TR-86-62

A Totally Automated Neural Spike Detection and Classification Scheme: A Preliminary Software System

by

Shihab A. Shamma
and
Xiaowei Yang

A Totally Automated Neural Spike Detection and Classification Scheme: A preliminary software system

Xiaowei Yang Dr. Shihab A. Shamma

26 November 1986

Abstract

A system for neural spike detection and classification is presented, which does not require a priori assumptions about spike presence or spike templates, and assumes only that the background noise has a Gaussian distribution. The system is divided into two parts: a learning subsystem and a real-time detection and classification subsystem. The former extracts templates of spikes for every class, which includes a feature learning phase and a template learning phase. The latter picks up 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. Performance of the system is illustrated by using it to classify spikes in a segment of neural activity recorded from monkey motor cortex. The system is implemented without human supervision so that it can be extended for multi-channel recording without loss of real-time property.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Edward M. Schmidt of the Laboratory of Neural Control, NINCDS, NIH, for kindly providing the monkey cortex data and Dr. James. W. Fleshman, Assistant Research Scientist with the Systems Research Center, University of Maryland, for his helpful comments and suggestions.

Contents

List of Figures

I Introduction

For studying nervous system organization at the cellular level, the analysis of extracellularly recorded neural spikes is essential [13], [14]. An extracellular electrode often records electrical activity from several adjacent neurons simultaneously. To analyze the contribution of each individual unit, one needs to distinguish the signals of each unit from the rest. Signals from different neurons can be classified by characteristic spike shapes, which are unpredictable functions of the neuron type, electrode construction, 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 weaker neural signals of distant units.

There are several techniques for classification of multiunit neural signals [1], [3]–[7], [9]–[12], [15], [16], [18]–[20]. Apart from their usefulness in particular situation, there are some common limitations. Some techniques require some information about spikes and epoches [3]–[7], which is often not available; some methods employ time consuming computations [1], [5], [7], [9], [11], [12], [15], [16], [18]–[20], making real-time implementation impossible except with specialized hardware; others, involve some human supervision [3], [4], [10]. Among them, the techniques using hardware equipment such as window discriminators, work only in low noise environments.

This paper proposes an efficient and convenient on-line multispike separation system which is totally automated from detection to classification, without presupposing knowledge of spike shapes. It is assumed that there are several classes of spikes in the Gaussian noise background on the neuron activity trace. First the Haar transformation is performed to locate the occurrence of the spikes in time. Then a two dimensional histogram is made to determine how many classes there are on the trace, based on two particular features of spikes: peak-to-peak amplitude and peak-to-peak time interval. Classifying each spike according to its features, templates of spikes of each class are generated by means of averaging. Once templates are ready, an optimal multi-threshold spike sorting algorithm is developed which minimizes minimum error probabilities while performing both detection and classification. The optimal thresholds are updated regularly thereafter to follow the slow variation of the spike template for each class estimated previously and the background noise level estimated by its statistics. An example of multiunit extracellular recording from primate motor cortex

is used to test the system.

The following section describes the whole system. The principle underlying the use of the Haar transformation to detect spikes in noise is given in section III. Section IV presents the feature selection and classification criteria. The multi-threshold sorting is detailed in section V.

II System Description

The system is divided into two parts as indicated in the block diagram Figure 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 picks up 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 the sampled data which contains spikes from several neurons corrupted noise is stored in a memory buffer. The algorithm begins with the detection of spikes from noise, using the discrete environment transformation (DHT). Then the extraction of features is performed on these detected spikes, to construct the feature histogram. The typical features for each class are selected according to the histogram, finishing the feature learning phase. In the template learning phase, the same DHT detection scheme is applied to detect spikes. Then they are sorted into classes by comparing them with the typical features. As a result, the typical templates for each class are averaged using all the templates of spikes within the class, which completes the learning subsystem.

The real-time detection and classification subsystem is supported by the the multi-threshold sorting scheme. Once the optimal multi-thresholds for every class are derived from the templates and the background noise, the real-time on-line implementation begins.

III Detection by Haar Transformation

The discrete Haar transform detection scheme plays the key role in the learning subsystem. Because its inherent characteristic of the transform basis is spike-like, the Haar transform is a powerful tool for detecting neural spikes. In transforming the original time function, there

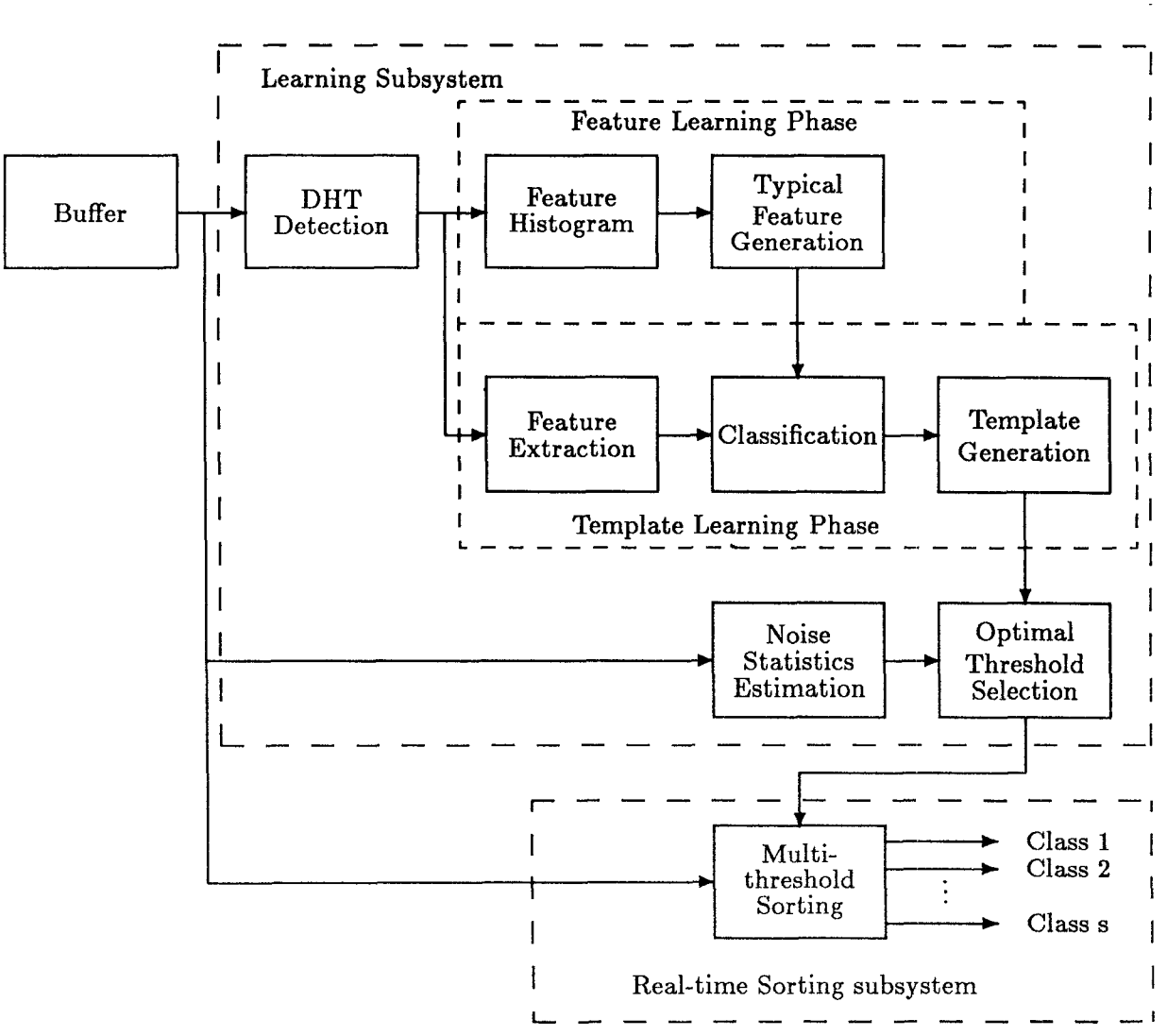

Figure 1: Schematic Diagram of the System

will be a large component 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 is defined in [0, 1) as [8]

$$h(0,\ 0,\ t)\ =\ 1,\quad t\ \epsilon\ [\ 0,\ 1\ )$$

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

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

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

where each element is a sample of the Haar function

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

$$w_k(i)\ =\ w_{2^m+n-1}(i) \qquad (3)$$

$$=\ h(m,\ n,\ i) = \begin{cases} 0, & i\ =\ 0,\ \cdots,\ N(n-1)/2^m\ -1 \\ 2^{m/2}, & i\ =\ N(n-1)/2^m,\ \cdots,\ N(n-1/2)/2^m\ -1 \\ -2^{m/2}, & i\ =\ N(n-1/2)/2^m,\ \cdots,\ Nn/2^m\ -1 \\ 0, & i\ =\ Nn/2^m,\ \cdots,\ N-1 \end{cases} \qquad (4)$$

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

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

Case 1 : suppose that

$$x_i = \begin{cases} a, & i = 0, 1 \\ -a, & i = 2, 3 \\ 0, & otherwise \end{cases}$$

there is a pulse corresponding to the spike in the transform domain

$$y_k = \begin{cases} a_N \ , & k = N/4 \\ 0 \ , & otherwise \end{cases}$$

where $a_N = 2a\sqrt{N}$.

Case 2 : suppose that

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

the resulting transformed sequence is

$$y_k = \begin{cases} 1.540\, a_N \ , & k = N/4+1 \\ 0.363\, a_N \ , & k = N/2+2 \\ -0.363\, a_N \ , & k = N/2+3 \\ 0 \ , & otherwise \end{cases}$$

where, again, $a_N = 2a\sqrt{N}$. The above examples correspond to the case where no noise is present. It is seen that the outstanding 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 spike. We continue to use Case 2 as an example to illustrate how this process work. The threshold t in the transform domain is set to be $|t| = 0.5a_N$, so the filtered transformed sequence $\hat{\mathbf{y}} = [\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_{N-1}]$ is expressed as

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

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

$$\hat{x}_i = \begin{cases} 1.54\, a_N\ , & i = 4,\ 5 \\ -1.54\, a_N\ , & i = 6,\ 7 \\ 0\ , & otherwise \end{cases}$$

The 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 i. i. d. with common Gaussian distribution $\mathcal{N}\ (0,\ \sigma^2)$, it can be easily show that the component of the transformed sequence $\{\ y_k\ :\ k\ =\ 0,\ 1,\ \cdots,\ N-1\ \}$ is identical distributed with common Gaussian distribution $\mathcal{N}\ (0,\ N\sigma^2)$. It is known that $P_{rob}[|y_k| \geq 3.0\sigma\sqrt{N}] < 0.0028$, and that $P_{rob}[|y_k| \geq 4.1\sigma\sqrt{N}] = 0.000042$. A nonlinear transform domain threshold is set to be $t = C(\sigma)\sigma\sqrt{N}$, where $C(\sigma)$ is between 3.0 and 4.15 and is dependent on the value of σ. If $\{\ y_k\ \}$ is the transformation of the noise only, then the probability that $|y_k|$ exceeds the threshold t is very small. That is the reason that we filter the noise by setting every component below the threshold to be zero. At this stage, the HT detection picks reliable spikes but may miss some. Our goal at this stage is simply to accumulate enough reliable spikes to construct the histogram and templates, and not to detect all spikes. Figure 2 demonstrates the HT detection.

In practice, $C(\sigma)$ is chosen as

$$C(\sigma) = \begin{cases} 3.00, & \sigma = 0.0 \\ 3.10, & \sigma = 0.8 \\ 3.90, & \sigma = 4.0 \\ 4.00, & \sigma = 6.0 \\ 4.15, & \sigma = 10.0 \end{cases} \qquad (5)$$

A polynomial of σ may be formed to approach the nonlinearity $C(\sigma) = \sum_{i=0}^{4} c_i\sigma^i$, with coefficients $c_0 = 3.00$, $c_1 = 0.0329$, $c_2 = 0.1370$, $c_3 = -0.0284$, $c_4 = 0.001557$.

IV Feature Extraction and Classification

In order to classify spikes, one may either use the complete templates, or instead base it on selected unique features of those spikes. In principle, spikes belonging to the same class

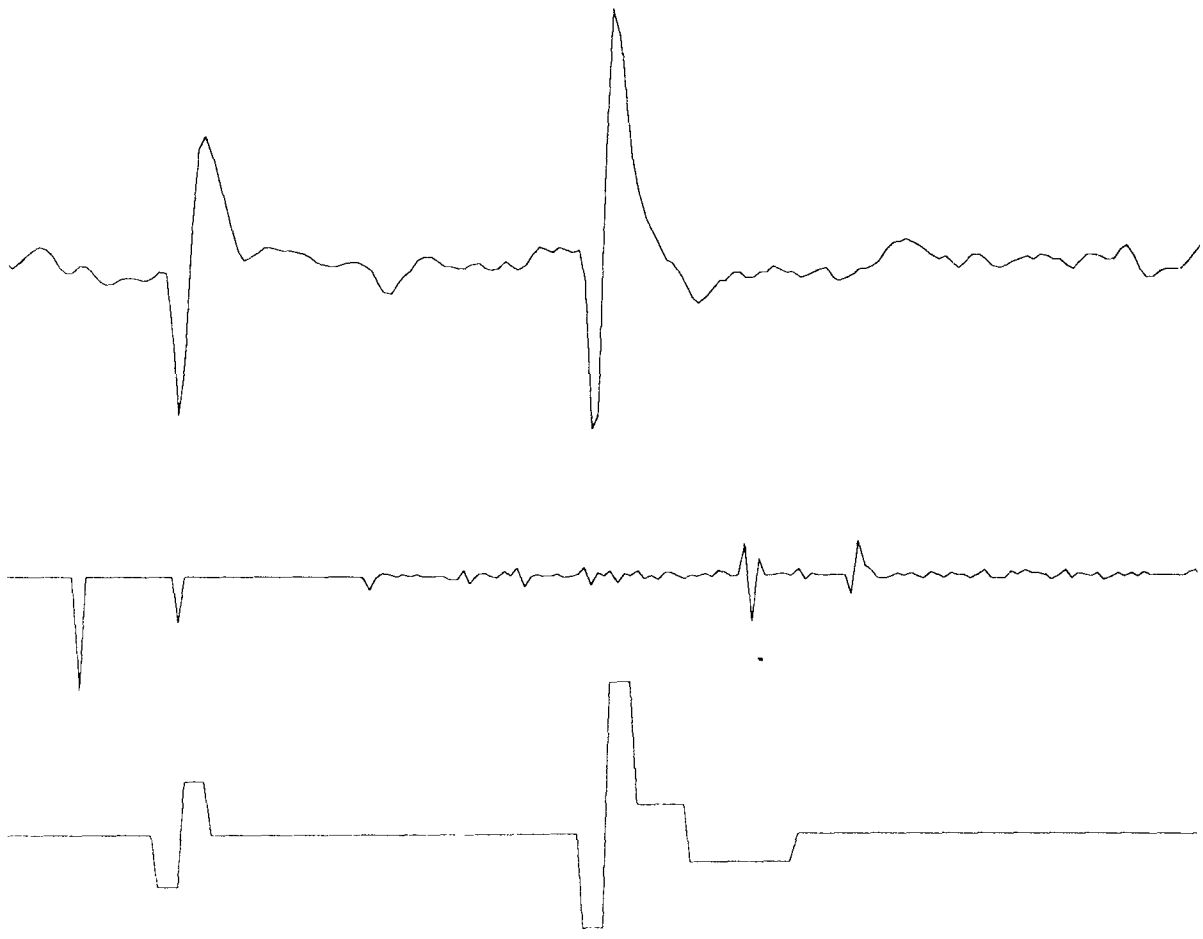

Figure 2: Spikers Detected by the Haar Transform: a) The origional trance. b) The transform domain equivalence, large components imply that there are spikes. c) Reconstructed spikes in the time domain, the noise is filled out, the spikes are detected by zero-crossing detection.

are supposed to be relatively close in features. To simplify the algorithm, two features are considered for spikes here. One is 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, then the histogram essentially reflects 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 features for some class. Of course, we only consider those peaks that are mutually apart from one another, i. e., the relative distance of each pair of peaks is larger than a specific value. For illustration, suppose $f(\xi)$ has peaks at ξ' and ξ''. If

$$\frac{|\xi' - \xi''|}{min(|\xi'|, |\xi''|)} > \epsilon \tag{6}$$

or equivalently,

$$\frac{|\xi' - \xi''|}{max(|\xi'|, |\xi''|)} > \frac{\epsilon}{1+\epsilon}. \tag{7}$$

where $\epsilon > 0$ is a specific value, then we say that the relative distance between the peak at ξ' and the peak at ξ'' is legal.

The question which naturally arises now is how accurate are the typical feature values determined in this way, assuming that there is no misclassification problem? Consider following example. For simplicity, only one feature is considered. Suppose that there are S classes with a prior probability 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 p. d. f.

$$f(\xi) = \sum_{i=1}^{S} P_i f_i(\xi|i) \tag{8}$$

where $f_i(\xi|i)$, is the p. d. f. 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 $f_i(\xi|i)$ has the peak at $\xi = \mu_i$. According to the feature selecting criterion, therefore, 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

$$\hat{\mu}_i = \frac{1}{n} \sum_{\xi_k \epsilon C_i} \xi_k \tag{9}$$

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

$$P_{rob}[\hat{\mu}_i - \frac{\sigma}{\sqrt{n_i}}t_\alpha \leq \mu_i \leq \hat{\mu}_i + \frac{\sigma}{\sqrt{n_i}}t_\alpha] = 1 - \alpha \qquad (10)$$

In words, one has $(1-\alpha)100$ percent confidence that the true value μ_i is in the interval $[\ \hat{\mu}_i - \frac{\sigma}{\sqrt{n_i}}t_\alpha,\ \ \hat{\mu}_i + \frac{\sigma}{\sqrt{n_i}}t_\alpha\ ]$. The length of the interval is $2\sigma t_\alpha/\sqrt{n_i}$, where t_α is chosen so that $P_{rob}[|N(0,1)| \leq t_\alpha] = 1-\alpha$, where $N(0,1)$ is a normalized Gaussian random variable. To obtain high confidence, choose $\alpha > 0$ small enough while increasing sample size n_i sufficiently large, still, the length of the interval can be made arbitrary small. Hence $\hat{\mu}_i$ is a consistent estimator of μ_i . The final step is to show that $\hat{\mu}_i$ is the value at which $f_i(\xi|i)$ 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 $P_{rob}[\lim_{n_i\to\infty}\hat{\mu}_i = \mu_i] = 1$ and it is apparently that $P_{rob}[\lim_{n_i\to\infty} Var(\hat{\mu}_i) = 0\ ] = 1$, i. e., $\hat{\mu}_i$ will approach to μ_i with no variation so that $f_i(\hat{\mu}_i|i)$ is definitely the peak as n_i goes to infinity.

Thus the question is answered in partial. 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. It is difficult to give an answer in general, but as long as the confidence interval estimation is possible to make and if the length of the interval shrinks as n increases, then we should arrive at the same conclusion.

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

$$\bar{h}(i,\ j) = \frac{1}{W}\sum_{(k,l)\epsilon N_b[(i,j),\ r]} w_{i-k\ j-l}h(k,\ l) \qquad (11)$$

where $W = \sum_{(k,l)\epsilon N_b[(0,0),\ r]} w_{kl}$, and $N_b[(i,\ j),\ r]$ is defined as the r-neighborhood of $(i,\ j)$.

Consider now the misclassification problem. S classes are assumed to have a prior probabilities P_i, $i = 1, 2, \cdots, S$. Using the risk function concept [17], we associate a risk r_{ij} with choosing class i when the correct classification is class j. The average risk is given by

$$R = \sum_{i=1}^{S}\sum_{j=1}^{S} r_{ij}P_jP(D_i|j) \qquad (12)$$

where $P(D_i|j)$ is the probability of choosing class i when class j is true. The feature space C is partitioned into exhaustive and mutually exclusive subset $C_i, i = 1, 2, \cdots, S$, so that spikes with feature ξ are classified to be in class i if $\xi \epsilon\ C_i$. Noting that $C_i = C - \bigcup_{j\neq i} C_j$

and $\int_C f_j(\xi|j)d\xi = 1$, $j = 1, 2, \cdots, S$, the average risk can be expressed as

$$R = \sum_{i=1}^{S}\sum_{j=1}^{S} r_{ij}P_j \int_{C_i} f_j(\xi|j)d\xi \quad (13)$$

$$= \sum_{i=1}^{S} r_{ii}P_i + \sum_{i=1}^{S} \int_{C_i} \sum_{j\neq i}^{S} (r_{ij} - r_{jj})P_j f_j(\xi|j)d\xi \quad (14)$$

$$= \sum_{i=1}^{S} r_{ii}P_i + \sum_{i=1}^{S} \int_{C_i} d_i(\xi)d\xi \quad (15)$$

where $d_i(\xi) = \sum_{j\neq i}^{S}(r_{ij} - r_{jj})P_j f_j(\xi|j)$ ($i = 1, 2, \cdots, ..., S$) are the decision variables. In order to minimize the risk R, we like to minimize $d_i(\xi)$ for every fixed ξ for each class. The classification strategy is simple, the observed value ξ is claimed to be in class i, if $i = arg\min_j\{d_j(\xi)\}$.

For performance measurement, it is mathematically convenient to derive first the probability that the classifier makes a correct decision. This is the probability that $d_i(\xi)$ is exceeded by all other decision variables $d_j(\xi)$ under class i, and it may be expressed as

$$P_c(i) = P_{rob}[d_i(\xi) \leq d_j(\xi), \ for\ all\ j|\ Class\ i\ ] \quad (16)$$

$$= \int_{A_i} f_i(\xi|i)d\xi \quad (17)$$

where $A_i = \{\xi : d_i(\xi) \leq d_j(\xi), \ for\ all\ j\}, (i = 1, 2, \cdots, S)$ are measurable subsets in the feature space. So the total probability of correct classification is $P_c = \sum_{i=1}^{S} P_i P_c(i)$, and the probability of error is $P_e = 1 - P_c$.

Of particular interest is the case where $r_{ij} = 1$ for $i \neq j$ and $r_{ii} = 0$ for all i and j, then $d_i(\xi)$ reduces to

$$d_i(\xi) = \sum_{j\neq i}^{s} P_j f_j(\xi|j) = \sum_{j\neq i}^{s} P(j|\xi)f(\xi) = [1 - P(i|\xi)]f(\xi) \quad (18)$$

The criterion becomes to be the minimum probability of error.

V Multi-threshold Sorting

The multi-threshold method is a fast real-time on-line spike detection scheme. Unlike the one-threshold detection that is commonly used, the multi-threshold 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 one-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 spike and noise. Denote observation, spike and noise as $x(t)$, $s(t)$ and $n(t)$, respectively. The hypotheses are stated as follows.

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

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* $t = 1, 2, ..., M$, 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. Under the assumption that the noise is white Gaussian distributed with zero-mean and variance σ^2, the probability of the false alarm P_F can be expressed as

$$P_F = P_{rob}(D_1|H_0) = P_{rob}[x(1) > \eta_1, x(2) > \eta_2, ..., x(M) > \eta_M|H_0] \qquad (19)$$

Because $n(t)$ is white, $x(t)$'s are mutually independent and with identical Gaussian distribution $\mathcal{N}(0, \sigma^2)$, therefore, P_F can be further expressed in terms of the error function $\Phi(\,)$.

$$P_F = \prod_{t=1}^{M} P_{rob}[x(t) > \eta_t|H_0] = \prod_{t=1}^{M} \Phi(-\frac{\eta_t}{\sigma}) \qquad (20)$$

where $\Phi(y) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$. Similarly, the probability of the missing P_M can be written

$$P_M = P_{rob}(D_0|H_1) = P_{rob}[x(t) < \eta_t, \; for \; some \; t \,|H_1] \qquad (21)$$

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

$$P_M = 1 - P_{rob}(D_1|H_1) = 1 - P_{rob}[x(1) > \eta_1, x(2) > \eta_2, ..., x(M) > \eta_M|H_1] \qquad (22)$$

By the same token, in terms of the error function $\Phi(.)$, P_M may be written

$$P_M = 1 - \prod_{t=1}^{M} \Phi(\frac{s(t) - \eta_t}{\sigma}) \qquad (23)$$

It is not difficult to see that if one likes 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 compromise them by constructing an objective function J,

$$J = \theta P_F + P_M \ , \quad for\ some\ \theta > 0 \qquad (24)$$

the goal is now to minimize J.

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

$$\frac{\partial J}{\partial \eta_k} = 0 \ , \quad k = 1, 2, ..., M \qquad (25)$$

This results in M simultaneous equations

$$\theta \prod_{l \neq k}^{M} \Phi(-\frac{\eta}{\sigma_l}) = e^{(2\eta_k - s(k))s(k)/2\sigma^2} \prod_{l \neq k}^{M} \Phi(\frac{s(l) - \eta_l}{\sigma}) \ , \quad k = 1, 2, ..., M \qquad (26)$$

The possible optimal set of thresholds η_k 's yields upon solving those equations.

The significance of θ may be explained as follows. If the false alarm is as costly as the missing detection, one sets $\theta = 1$. And $\theta > 1$ means that the false alarm is more costly than the missing, 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 probability of error criterion. Obviously, the higher the signal to noise ratio $s(t)/\sigma$, the smaller the error probabilities P_F and P_M.

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

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

Only partial information about the template of spikes $s(t), t = 1, 2, ..., M$ is needed to generate the optimal multi-thresholds. The real time on-line implementation is simple: Compare the 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.

Example: Suppose that the spike with template $s_c(\tau) = 8\sin(2\pi f\tau)$ is contaminated by white Gaussian noise with zero-mean and variance $\sigma^2 = 3.5^2$. The data is sampled at frequence $12f$ Hz. Let $M = 3$, i. e., only three points are taken from the template, $s(1) = 8$, $s(2) = 6.9282$, and $s(3) = 8$. Choose $\theta = 1$, the optimal threshold values are calculated to be $\eta_1 = 1.47$, $\eta_2 = -0.215$, and $\eta_3 = 1.45$, resulting in the error probabilities $P_F \leq 0.0599$ and $P_M \leq 0.0802$. While the one-threshold detection ($M = 1$) with $s(1) = 8$, $\sigma = 3.5$ and the optimal threshold $\eta_1 = s(1)/2 = 4$ gives $P_F = P_M = 0.1265$.

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 if the peak-to-peak time interval are similar. Thus, spikes in class 1 are those with the largest amplitude and those of class 2 are the second largest, and so forth, and by taking the template value of class i 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.

VI A Testing Example

An epoch of extracellular recording from monkey motor cortex was used to test the system. The data were stored on cassette tape and sampled by a MASSCOMP minicomputer at 10 kHz, guaranteeing that a spike was represented by at least 8 samples, necessary for accurate thresholding.

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 programming language, it took 5 min to complete the extraction of templates and 3 min to finish the calculation of the optimal multi-thresholds.

Five peaks appeared in the two-dimensional histogram. One of them was very small, which meant that the number of spikes in that region was less than one percent of the total; another peak failed the distance test. Therefore, these two peaks were disregarded, resulting in three classes of neural spikes.

A display routine has been implemented as part of the real-time sorting subsystem in which different classes of spikes are highlighted in different colors on the neural activity trace as shown in Fig 3 . A comparison of the performance of the automatic system with

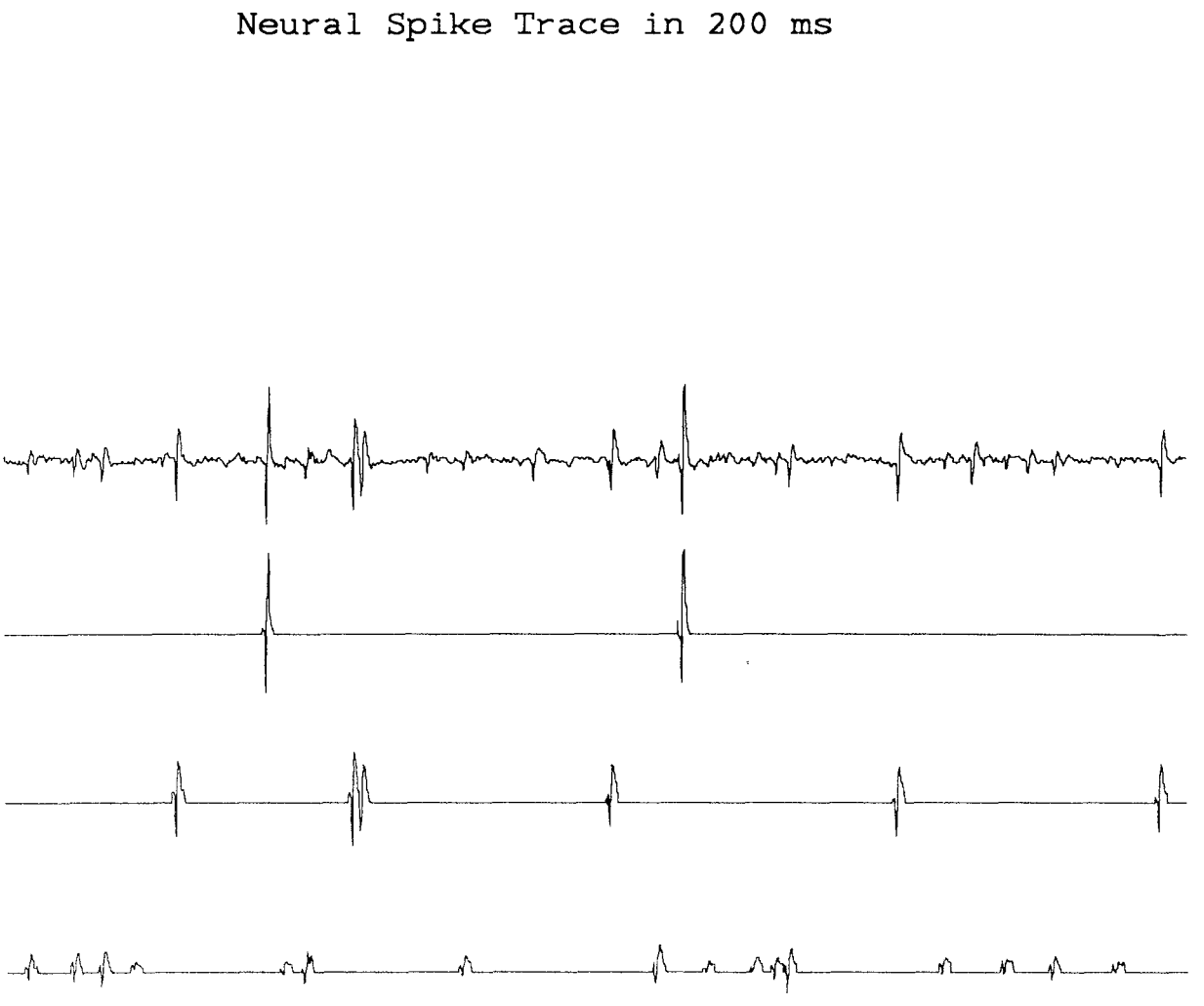

Figure 3: Spikers Classified by the System: a) The origional trance. b) Class 1. c) Class 2. d) Rest of spikes.

that of experienced human observers indicates that most larger spikes are detected and the discrepancies between classifications done by human observers and the recognition system are small.

VII Conclusion

A preliminary software 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 multi-thresholds are selected based on the templates and the noise level, a process which requires several minutes, the real-time subsystem starts spike sorting. A testing example shows the potential of the system. Further improvements under consideration include expanding the dimension of the feature space by using template matching techniques in the template learning subsystem and changing from multi-threshold to multi-window criteria in the real-time sorting subsystem.

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

$$\frac{\partial J}{\partial x_n} = 0, \; n = 1, 2, \cdots, M \tag{27}$$

we have

$$\theta\phi(-x_n) \prod_{m \neq n}^{M} \Phi(-x_m) \; = \; \phi(\alpha_n - x_n) \prod_{m \neq n}^{M} \Phi(\alpha_m - x_m), \; n = 1, 2, \cdots, M \tag{28}$$

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

$$\nu_n e^{-x_n^2/} \; = \; e^{-(x_n - \alpha_n)^2/2}, \; for \; some \; \nu_n > 0, \; n = 1, 2, \cdots, M \tag{29}$$

Suppose that $s(n) \neq 0$, hence $\alpha_n \neq 0$, then (2.3) has one and only one solution, which implied that there is only one extremum for J.

To see the extremum is the minimum, notice that

$$\lim_{x_m \to \infty} J \; = \; 1, \; for \; all \; m \tag{30}$$

and

$$\lim_{x_1 \to -\infty} \lim_{x_2 \to -\infty} \cdots \lim_{x_n \to \infty} J \; = \; \theta \tag{31}$$

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

$$J = \theta\gamma \int_{-\infty}^{-x_1} \phi(x) dx + 1 - \beta \int_{-\infty}^{\alpha_1 - x_1} \phi(x) dx \tag{32}$$

$$= (\theta\gamma - \beta) \int_{-\infty}^{-x_1} \phi(x) dx - \beta \int_{-x_1}^{\alpha_1 - x_1} \phi(x) dx + 1 \tag{33}$$

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

$$J \; < \; (\theta\gamma - \beta) e^{-2x_1^2} - \frac{\beta\alpha_1}{\sqrt{2\pi}} e^{-x_1^2/2} + 1 \; < \; 1, \; for \; x_1 > x_1^* \tag{34}$$

This completes the proof.

Bibliography

[1] M. Abeles and M. H. Goldstein, "Multispike train analysis," *Proc. IEEE*, Vol. 65, pp. 762–772, 1977.

[2] N. Ahmed and K. R. Rao , *Orthogonal Transforms for Digital Signal Processing*, 1975.

[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] J. Bak and E. M. Schmidt, "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," *Electroenceph. 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(1910), pp. 334–371, 1910; Vol. 71(1912), pp. 38–53, 1912.

[9] G. D. McCann, "Interactive computer strategies for living nervous system research," *IEEE Trans. Biomed. Eng.*, Vol. BME-20, pp. 1–11, 1973.

[10] 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.

[11] 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.

[12] M. N. Oguztoreli and R. B. Stein, "Optimal filtering of nerve signals," *Biol. Cyber.*, Vol. 27, pp. 41–48, 1977.

[13] 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.

[14] D. H. Perkel, G. L. Gerstein and G. P. Moore, "Neuronal spike trains and stochastic point processes, II. Simultaneous spike train," *Biophys. J.*, Vol. 7, pp. 419–440, 1967.

[15] V. J. Prochazka and H. H. Kornhuber, "On-line multi-unit sorting with resolution of superposition potentials," *Electroenceph. Clin. Neurophysiol.*, Vol. 34, pp. 91–93, 1973.

[16] W. M. Roberts, "Optimal recognition of neuronal waveforms," *Biol. Cyber.*, Vol. 35, pp. 73–80, 1979.

[17] M. D. Srinath and P. K. Rajasekaran, *An Introduction to Statistical Signal Processing with Applications*, John Wiley & Sons, New York, 1979.

[18] D. Stagg, "Computer acquisition of multiunit nerve-spike signals," *Med. Biol. Eng.*, Vol. 11, pp. 340–347, 1973.

[19] R. B. Stein, S. Andreassen and M. N. Oguztoreli, "Mathematical analysis of optimal multichannel filtering for nerve signals," *Biol. Cyber.*, Vol. 32, pp. 19–24, 1979.

[20] R. B. Stein, S. Andreassen and M. N. Oguztorelia, "Application of optimal multichannel filtering to simulated nerve signals," *Biol. Cyber.*, Vol. 32, pp. 25–33, 1979.