A Totally Automated System for the Detection and Classification of Neural Spikes

XIAOWEI YANG, STUDENT MEMBER, IEEE, AND SHIHAB A. SHAMMA, MEMBER, IEEE

Abstract—**A system for neural spike detection and classification is presented which does not require *a priori* assumptions about spike shape or timing. The system is divided into two parts: a learning subsystem and a real-time detection and classification subsystem. The learning subsystem, comprised of a feature learning phase and a template learning phase, extracts templates for each separable spike class. The real-time detection and classification subsystem identifies spikes in the noisy neural trace and sorts them into classes, based on the templates and the statistics of the background noise. Comparisons are made among three different schemes for the real-time detection and classification subsystem. Performance of the system is illustrated by using it to classify spikes in segments of neural activity recorded from monkey motor cortex and from guinea pig and ferret auditory cortexes. The system is implemented without human supervision and therefore is suitable for real-time multichannel recording.**

I. Introduction

TO STUDY the functional connectivity of neural systems, reliable observation of the simultaneous activity of a group of neurons is essential [14], [15]. An extracellular electrode often records such electrical activity from several adjacent neurons. To analyze the contribution of each individual unit, one needs to distinguish the signals of each unit from the rest. In principle, signals from different neurons can be classified by their characteristic spike shapes. However, these shapes are often unpredictable functions of neuron type, electrode construction and placement, and the electrical characteristics of the intervening tissue [6]. In addition, multiunit recordings are always contaminated by noise, which comes both from external sources and from the weaker neural signals of more distant units.

There are several techniques available for the classification of multiunit neural signals [1], [3]–[7], [10]–[13], [16], [17], [19]–[21]. Some techniques require prior information about spike shapes and epochs [3]–[7], which is often not available; other methods employ time consuming computations [1], [5], [7], [10], [12], [13], [16], [17], [19]–[21], making real-time implementation impossible without specialized hardware; and others involve human supervision [3], [4], [11]. With the increasing availability of multichannel extracellular microelectrode arrays [9] and the potential large number of simultaneous recordings, data processing capability and capacity will become very important, and on-line neural spike-separating techniques must strike a balance between performance and speed.

The fundamental motivation behind this work is the desire to overcome some of the limitations presented by the newly developed extracellular microelectrode arrays. In most such electrodes, recording sites are fabricated on the same substrate. Consequently, the quality of the recordings obtained from a given channel cannot be improved by a simple adjustment of the electrode position (as is the case with single electrodes) since this will necessarily affect the position (hence, the recording quality) on all other channels. Therefore, the best strategy is to utilize these recordings as best as possible instead of constantly attempting to adjust them. Another serious problem concerns the large number of recording sites and the presence of multiple spike shapes in every channel. Using traditional manual spike detection methods such as threshold and window discriminators to determine which channels carry useful information and to separate out the different spikes on each channel is tedious, extremely time consuming, and unlikely to succeed if adjustments have to be made during a recording session as is often the case.

To reiterate it, the overriding goal of the spike detection algorithm to be used with multielectrode arrays is *not* so much to detect the smallest spikes in the midst of noisy traces, but rather to isolate the most reliable spikes with no or mimimal human intervention. This paper proposes an efficient and convenient on-line multispike separation system which is totally automated from detection to classification, without presupposing any knowledge of spike shapes or of interspike intervals. The system consists of two parts: a learning module, which extracts templates for each class of spikes in the neural trace, and a real-time classification module, which detects the spikes and sorts them into classes. In the learning module, a Haar transformation is first performed to locate the occurrences of spikes. A two-dimensional histogram is then constructed to determine how many classes of spikes the neural signal contains based on two particular features of spikes: peak-to-peak amplitude and peak-to-peak time interval. Classifying each spike according to these features, templates

Manuscript received August 31, 1987; revised May 16, 1988. This work was supported in part by grants from the Whitaker Foundation and the Naval Research Laboratory.

X. Yang is with the Department of Electrical Engineering, University of Maryland, College Park, MD 20742.

S. A. Shamma is with the Department of Electrical Engineering Systems Research Center, and the University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742.

IEEE Log Number 8822614.

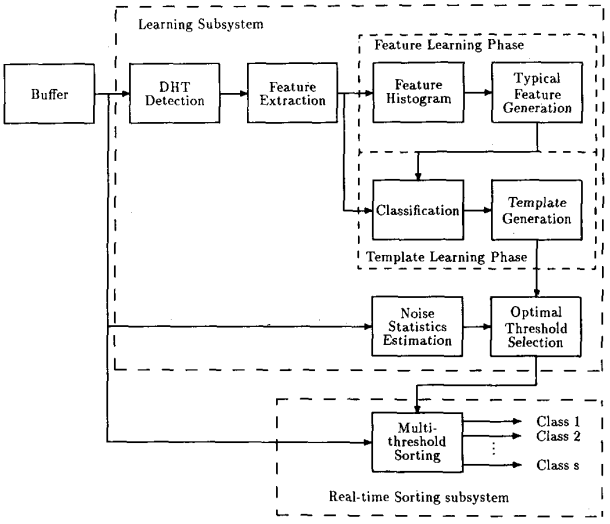


Fig. 1. Schematic diagram of the system.

of each class are finally generated by means of averaging. Once templates are ready, a spike-sorting algorithm with the determined optimal multithresholds for each template performs real-time detection and classification. The optimal thresholds are generated by minimizing the total error probability of detection and classification. Examples of multiunit extracellular recording from auditory cortex and primate motor cortex are used to test the system.

The following section describes the complete system. The principles underlying the use of the Haar transformation to detect spikes in noise are given in Section III. Section IV presents the feature selection and classification criteria. The real-time sorting subsystem is detailed in Section V, and several testing examples are given in Section VI.

II. System Description

The system is divided into two parts as indicated in the block diagram of Fig. 1. The first part is a learning subsystem which extracts templates of spikes for every class. This subsystem includes a feature learning phase and a template learning phase. The second part is a real-time detection and classification subsystem which detects spikes in the noisy trace and sorts them out into classes based on the templates that the learning subsystem provides and the statistics of the background noise.

Assume that a segment of sampled data containing spikes from several neurons is stored in a memory buffer. The learning subsystem begins with the detection of spikes in the noise, using the discrete Haar transformation (DHT). In the feature learning phase, specific features, the peak-to-peak amplitude and the peak-to-peak time interval, are measured for each spike detected and are used to construct the feature histogram. The histogram serves to determine the number of separable unit classes and typical feature values of each class. In the template learning phase, the same DHT detection scheme is applied to detect spikes which are sorted into classes by comparing their features with the typical features. As a result, typical templates for each class are formed by averaging all the classified spikes.

The real-time detection and classification subsystem is supported by a multithreshold sorting scheme, a multiwindow sorting scheme, or a soft-decision sorting scheme. Once the optimal threshold parameters for all classes are derived, based on the templates and the background noise, the real-time on-line processing begins.

III. Detection by Haar Transformation

The discrete Haar transform detection scheme plays a key role in the learning subsystem. Because of the spike-like characteristics of the transform bases, the Haar transform is a powerful tool for detecting neural spikes. In transforming the original neural trace, a large component appears in the Haar-transformed domain if the basis for that component is similar in width and phase to the spike in the time trace. By detecting these outstanding components, spikes are detected in the corresponding time trace.

To elaborate, let N be a positive integer that is a power of two. The Haar function [8] is defined in $[0, 1)$ as

$$\text{har}(0, 0, t) = 1, \quad t \in [0, 1)$$

$$\text{har}(r, m, t) = \begin{cases} 2^{r/2}, & (m - 1)/2^r \le t < (m - 1/2)/2^r \\ -2^{r/2}, & (m - 1/2)/2^r \le t < m/2^r \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $r = 0, 1, \cdots, \log_2 N$, and $m = 1, 2, \cdots, 2^r$. It is shown that the Haar function $\text{har}(n, m, t)$ is orthogonal and complete for any N which is a power of two. If $\text{har}(n, m, t)$ is sampled at rate N, than an $N \times N$ orthogonal matrix is obtained [2]

$$\boldsymbol{W} = \begin{bmatrix} w_0(0) & w_0(1) & \cdots & w_0(N-1) \\ w_1(0) & w_1(1) & \cdots & w_1(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-1}(0) & w_{N-1}(1) & \cdots & w_{N-1}(N-1) \end{bmatrix} \tag{2}$$

where each element is a sample of the Haar function

$$w_0(i) = h(0, 0, i) = 1, \quad i = 0, 1, \cdots, N - 1$$

$$w_k(i) = w_{2^m+n-1}(i) =$$

$$h(m, n, i) = \begin{cases} 0, & i = 0, \cdots, \\ & N(n-1)/2^m - 1 \\ 2^{m/2}, & i = N(n-1)/2^m, \cdots, \\ & N(n-1/2)/2^m - 1 \\ -2^{m/2}, & i = N(n-1/2)/2^m, \cdots, \\ & Nn/2^m - 1 \\ 0 & i = Nn/2^m, \cdots, N-1 \end{cases} \tag{3}$$

for $k = 1, 2, \cdots, N-1$ where the ranges of m and n are $0 \le m \le \log_2 N - 1$ and $1 \le n \le 2^m$, respectively. Every row of the transform matrix is a basis of the Haar transform.

Denote the sampled time trace as $\boldsymbol{x} = [x_0, x_1, \cdots, x_{N-1}]$ and the discrete Haar transformed sequence as $\boldsymbol{y} = [y_0, y_1, \cdots, y_{N-1}]$; they are related by $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. The following simple cases shows how the Haar transform locates a spike.

Fig. 2 illustrates the following example: suppose that an artificial spike has the shape

$$x_i = \begin{cases} a \sin [2\pi(i-3)/5], & i = 4, 5, 6, 7 \\ 0, & \text{otherwise.} \end{cases}$$

The resulting transformed sequence is

$$y_k = \begin{cases} 1.54a_N, & k = N/4 + 1 \\ 0.36a_N, & k = N/2 + 2 \\ -0.36a_N, & k = N/2 + 3 \\ 0 & \text{otherwise} \end{cases}$$

where $a_N = a\sqrt{N}$. This example corresponds to the case where no noise is present. It is seen that large components occur in the transform domain when there is a spike in the time domain. Experimentally, there is always an additive noise process in the background of neural recordings. The noise may be removed by a "threshold filtering" method, using a properly selected threshold in the transform domain. To perform the threshold filtering, set every component below the threshold equal to zero and leave components above the threshold unchanged. Then the threshold filtered sequence is transformed back to the original time domain, resulting in noise-free, step-like spikes at their original positions in the trace. Time of occurrence is defined by the zero-crossing of the step-like spikes. We continue to use this example to illustrate how the process works. The threshold t in the transform domain is set to be $|t| = 0.5a_N$, so the filtered transformed sequence $\hat{\boldsymbol{y}} = [\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_{N-1}]$ is expressed as

$$\hat{y}_k = \begin{cases} 1.54a_N, & k = N/4 + 1 \\ 0, & \text{otherwise} \end{cases}$$

which is transformed back to the time domain by $\hat{\boldsymbol{x}} = \boldsymbol{W}^{-1}\hat{\boldsymbol{y}}$, yielding the reconstructed spike $\hat{\boldsymbol{x}} = [\hat{x}_0, \hat{x}_1, \cdots, \hat{x}_{N-1}]$ as

$$\hat{x}_i = \begin{cases} 0.77a, & i = 4, 5 \\ -0.77a, & i = 6, 7 \\ 0, & \text{otherwise.} \end{cases}$$

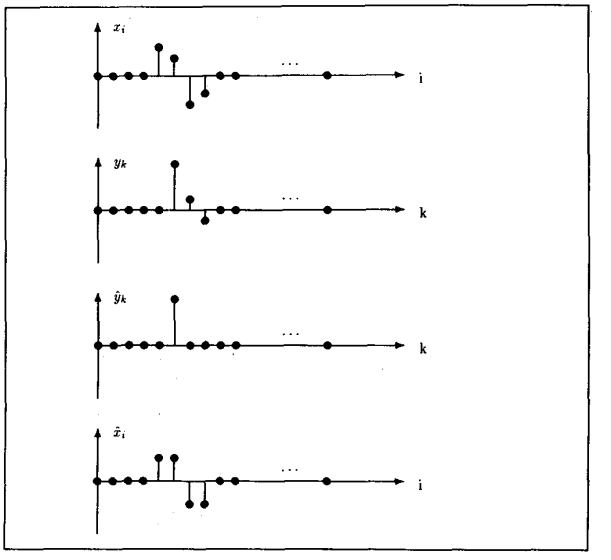


Fig. 2. Illustration of Haar transform detection by using an artificial spike.

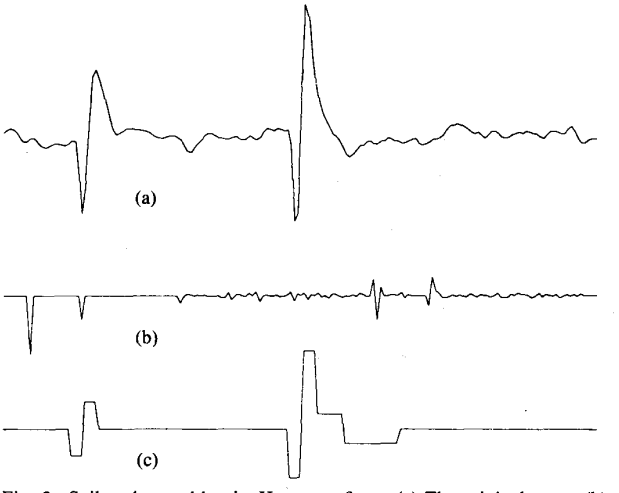


Fig. 3. Spikes detected by the Haar transform. (a) The original trace. (b) The transform domain equivalence; large components indicate presence of spikes. (c) Reconstructed spikes in the time domain with noise filtered out. Spike times are determined by zero-crossing detection.

Threshold for the filter is selected by variance estimation of the noise. Assume that the noise sequence $\{n_i : i = 0, 1, \cdots, N-1\}$ is iid with common Gaussian distribution $\mathfrak{N}(0, \sigma^2)$. It can easily be shown that the components of the transformed sequence $\{y_k : k = 0, 1, \cdots, N-1\}$ are identically distributed with common Gaussian distribution $\mathfrak{N}(0, N\sigma^2)$. Since $P_{\text{rob}}[|y_k| \ge 3.0\sigma\sqrt{N}] < 0.0028$, and $P_{\text{rob}}[|y_k| \ge 4.1\sigma\sqrt{N}] = 0.000042$, a nonlinear transform domain threshold is set

to be $t = C(\sigma)\,\sigma\sqrt{N}$ where $C(\sigma)$ is a nondecreasing function with range [3.0, 4.1]. If $\{y_k\}$ is the transformation of the noise only, then the probability that $|y_k|$ exceeds the threshold t is very small. The noise will, therefore, be filtered out by setting every component below the threshold to be zero.

The DHT at this stage serves to detect spikes with the goal of accumulating enough of them to construct the histogram and templates. Fig. 3 demonstrates the DHT detection. A segment of data from a neural recording in Fig. 3(a) was transformed by the DHT. In the transform domain [Fig. 3(b)] there are two large components in the low "frequency" portion and two in the high "frequency" portion, indicating two spikes in the original trace. After filtering our noise in the transform domain, the spikes were reconstructed without the high-frequency components and noise, as shown in Fig. 3(c).

IV. Feature Extraction and Classification

In order to classify the spikes, one can either use complete templates, or selected unique features of these templates. In principle, spikes belonging to the same class possess relatively similar features. To simplify the algorithm, two features are considered for the spikes here. One is the peak-to-peak amplitude, the other is the peak-to-peak time interval.

A two-dimensional feature histogram $h(i, j)$ is generated with index i as the ordered bin number for one feature and with index j for the other. If the resolution of the bins is high enough, the histogram will approximately reflect the joint probability density function $f(\xi_1, \xi_2)$ of the two features ξ_1 and ξ_2. By the motivation of the maximum likelihood estimation, we pick the particular values of ξ_1 and ξ_2 at which $f(\xi_1, \xi_2)$ has a peak as the typical feature for a class. We consider only those peaks that are separated from a smaller peak by a minimum distance ϵ. For example, suppose $f(\xi)$ has peaks at ξ' and ξ''. If

$$\frac{\|\xi' - \xi''\|}{\min\left(\|\xi'\|, \|\xi''\|\right)} > \epsilon \tag{4}$$

where $\epsilon > 0$ is a specific value, then we say that peaks at ξ' and at ξ'' represent different units. This idea can be generalized to multidimensional feature space.

An important question which arises here is how accurate are the typical feature values determined in this way, assuming there are no classification errors? Consider the following example where for simplicity only one feature ξ is chosen. Suppose that there are S classes with *a priori* probabilities P_i, $i = 1, 2, \cdots, S$, and that the histogram is made from n observations of feature ξ of which approximately $n_i = nP_i$ observations belong to class i. The feature has a pdf

$$f(\xi) = \sum_{i=1}^{S} P_i f_i(\xi) \tag{5}$$

where $f_i(\xi)$, is the pdf of ξ under class i. In the Gaussian case, let ξ have a Gaussian distribution with unknown mean μ_i and variance σ_i^2 under class i. Thus $f_i(\xi)$ has its peak at $\xi = \mu_i$. Therefore, according to the feature selection criterion, μ_i is chosen as the typical feature value for class i. The maximum likelihood estimator of μ_i is the sample mean of class i, and it can be written as

$$\hat{\mu}_i = \frac{1}{n} \sum_{\xi_k \in C_i} \xi_k. \tag{6}$$

If one uses the confidence interval estimation argument, it is easy to see that

$$P_{\text{rob}}\left[\hat{\mu}_i - \frac{\sigma}{\sqrt{n_i}} t_\alpha \le \mu_i \le \hat{\mu}_i + \frac{\sigma}{\sqrt{n_i}} t_\alpha\right] = 1 - \alpha. \tag{7}$$

In words, one has $(1 - \alpha)$ 100 percent confidence that the true value μ_i is in the interval $[\hat{\mu}_i - (\sigma/\sqrt{n_i})\, t_\alpha, \hat{\mu}_i + (\sigma/\sqrt{n_i})\, t_\alpha]$. The length of the interval is $2\sigma t_\alpha/\sqrt{n_i}$ where t_α is chosen so that $P_{\text{rob}}[|N(0, 1)| \le t_\alpha] = 1 - \alpha$ where $N(0, 1)$ is a normalized Gaussian random variable. Clearly, $\hat{\mu}_i$ is a consistent estimator of μ_i. The final step is to show that $\hat{\mu}_i$ is the value at which $f_i(\xi)$ has a peak with probability 1. This is true because $E(\hat{\mu}_i) = \mu_i$ and var $(\hat{\mu}_i) = \sigma_i^2/n_i$. The strong law of large numbers says that $\hat{\mu}_i$ converges to μ_i almost surely, and it is apparent that $P_{\text{rob}}[\lim_{n_i \to \infty} \text{var}\,(\hat{\mu}_i) = 0] = 1$; i.e., $\hat{\mu}_i$ will approach μ_i with no variation so that $f_i(\hat{\mu}_i)$ is definitely the peak as n_i goes to infinity. Now, it is in confirmation of Gaussian cases that one can make the typical feature values as accurate as one wants providing that the sample size n is sufficiently large.

In the case of insufficient sample size, local averaging is used to smooth the histogram. The resulting histogram is revised by

$$\bar{h}(i, j) = \frac{1}{W} \sum_{(k,l) \in N_b[(i,j), r]} w_{i-kj-l} h(k, l) \tag{8}$$

where

$$W = \sum_{(k,l) \in N_b[(0,0), r]} w_{kl}$$

and $N_b[(i, j), r]$ is defined as the r neighborhood of (i, j).

Consider now the classification problem. S classes are assumed to have *a priori* probabilities P_i, $i = 1, 2, \cdots, S$, and the noise class has *a priori* probability P_0. Using the risk function concept [18], we associate a risk r_{ij} with choosing class i when the correct classification is class j. The average risk is given by

$$R = \sum_{i=0}^{S} \sum_{j=0}^{S} r_{ij} P_j P(D_i \,|\, H_j) \tag{9}$$

where $P(D_i \,|\, H_j)$ is the probability of choosing class i when the true hypothesis is H_j. The feature space A is partitioned into exhaustive and mutually exclusive subsets A_i, $i = 0, 1, \cdots, S$, so that spikes with features $\boldsymbol{x}$ are classified to be in class i if $\boldsymbol{x} \in A_i$. Noting that $A_i = A - \cup_{j \neq i} A_j$ and $\int_A f_j(\boldsymbol{x})\, d\boldsymbol{x} = 1$, $j = 0, 1, \cdots, S$, the

average risk can be expressed as

$$R = \sum_{i=0}^{S}\sum_{j=0}^{S} r_{ij}P_j \int_{A_i} f_j(\boldsymbol{x})\, d\boldsymbol{x}$$
$$= \sum_{i=0}^{S} r_{ii}P_i + \sum_{i=0}^{S}\int_{A_i}\sum_{j\neq i}^{S}(r_{ij} - r_{jj})P_j f_j(\boldsymbol{x})\, d\boldsymbol{x}$$
$$= \sum_{i=0}^{S} r_{ii}P_i + \sum_{i=0}^{S}\int_{A_i}\lambda_i(\boldsymbol{x})\, d\boldsymbol{x} \tag{10}$$

where $\lambda_i(\boldsymbol{x}) = \Sigma_{j\neq i}^{S}(r_{ij} - r_{jj})P_j f_j(\boldsymbol{x})$ $(i = 0, 1, \cdots, S)$ are the decision variables. In order to minimize the risk R, we minimize the integrals for each class. The classification strategy is simple. The observed value $\boldsymbol{x}$ is claimed to be in class i if $i = \arg\min_j \{\lambda_j(\boldsymbol{x})\}$. Therefore, the optimal partition of A is

$$A_i^* = \{\boldsymbol{x}:\lambda_i(\boldsymbol{x}) \leq \lambda_j(\boldsymbol{x}), \text{ for all } j\},$$
$$i = 0, 1, \cdots, S. \tag{11}$$

The risk R is equal to the probability of error P_e if the risk r_{ij} is set to be

$$r_{ij} = \begin{cases} 1, & i \neq j \\ 0, & i = j. \end{cases} \tag{12}$$

In this case we have

$$P_e = R = \sum_{i=0}^{S}\int_{A_i}\lambda_i(\boldsymbol{x})\, d\boldsymbol{x}. \tag{13}$$

It is well known that under minimum error probability criterion, a matched filter performs the optimal classification if the noise is white Gaussian. Except with specialized hardware, implementing a matched filter is costly because of its computational complexity (multiplication and addition operations are proportional to template points). Faster implementation speeds are possible, however, if performance is comprised by using the following real-time sorting schemes.

V. Real-Time Sorting

Three different schemes are proposed for real-time sorting. The theoretical comparisons will be made at the end of the section.

A. Multithreshold Scheme

The multithreshold method is a fast real-time, on-line spike detection scheme. Unlike the single-threshold detection that is commonly used, the multithreshold technique provides more reliable detection, especially when the signal-to-noise ratio is low. To see the effectiveness of the scheme, a comparison is made with the single-threshold scheme by an example later in this section.

The method can be described as a hypothesis test problem. The underlying assumption is that the observed data trace is an additive combination of spikes and noise. Denote the observation, the spike and the noise as $x(t)$, $s(t)$, and $n(t)$, respectively. The hypotheses are stated as follows:

$$H_0: x(t) = n(t), \qquad t = 1, 2, \cdots, M$$
$$H_1: x(t) = s(t) + n(t), \qquad t = 1, 2, \cdots, M.$$

The null hypothesis H_0 is that there is no spike in the observation and the alternative hypothesis H_1 means that there is a spike in the observation. The decision as to whether there is a spike is made by M comparisons with the following rule. If $x(t) > \eta_t$, for all $t = 1, 2, \cdots, M$, then we assume that H_1 is true, otherwise H_0 is true, where η_t's are M independent thresholds. Our aim is to optimize the thresholds according to some criterion. There are two types of error in a statistical hypothesis test. The false alarm is the first type of incorrect decision, rejecting the hypothesis H_0 when that hypothesis is true. The second type of error, accepting H_0 when H_0 is false, is called the missing detection. For convenience, it is assumed that the noise is white Gaussian distributed with zero-mean and variance σ^2. This assumption does not lose generality because 1) the noise consists of many weak signals of distant units so that it is Gaussian distributed by the central limit theorem, and 2) the colored noise can be transformed into white by passing it through a whitening filter. In practice, therefore, we can pass $x(t)$ through a whitening filter of which the output $\tilde{x}(t)$ is composed by $\tilde{n}(t)$, the white version of the noise, and $\tilde{s}(t)$, a transformed version of the spike. Under this assumption, the probability of the false alarm P_F can be expressed as

$$P_F = P_{\text{rob}}(D_1|H_0) = P_{\text{rob}}[x(1) > \eta_1, x(2) > \eta_2, \cdots, x(M) > \eta_M|H_0]. \tag{14}$$

Because $n(t)$ is white, $x(t)$'s are mutually independent and with identical Gaussian distribution $\mathfrak{N}(0, \sigma^2)$; therefore, P_F can be further expressed in terms of the error function $\Phi(y) = \int_{-\infty}^{y} 1/\sqrt{2\pi}\, e^{-x^2/2}\, dx$ as

$$P_F = \prod_{t=1}^{M} P_{\text{rob}}[x(t) > \eta_t|H_0] = \prod_{t=1}^{M}\Phi\left(-\frac{\eta_t}{\sigma}\right). \tag{15}$$

Similarly, the probability of the missing P_M can be written as

$$P_M = P_{\text{rob}}(D_0|H_1)$$
$$= P_{\text{rob}}[x(t) < \eta_t, \text{ for some } t|H_1] \tag{16}$$

For the mathematical manipulation, we write it in terms of the probability of the complement of the missing

$$P_M = 1 - P_{\text{rob}}(D_1|H_1) = 1 - P_{\text{rob}}[x(1) > \eta_1, x(2) > \eta_2, \cdots, x(M) > \eta_M|H_1]. \tag{17}$$

Therefore, in terms of the error function, P_M may be written as

$$P_M = 1 - \prod_{t=1}^{M}\Phi\left(\frac{s(t) - \eta_t}{\sigma}\right). \tag{18}$$

It is not difficult to see that if one wishes P_F to be small, then one chooses thresholds η_t to be large, thus increasing P_M. Conversely, by choosing η_t to be small, P_M decreases while P_F increases. This implies that minimizing both P_F and P_M is a conflicting objective. One may reach a compromise by constructing an objective function J,

$$J = \theta P_F + P_M, \qquad \text{for some } \theta > 0. \tag{19}$$

The goal is now to minimize J.

The necessary condition for achieving a minimum of J is to set

$$\frac{\partial J}{\partial \eta_k} = 0, \qquad k = 1, 2, \cdots, M. \tag{20}$$

This results in M simultaneous equations

$$\theta \prod_{l \neq k}^{M} \Phi\left(-\frac{\eta}{\sigma_l}\right) = e^{(2\eta_k - s(k))s(k)/2\sigma^2} \prod_{l \neq k}^{M} \Phi\left(\frac{s(l) - \eta_l}{\sigma}\right), \quad k = 1, 2, \cdots, M. \tag{21}$$

The possible optimal set of thresholds η_k's is obtained by solving those equations.

Proposition: There exists a unique minimum of the objective function J if a spike is detected, and if a false alarm is more costly than a missing detection.

A proof of the proposition is given in the Appendix. The existence and uniqueness of the minimum of J guarantees the risk-free solution of the M simultaneous equations.

The significance of θ may be explained as follows. If the false alarm is as costly as the missing detection, one sets $\theta = 1$; $\theta > 1$ means that the false alarm is more costly than the missing detection, and $\theta < 1$ otherwise. If the *a priori* probability $P(H_0)$ is known, then the total error probability is $P_e = P(H_0)\,P_F + P(H_1)\,P_M$ and the objective function becomes $J = P_e/P(H_1)$ with $\theta = P(H_0)/P(H_1)$. Thus, minimizing J is equivalent to the minimum error probability criterion. Obviously, the higher the signal-to-noise ratio $s(t)/\sigma$, the smaller the error probabilities P_F and P_M.

Only partial information about the template of spikes $s(t)$, $t = 1, 2, \cdots, M$ is needed to generate the optimal multithresholds. The real-time, on-line implementation is simple: compare the observation datum at every instant t with the corresponding threshold η_t; we announce that there is a spike if all thresholds are exceeded by the data.

Thus, for each class of spikes we recognized in the earlier sections, we shall compute the optimal thresholds. This proceeds by first ranking the classes according to the amplitude of their spikes is the peak-to-peak time intervals are similar. Thus, spikes in class 1 are the largest and those of class 2 are the second largest, and so forth. The largest value of class i is taken as the noise level of class $i - 1$. For classes of spikes with unique peak-to-peak time interval, the background noise in the trace is taken as the noise level.

B. Multiwindow Scheme

The multiwindow method is another fast real-time, on-line spike classification scheme since no arithmetic operations other than comparisons are involved. Here, for each test, an upper and a lower threshold is chosen to construct a window. As in the previous case, the method can also be described as a multiple hypothesis test problem. The underlying assumption about the noise is the same as in the multithreshold scheme. The decision as to which class a spike belongs is made by M comparisons with the following rule.

There are M amplitude windows for each class. The mth window for class i is $[l_{im}, u_{im})$. If all selected components in the observed time series data $\{x_m\}_{m=1}^{M}$ pass through their windows for class i, we claim that a spike in class i occurs. This decision rule may be expressed as

$$D = H_i, \text{ if } \boldsymbol{x} \in A_i \tag{22}$$

where

$$A_i = \left\{\boldsymbol{x} : x_m \in [l_{im}, u_{im}),\ m = 1, 2, \cdots, M\right\} \tag{23}$$

is another exhaustive and mutually exclusive partition of A.

As before, we can compute the probabilities of correct decision for each class

$$P_c(j) = \int_{A_j} f_j(\boldsymbol{x})\,d\boldsymbol{x} = \prod_{m=1}^{M} \left[\Phi\left(\frac{u_{im} - s_{im}}{\sigma}\right) - \Phi\left(\frac{l_{im} - s_{im}}{\sigma}\right)\right], \quad j = 1, 2, \cdots, S \tag{24}$$

and

$$P_c(0) = \int_{A_0} f_0(\boldsymbol{x})\,d\boldsymbol{x} = 1 - \int_{\cup_{i=1}^{S} A_i} f_0(\boldsymbol{x})\,d\boldsymbol{x} = 1 - \sum_{i=1}^{S} \prod_{m=1}^{M} \left[\Phi\left(\frac{u_{im}}{\sigma}\right) - \Phi\left(\frac{l_{im}}{\sigma}\right)\right] \tag{25}$$

where $\Phi(y) = \int_{-\infty}^{y} 1/\sqrt{2\pi}\, e^{-t^2/2}\,dt$.

Our aim is to optimize the window thresholds according to some criterion. Since the best performer is the matched filter, the probabilities of correct classification for each class, $P_c^*(j)$, $j = 0, 1, \cdots, S$ are the ultimate performance that can be achieved. Therefore, one chooses window thresholds such that the $P_c(j)$'s are arbitrarily close to the corresponding $P_c^*(j)$'s. Suppose that the criterion is chosen to be the weighted squared error. Then we want to minimize

$$\sum_{j=0}^{S} P_j \left[\int_{A_j} f_j(\boldsymbol{x})\,d\boldsymbol{x} - P_c^*(j)\right]^2 \tag{26}$$

with respect to l_{im}'s and u_{im}'s subject to constraints

$$A_i \neq \phi, \quad i = 1, 2, \cdots, S \tag{27}$$

and

$$A_i \cap A_j = \phi, \text{ for } i \neq j,\ i, j = 1, 2, \cdots, S. \quad (28)$$

By the Lagrange multiplier method, it is equivalent to minimize the objective function J

$$J = \sum_{j=0}^{S} P_j \left[\int_{A_j} f_j(x)\, dx - P_c^*(j) \right]^2 + a \sum_{j=0}^{S} \sum_{m=1}^{M} (u_{jm} - l_{jm} - \alpha_{jm}^2)^2 + b \sum_{i=1}^{S} \sum_{j=0}^{S} \sum_{m=1}^{M} (l_{im} - u_{jm} - \beta_{ij}^2)^2 I_{ijm} \quad (29)$$

with respect to l_{im}, u_{im}, α_{im} and β_{ij}, $i, j = 1, 2, \cdots, S$; $m = 1, 2, \cdots, M$. Where

$$I_{ijm} = \begin{cases} 1, & m = m_{ij} \text{ and } s_{im} \geq s_{jm} \\ 0, & \text{otherwise} \end{cases} \quad (30)$$

where m_{ij} is the index for the component in which the templates between class i and class j differ the most.

The optimization of the multiwindow thresholds has an interesting geometric interpretation. Imagine that the M dimemsional Euclidean space $R^m = A$ consists of $S + 1$ exhaustive and mutually exclusive subspaces A_i^*, $i = 0, 1, \cdots, S$ with arbitrary shapes. The multiwindow threshold partitions the same space with $S + 1$ super-rectangles A_i; moreover, there is a one-to-one correspondence between A_i^* and A_i. The optimization of the multiwindow thresholds tends to make each super-rectangle A_i closely approach its corresponding optimally partitioned subspace A_i^*. It is apparent therefore that the multiwindow threshold sorting scheme would perform as well as the matched filter if the template s_i's are such that every optimal subspace A_i^* is rectangular.

C. Soft-Decision Scheme

Now borrowing from communication technology, we turn to another economical detection and classification scheme called soft-decision sorting. The sums of the selected points of templates, rather than individuals of these points as in previous schemes, are taken as the decision variables that will be examined in order to make a decision. As in previous cases, S classes of spikes are to be sorted. The decision rule is described as

$$D = H_i, \text{ if } d(x) \in [\eta_{i-1}, \eta_i) \quad (31)$$

where $d(x)$ is the decision variable. The decision is made in favor of the spike class corresponding to the decision variable within the partition of the decision space for that class. Using the same underlying assumptions about the noise, the decision variable is Gaussian distributed with mean ν and variance $M\sigma^2$, leading to $S + 1$ hypotheses.

Under H_i

$$d(x) = \sum_{m=1}^{M} x_m \sim \mathfrak{N}(\nu_i, M\sigma^2) \quad (32)$$

where $\nu_i = \Sigma_{m=1}^{M} s_{im}$ is defined as the signal intensity for class i. Having these statistics, one can compute the probabilities of a correct decision ($j = i$) and of an error ($j \neq i$)

$$P_r[D_j | H_i] = P_r[d(x) \in [\eta_{j-1}, \eta_j) | H_i] = \int_{\eta_{j-1}}^{\eta_j} \phi_i(t)\, dt = \Phi\left(\frac{\eta_j - \nu_i}{\sqrt{M}\sigma}\right) - \Phi\left(\frac{\eta_{j-1} - \nu_i}{\sqrt{M}\sigma}\right) \quad (33)$$

with $\eta_{-1} = -\infty$, and $\eta_S = \infty$. Therefore, the total probability of error can be written in terms of $P_r[D_j | H_i]$ as

$$P_e = \sum_{i=0}^{S} \sum_{j \neq i}^{S} P_j P_r[D_i | H_j] = 1 - \sum_{i=0}^{S} P_i P_r[D_i | H_i] = 1 - \sum_{i=0}^{S} P_i \left[\Phi\left(\frac{\eta_i - \nu_i}{\sqrt{M}\sigma}\right) - \Phi\left(\frac{\eta_{i-1} - \nu_i}{\sqrt{M}\sigma}\right) \right]. \quad (34)$$

To optimize the scheme, the necessary condition for achieving a minimum of P_e is to set

$$\frac{\partial P_e}{\partial \eta_k} = 0, \quad k = 0, 1, \cdots, S - 1. \quad (35)$$

This results in S simultaneous equations

$$P_k \phi\left(\frac{\eta_k - \nu_k}{\sqrt{M}\sigma}\right) = P_{k+1} \phi\left(\frac{\nu_{k+1} - \eta_k}{\sqrt{M}\sigma}\right) \quad k = 0, 1, \cdots, S - 1. \quad (36)$$

Notice that $\phi(t) = d/dt\ \Phi(t)$ and the even property of $\phi(t)$ has been used. For equal probabilities, $P_k = P_{k+1}$, then we have

$$\eta_k = \frac{\nu_k + \nu_{k+1}}{2}. \quad (37)$$

With no loss of generality, one can rank the classes so that

$$\nu_1 \leq \nu_2 \leq \nu_3 \cdots \leq \nu_S$$

hence,

$$\eta_0 \leq \nu_1 \leq \eta_1 \leq \nu_2 \leq \eta_2 \cdots \leq \nu_{S-1} \leq \eta_{S-1} \leq \nu_S.$$

For the equal probability case, (37) is valid for $k = 0, 1, \cdots, S - 1$, so the minimal error probability is calculated to be

$$\min P_e = 1 + \frac{S - 1}{S + 1} - \frac{2}{S + 1} \sum_{k=0}^{S-1} \Phi\left(\frac{\nu_{k+1} - \nu_k}{2\sigma\sqrt{M}}\right). \quad (38)$$

We can see in the extreme cases

$$\min P_e \to 0 \quad \text{as } \sigma \to 0$$

and

$$\min P_e \to \frac{S}{S+1} \quad \text{as } \sigma \to \infty.$$

If there is only one class $S = 1$, it becomes a detection problem, min P_e is reduced to

$$\min P_e = 1 - \Phi\left(\frac{\nu_1}{2\sigma\sqrt{M}}\right) = 1 - \Phi\left(\frac{\gamma}{2}\right) \quad (39)$$

where $\gamma = (1/\sqrt{M}) \sum_{m=1}^{M} s_{1m}/\sigma$ is the signal-to-noise ratio. We can also show that the error probability of the matched filter is

$$P_e^* = 1 - \Phi\left(\frac{\gamma^*}{2}\right) \quad (40)$$

where $\gamma^* = \sqrt{\sum_{m=1}^{M} s_{1m}^2}/\sigma$ is the signal-to-noise ratio for the matched filter in the same situation. It is important to note that

$$\gamma \le \gamma^* \quad (41)$$

and equality holds iff the signal is time invariant, i.e., $s_{11} = s_{12} = \cdots = s_{1M}$, which implies that the matched filter is always superior to the soft-decision except for time-invariant signals, in which case the soft-decision becomes a matched filter too.

D. Comparisons Among the Different Classification Schemes

In order to evaluate the performance of these different schemes, the following comparisons were made. Consider an artificial spike with template $s_c(\tau) = 8 \sin(2\pi k\tau)$ contaminated by white Gaussian noise with zero-mean and variance $\sigma^2 = 3.5^2$. The data are sampled at frequency 12 kHz. Let $M = 3$, i.e., only three points are taken from the template, $s(1) = 8.0000$, $s(2) = -8.0000$, and $s(3) = 6.9282$. For the detection case, the comparisons of false detection P_F, missing detection P_M, and the computational complexity are given in Table I. For example, with $\theta = 1$, the optimal threshold values of the multithreshold scheme are calculated to be $\eta_1 = 1.470$, $\eta_2 = -0.215$, and $\eta_3 = 1.450$. This results in the error probabilities $P_F \le 0.0599$ and $P_M \le 0.0802$. For single-threshold detection with $s(1) = 8.0$, $\sigma = 3.5$ and the optimal threshold $\eta_1 = s(1)/2 = 4.0$, the error probabilities are $P_F = P_M = 0.1265$. It can also be seen from Table I that the soft-decision scheme performs almost as optimally as the matched filter while saving M multiplication operations. This increases implementation speed considerably.

Table II illustrates the classification comparisons, calculated using data from an epoch of extracellular recording from monkey motor cortex. The noise level was estimated to be $\sigma = 2.8604$, and three classes were extracted with three most significant points (in terms of signal-to-noise ratio) in each template: $s_1 = (24.1917, -14.7333, 19.8833)$ for class 1, $s_2 = (12.6188, -8.8000, 12.0250)$ for class 2, $s_3 = (6.6200, -4.2800, 6.3600)$ for class 3. Also estimated was a prior probability for every class to be $P_1 = 0.1$, $P_2 = 0.1$, $P_3 = 0.2$, and that for noise class to be $P_0 = 0.6$.

TABLE I
PERFORMANCE COMPARISONS FOR DETECTION WITH NOISE LEVEL $\sigma = 3.5$ AND M (=3) POINT FROM THE TEMPLATE $s = (8.0000, -8.0000, 6.9282)$

Schemes	P_F	P_M	Speed (in terms of computational operations)
Single-threshold	.1265	.1265	1 comparison
Multi-threshold ($\theta = 1$)	.0599	.0802	M comparisons
Soft-decision	.0599	.0140	$M-1$ additions and 1 comparison
Matched filter	.0599	.0139	M multiplications, $M-1$ additions and 1 comparison

TABLE II
PERFORMANCE COMPARISONS FOR CLASSIFICATION WITH NOISE LEVEL $\sigma = 2.8604$. S (=3) UNITS ARE SEPARATED WITH EACH HAVING M (=3) POINTS FROM THE TEMPLATES $s_1 = (24.1917, -14.7333, 19.8833)$, $s_2 = (12.6188, -8.8000, 12.0250)$, $s_3 = (6.6200, -4.2800, 6.3600)$ WITH PRIOR PROBABILITIES $P_0 = 0.6$, $P_1 = 0.1$, $P_2 = 0.1$, $P_3 = 0.2$.

Schemes	$P_c(0)$	$P_c(1)$	$P_c(2)$	$P_c(3)$	min P_e	Speed
Multi-window	.9318	.9527	.8318	.7778	.1069	$2MS$ comparisons
Soft-decision	.9802	.9948	.9171	.8907	.0426	$(M-1)S$ additions and S comparisons
Matched filter	.9802	.9967	.9175	.8909	.0423	MS multiplications, $(M-1)S$ additions and S comparisons

To classify one spike, the matched filter requires MS multiplications, MS additions and S comparisons, and the soft-decision operates MS additions and S comparisons while the multiwindow sorting needs only $2MS$ comparisons.

The matched filter is the best scheme for classification in the additive white Gaussian case. Its computational complexity, however, leads to slower processing speed. The alternative classification schemes are inferior in performance but superior in speed to the matched filter. In particular cases such as shown in Table II, however, they can perform as well as the matched filter.

VI. TESTING EXAMPLES

Several epochs of extracellular recording from guinea pig and ferret auditory cortexes were used to test the system. The neural signals were recorded via distributed microelectrode sensor arrays developed in collaboration with the Microelectronics Facility of the Naval Research Laboratories (NRL). The data were stored on tape and later sampled at 10 kHz. This guarantees that a spike was represented by sufficient samples, necessary for choosing M points in the templates properly to get maximal signal-to-noise ratio.

The first 60 000 samples (corresponding to 6 s of data) were used to generate templates for each class. The whole algorithm was written in a high level programming language and run on a Masscomp 5500 computer. The computer required about 1–5 min (depending on the length of learning data) to complete the extraction of templates. It took few seconds to determine the optimal thresholds for the soft-decision scheme, or about 0.5 min to finish the calculation of the optimal multithresholds for multi-

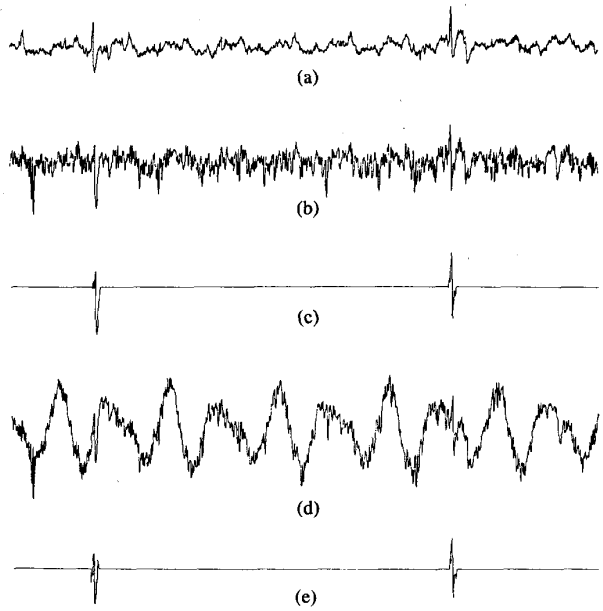


Fig. 4. Neural spikes recorded by a silicon-based multielectrode from guinea pig auditory cortex. (a) 200 ms of original data. (b) The original data contaminated by band-limited noise with a bandwidth of 2.5 kHz. (c) Spikes in noisy trace (b) detected by the multithreshold scheme. (d) Much noisier trace formed by adding 60 and 90 Hz sinusoids to the trace in (b); both spikes are totally buried in the waves. (e) Both spikes in (d) are detected by the soft-decision scheme.

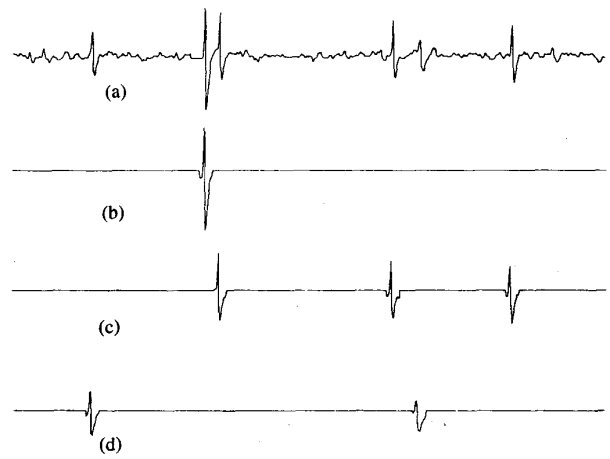


Fig. 5. Classes of neural spikes from monkey motor cortex identified by multithreshold sorting; only the first three most significant units are plotted. (a) 100 ms of original data. (b) Unit 1. (c) Unit 2. (d) Unit 3.

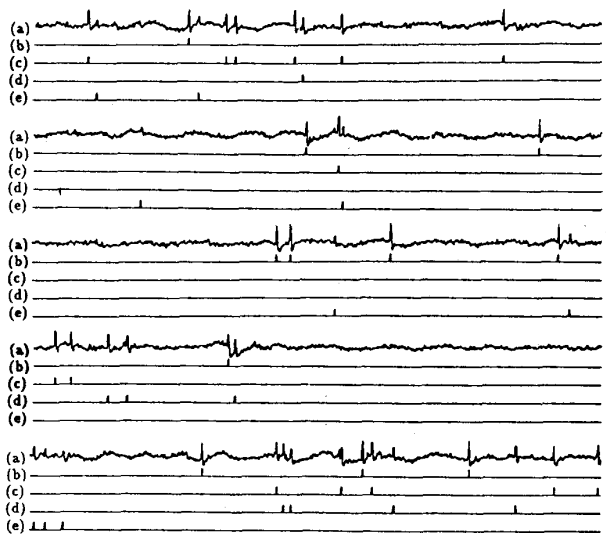


Fig. 6. One second of data containing neural spikes from ferret auditory cortex separated by soft-decision sorting, chopped into five frames with 200 ms of data each; only the first four most significant units are plotted. (a) 1 s of original data. (b) Unit 1. (c) Unit 2. (d) Unit 3. (e) Unit 4.

threshold scheme. The mean value and the variance of noise were then estimated. After that, the real-time sorting subsystem was invoked to detect and classify spikes simultaneously.

A display routine was implemented as part of the real-time sorting subsystem in which different classes of spikes are separated and highlighted in different colors on the neural activity trace as shown in Figs. 4–6. Fig. 4 shows an epoch of extracellular recording from guinea pig auditory cortex using one channel of a silicon-based, 40-channel microelectrode. Fig. 4(a) shows 200 ms of original data. Fig. 4(b) shows the original data contaminated by band-limited noise with a bandwidth of 2.5 kHz. Clearly, it is impossible to detect both spikes in this trace by a single-threshold device. Fig. 4(c) demonstrates the detection of the spikes in noisy trace Fig. 4(b) by the multithreshold scheme. In Fig. 4(d), a much noisier trace is formed by adding 60 Hz and 90 Hz sinusoids to the trace in Fig. 4(b) and both spikes are now totally buried in the waves. We see in Fig. 4(e) that both spikes in Fig. 4(d) can still be detected by the powerful soft-decision scheme. Fig. 5 shows spikes from monkey motor cortex extracellular recordings, kindly provided by Dr. E. M. Schmidt. Classes were determined in the learning subsystem and separated in the real-time sorting subsystem by the multithreshold scheme, we are only interested in the first three most significant classes for this data. Fig. 6 shows 1 s of data extracellularly recorded from ferret auditory cortex with corresponding sorted four classes performed by the soft-decision scheme; again, the first four most significant classes are interesting in this case.

A comparison of the performance of the automatic system with that of experienced human observers indicates that most spikes are detected and that the discrepancies between classifications done by human observers and the recognition system are small. In particular, for the first four most significant classes of the ferret data, the rate of missing detection was 2 percent and that of false detection was 1 percent and the discrepancies between classifications were 4 percent.

VII. Conclusion

A software-based neural spike sorting system has been developed to implement detection and classification of

signals from multiple neurons without human supervision. After the spike templates for each class are estimated from original data and the optimal multithresholds are selected based on the templates and the noise level, the real-time subsystem starts spike sorting. Testing examples show the potential of the system. Further improvements under consideration include expanding the dimension of the feature space in the template learning subsystem and developing classifiers using recursive algorithm for Haar transform detection.

Appendix

In this appendix, we prove the following

Proposition: There exists a unique minimum of the objective function $J = \theta P_F + P_M$, if not all $s(k)$ are zero and if $\theta > 1$.

Proof: Since $P_F = \Pi_{m=1}^{M} \Phi(-x_m)$, and $P_M = 1 - \Pi_{m=1}^{M} \Phi(\alpha_m - x_m)$ where $\Phi(y) = \int_{-\infty}^{y} (1/\sqrt{2\pi})\, e^{-x^2/2} dx$, $x_m = \eta_m/\sigma$, and $\alpha_m = s(m)/\sigma$. Set

$$\frac{\partial J}{\partial x_n} = 0, \qquad n = 1, 2, \cdots, M. \tag{42}$$

We have

$$\theta\phi(-x_n) \prod_{m \neq n}^{M} \Phi(-x_m) = \phi(\alpha_n - x_n) \prod_{m \neq n}^{M} \Phi(\alpha_m - x_m), \quad n = 1, 2, \cdots, M \tag{43}$$

and since $\theta > 0$, $0 < \Phi(x) < 1$, and $\phi(x) = (1/\sqrt{2\pi})\, e^{-x^2/2}$, therefore, the M simultaneous equations are equivalent to

$$\nu_n e^{-x_n^2/} = e^{-(x_n - \alpha_n)^2/2}, \quad \text{for some } \nu_n > 0,\ n = 1, 2, \cdots, M. \tag{44}$$

Suppose that $s(n) \neq 0$; hence $\alpha_n \neq 0$, then (43) has one and only one solution, which implies that there is only one extremum for J.

To see the extremum is the minimum, notice that

$$\lim_{x_m \to \infty} J = 1, \text{ for all } m \tag{45}$$

and

$$\lim_{x_1 \to -\infty} \lim_{x_2 \to -\infty} \cdots \lim_{x_n \to \infty} J = \theta. \tag{46}$$

It suffices to show that there exists an x^*, such that $J(x^*) < 1$. It is easy to see that for fixed $x_2, x_3, \cdots, x_M$, J can be expressed as

$$J = \theta_\gamma \int_{-\infty}^{-x_1} \phi(x)\, dx + 1 - \beta \int_{-\infty}^{\alpha_1 - x_1} \phi(x)\, dx \tag{47}$$

$$= (\theta_\gamma - \beta) \int_{-\infty}^{-x_1} \phi(x)\, dx - \beta \int_{-x_1}^{\alpha_1 - x_1} \phi(x)\, dx + 1 \tag{48}$$

where $0 < \gamma < \beta < 1$, and $\alpha_1 > 0$. Because $\Phi(-x) < e^{-2x^2}$, for $x > 0$; hence, we have

$$J < (\theta_\gamma - \beta) e^{-2x_1^2} - \frac{\beta\alpha_1}{\sqrt{2\pi}} e^{-x_1^2/2} + 1 < 1, \quad \text{for } x_1 > x_1^*. \tag{49}$$

This completes the proof.

Acknowledgment

The authors wish to thank Dr. J. W. Fleshman, Assistant Research Scientist with the Systems Research Center, University of Maryland, for his contribution in the guinea pig and ferret experiments and for his helpful comments and suggestions. Thanks are also due to Dr. E. M. Schmidt of the Laboratory of Neural Control, NINCDS, NIH, for kindly providing the monkey motor cortex data.

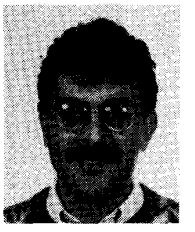
References

[1] M. Abeles and M. H. Goldstein, "Multispike train analysis," *Proc. IEEE*, vol. 65, pp. 762–772, 1977.
[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Some considerations of the modified Walsh–Hadamard and Haar transforms," in *Proc. 1973 Symp. Appl. Walsh Functions*, pp. 91–95.
[3] J. Bak and E. M. Schmidt, "An analog delay circuit for on-line visual confirmation of discriminated neuroelectric signals," *IEEE Trans. Biomed. Eng.*, vol. BME-24, pp. 69–71, 1977.
[4] —, "An improved time-amplitude window discriminator," *IEEE Trans. Biomed. Eng.*, vol. BME-24, pp. 486–489, 1977.
[5] J. C. Dill, P. C. Lockeman, and K. Naka, "An attempt to analyze multiunit recordings," *Electroencephalogr. Clin. Neurophysiol.*, vol. 28, pp. 79–82, 1970.
[6] G. J. Dinning and A. C. Sanderson, "Real-time classification of multiunit neural signals using reduced feature sets," *IEEE Trans. Biomed. Eng.*, vol. BME-28, pp. 804–812, 1981.
[7] G. L. Gerstein and W. A. Clark, "Simultaneous studies of firing patterns in several neurons," *Science*, vol. 143, pp. 1325–1327, 1964.
[8] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Math. Ann.*, vol. 69, pp. 334–371, 1910; vol. 71, pp. 38–53, 1912.
[9] J. Krüger, "Simultaneous individual recordings from many cerebral neurons: Techniques and results," *Rev. Physiol. Biochem. Pharmacol.*, vol. 98, pp. 177–233, 1983.
[10] G. D. McCann, "Interactive computer strategies for living nervous system research," *IEEE Trans. Biomed. Eng.*, vol. BME-20, pp. 1–11, 1973.
[11] D. J. Mishelevich, "On-line real-time digital computer separation of extracellular neuroelectric signals," *IEEE Trans. Biomed. Eng.*, vol. BME-17, pp. 147–150, 1970.
[12] R. O'Connell, W. A. Kocsis, and R. L. Schoenfeld, "Minicomputer identification and timing of nerve impulses mixed in a single recording channel," *Proc. IEEE*, vol. 61, pp. 1615–1621, 1973.
[13] M. N. Oguztoreli and R. B. Stein, "Optimal filtering of nerve signals," *Biol. Cybern.*, vol. 27, pp. 41–48, 1977.
[14] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes—I: The single spike train," *Biophys. J.*, vol. 7, pp. 391–418, 1967.
[15] —, "Neuronal spike trains and stochastic point processes—II: Simultaneous spike trains," *Biophys. J.*, vol. 7, pp. 419–440, 1967.
[16] V. J. Prochazka and H. H. Kornhuber, "On-line multiunit sorting with resolution of superposition potentials," *Electroencephalogr. clin. Neurophysiol.*, vol. 34, pp. 91–93, 1973.
[17] W. M. Roberts, "Optimal recognition of neuronal waveforms," *Biol. Cybern.*, vol. 35, pp. 73–80, 1979.
[18] M. D. Srinath and P. K. Rajasekaran, *An Introduction to Statistical*

Signal Processing with Application. New York: Wiley, 1979, pp. 79-83.
[19] D. Stagg, "Computer acquisition of multiunit nerve-spike signals," *Med. Biol. Eng.*, vol. 11, pp. 340-347, 1973.
[20] R. B. Stein, S. Andreassen, and M. N. Oguztoreli, "Mathematical analysis of optimal multichannel filtering for nerve signals," *Biol. Cybern.*, vol. 32, pp. 19-24, 1979.
[21] R. B. Stein, S. Andreassen, and M. N. Oguztorelia, "Application of optimal multichannel filtering to simulated nerve signals," *Biol. Cybern.*, vol. 32, pp. 25-33, 1979.

Xiaowei Yang received the diploma of physics from Wuhan University, Wuhan, China, in 1978, the M.S. degree in radio-electronics from East China Normal University, Shanghai, in 1982, and the M.S. degree in electrical engineering in 1988. He is currently a Ph.D. candidate in electrical engineering at the University of Maryland, College Park.

From 1978 to 1979 he worked at Wuhan University where he performed research in ultrahigh frequency circuits. From 1982 to 1983 he served as an Instructor at East China Normal University. While studying at the University of Maryland, he was a Graduate Teaching Assistant from 1984 to 1986 and has been a Graduate Research Assistant since June 1986. His research interests include various aspects of statistical signal processing and system identification.



Shihab A. Shamma (M'85) lived in Baghdad, Iraq. He received the B.Sc. degree in electrical engineering from the Imperial College, London University, London, England, in 1976, the M.S. degree in electrical engineering, the M.A. degree in Slavic Languages and Literature, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1977, 1980, and 1980, respectively.

From 1981-1984, he was at the National Institutes of Health involved in neurophysiological research and mathematical modeling of the mammalian auditory system. Since 1984, he has been with the Department of Electrical Engineering, University of Maryland, College Park. His research work has focused on computational models of the auditory nervous system, and their applications to automatic speech recognition, learning, and the understanding of auditory processing of complex sounds.