

Demodulation Methods for an Adaptive Neural Encoder Model

A. M. Bruckstein¹, M. Morf¹, and Y. Y. Zeevi²

¹ Stanford University, Stanford, CA, USA

² Technion, Israel Institute of Technology, Haifa, Israel

Abstract. An adaptive version of the integrate and fire-at-threshold model for the neural coding process is presented. The encoder transforms stimulus intensity information into sequences of identical membrane depolarization spikes, their times of occurrence defining a modulated point process. Several theoretical decoding schemes are then introduced and their performance analyzed. These implement simple, recursive parameter estimation algorithms and their output reproduces reliably the encoded time-varying stimulus level.

1 Introduction

The nervous system employs for long distance transmission of information a pulse frequency modulation technique; membrane depolarization spikes, essentially identical in shape, propagate along neuronal axons carrying messages encoded in patterns of interspike intervals (Katz, 1966). Neural encoders are in fact complex pulse frequency modulators transforming analog signals, usually slowly varying in time, into sequences of “all-or-none” events. The coding process is far from being completely understood, in spite of many important discoveries such as the Hodgkin-Huxley (1962) membrane theory, and of steady improvements in experimental techniques (see Jack et al., 1975).

Since the intricate, distributed, spatial and temporal processing which takes place at spike initiation loci is, in most cases, affected by a large number of factors, a practical and meaningful way to analyze the experimentally recorded neural activity is to provide some statistical description for it (Gestri, 1971; Knight, 1972). Adopting this approach, the sequences of spikes propagation along the axonal channels are regarded as realizations of stochastic point processes, the suc-

cessive interoccurrence intervals being series of positive random variables. It is the task of experimental research in neurophysiology to provide ensemble statistics for the neural activity in response to specific stimuli and environmental conditions. Once such probabilistic descriptions become available, one can define and analyze mathematical models for the coding process. From an engineering point of view, these models are stochastic pulse frequency modulators that summarize experimental results and predict, with varying degrees of success, the neural response under yet untested conditions. It is clear that such models need not have structural relations to the actual physiological processes that lead to depolarization spikes in response to stimuli. Most models under study do, however, incorporate some rough and abstract mathematical descriptions of the basic phenomena physiologists believe to constitute the coding process. These are the integration of ionic currents at the membrane level resulting in changes of polarization and the initiation of the neuron firing at a certain threshold level of membrane depolarization (Gestri, 1971; Knight, 1972; Jack et al., 1975). If some structural assumptions lead to improved behavior of the encoder model in terms of matching and predicting experimental results, one may gain valuable insights into the actual coding process and also provide guidelines for further physiological research.

This paper briefly presents an adaptive “integrate and fire at threshold” neural encoder model the output of which is a selfexciting point process (Bruckstein, 1980; Bruckstein and Zeevi, 1983). The slowly varying stimulus together with two feedback effects, selfinhibition and refractoriness, determine the interspike interval distributions. Subsequently, some simple decoding techniques are introduced and analyzed. These demodulation schemes are in fact recursive parameter estimation procedures; the algorithms derived are based on the assumption that the decoder has infor-

mation concerning the nature of the encoding process, the level of the stimulus being the parameter to be recovered. Computer simulations of the encoding process as well as of the demodulation algorithms complement the performance analysis of the neural communication model under study.

2 The Integrate and Fire at Threshold Model

Let the neural encoder input, representing the intensity of a “generator current” which depends on primary stimuli, be a slowly varying time function $\lambda(t)$. We will assume that $\lambda(t)$, representing stimulus intensity, is always positive. The encoder output is the realization of a stochastic point process $\{t_n\}_{n \in \mathbb{N}}$ where the n -th interoccurrence interval is given by:

$$i_n = t_{n+1} - t_n. \quad (1)$$

Clearly i_n is the realization of a random variable and the output point process can be characterized by the statistics of the sequence $\{I_n\}_{n \in \mathbb{N}}$ of successive random interspike intervals.

Consider now the following encoder model: starting at an occurrence say t_n an “effective input” $\lambda^*(t) > 0$ derived from the input $\lambda(t)$ is integrated until the nondecreasing function $u(t) = \int_{t_n}^t \lambda^*(\xi) d\xi$ first upcrosses a threshold level a_n (Fig. 1).

The equation:

$$\int_{t_n}^{t_{n+1}} \lambda^*(\xi) d\xi = a_n \quad (2)$$

determines the next neural event's occurrence time t_{n+1} . At time t_{n+1} the process restarts – thus given $\lambda^*(t)$ and the sequence $\{a_n\}$ the output is “recursively” generated. The a_n 's are considered realizations of random variables A_n (the threshold process) and we assume that their distribution depends on the past history of the output process up to $t = t_n$

$$p_{A_n|\Sigma_n}(a|\sigma_n) = \kappa(a, m_{\sigma_n}). \quad (3)$$

Here Σ_n represents the past of the output process up to t_n and σ_n is the sigma field associated with the event $\{t_0, t_1, \dots, t_n\}$. The function m_{σ_n} is, as indicated, measurable on σ_n and determines the memory depth (in fact the past dependence) of the threshold process.

Under the assumption that $\lambda^*(t)$ changes slowly (being almost constant over interspike intervals) we can write from (1) and (2) that:

$$\lambda^*(t_k) i_k \approx a_k. \quad (4)$$

Therefore i_k and a_k are related through a $\lambda^*(t_k)$ -scaling and one can immediately write the following ex-

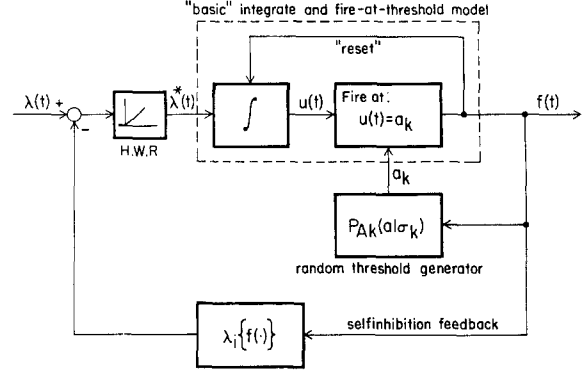


Fig. 1. The adaptive “integrate and fire-at-threshold” neural encoder model

pression for the interspike interval distribution:

$$p_{I_n|\Sigma_n}(i|\sigma_n) = \lambda^*(t_n) \kappa(i\lambda^*(t_n), m_{\sigma_n}). \quad (5)$$

Let us now turn to the issue of the dependence between the effective input and the primary stimulus intensity. Some models simply assume that $\lambda^*(t) = \lambda(t)$. A better, more general, assumption would be that the effective input depends not only on the primary stimulus strength but also on the past output of the encoder. We can formally write that

$$\lambda^*(t_n) = H(\lambda(t_n), s_{\sigma_n}),$$

where H is some function of the input and of the σ_n -measurable quantity s_{σ_n} . This assumption turns out to be well motivated physiologically.

A specific form for the kernels $\kappa(\cdot, \cdot)$, that allows for considerable flexibility in describing experimental data is the following:

$$\kappa(a, m_{\sigma_n}) = \frac{(m_{\sigma_n})^\Omega a^{\Omega-1} \exp[-a(m_{\sigma_n})]}{\Gamma(\Omega)}. \quad (6)$$

This is a Gamma distribution of order Ω with parameter m_{σ_n} which provides immediately for the moments of the random variable A_n (Karlin and Taylor, 1975):

$$E[A_n|\Sigma_n] = \frac{\Omega}{m_{\sigma_n}}, \quad (7)$$

$$\text{var}[A_n|\Sigma_n] = \frac{\Omega}{(m_{\sigma_n})^2}. \quad (8)$$

Now using (5) one obtains for the interval transition kernels:

$$p_{I_n|\Sigma_n}(i|\sigma_n) = \frac{[\lambda^*(t_n)(m_{\sigma_n})]^\Omega i^{\Omega-1} \exp[-i\lambda^*(t_n)(m_{\sigma_n})]}{\Gamma(\Omega)}. \quad (9)$$

It is clear from the above that the $\{I_k\}_{k \in \mathbb{N}}$ process has statistics governed by Gamma type transition kernels having order Ω and parameters:

$$\Phi_n = (m_{\sigma_n}) \lambda^*(t_n). \quad (10)$$

From (9) it readily follows that:

$$E[I_n|\Sigma_n] = \frac{\Omega}{\Phi_n}, \quad (11)$$

$$\text{var}[I_n|\Sigma_n] = \frac{\Omega}{\Phi_n^2}. \quad (12)$$

The above results clearly display the fact that the interval process has realization-dependent evolution, thus the output point process is selfexciting (Snyder, 1975).

The above described model is yet quite general, more specific and commonly used models being obtained through further assumptions on the threshold behavior and on the dependence of the effective input on past firing activity.

Considering a wide range of experimental observations on the overall behavior of neural coding units as well as physiological evidence on ionic flows through the neuronal membrane during the firing activity (Katz, 1960; Jack et al., 1975; Sokolove, 1972; Bromm and Tagmat, 1977; Gestri et al., 1980; Walloe, 1969; Sanderson et al., 1973) we decided on a set of assumptions on the past-dependence of the encoding process. These are:

1) The effective input has the following form:

$$\lambda^*(t_n) = (\lambda(t_n) - s_{\sigma_n})^+ = \begin{cases} \lambda(t_n) - s_{\sigma_n} & \text{for } \lambda(t_n) - s_{\sigma_n} > \lambda_{\min} \\ \lambda_{\min} & \text{otherwise.} \end{cases} \quad (13)$$

Here s_{σ_n} is a selfinhibitory current – measured at t_n – resulting from the activity of an active electrogenic “ion pump” that evacuates the surplus of ions which diffuse across the membrane and cumulate within the nerve cell due to the repetitive firing. The selfinhibitory current is considered to be described by first order dynamics (Sokolove, 1972; Bruckstein, 1980)

$$s_{\sigma_n} = S \sum_{i \leq n} \exp\left[-\frac{(t_n - t_i)}{\tau_i}\right], \quad (14)$$

where S represents the quantile charge diffusing into the cell during the time course of a single action potential and τ_i is the time-constant of the selfinhibition process.

2) The threshold control function m_{σ_n} has to account for the trend observed when correlating successive intervals, therefore we choose

$$m_{\sigma_n} = M(1 - \beta) \sum_{x=0}^{\infty} \beta^x i_{n-x-1} \quad \text{with } \beta < 1. \quad (15)$$

Here M is a gain parameter and β determines the memory depth of the threshold control function. This moving average relationship for m_{σ_n} clearly implies

that most recent intervals determine its value and since $E[I_n|\Sigma_n] = \Omega/m_{\sigma_n}$, one obtains that short intervals increase the expected length of the following ones, as desired (Walloe, 1969).

Under the above assumptions one gets the following description for the encoder output: $\{t_k\}$ is the sample path of a selfexciting point process, the evolution of which in terms of the interval process is given by the following transition kernels:

$$p_{I_n|\Sigma_n}(i|\sigma_n) = \frac{\Phi_n^\Omega i^{\Omega-1} \exp[-\Phi_n i]}{\Gamma(\Omega)}. \quad (16)$$

Here $\Phi_n = m_{\sigma_n}(\lambda(t_n) - s_{\sigma_n})^+$ and s_{σ_n} and m_{σ_n} are given by (14) and (15) respectively.

The encoder output can easily be simulated by generating realizations of the sequence $\{I_k\}$ according to (16), however an analytical approach to the study of the ensemble behavior of the output process is practically impossible. For selfexciting processes of this type only some local, sample-path characterizations can be obtained (Snyder, 1975). In biological investigations we are mainly interested in the ensemble behavior, this being the description obtained through experiments. In this case, computer simulations can provide a good idea of the typical encoder response. An alternative approach leading directly to an approximate description of the ensemble behavior (through a nonlinear dynamic system, or ordinary differential equations) also proved to be feasible (Bruckstein and Zeevi, 1983; Zeevi and Bruckstein, 1981).

The above introduced encoder model accounts for the experimental findings in terms of general trends in the behavior of neural coding units. In order to apply the model to a particular neurosensory system and thereby explain its responses, one must introduce some specific assumptions. The model is however readily seen to reproduce the characteristics of adaptive behavior. If increased stimulus leads to a decrease in interoccurrence intervals (higher frequency) for a certain period of time the resulting probabilistic increase in threshold will tend to increase the following interoccurrence intervals in an “automatic-gain-control”-like mechanism. Increased neural activity over a longer period will lead to a decrease in the effective generator current. This is an output feedback mechanism that also reduces the firing frequency through the action of the selfinhibition process. If the threshold control or refractoriness mechanism is fast compared to the selfinhibition process one obtains two time constant transients of adaptation when step-like changes in the stimulus occur. The integrate to threshold scheme has as a built in characteristic the scaling property concerning the output variability. Transient changes in the stochastic structure of the response can

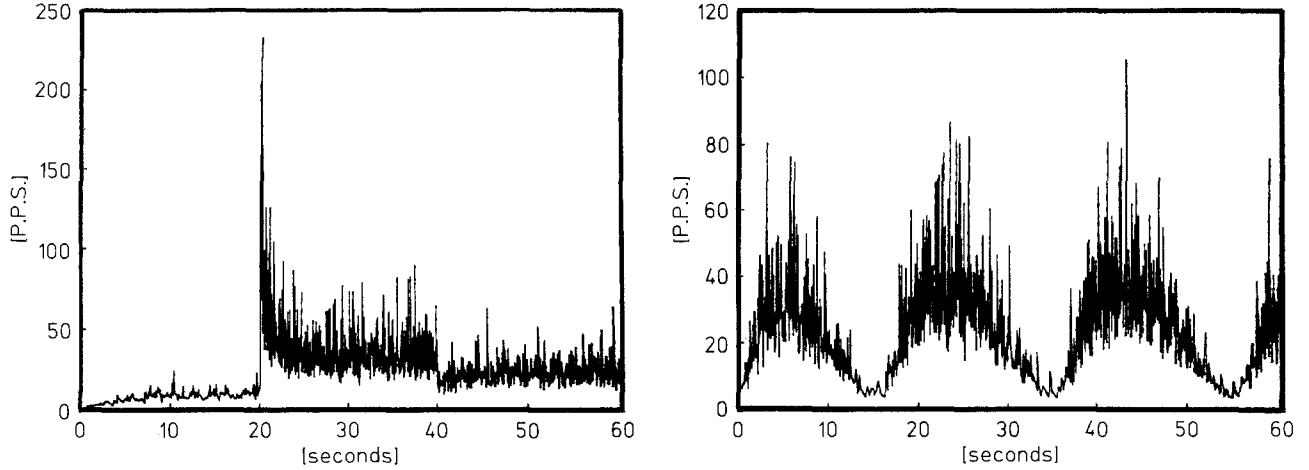


Fig. 2a and b. Examples of encoder output: instantaneous frequency *vs* time for the following inputs: **a** double step of amplitude 1-10-5 (encoder test sequence) and **b** sinewave of span 1-10

be accounted for only by assuming changes in the order of the transition kernel, induced by high output activity (Bruckstein and Zeevi, 1981).

The above encoder model was simulated using the following values for the parameters: $\Omega = 10.0$, $\beta = 0.99$, $M = 0.1$, $\tau_s = 5.0$, $S = 0.05$. The inputs tested were double steps and sinewaves. It is to be noted that, with the chosen parameters, the selfinhibition was only rarely high enough to “cut off” the effective input to λ_{\min} and this is indeed the case in physiology. The cutoff becomes obvious only when a sudden significant drop in the stimulus level occurs in which case a silencing of the neural activity is observed. Typical simulation results are presented in Fig. 2 indicating that the physiological characteristics of encoder behavior are well reproduced. Further discussion of the model and its responses is beyond the scope of this study and can be found in the references (Bruckstein, 1980; Bruckstein and Zeevi, 1983).

3 The Decoding Algorithms

Suppose that at the “receiving end” sequences of spikes are observed and it is required to recover the primary input to the encoder $\lambda(t)$. In the nervous system there may, of course, be no need to demodulate the sequences of neural events in order to continue their processing. The question is, therefore, mainly of theoretical importance: it is to evaluate the amount of information on the stimulus time course $\lambda(t)$ that is retained in the random patterns of spikes that form the encoder response.

In order to recover $\lambda(t)$, the demodulator observes the sequence of occurrence times and it is assumed to have knowledge of the encoding mechanism, i.e. the

σ_n -measurable functions m_{σ_n} and s_{σ_n} . This is not a restrictive assumption since even in the biological setting at the decoding sites one has local encoders that could simulate the behavior of a far-end coding unit, providing a model-reference for the decoding mechanism (Bruckstein and Zeevi, 1983).

There are several possible ways to approach the decoding problem, either through the use of classical estimation techniques or by model reference and stochastic approximation methods. The algorithms that will subsequently be presented were tested in simulation as follows: a stimulus function (double step) was applied to the encoding model and the encoder output was used as a test input to the decoding schemes. The demodulation method performance could then be judged by the quality of stimulus signal recovery (in terms of acquisition times and tracking capability).

3.1 Maximum Likelihood and “Bayesian” Estimation

At time $t = t_n$ the information available to the decoder is σ_n and the σ_n -measurable functions s_{σ_n} and m_{σ_n} can be computed since, by assumption, the decoder has knowledge of the encoder behavior and has clearly seen the past encoded sequence. Thus at $t = t_n$ the receiving end knows that the next interval to be observed, i_n , is the realization of a random variable I_n with distribution given by (16). If no prior information on $\lambda(t_n)$ is assumed to be available at time t_n , the natural choice would be to estimate it by the value that maximizes the likelihood of the observed interval i_n . This Maximum Likelihood Estimator (MLE) for $\lambda_n = \lambda(t_n)$ is readily obtained as:

$$\hat{\lambda}(t_n) = s_{\sigma_n} + \frac{\Omega}{i_n m_{\sigma_n}} \quad \text{if } \hat{\lambda}(t_n) - s_{\sigma_n} > \lambda_{\min}. \quad (17)$$

In case of large i_n , that yields $\hat{\lambda} - s_{\sigma_n} < \lambda_{\min}$ we shall set the λ -estimate to $s_{\sigma_n} + \lambda_{\min}$. This estimator is extremely noisy for the obvious reason that it does not take into account prior knowledge on either the behavior of the input (for example that it changes very slowly most of the time) or on its level as gained through previous observations. The prior information can be assumed to be summarized in the current estimate $\hat{\lambda}(t_{n-1})$. Indeed if we take into account that $\lambda(t)$ has a smooth evolution

reflect our “degree of confidence” in the previous estimate. After having assigned the prior, one is in a position to find the “optimal Bayes estimate” of $\lambda(t)$ as the mean of the λ -distribution conditioned on the observed interval. From Bayes’ rule we have:

$$p_{A|I_n}(\lambda|i_n) = \frac{p_{I_n|A}(i_n|\lambda)p_A(\lambda)}{p_{I_n}(i)} \quad (22)$$

providing after some algebraic manipulation:

$$p_{A|I_n}(\lambda|i) = \frac{(m_{\sigma_n}i_n + \Psi)^{\Theta + \Omega} (\lambda - s_{\sigma_n})^{\Theta + \Omega - 1} \exp[-(m_{\sigma_n}i_n + \Psi)(\lambda - s_{\sigma_n})]}{\Gamma(\Theta + \Omega)} \quad (23)$$

most of the time (although jumps can occur), we may consider a technique that does not discard the information contained in previous estimates of the input level up to time t_n . Therefore we shall concentrate on the following demodulation method: the current estimate $\hat{\lambda}_{n-1}$ determines a prior distribution for the next one. Choosing this prior involves a tradeoff between contradicting requirements; the prior should constrain the next estimate to the neighborhood of $\hat{\lambda}_{n-1}$ but also allow it to track possible jumps in the input. For simplicity we consider the following distribution type for the prior:

$$p_A(\lambda) = \frac{\Psi^{\Theta} (\lambda - s_{\sigma_n})^{\Theta - 1} \exp[-\Psi(\lambda - s_{\sigma_n})]}{\Gamma(\Theta)} \quad (18)$$

In case the previous estimate is lower than $s_{\sigma_n} + \lambda_{\min}$ (which rarely occurs) we shall set it equal to this value. Now one has to specify the parameters Θ and Ψ to set the expected value of the prior to $\hat{\lambda}_{n-1}$. The additional constraint will be the assignment of the prior variance. Since (18) is a s_{σ_n} -displaced Gamma distribution we have:

$$E[\lambda] = s_{\sigma_n} + \frac{\Theta}{\Psi} \quad (19)$$

and therefore one readily gets

$$\frac{\Theta}{\Psi} = \hat{\lambda}_{n-1} - s_{\sigma_n} \quad (20)$$

The variance of the prior distribution is given by:

$$\text{var}[\lambda] = \frac{\Theta}{\Psi^2} = \frac{(\hat{\lambda}_{n-1} - s_{\sigma_n})}{\Psi} \quad (21)$$

and one must choose a procedure to assign this parameter in order to determine the values of Θ and Ψ . The variance of the prior actually reflects the relative weight assigned to the next observation in the process of updating the estimate of the input. Two methods to choose the prior variance can be thought of: the variance can be set to a fixed value or can be chosen to

This result shows a nice reproducing property of Gamma type distributions. Since (23) is a s_{σ_n} -displaced Gamma density, one obtains:

$$\hat{\lambda}_n = E[\lambda|i_n] = s_{\sigma_n} + \frac{\Omega + \Theta}{m_{\sigma_n}i_n + \Psi} \quad (24)$$

Suppose at time t_n we have chosen the prior parameters Θ_n and Ψ_n so that we have

$$\frac{\Theta_n}{\Psi_n} = \hat{\lambda}_{n-1} - s_{\sigma_n} \quad \text{and} \quad \frac{\Theta_n}{\Psi_n^2} = V_n \quad (25)$$

yielding

$$\Psi_n = \frac{\hat{\lambda}_{n-1} - s_{\sigma_n}}{V_n} \quad \text{and} \quad \Theta_n = \frac{(\hat{\lambda}_{n-1} - s_{\sigma_n})^2}{V_n} \quad (26)$$

Now the estimation algorithm becomes:

$$\hat{\lambda}_n = s_{\sigma_n} + \frac{\Omega + \Theta_n}{m_{\sigma_n}i_n + \Psi_n} \quad (27)$$

In order to complete the recursion we need to specify V_n . If fixed prior variance is assigned then $V_n = V_0$, however, if we wish to adjust V_n according to a measure of confidence in previous estimates, we may choose V_n to be a cumulative function of the errors in predicting the interspike intervals based on previous estimates. This method will assign a prior variance that measures the likelihood of previous estimates. The rationale: if the evolution of $\hat{\lambda}$ did not closely follow that of the true input the prediction errors will cumulate and thus assign a higher prior variance resulting in the assignment of more weight to the new information. In the algorithm that was implemented the prior variance was

$$V_n = \frac{|\text{a sliding mean of prediction errors}|}{\text{a sliding mean of intervals}} V_0 \quad (28)$$

providing a relative measure of confidence in $\hat{\lambda}$. In the above-discussed algorithms the prior variance acts as an “inertia” parameter since, for example, a small V_0 will confine the next decoder output to a value close to

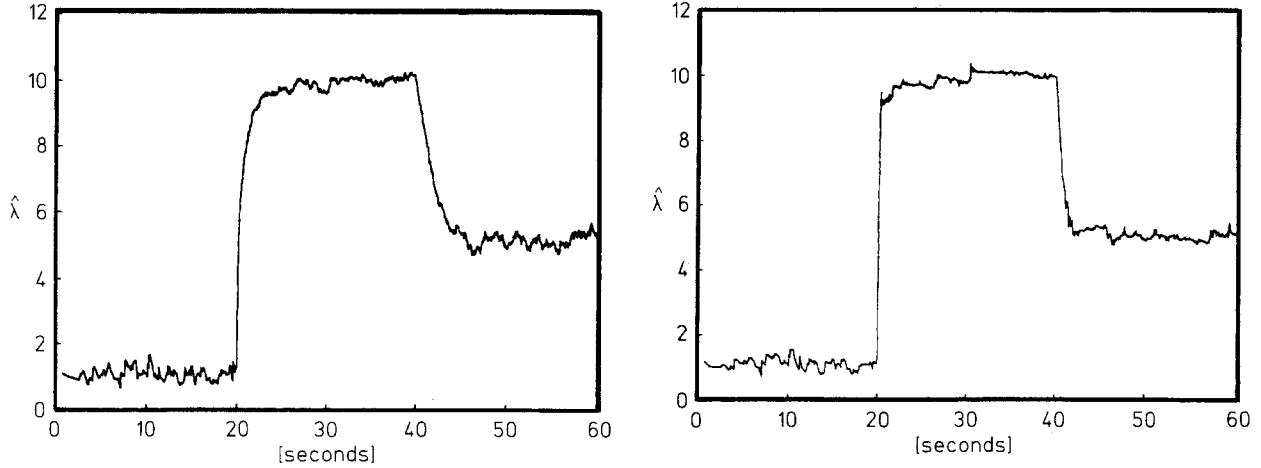


Fig. 3a and b. Quasi-Bayes decoder performance a fixed prior variance: 0.1 and b adaptive prior variance

the previous one. This improves steady-state “identification” accuracy at the expense of reducing convergence rate and thus tracking capability. Simulation results with these “quasi-Bayes” demodulation techniques are presented in Fig. 3.

3.2 Prediction Error Demodulation Techniques

The general philosophy of prediction error decoding is the following: using the current, local estimate of $\lambda(t)$, and taking advantage of some knowledge on the encoder structure, a prediction of the next interval is generated. This can be done for example by re-encoding the local estimate through a locally available encoder. Theoretically this prediction may be the expected value for i , given the local input estimate, or alternatively the value that has maximum occurrence likelihood. The prediction is then compared to the actually observed interval and the estimate of the input is corrected (updated) by an amount depending on the resulting prediction error (Ljung, 1981). Since at $t=t_n$ we know s_{σ_n} and m_{σ_n} we can predict the next interval conditioned on $\hat{\lambda}_{n-1}$ as:

$$\hat{i}_n = E[i|\sigma_n, \hat{\lambda}_{n-1}] = \frac{\Omega}{m_{\sigma_n}(\hat{\lambda}_{n-1} - s_{\sigma_n})^+}. \quad (29)$$

The prediction error demodulator will, therefore, implement variations of the following general procedure:

- 1) predict next interval: $\hat{i}_n = \frac{\Omega}{m_{\sigma_n}(\hat{\lambda}_{n-1} - s_{\sigma_n})^+}$,
- 2) compute prediction error: $e_n = i_n - \hat{i}_n$,
- 3) update local estimate: $\hat{\lambda}_n = \hat{\lambda}_{n-1} + \gamma_n f(e_n)$, where γ_n is a gain sequence and $f(e_n)$ is some function of the prediction error.

The gain sequence in the above procedure may depend on the history of the observations and on the

past evolution of the prediction error. In the following section we define and analyze some specific decoding algorithms which are particular cases of the above general procedure.

3.2.1 Prediction Error Random Walk Decoder

A very simple prediction error decoder can be obtained from the general procedure by assigning a constant gain sequence, $\gamma_n = \gamma_0$, and defining $f(e_n) = -\text{sign}(e_n)$. Now the algorithm becomes:

$$\hat{\lambda}_n = \hat{\lambda}_{n-1} - \gamma_0 \text{sign}(e_n). \quad (30)$$

A γ_0 -adjustment is thus made in the local estimate according to the sign of the prediction error. Assuming, for the moment, that $\lambda(t) = \lambda_0$, i.e. constant stimulus level applied to the input, we can readily compute the probability of “up” or “down” adjustments in the estimate of λ as a function of its current value. We have

$$Pr\{i_n > \hat{i}_n | \lambda(t) = \lambda_0, \hat{\lambda}_{n-1}, \Sigma_n\} = \int_{\hat{i}_n}^{\infty} p_{I_n | \Sigma_n, \lambda_0}(i | \sigma_n) di. \quad (31)$$

Using the Gamma distribution function defined as:

$$F_I(x) = \frac{1}{\Gamma(\Omega)} \int_{-\infty}^x \xi^{\Omega-1} \exp(-\xi) d\xi$$

one gets, after some straightforward manipulations:

$$Pr\{i_n > \hat{i}_n | \lambda_0, \hat{\lambda}_{n-1}, \Sigma_n\} = 1 - F_I \left[\frac{\Omega(\lambda_0 - s_{\sigma_n})^+}{(\hat{\lambda}_{n-1} - s_{\sigma_n})^+} \right]. \quad (32)$$

Now assuming s_{σ_n} very small compared to both input level and its local estimate (which is to be expected at higher input levels) we can make the approximation

$$\frac{(\lambda_0 - s_{\sigma_n})^+}{\hat{\lambda}_{n-1} - s_{\sigma_n}} = \frac{\lambda_0}{\hat{\lambda}_{n-1}}$$

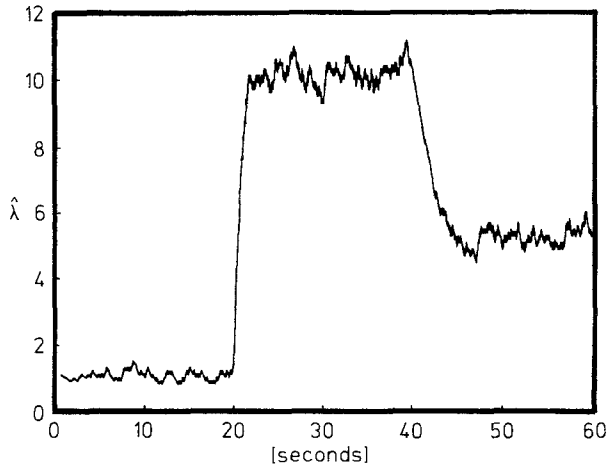


Fig. 4. Prediction error random walk decoder performance

yielding the following result:

$$Pr\{\hat{\lambda}_n = \hat{\lambda}_{n-1} + \gamma_0\} = Pr\{i_n < \hat{i}_n | \dots\} = F_F\left(\frac{\Omega\lambda_0}{\hat{\lambda}_{n-1}}\right), \quad (33)$$

$$Pr\{\hat{\lambda}_n = \hat{\lambda}_{n-1} - \gamma_0\} = Pr\{i_n > \hat{i}_n | \dots\} = 1 - F_F\left(\frac{\Omega\lambda_0}{\hat{\lambda}_{n-1}}\right). \quad (34)$$

In fact the decoder output takes values only on the levels $\{k\gamma_0\}$ and it performs a random walk with transition probabilities given by (33) and (34). A plot of the “up” and “down” adjustment probabilities as a function of the state, k , reveals that the transition matrix of the random walk drives the encoder output towards a value close to λ_0 a “crossover” region of the curves defined by (33) and (34). In order to roughly estimate the variance of the decoder output when the markov chain is already in stationary regime, we can approximate the crossover interval by considering up and down adjustment probabilities that vary linearly with k over this region, say $(k_L, k_L + R)$. The stationary probability distribution for the resulting markov chain is:

$$Pr\{\text{state} = k_L + X\} = 2^{-R} \binom{R}{X} \quad (35)$$

with corresponding expected value and variance of

$$E[X] = \frac{R}{2} \quad \text{var}[X] = \frac{R(R-1)}{4}. \quad (36)$$

This result, together with the fact that the crossover is near the value of k that yields $k\gamma_0 = \lambda_0$ and its span is proportional to λ_0 show that the expected decoder output is approximately the true stimulus intensity whereas the variance about this value is proportional to λ_0 , but independent of γ_0 . Small values of γ_0 will result in slow tracking dynamics; the value of this

parameter should, thus, be chosen to allow for good tracking capability. Simulations with this demodulation algorithm clearly exhibit the above discussed characteristics (Fig. 4).

3.2.2 The Stochastic Approximation Approach

Originally introduced by Robbins and Monro, the stochastic approximation (R–M) technique is a particularly simple but powerful and important method of parameter identification (Ljung, 1981). A stochastic approximation decoder can be obtained in the prediction error method framework by setting $f(e_n) = -e_n$. At $t = t_n$ the error is given by

$$e_n = i_n - \frac{\Omega}{m_{\sigma_n}(\hat{\lambda}_{n-1} - s_{\sigma_n})^+}$$

and it follows that

$$E[e_n | \Sigma_n, \hat{\lambda}_{n-1}] = \frac{\Omega}{m_{\sigma_n}} \left[\frac{1}{(\lambda(t_n) - s_{\sigma_n})^+} - \frac{1}{(\hat{\lambda}_{n-1} - s_{\sigma_n})^+} \right]. \quad (37)$$

Now, clearly, the expected value of the error is zero if and only if the input estimate coincides with the true input level, provided they are both above the selfinhibitory current. In this case the error expected value is positive for $\lambda(t_n) < \hat{\lambda}_{n-1}$ and negative otherwise, thus the procedure of updating the local estimate:

$$\hat{\lambda}_n = \hat{\lambda}_{n-1} - \gamma_n e_n \quad (38)$$

will indeed produce a correction which, on the average, is in the desired direction (it is assumed here that the input does not vary considerably over successive intervals). Note that here the application of the stochastic approximation procedure differs slightly from the usual case since the error process is causally realization-dependent. In order to have good tracking capabilities in an algorithm like (38) but also good accuracy in identifying a constant input level, if the case arises, the gain sequence must be chosen as follows:

1) When predictions are good, on the average, γ_n must approach 0 as $1/n$ (the estimate is, in this case, close to the true value and this method improves accuracy as in the usual R–M procedure).

2) When a cumulative process defined on the prediction error evolution (a forgetting-factor weighted average, for example) reaches a certain threshold, indicating that the procedure consistently either under or overestimates the input level, the gain sequence must be restarted at some reasonably high value in order to give more weight to the new information.

In the algorithms implemented a somewhat more sophisticated gain restart scheme, based on the above principle, was used (multiple thresholds and several gain restart points were chosen). Following Nevelson

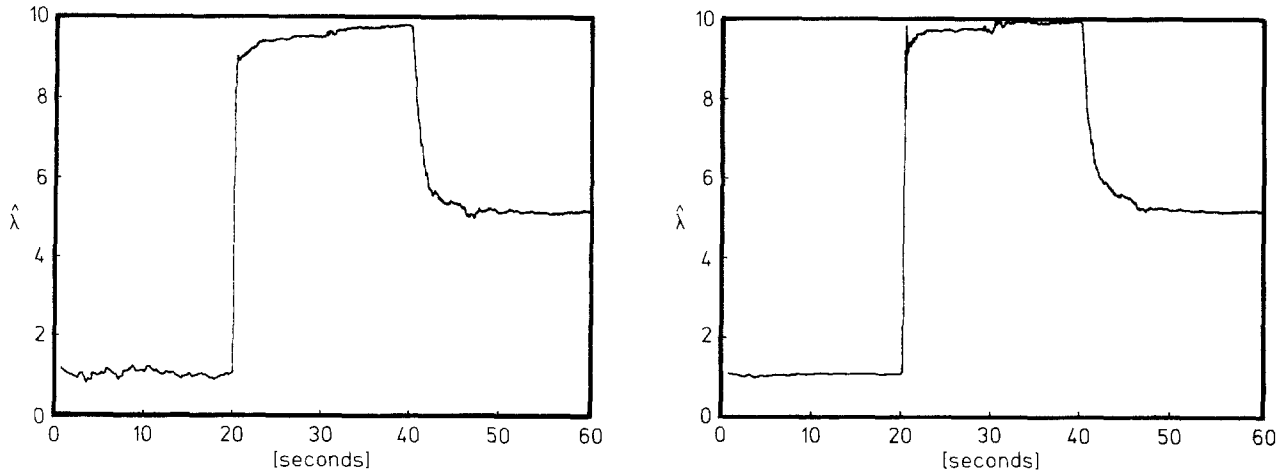


Fig. 5a and b. Stochastic approximation decoder performance a and “adaptive” stochastic approximation decoder performance b

and Hasminski’ (1973), an “adaptive” stochastic approximation method was also implemented and proved to be the best of all decoding algorithms that were tested in this study. The modified estimate update formula is:

$$\hat{\lambda}_n = \hat{\lambda}_{n-1} + \left[\frac{d}{d\lambda} m_{\sigma_n}(\lambda - s_{\sigma_n}) \Big|_{\lambda = \hat{\lambda}_{n-1}} \right]^{-1} \gamma_n e_n. \quad (39)$$

The better performance of this procedure can be explained by the fact that, while the usual R–M method is a “gradient”-type algorithm, the adaptive scheme is close to the “stochastic Newton” recursive identification method. We note, however that the true stochastic gradient and Newton algorithms would be obtained by defining the above demodulation problem as a process of minimizing the expected mean square prediction error. This latter approach leads to slightly different decoding algorithms that are, however, close to the ones described. The performance of the above algorithms is illustrated in the simulation results presented in Fig. 5. Comparing the decoder performance to the simulations using other approaches it is clear that the adaptive method has the fastest level acquisition time and accuracy and also better tracking capability.

4 Concluding Remarks

A model for the neural encoder was presented and briefly analyzed. Several recursive demodulation methods were then introduced to test possible ways to recover stimulus strength information from the encoder output, a selfexciting point process. Such idealized decoding schemes enable us to evaluate the amount of information on the stimuli carried by the rate-modulated random sequences of spikes.

Among the different decoding schemes, best performance was obtained with the adaptive stochastic approximation algorithm. However, all proposed decoding techniques performed well on the tests that required the demodulation of spike trains resulting from the encoding of step and sinewave functions. This indicates that, from a communication-theoretic point of view the encoded sequences are able to reliably convey the relevant information, in spite of the noisy threshold behavior. (Note that the threshold noise can be regarded as summarizing all possible sources of randomness in the coding process.) In this context, it is also important to notice that the adaptive characteristics of the encoder model significantly contribute to a better decoding performance since sharp changes in stimulus level are enhanced and this leads to better tracking of sudden changes in input.

Acknowledgement. The work of the first author was supported in part by a Rothschild Fellowship for 1980/1981.

References

- Bromm, B., Tagmat, A.T.: Regression analysis of non-stationary discharges in a neuron; adaptation in the electrosensory afferent of dogfish. *Biol. Cybern.* **28**, 41 (1977)
- Bruckstein, A.M., Zeevi, Y.Y.: Analysis of integrate to threshold neural coding schemes. *Biol. Cybern.* **34**, 63 (1979)
- Bruckstein, A.M.: Models of coding in the nervous system. M. Sc. Thesis, Dept. of E.E., Technion, I.I.T., Haifa 1980
- Bruckstein, A.M., Zeevi, Y.Y.: An adaptive stochastic model for the neural coding process (submitted, 1983)
- Gestri, G.: Pulse frequency modulation in neural systems. *Biophys. J.* **11**, 98 (1971)
- Gestri, G., Mastebroek, H.A.K., Zaagman, W.H.: Stochastic constancy, variability and adaptation of spike generation: performance of a giant neuron in the visual system of the fly. *Biol. Cybern.* **38**, 31 (1980)

- Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **117**, 500 (1952)
- Jack, J.J.B., Noble, D., Tsien, R.W.: Electric current flow in excitable cells. Oxford: Clarendon Press 1975
- Katz, B.: Nerve, muscle, synapse. New York: McGraw-Hill 1966
- Karlin, S., Taylor, H.M.: A first course in stochastic processes. New York: Academic Press 1975
- Knight, B.W.: Dynamics of encoding in a population of neurons. *J. Gen. Physiol.* **59**, 734 (1972)
- Ljung, L.: Lecture notes on system identification. Stanford University 1981
- Nevelson, M.B., Hasminskii, R.Z.: Stochastic approximation and recursive estimation. Math. Monographs, 47, Providence, Rh. Isl., 1973
- Sanderson, A.C., Kozak, W.M., Calvert, T.W.: Distribution coding in the visual pathway. *Biophys. J.* **13**, 218 (1973)
- Snyder, D.L.: Random point processes. New York: Wiley 1975
- Sokolove, P.G.: Computer simulation of after inhibition in crayfish slowly adapting stretch receptor. *Biophys. J.* **12**, 1249 (1972)
- Walloe, L.: Transfer of signals through a second order neuron. Oslo University Thesis 1968
- Zeevi, Y.Y., Bruckstein, A.M.: Adaptive neural encoder model with selfinhibition and threshold control. *Biol. Cybern.* **40**, 79 (1981)

Received: May 3, 1983

Dr. A. Bruckstein
Stanford University
Department EE/ISL
Stanford, CA 94305
USA