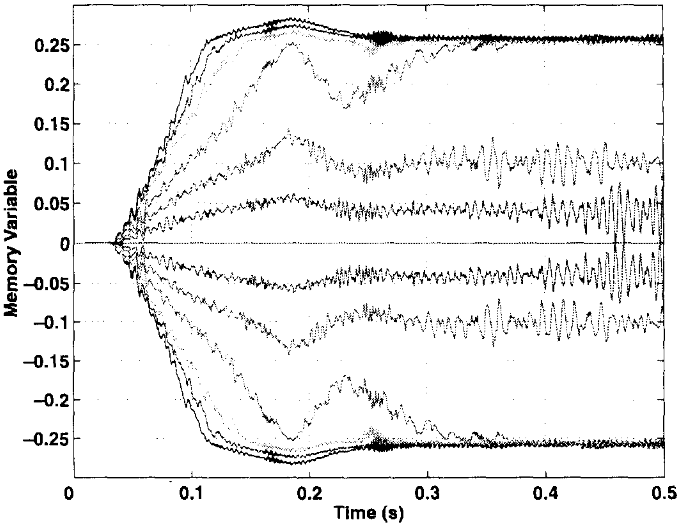


Figure 12. A memory circuit realized as BNN retaining several distinct values of the analog variable $x$.

by Figure 10. The reason for this lies in the fact that the fixed points in Figure 10 are found for the time $t$ sufficiently large (steady state), and for the feedforward circuit. Such analysis does not apply to a dynamic feedback system such as memory.

## 6. CONCLUSION

In this paper, we have described, in fairly complete details, how a population of neurons can encode analog signals. The basic idea of signal encoding is extended from a single neuron and *on/off* pair to a population of neurons. We demonstrate how such a population can be utilized for a specific computational task, *viz.*, a function representation and solving a differential equation. Despite the nonlinear encoding process, we introduce the decoding rule which is linear. The concept of linear decoding represents a basis for solving differential equations using neurons, a process of transferring the dynamics from the space of external variables to the space internal variables—the activities of neurons. In this process, a further generalization of the population concept is introduced, i.e., the discussion includes a population of spiking neurons and the use of a low-pass filter. Finally, we investigate how these ideas can be implemented in a framework of biologically realistic neural networks, where the role of filters has to be replaced by a synapse. We propose a new constrained optimization scheme that gives rise to a set of synaptic weights that will preserve the information about the input at the level of presynaptic layer in a two layered BNN. Such a network can be utilized in building a memory, and the simulations have successfully been implemented in GENESIS, indicating a certain number of stationary points or distince values that can be memorized.

## 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. (Lond.)* **117**, 500–544, (1952).
2. F. Rieke, D. Warland, R.R. Steveninck and W. Bialek, *Spikes: Exploring the Neural Code*, MIT Press, Cambridge, (1997).
3. L.F. Abbott and S. Song, Temporally asymmetric Hebbian learning, spike timing and neuronal response variability, In *Advances in Neural Information Processing Systems, Volume 11*, (Edited by M.S. Kearns, S.A. Solla and D.A. Cohn), pp. 69–75, MIT Press, Cambridge, (1999).
4. C. Koch, T. Poggio and V. Torre, Nonlinear interactions in a dendritic tree: Localization, timing, and role in information processing, *Proc. Natl. Acad. Sci.* **80**, 2799–2802, (1983).
5. E.D. Adrian, The impulses produced by sensory nerve endings: Part I, *J. Physiol. (Lond.)* **61**, 49–72, (1926).
6. P. Dayan and L.F. Abbott, *Theoretical Neuroscience*, MIT Press, Cambridge, (2001).
7. Z. Nenadic and B.K. Ghosh, Dynamic-control problems with on/off cells, In *Proc. of the 38th Conference on Decision & Control*, pp. 399–404, (1999).
8. Z. Nenadic and B.K. Ghosh, Signal processing and control problems in the brain: A neuroscientific approach, *IEEE Control Systems Magazine* **21** (4), 28–41, (August 2001).
9. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*, Springer-Verlag, New York, (1983).
10. C. Eliasmith and C.H. Anderson, Developing and applying a toolkit from a general neurocomputational framework, *Neurocomputing* **26/27**, 1013–1018, (1999).
11. Q. Huang, C.H. Anderson and J. Clark, Harmonic analysis of spiking neuronal ensembles, *Neurocomputing* **32/33**, 279–284, (2000).
12. A.D. Polpitiya, Z. Nenadic and B.K. Ghosh, Optimal filtering in biological neural networks, In *Proc. American Control Conference*, pp. 3539–3542, (2001).
13. Z. Nenadic and B.K. Ghosh, Computation with biological neurons, In *Proc. American Control Conference*, pp. 257–262, (2001).
14. M.A. Arbib, P. Erdi and J. Szentagothai, *Neural Organization: Structure, Function, and Dynamics*, MIT Press, Cambridge, (1998).
15. F. Delcomyn, *Foundations of Neurobiology*, W. H. Freeman & Co., New York, (1998).
16. H.R. Wilson, *Spikes, Decisions, and Actions: The Dynamical Foundation of Neuroscience*, Oxford University Press, New York, (1999).