AN AUTOMATED LEARNING PHASE SYSTEM FOR MULTIUNIT NEURAL CLASSIFICATION

BY

AMODEO WILLIAM VALENTI

B.S., University of Wisconsin-Milwaukee, 1985

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1987

Urbana, Illinois

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Bruce Wheeler, for introducing me to the field of bioengineering and for always finding the time to answer all my questions. I thank Scott Smith for his help and guidance and whose spirit always kept me on top of things. Above all, I would like to thank my parents and Sue Gegenhuber whose love and support I consider a blessing.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

One of the most challenging tasks in biology today is to improve our present knowledge of how the nervous system communicates information. Reliable observation of a small population of cells is an important first step toward such an understanding. To accomplish this, the neurophysiologist uses a microelectrode made of metal, insulated to the tip, or a fluid-filled micropipette, to record the electric signals produced by the neurons. Often, when the microelectrode is inserted among the nerve cells or fibers, it is possible to observe a number of spike waveforms clearly attributable to single nerve cells. The signal from the electrode is usually digitized to facilitate processing and storage. The waveforms are then separated by shape and amplitude into classes which, hopefully, correspond to each of the nerve cells. Hence the experimenter has multiple neuron data with which to test biologically relevant hypotheses.

With the growth in interest in multiple element electrode arrays for neural recording [7,17,30,31], researchers are more interested in utilizing multiple unit separation techniques to aid in their analyses. Multiple channel data also make it increasingly important to develop, understand, and utilize automated spike separation techniques. The combination of multiple

channels and high sample and event rates makes it impossible for a single researcher to follow all of the neural activity and to make adjustments to instruments in order to optimize data acquisition. Further, data compression utilizing automatic spike recognition is necessary for logging data, especially in continuous time recording.

The overall spike separation task can be divided into two phases: an operational phase in which data are analyzed, either on- or off- line, and a learning phase, in which the parameters for the operational phase are determined.

This thesis addresses the automation of the learning phase of the multiunit data separation problem. The specific goals of the project were: 1) the creation of a software tool which would automatically learn to classify action potentials based on several algorithms reported to be most efficient; and, 2) the comparison of the performance of these algorithms for spike detection and separation for test data with varying signal-to-noise ratios as well as on sample data from several representative neurobiological preparations.

1.1 Problem Definition

Inherent in some computer separation systems is the interactive learning phase. In some systems this is defined as the visual selection of spike shapes or the mean of similar spikes selected by the experimenter to be used as "standards" or templates [10]. Other systems sort spikes during the learning phase by utilizing measures such as amplitude, latency, slope and

area. In either case, hand quantification of such parameters is not only time-consuming but often introduces both bias and random errors [21]. The purpose of this thesis is to present a solution to the supervised learning phase problem by offering an automated learning phase algorithm.

The balance of this chapter introduces the characteristics of neural data, summarizes the state-of-the-art in the separation of multiunit recordings, including learning phase algorithms, and outlines the specific tasks undertaken.

1.2 Nature of Neural Data

A typical action potential used in this thesis is shown in Figure 1.1. Most spike waveforms have durations which range from 0.5 to 5.0 ms and amplitudes of from 50 to 500 μV, both of which vary somewhat from species to species. An intracellular recordings of electrical activity is obtained using an electrode that penetrates the membrane of the neuron. Extracellular recordings are obtained using an extracellular electrode that does not penetrate the neuron membrane.

An intracellular recording often permits the observation of the excitatory and inhibitory post-synaptic potentials that are generated by stimulation of the cell by presynaptic neurons. Because the propagation of the post-synaptic potentials is in a decremental fashion, those originating from more distant synapses will be attenuated and more difficult to observe through the electrode noise than those from nearby synapses [11]. Many

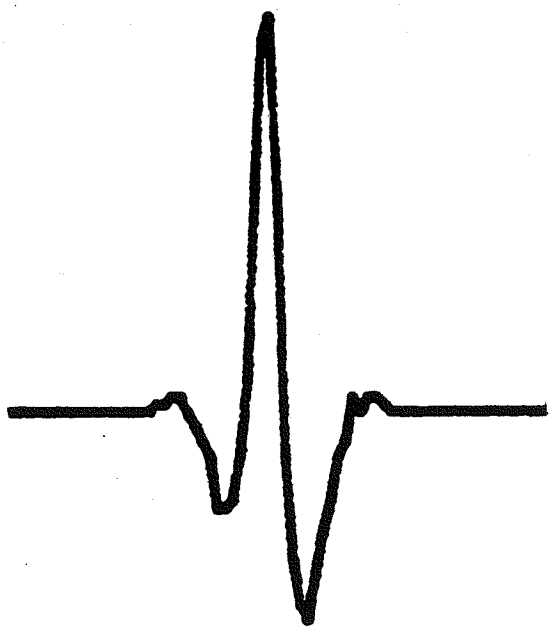


Figure 1.1. Spike waveform obtained from a giant interneuron of the cockroach.

systems use this property to distinguish different synapses by their observed amplitudes.

The extracellular potentials are small compared to the full-action potential observed with an intracellular electrode. The waveforms vary with the axon-electrode distance and with the diameter of the axon. Fortunately, the variations in the structure of the neurons and their locations with respect to the recording electrode give rise to consistent differences in the shapes of the action potentials. These differences allow one to extract multidimensional features for use in classification.

It is generally accepted that the bandwidths of neural data are in the range 1-10 kHz, with 6 kHz being representative [23]. This figure suggests a value for the high frequency cutoff of the analog amplifier system used. Further, digital sampling must be greater than 12 kHz (the Nyquist rate). On the other side of the spectrum, slow wave activity tends to impede computer separation of extracellular neuronal activity into single-unit activity. The effects of slow wave activity on such separation can be reduced by a highpass filter with low frequency cutoff between 100 and 600 Hz. Once the parameters of the data are known, linear filtering can then be introduced to enhance computer detection and separation.

1.3 Background

1.3.1 Computer Separation

Gerstein and Clark [10] were among the first to implement the

use of computers for separating multineuronal data. Since then, a number of computer classification techniques have been developed ranging in complexity and versatility. Wheeler and Heetderks compared such techniques as the use of amplitude and conduction time measures, template matching, the principal components (PC) method, optimal filtering, and maximin discrimination [28].

Amplitude classification is the simplest and fastest technique, yet is only effective for neural signals with high signal-to-noise ratios. Pulse height or window discriminators have been developed to select only those spikes with amplitude falling between preset limits [13]. While such window discriminators may be used for the detection of neural signals [30], it is not recommended for classification since spikes of equal amplitude but originating from different nerve fibers cannot be separated.

In template matching classification is done according to the best rms fit between an observed waveform and preselected standard [20]. This "standard" is the average of visually selected waveforms for each neural unit. Gerstein and Clark's implementation [10] uses this method to develop histograms which display the population as a function of dissimilarity number calculated as the rms difference between each waveform and a standard template. Usually, the histogram shows sufficent clustering of populations to permit the experimenter to classify the action potential waveforms generated from the same unit as the standard. Each of the sample points comprising a waveform can be individu-

ally weighted during computation of the dissimilarity number, thus emphasizing various portions of the waveform. O'Connell et al. [18] found that this rms measure has an effectiveness of 97-98% correct selection as compared to an experienced observer.

Roberts and Hartline [22] applied a technique involving multichannel linear filtering in which filters are constructed, each of which is optimal for distinguishing one neural signal from other signals and noise. Roberts and Hartline claimed that the number of recording electrodes must approximate the number of units present. This optimal filter method is a variation on the traditional matched filter in colored noise where the noise not only includes the instrumentation noise, but also the waveforms of units which are to be rejected. The linearity of the filters makes it possible to resolve superimposed waveforms under some circumstances.

The principal component method generates templates for each unit by projecting the waveforms onto an orthogonal set of basis functions that provide the least-mean-square error for representing the original spikes in the data set. Each detected waveform is transformed and defined as a point in a two-dimensional PC plane. As more events are transformed into points, dense clusters form which likely correspond to a single neural unit. Abeles and Goldstein [1] suggest that more than two PC vectors are generally not needed since they typically represent 93% or more of the waveform's energy. Separation is usually done by defining classification boundaries around the projected clusters (see Figure 1.2). This technique has been

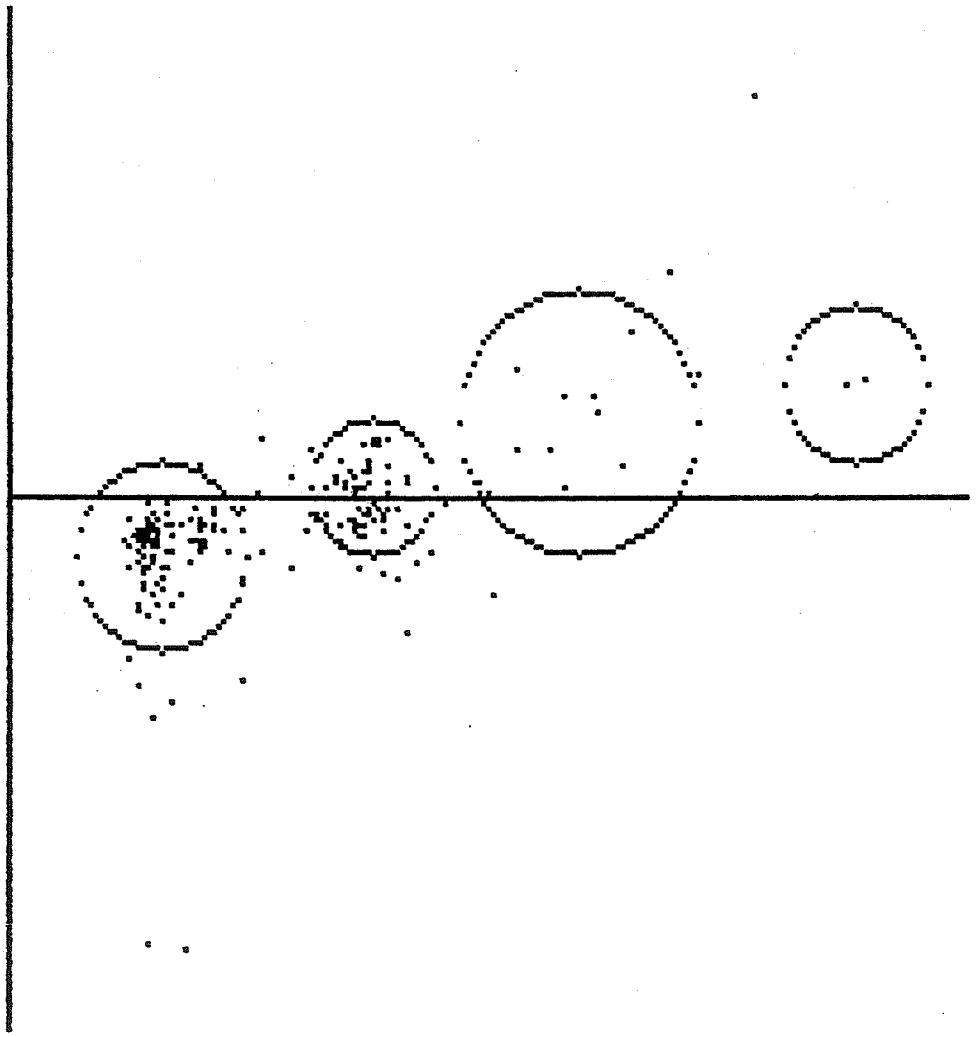


Figure 1.2. Two-dimensional principal component cluster plot showing four possible clusters (defined by the circles). The horizontal axis corresponds to the first principal component direction and the vertical axis to the second principal component direction. Each dot represents one action potential as described by the derived principal component coefficients.

used on line in real time in our laboratory [26].

1.3.2 Automatic Learning Phase Algorithms

Dinning and Sanderson [6] approached the problem of an unsupervised learning phase using a reduced feature set algorithm. Using limited knowledge of the signals, a compressed learning set containing all the still unclassified threshold-detected shapes is analyzed by a computer to select several samples of each spike event. An automatic clustering algorithm is applied to the aperiodic samples of the learning set, producing a set of clusters whose means are used as reduced templates for real-time classification in the operational phase [6].

An on-line spike separation system was developed by D'Hollander and Orban [5] in which the learning phase is also unsupervised. Data samples of each waveform in the learning set are used as coordinates in a multidimensional feature space. Cluster configurations which hopefully correspond to a single unit are then determined using a nearest neighbor technique. The output of the learning program, i.e., the mean waveforms, constitute the templates for the operational phase.

Another system uses two-dimensional projection of the action potential shape on the first (abscissa) and second (ordinate) principal components of eigenvectors [9]. Once again, clusters of points form which correspond to the repeated occurrence of action potentials of a particular shape, presumably coming from a particular single neuron. A third dimension is used to indicate

the number of points falling into each cluster. Interaction of the learning phase program allows one to define numbered elliptical boundaries that are used to separate all waveforms falling into a cluster for subsequent operation phase analysis.

A "smart trigger" for automatically sorting action potential trains was developed by McGill and McMillan [16]. Here, the learning phase and operation phase are synonymous. The system uses two fast digital signal-processing microprocessors to detect and sort up to eight different waveforms on a single input channel in real time. When a spike is detected, it is compared with all the existing templates using the squared error criterion. The smart trigger uses a threshold-crossing detector set at a multiple of the standard derivation of noise. If an existing template fails to get four matches within 2000 spike detections, it is erased to make room for new templates. The system uses a discrete Fourier transform algorithm [15] to interpolate waveforms in the background and is spread out over several sampling intervals.

1.4 Summary of System Development Efforts

A user-friendly system was developed that automatically performs the learning phase in real time. The main program was written in Fortran with assembly language subroutines for fast data transfer and quick graphic capabilities. Five separation algorithms are made available to the neurophysiologist for use under different experimental conditions. A real-time cubic spline algorithm [29] is also available for interpolation of

detected waveforms. Simulated test data were created and used to test the efficiency of five separation algorithms with varying signal-to-noise ratios. The effects of digital filtering on detection and classification are also studied. A test program written in Fortran summarizes the performance of the five algorithms tested as a function of noise and filtering. Tests on actual data from giant interneurons of the cockroach are also evaluated and studied. The results indicate which algorithm is most efficient and suggest optimal installation parameter settings for certain experimental conditions. With the knowledge gained from the performance of each algorithm, one can decide which algorithm is likely to be most efficient for one's own purpose.

CHAPTER 2

THEORY

2.1 Problem Model and Definitions

When recording the activity of groups of neurons, one can expect the recorded signal to be corrupted by noise. Electrodes, due to their ohmic component, generate Johnson noise. The microelectrode amplifier also generates noise that is added to that generated by the electrodes. Another source of electrical interference that contaminates neuronal recordings is electrostatic and/or electromagnetic pick-up from the power lines in the laboratory. Slow wave activity and the activity of neurons some distance from the electrode contribute only to the baseline hash [23]. With this is mind, we now formulate a simple additive model implemented by Wheeler and Heetderks in their extensive comparison of multiunit separation techinques [28].

If v(t) represents the recorded voltage from a particular electrode and n(t) is the noise, we observe:

$$v(t) = \sum_{i,j} w_i(t-\tau_{ij}) + n(t) \qquad (1)$$

where w_i denotes the signal contribution of the ith waveform unique to each unit and limited in duration, and τ_{ij} denotes the time of firing of the ith unit for the jth time measured from some starting instant. The determination of the times τ_{ij} when

each of the units fired is assumed known. Therefore, in the classification problem described below, it is assumed that the detection problem has been solved.

Once a spike has been detected, assuming no superposition of waveforms from two or more spikes, the recorded voltage is

$$v(t) = w_i(t-\tau_{ij}) + n(t) \tag{2}$$

Since all of the classification techniques in this thesis are digital, the recorded signal over the spike duration can be represented by writing the discrete vector equation.

$$\underline{v} = \underline{w}_i + \underline{n} \tag{3}$$

where the elements of $\underline{v}$ are the voltage samples that straddle the peak amplitude of the spike immediately following epoch τ_{ij}, $\underline{w}_i$ includes the sample values of unit i's waveform, and $\underline{n}$ represents the sampled noise. Note that the noise generated in this model is assumed to be bandlimited zero mean Gaussian. In solving the classification problem, one uses the information gathered from the waveform vector $\underline{w}_i$ and the statistics of noise $\underline{n}$ to identify which unit i was most likely responsible for the recorded voltage samples $\underline{v}$.

In the development of the classification techniques that follow, different features extracted from the vector $\underline{v}$ will be used. A multivalued transformation vector $\underline{F}$, a bivariate function of classification technique and $\underline{v}$, is applied to the variables in (3), which, if $\underline{F}$ is linear, yields

$$\underline{F}(\underline{v}) = \underline{f} = \underline{f}_i + \underline{n} \tag{4}$$

where $\underline{f}$ and $\underline{f}_i$ are feature vectors extracted from $\underline{v}$ and $\underline{w}_i$ and $\underline{\nu}$ is the transformed noise vector in the feature space. Since each unit in this thesis is assumed a priori equally likely to be active, then a Bayesian analysis recommends classification according to whichever set of features $\underline{f}_i$ best fits the observed features $\underline{f}$; i.e.,

$$\min_i \| \underline{f}_i - \underline{f} \| \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm.

The Euclidean norm is a simple conservative criterion appropriate for use provided that the features have been "whitened" to remove the correlation of the noise process [8]. This least distance measure will use classification features extracted from the Peak Amplitude, Template Matching and Weighted Template Matching Algorithms. Although the Principal Components algorithm could also be done this way, in this thesis classification is done by a lookup table established by eye from a scatter plot of features.

To measure the effectiveness of any set of features in classification, a separation matrix is constructed whose entries are the statistical distances between pairs of units as measured in the feature space. Each entry is given by

$$d_{ij} = \| \underline{f}_i - \underline{f} \| \tag{6}$$

provided the feature's variations are independent of noise and have been normalized. When this distance is greater than five

standard deviations, two units can be considered well separated implying at least a 95 percent correct classification [12].

If a pair of feature sets is derived from the mean of a class of action potentials and another action potential by any one of the linear transformations, the maximum separation allowed for the new action potential to be considered from the same unit is a multiple (usually 2.5) of the standard deviation of the baseline noise. Classification is done by measuring the distance between the existing template features $\underline{f}_i$ and the observed feature $\underline{f}$. If the distance between the existing template features $\underline{f}_i$ and the observed feature $\underline{f}$ are all greater than the allowed maximum separation standard, a new template is generated from the observed vector $\underline{v}$. Otherwise, classification is done according to whichever template feature $\underline{f}_i$ best fits the observed feature $\underline{f}$ (see equation (5)). The template for each neural unit is constructed only from the mean waveform vectors $\underline{v}$ in which their feature distance, calculated above, is less than a multiple (usually 1.0) of the standard deviation of the baseline noise. Therefore, it is possible for an observed vector $\underline{v}$ to be classified yet not be averaged into the unit template $\underline{w}_i$.

2.2 Development of Classification Features

2.2.1 Peak Amplitude Classification

The simplest, most obvious feature for use in classification relies on the peak amplitude of neural waveforms [2,3,24]. This difference in spike amplitude exists in some records due to the

neuron's size and distance from the recording electrode. As the name implies, the feature $\underline{f}$ extracted from the vector $\underline{v}$ is its peak amplitude. Other simple features such as spike width and peak-to-peak-crossing time were not extracted for classification since they appear not to improve reliability [3,27].

2.2.2 Template Matching Classification

This technique compares the mean template $\underline{w}_i$ of each unit to the observed spike vector $\underline{v}$ by computing the rms distance; i.e.,

$$RMS = [(1/N)\{\sum_{j=1}^{N}(w_{ij}-v_j)^2\}]^{1/2} = N^{-1/2} \parallel \underline{w}_i-\underline{v} \parallel \qquad (7)$$

where N is the total samples in the $\underline{v}$ vector. In the past, this classification technique was only considered for off-line separation because of the large number of squaring operations needed to compute the rms errors for all neural units. With the vast improvements in fast digital signal processors, this statistically optimal technique can be implemented in real time. For later reference, the notation RMS will refer to template matching.

2.2.3 Weighted Template Matching Classification

This technique, basically the same as Template Matching, individually weighs each difference $(w_{ij}-v_j)$ in (7);i.e.,

$$W\text{-}RMS_i = \{(1/K_i) \sum_{j}^{N} H_{ij}(w_{ij}-v_j)^2\}^{1/2} \quad (8)$$

where H_{ij} is the weight for the jth sample of unit i and K_i is the sum of the weights H_{ij} for unit i. The elements H_{ij} are the mean square sample values for the waveforms previously classified as unit i. Therefore, the weight vector emphasizes various portions of the waveform. Hand selection of the elements of a generalized weight vector for all units has been implemented but was not tested here. The notation W-RMS will refer to weighted template matching for later reference. Note that if the elements H_{ij} are chosen equal to the w_{ij} the procedure is equivalent to matched filtering.

2.2.4 Principal Components

The Principal Component (PC) technique is based upon calculating a set of orthogonal basis functions (refered to as PC vectors) $\underline{P}_j$, that provide the least-mean-square error for representing the original spikes in the data set. The original waveforms can then be approximated by using the scaled sum of two or three optimally chosen PC vectors. These scaling factors f_{ij} (PC coefficients) are very effective features for classification.

$$f_{ij} = \underline{P}_i^T \underline{w}_i$$
$$f_j = \underline{P}_i^T \underline{v} \quad (9)$$

Two PC vectors will be tested in this model since 93% or more of the waveform's energy are generally represented by two PC vectors [1]. The two optimal PC vectors are the eigenvectors of

the correlation matrix $\underline{C}$ ordered by the sizes of their associated eigenvalues [1].

$$\underline{C} = \sum_i \underline{w}_i \underline{w}_i^T \tag{10}$$

If one normalizes the waveforms $\underline{w}_i$ before calculating the PC vectors, the PC coefficients are said to be more sensitive to the smaller amplitude waveforms [28]. In the testing that follows, two cases will be referred to: 1) PCA, no normalization, and 2) PCB, with normalization of the waveforms before the PC vectors are calculated. Classification of both cases can be done by eye (see Figure 1.2) though automatic learning of the PC technique can be accomplished [9].

2.3 Interpolation Formulae

When a waveform is detected, the time of occurrence plus 31 sample points which straddle the peak are saved. Because action potentials are generated asynchronously with respect to the finite rate sampling process, the waveforms show jitter due to alignment error. Figure 2.2 clearly shows an example of the sampling problem. This is equivalent to the addition to the waveform of noise whose variance is proportional to the square of the slope [29]. The desired temporal resolution [7] may be obtained by sampling faster than the Nyquist rate even though, theoretically, oversampling adds no information [15]. A cubic spline interpolation technique is applied to the problem of aligning the detected waveform. Wheeler and Smith [29] developed

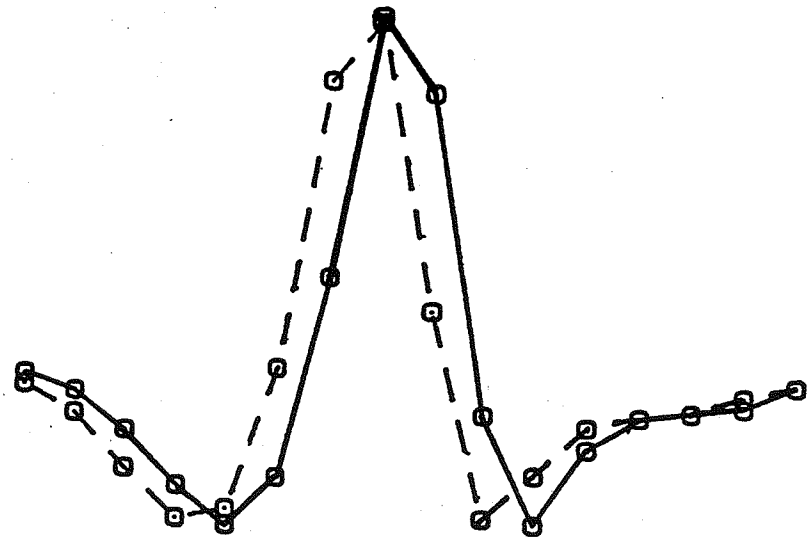


Figure 2.2. Two 16-point signatures from the same unit superimposed show the alignment problem. (From Wheeler and Smith [29])

this technique as a more efficient, time domain alternative to the discrete Fourier transform approach [15]. Alignment is done by finding the peak of the interpolated waveform using Newton's method. The waveform is then reconstructed with the 31 sample points now straddling the interpolated peak.

2.3.1 Cubic Spline Calculations

Let $q_i(t)$ be the cubic spline function which interpolates the value of the continuous waveform x(t) between sample points x_i and x_{i-1} at times t_i and t_{i-1}, respectively. The equations without proof are [4]:

$$q(t) = A + \Delta t\ B + (\Delta t)^2\ C + (\Delta t)^3\ D \qquad (11)$$

where $\Delta t = t_i - t$,

$A = X_{i-1}$,

and B,C, and D can be adequately approximated [29] by

$B = [.216,-.804,0,.804,-.216]\ X$

$C = [-.413,1.392,-2.196,1.392,-.372,.216]\ X$

$D = [.216,-.588,1.196,-1.196,.588,.216]\ X$

where $X = [x_{i-3},x_{i-2},x_{i-1},x_i,x_{i+1},x_{i+2}]$

In this form, one can quickly solve for $q_i(t)$ with a minimum of multiplications.

2.3.2 Alignment Criterion

Alignment of the sampled action potential is done by first locating the interpolated peak of the waveform. Figure 2.3 shows that the location of this peak must lie in one of the sample

intervals straddling the discrete sample time peak. The interpolated peak is found using Newton's method, i.e.,

$$t_{n+1} = t_n - f(t_n)/f'(t_n), \quad n=1,2,3,... \tag{12}$$

where $f(t)=dq(t)/dt$ $f'(t_n)=df(t_n)/dt$

After four iterations of equation (12), the root t is sufficiently accurate as the time of the interpolated peak. Two iterations can be used assuming a good initial value t_1 is used (t_1 = 1/4 for case I and t_1 = 3/4 for case II as shown in Figures 2.3a and Fig. 2.3b, respectively). Reconstruction of the sampled action potential waveform is done using equation (11) such that the new interpolated sample points straddle the interpolated peak.

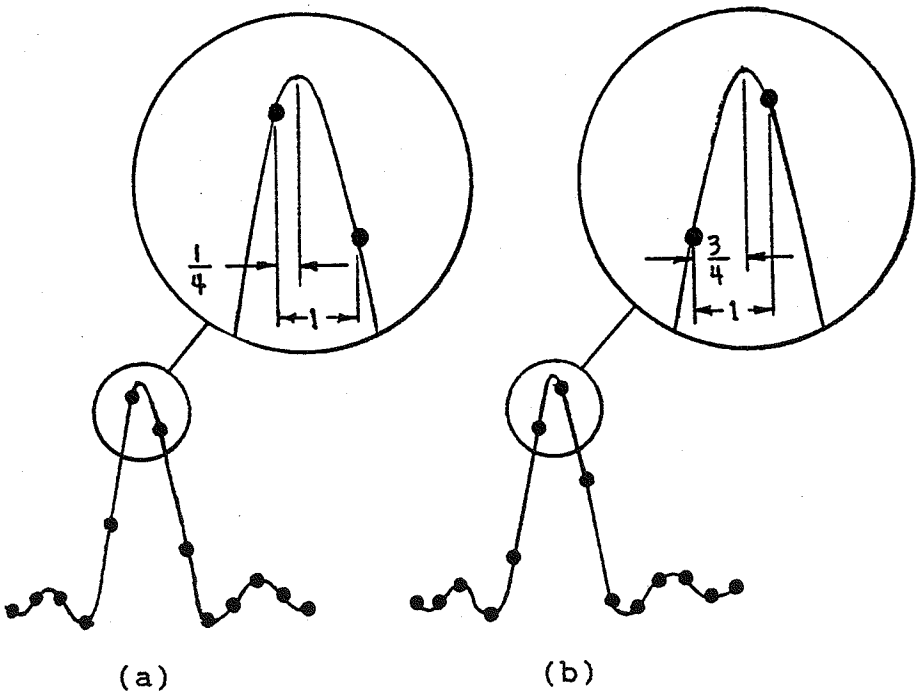


Figure 2.3. Two possible locations of the waveform's continuous peak. (a) Peak lies approximately 1/4 unit to right of sample point, and (b) Peak approximately 3/4 unit to right of sample point.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 Computer System and Hardware

The learning phase described in this thesis was designed to simplify the neurophysiologist's tasks of downloading operation phase parameters to a peripheral processor which performs real-time spike recognition. The system diagram (Figure 3.1) shows that the raw data from each electrode is amplified, filtered, and digitized before being sent to the peripheral processor [26]. Assuming the parameters from the learning phase have been downloaded, the peripheral processor performs action potential detection and perhaps classification. The compressed data are then sent to the host computer which is an IBM PC compatible AT&T PC6300. It is this system that is used to digitize a small sample of the raw data to be used for the learning phase. In this case, the raw data sent to the host computer are not compressed, since the parameters of the noise must be found. The AT&T PC6300 is then used to perform all of the learning phase operations.

3.2 System Software and Functions

A useful multispike separation system must be user friendly and efficient. The operational phase of separating spike waveforms and processing data is secondary to its procurement. When conducting real-time experiments, the monitoring of

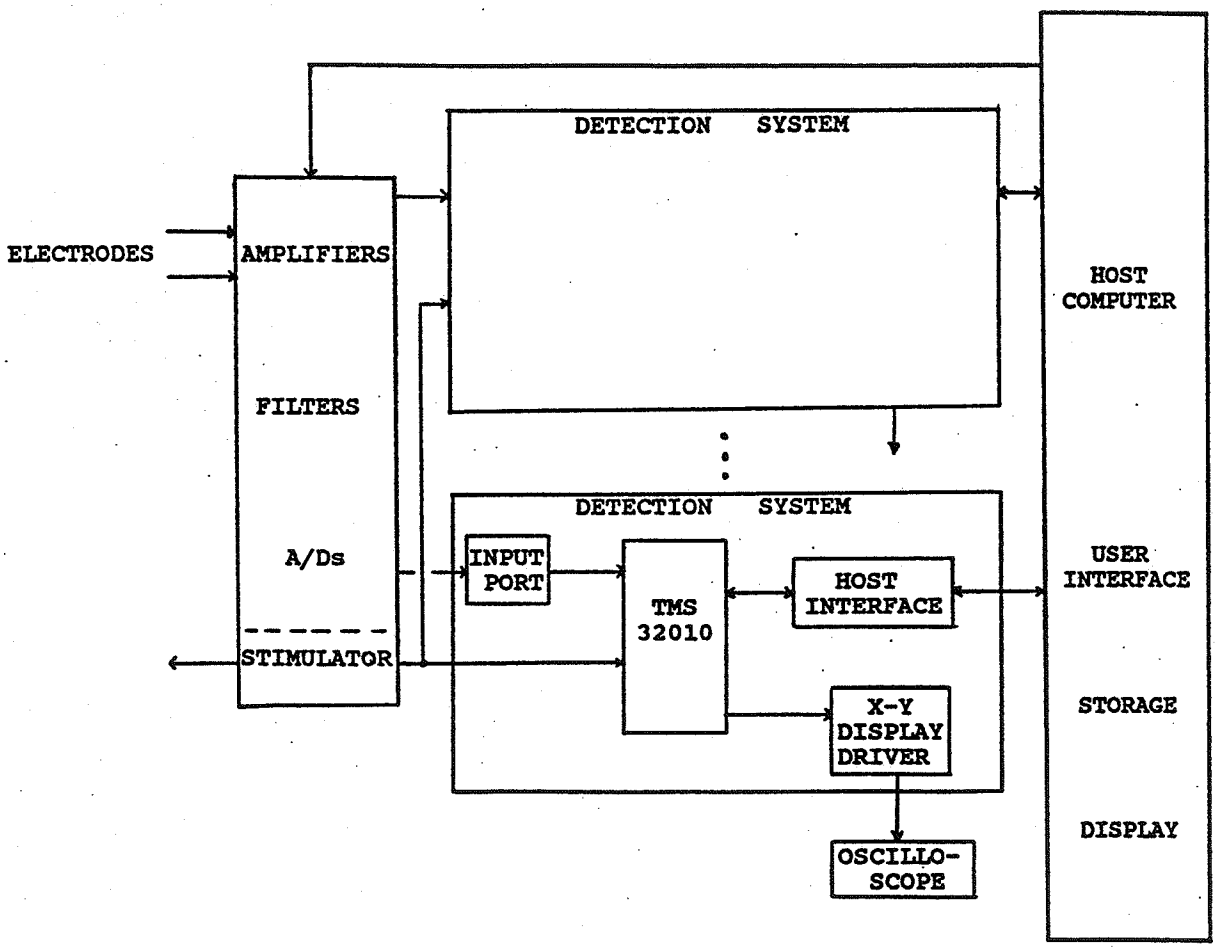


Figure 3.1. System diagram showing flow of processing from electrode to host computer. (From Smith and Wheeler [26])

biological preparation, the presentation of stimuli, the keeping of records, and the recording of raw data generally take precedence over the spike classification process. Examination of an initial learning set through a fast automated learning phase program is necessary to establish classification boundaries. Once this is done, the classification of data via an operational phase can be executed in real time for experiments using interactive stimulus presentation. With this on-line capability, the experimenter obtains immediate feedback about multiple neuron responses permitting expedient analysis of physiological parameters.

A learning phase algorithm was developed that automatically detects and separates up to 16 different spike waveforms. The program displays statistical data, sets the detection threshold automatically (if desired), allows operator intervention (also, if desired), gives temporary classification boundaries, and the ability to change separation algorithms and parameters used in detection and classification via an installation menu. The following outlines such features and why they are needed.

3.2.1 Algorithms Implemented in Learning Phase

Different types of neural data can be classified using different separation algorithms. For instance, if the neural data have high signal-to-noise ratios (i.e., giant interneuron signals from the cockroach ventral nerve cord), then they can be classified using a fast and simple algorithm such as Peak Amplitude. On the other hand, if the neural data have low signal-to-

noise ratios, then they should be classified using a more complex yet more time consuming technique such as Template Matching or Principal Components. The program provides six different algorithms: 1) Peak Amplitude, 2) RMS Template Matching, 3) Hand Weighted RMS Template Matching, 4) Auto Weighted Template Matching, 5) Principal Components and, 6) Principal Components Using Normalized Templates.

There are 16 template buffers on hard disk which are initially empty. When a spike is detected, it is compared with all the existing templates using whichever separation algorithm selected. Whenever more than 16 templates are encountered, the program erases an existing template with the least amount of matches to make room for the new template. A flowchart outlining the Learning Phase algorithm is shown in Figure 3.2. The features extracted are a function of the separation process chosen.

3.2.2 System Control Menus

Of the Learning Phase Algorithms mentioned above, only Peak Amplitude and RMS Template Matching are present on the Main Menu (see Figure 3.3). Since Weighted and Auto RMS Template Matching is a variation of RMS Template Matching, execution is implemented through the Install Menu under selection number 6 (see Figure 3.4). Both Principal Component methods are also not on the Main Menu since templates must be selected (via Peak Amplitude, etc.) prior to their executions.

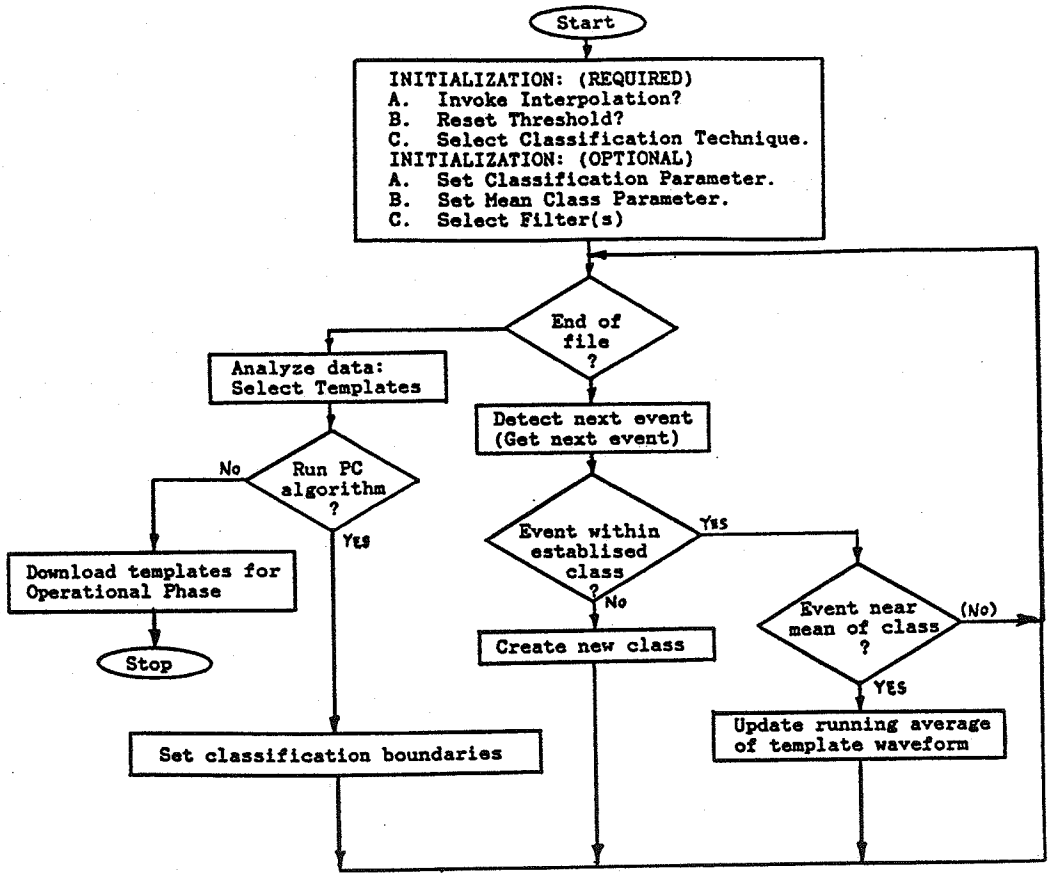


Figure 3.2. Auto learning phase algorithm flowchart.

```
                < < MAIN MENU >>
 1. SET Threshold, PEAK Classify, MANUAL rms display.
 2. SET Threshold, AUTO PEAK Classify with no intervention.
 3. AUTO Threshold, PEAK Classify, MANUAL rms display.
 4. AUTO Threshold, AUTO PEAK Classify with no intervention.
 5. SET Threshold, RMS Classify, MANUAL rms display.
 6. SET Threshold, AUTO RMS Classify with no interven tion.
 7. AUTO Threshold, RMS Classify, MANUAL rms display.
 8. AUTO Threshold, AUTO RMS Classify with no intervention.
 9. Recover RMS file.
10. Recover PC file.
 0. INSTALL MENU
       Input an integer (0-8):
```

Figure 3.3. Main Menu of the learning phase algorithm.

```
                << INSTALL MENU>>
 1. SET AUTO THRESHOLD (NUMBER OF STANDARD DERIVATIONS).
 2. SET AUTO PEAK (NUMBER OF STANDARD DERIVATIONS).
 3. SET RMS DIFFERENCE (NUMBER OF STANDARD DERIVATIONS).
 4. SET AUTO AVERAGE (NUMBER OF STANDARD DERIVATIONS).
 5. SET NUMBER OF BLOCKS OF 3000 FOR AUTO THRESHOLD.
 6. SET WEIGHT FACTORS FOR RMS CALCULATIONS.
 7. FILTER TYPE (FOR DETECTION AND/OR CLASSIFICATION).
 8. NEGATE DATA.
 9. DATA SIZE.
 0. BACK TO MAIN MENU.
       Input an integer (0-5):
```

Figure 3.4. Installation Menu for adjusting learning phase parameters.

Selections 1 and 5 require the most user intervention. Here, the user must manually set the threshold via a threshold-probability display and must decide whether or not waveforms which do not match any established class should be manually classified or picked as new templates using an rms waveform-template display. The time consuming manual intervention of selections 1 and 5 generally restrict their use for off-line experiments only. On the other hand, selections 4 and 8 require no user intervention. Here, all detection and classification are done automatically. Other selections 2,3 and 6,7 vary in the amount of user intervention. Selections 9 and 10 allow one to recover from the previous RMS/PEAK or PC experiment without having to rerun the program. The main menu allows the user to decide the amount of manual intervention, if any, needed to execute the learning phase efficiently.

The Install Menu shown in Figure 3.4 allows one to vary parameters used in the execution of detection and classification and is called from the Main Menu using selection 0. Selections 1-4 are used to input a multiple used in the sensitivity of detection and classification. Selection 5 sets the number of 3000 sample point blocks used in the calculation of the standard derivation (used in selections 1-4). As mentioned above, selection 6 is used to alter the rms template matching routine where one can choose auto or manual weighing of the waveform signature and how many sample points to use in the rms calculation (i.e., vector window size). Selection 7 allows one to pick between two simple difference filters [14] for detection

and one 3rd order butterworth filter [19] used for classification. Also, both detection and classification filters can be used together or alone. Selection 8 is used for negating (or flipping) the data since different recording methods and/or filtering can affect the positive detection threshold. The length of data to be studied, also a function of recording methods, can be set using selection 9. When entering any of the selections above, the current (or default) value is displayed. Control is returned back to the main menu via selection 0 in which one can now execute one of the algorithms as a function of the new or unchanged parameters.

3.2.3 User/System Interaction Displays

Interactive user-friendly video displays were developed to assist in setting the threshold level, picking a filter type, analyzing the classified waveforms, choosing templates for the principal component techniques, and drawing classification boundaries (used only for principal component analysis).

When manually setting the threshold level, the first video display presented is the Threshold-Probability display (see Figure 3.5). This display shows the sample population (abscissa) as a function of voltage level (ordinate) and is in essence a probability curve. Note that the population as a function of voltage levels less than one standard deviation and greater than one negative standard deviation is not plotted. Because the plot is similar to a Gaussian probability curve, the data plus and minus one standard derivation would be lost when the plot is com-

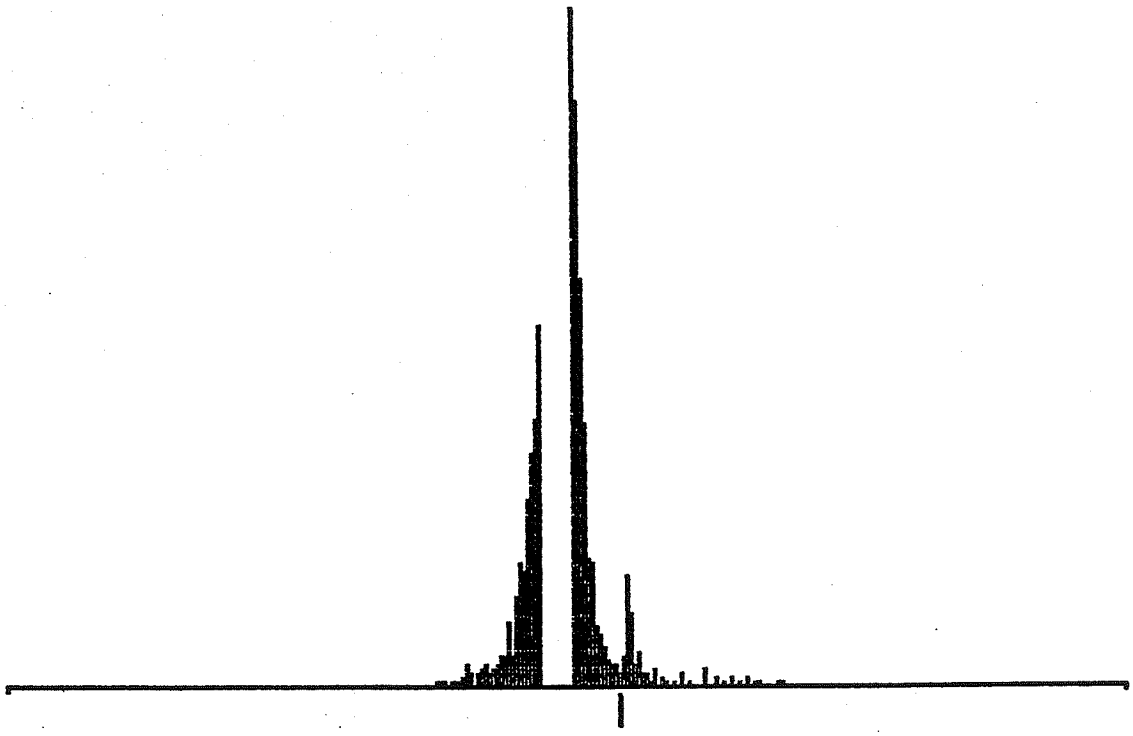


Figure 3.5. Threshold-Probability Display gives standard deviation of data (STD), number of sample points used for display (POPL), and mean of data (XBAR).

pressed for the video display. Since most cases require an absolute threshold level greater than one standard deviation, it was decided not to plot the population in that interval. Hence, important information is preserved after the compression of the plot to the video display. The left and right arrow keys of the AT&T PC6300 keyboard are used to move the display arrow which indicates the threshold level. Possible threshold level settings are shown in Figure 3.6 for four different cases. Note that in each case the threshold is set where the plot deviates from that expected of Gaussian noise. The population points greater than the displayed arrow setting suggest the presence of action potentials in the data. The standard deviation and the displayed arrow position as a multiple of the standard derivation are displayed to assist the user. In most cases, the level of threshold is set at approximately three standard derivations.

Once the threshold is set, a filter display is presented showing the raw data and filtered data in reference to the threshold level (see Figure 3.7). If desired, one can implement or change the detection filter by entering "c" on the keyboard. By entering "x" on the keyboard, one can either reset the threshold, return to the main menu or continue to the classification stage.

If a manual classification algorithm is being used (selections 1,3,5 and 7 of the Main Menu), then the next video display is the RMS Classification Display (see Figure 3.8). Whenever a waveform is detected that is not automatically classified, it is compared to the existing templates using the RMS Classifi-

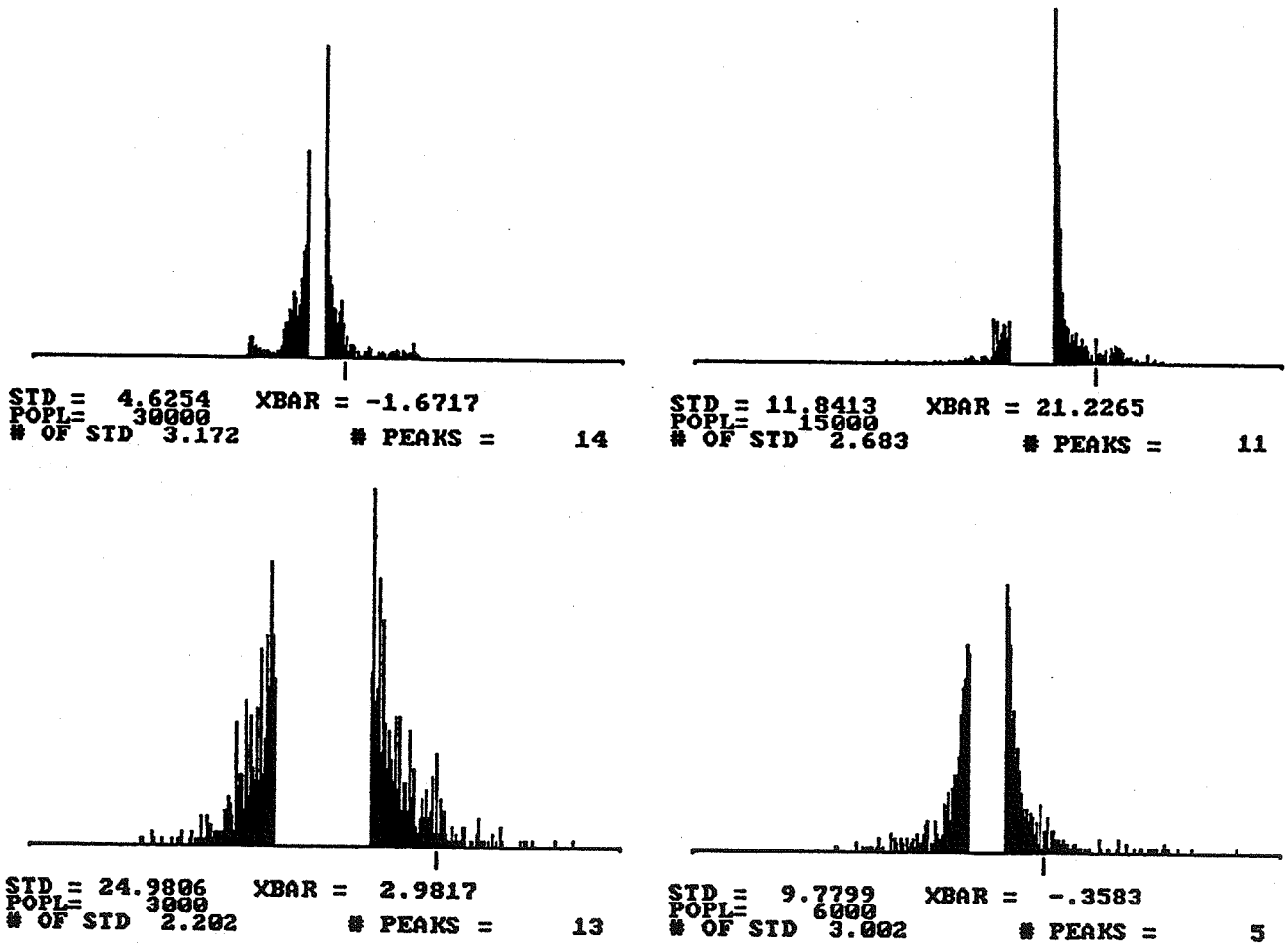


Figure 3.6. Four possible threshold settings of the Threshold-Probability Display using four different data sets.

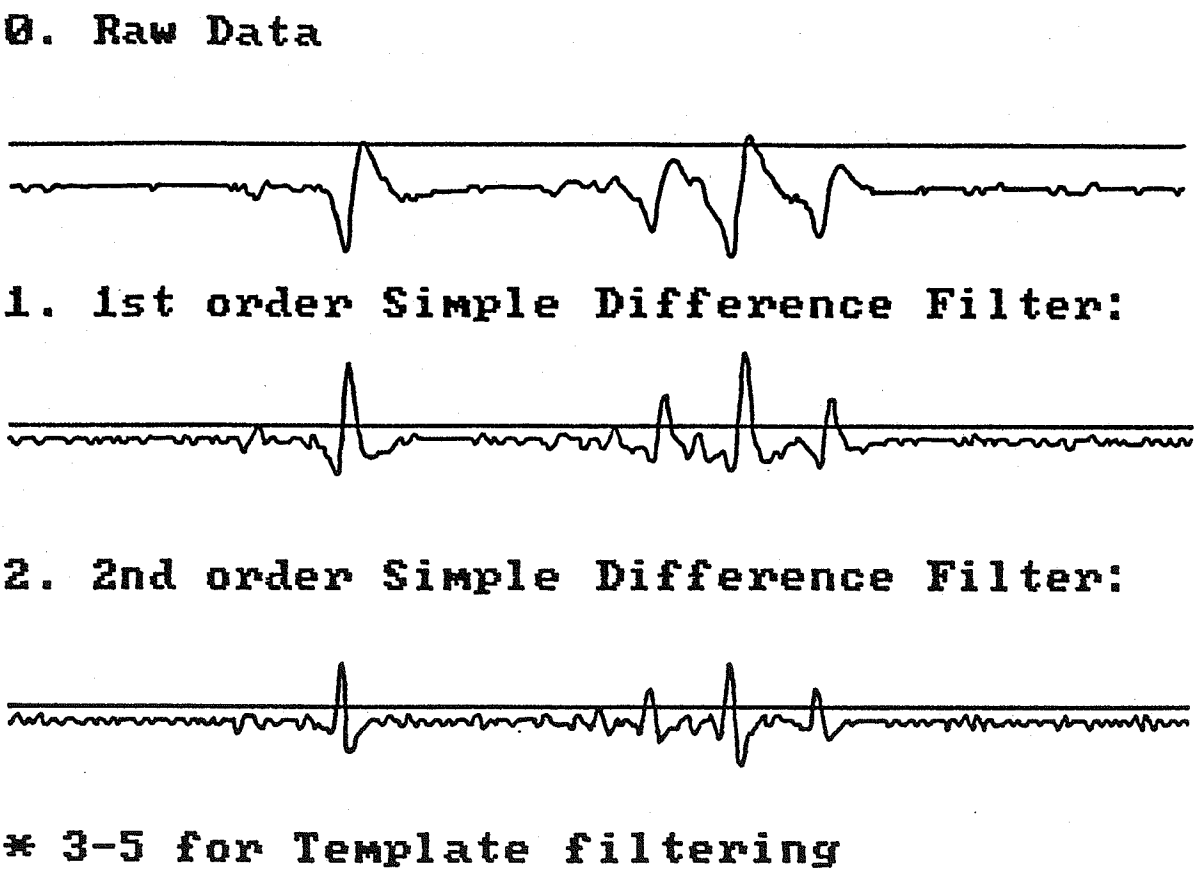


Figure 3.7. Filter Display showing the raw data (top), first order filtered data (middle), and second order filtered data (bottom). In each case the set threshold is shown with respect to the data.

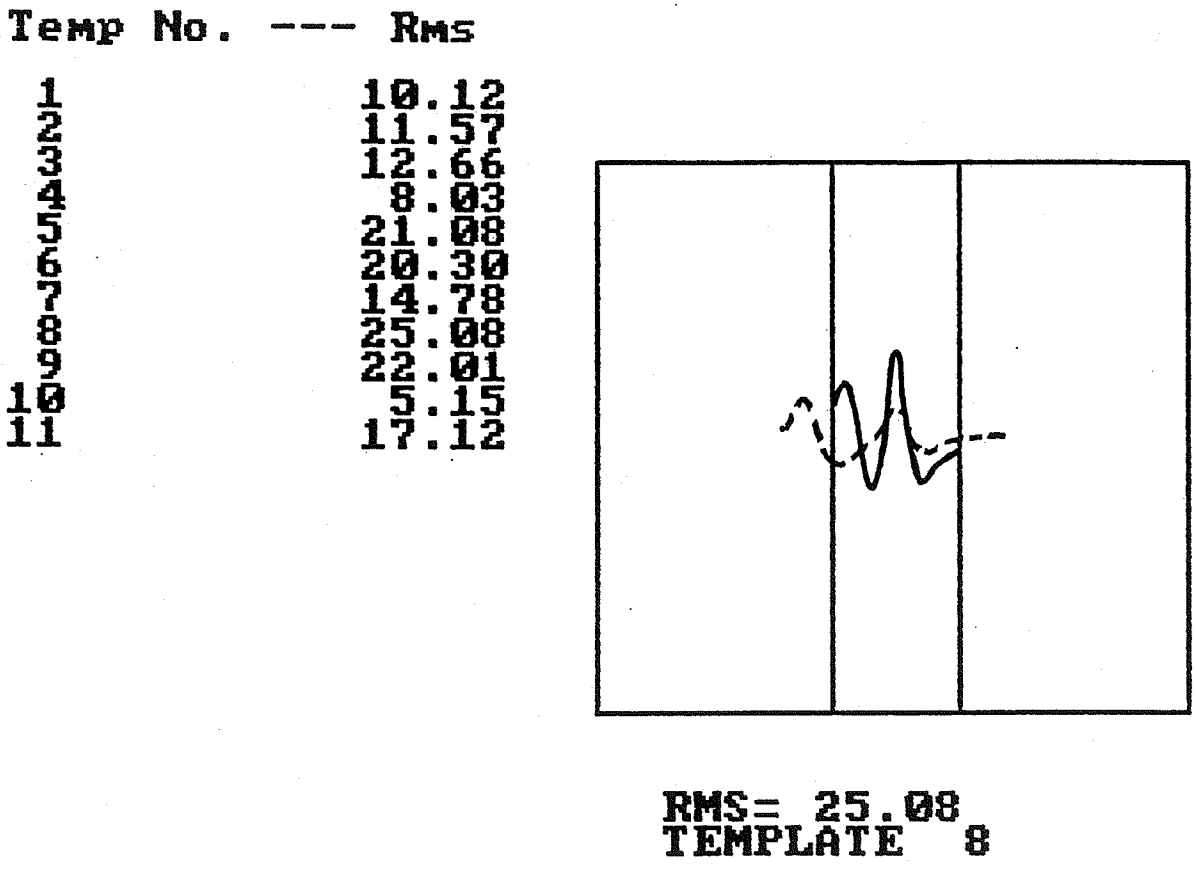


Figure 3.8. RMS Classification Display shows the observed waveform superimposed on template number 8. To the left of the waves, the rms difference of each existing template is given.

cation Display. Using the left and right arrow keys, one can slide the observed waveform over the existing templates. Different existing templates are displayed by using the up and down arrow keys allowing fast visual comparison with all templates. The rms difference between the existing waves and the observed waveform is displayed to the left of the waveforms to aid the user in deciding whether the observed waveform has a match or not. The size of the rms calculation window can be increased or decreased be entering "i" or "d", respectively. If the user decides there is no match, the observed waveform is used as a new template for future comparisons.

After execution of a separation algorithm or if selections 9 or 10 of the main menu are entered, the user is presented with the Temporary Classification Display (see Figure 3.9). Here, the user can easily analyze the performance of the algorithm by scanning through the classified waveforms of each template. This allows one to evaluate how effective the algorithm performed its separation as a function of the pre set parameters and whether one could change these parameters or try a different algorithm. Once satisfied with the separation process, the user then picks the templates best suited for the operational phase. The number of matched waveforms is displayed under each template for quick selection of permanent operation phase templates. These templates can also be used to run the principal component techniques of the learning phase.

If one decides to run one of the principal component algorithms (PCA or PCB), the templates selected above are then

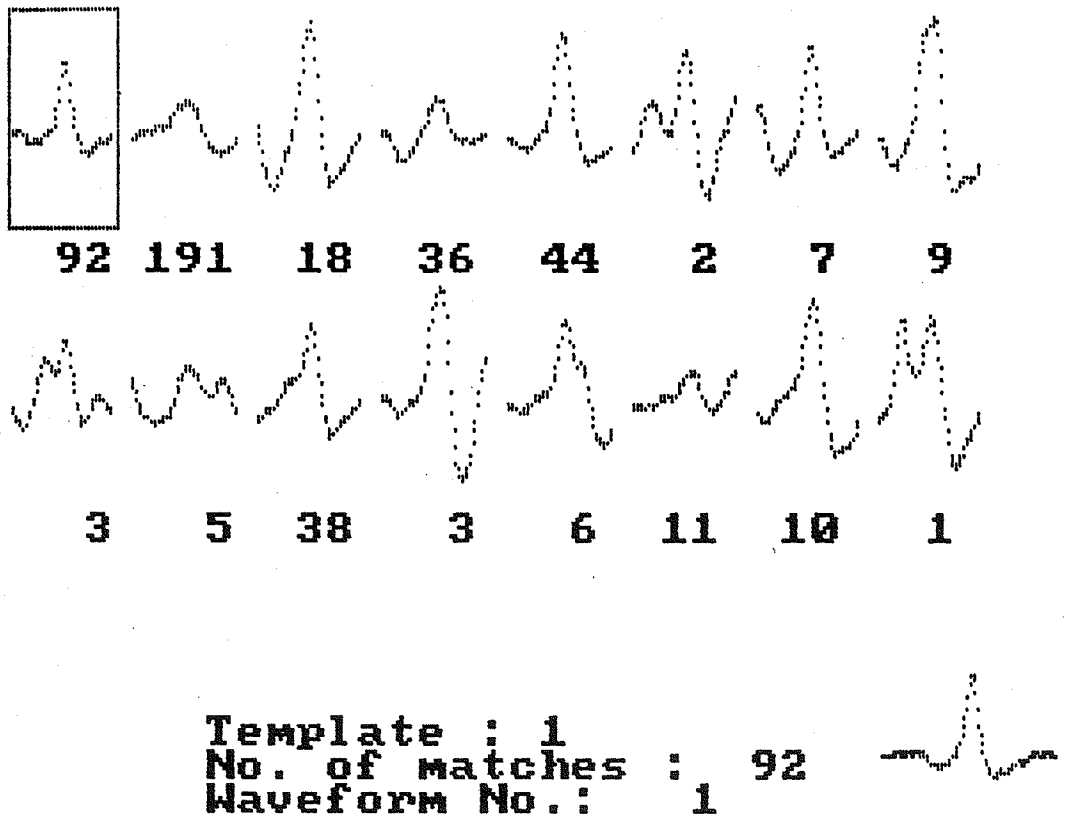


Figure 3.9. Temporary Classification Display allows user to easily analyze the performance of the separation algorithm used.

used in calculating the two optimal PC vectors (see Section 2.2). The next video display is the PC-Boundary Display shown in Figure 3.10. The user can draw lines as shown in Figure 3.10 or plot circles (Figure 1.2) around clusters which hopefuly correspond to a particular unit. A circle is plotted by first selecting its center position by moving a displayed cross using the arrow keys. Once the cross is moved to where the center is desired, it is set by entering the "s" key. The cross disappears and a circle is displayed with its center exactly where the cross was. The radius of the circle can then be increased or decreased by entering "i" and "d", respectively. Finally, when the desired radius is set, the circle classification boundary is saved by once again entering the "s" key. A contour line is drawn by by moving the displayed cross using the arrows keys to where one wants the line to begin and is marked by entering the "s" key. Once again the cross disappears and a contour, beginning at where the cross was, is drawn using the four arrow keys. Once the contour boundary line is finished, it is saved by entering the "s" key. After all classification boundaries are plotted using circles or lines, the "x" key is entered and classification is then executed and control is returned back to the Temporary Classification Display (see above).

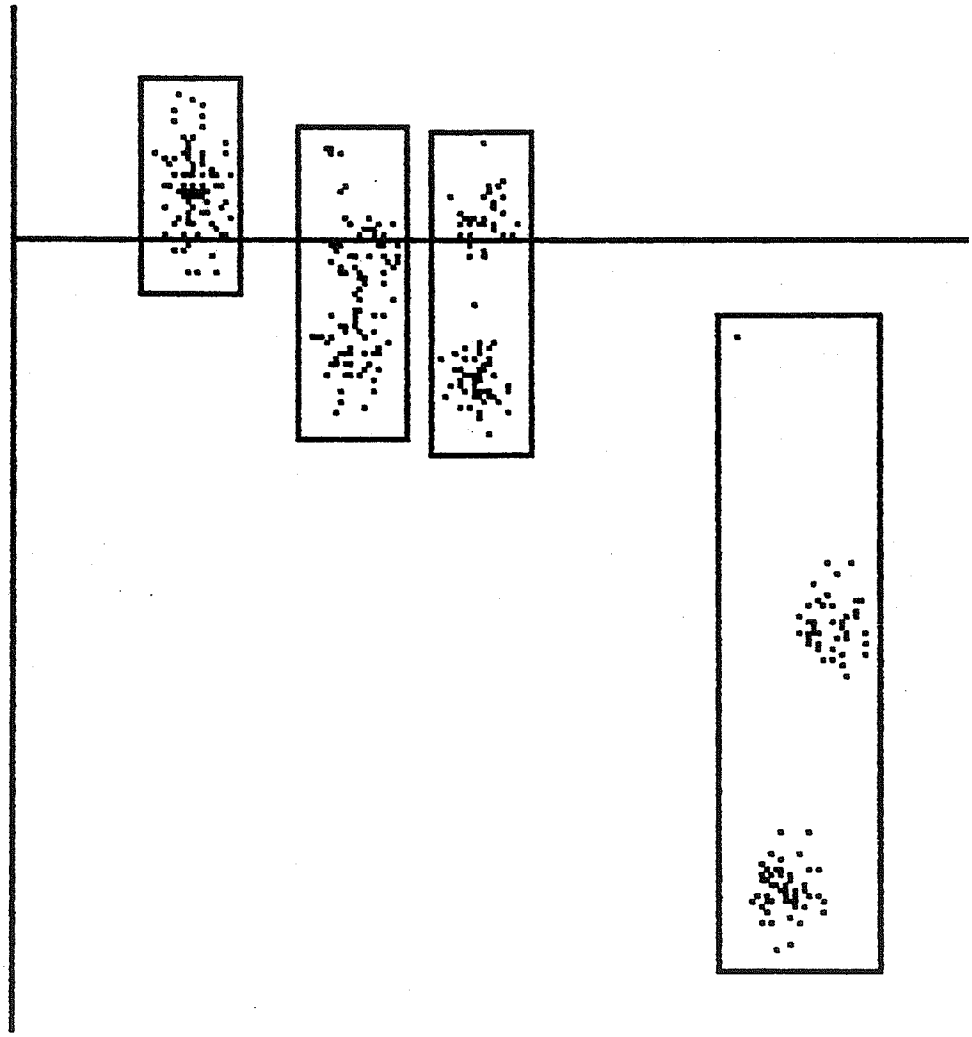


Figure 3.10. PC-Boundary Display in which lines are plotted around clusters presumed to correspond to four particular units.

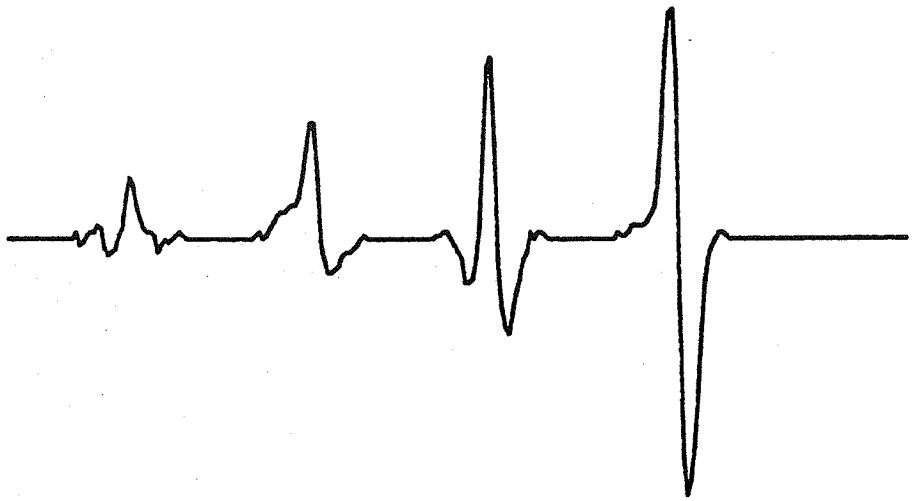
CHAPTER 4

EXPERIMENTAL METHODS

4.1 Introduction

The experiments described in this chapter are designed to test the performance of the separation algorithms discussed in Chapter 3. The experiments are such that questions regarding system performance under different signal-to-noise ratios can be answered. The effects of filtering and interpolation were examined along with the effects of window vector size on separation performance. Since filtering, interpolation and vector size determine the execution time of each algorithm, testing how they affect performance can be useful in selecting such parameters and whether or not they should be used.

Two sets of action potentials were used as data for these experiments. The first set, shown in Figure 4.1a, uses four distinguishable giant interneurons of the cockroach with the maximum zero to peak amplitude being approximately 500 μV. Each of the four waveforms was inserted 100 times in a noiseless medium to be used as a standard data set. Other data sets (test sets) were created as above except with the addition of varying amounts of bandlimited Gaussian noise. The sets of test data were then detected and classified using the five separation techniques described above and compared to the standard data set using a test program written in Fortran. The noise used for these test



(a)

(b)

Figure 4.1. Two sets of four action potentials from the giant interneurons of the cockroach. The top trace (a) shows four selected waveforms used 100 times each. The bottom trace (b) shows the four most detected waveforms of an unaltered sample of actual data.

files was created using a Wavetek Model 132 VCG/Noise Generator and is bandlimited to 300 Hz [25]. The noise was scaled such that the standard deviation of noise added to each test file was different. The SNR defined in this thesis is the ratio of the minimum waveform's maximum positive sample (peak) to the standard derivation of the noise.

The second data set was unaltered and obtained from the giant interneurons of another cockroach for the purpose of testing actual data. Figure 4.1b shows the four most prominent neural units present. In this case, the standard data set was classified by eye and compared to other data sets (test sets) obtained from each separation algorithm.

The cockroach recordings were obtained from S. Smith of our laboratory. Wire hook electrodes were placed around one connective of the neutral nerve cord of the cockroach. The animal was simulated with puffs of wind. The electrode was interfaced to a differential amplifier with 300 Hz-10 kHz bandwidth and a gain of 1000 to record the giant interneurons. A 15 kHz sampling rate was used to digitize the data for storage on disk [25].

4.2 Test Program

A test program was written in Fortran that automatically compares a test set to a standard set and outputs the percent of correct detections and classifications. The program outputs a Classification Matrix that displays how the separation algorithm sorted its test waveforms among its templates as compared to the list of templates of the standard data. The test program also

gives such information as the number of times each template in the standard was falsely classified and detected and the number of times each template was missed (not detected) and hence never classified. Figure 4.2 shows the flowchart logic implemented by the test program.

4.3 Performance of Separation Algorithms as Functions of Noise

The first experiment performed was designed to test the performance of detection and the basic classification algorithms on the first data set (Figure 4.1a) as a function of noise without the use of filtering or interpolation. The window vector size was held constant at 17 sample points (except for peak classification). Test runs were done on data of 60,000 sample points in which 400 thirty-one point action potentials were present (100 of each waveform). More sample points were not considered since Learning Phase data are generally a small sample of the complete data set. The percent noise (i.e., 1/SNR) present on the files tested ranged from 0%-100% at 10% intervals. The Learning phase parameters which produced the best results for each separation algorithm were found via trial and error and recorded along with the level of the detection threshold. The classification algorithm which performed the best at each noise level was used to select templates for the PC algorithms. The test results allow one to make a confident decision on which separation algorithm is most efficient as a function of noise and frees the user from the time-consuming "fine tuning" of the learning phase parameters.

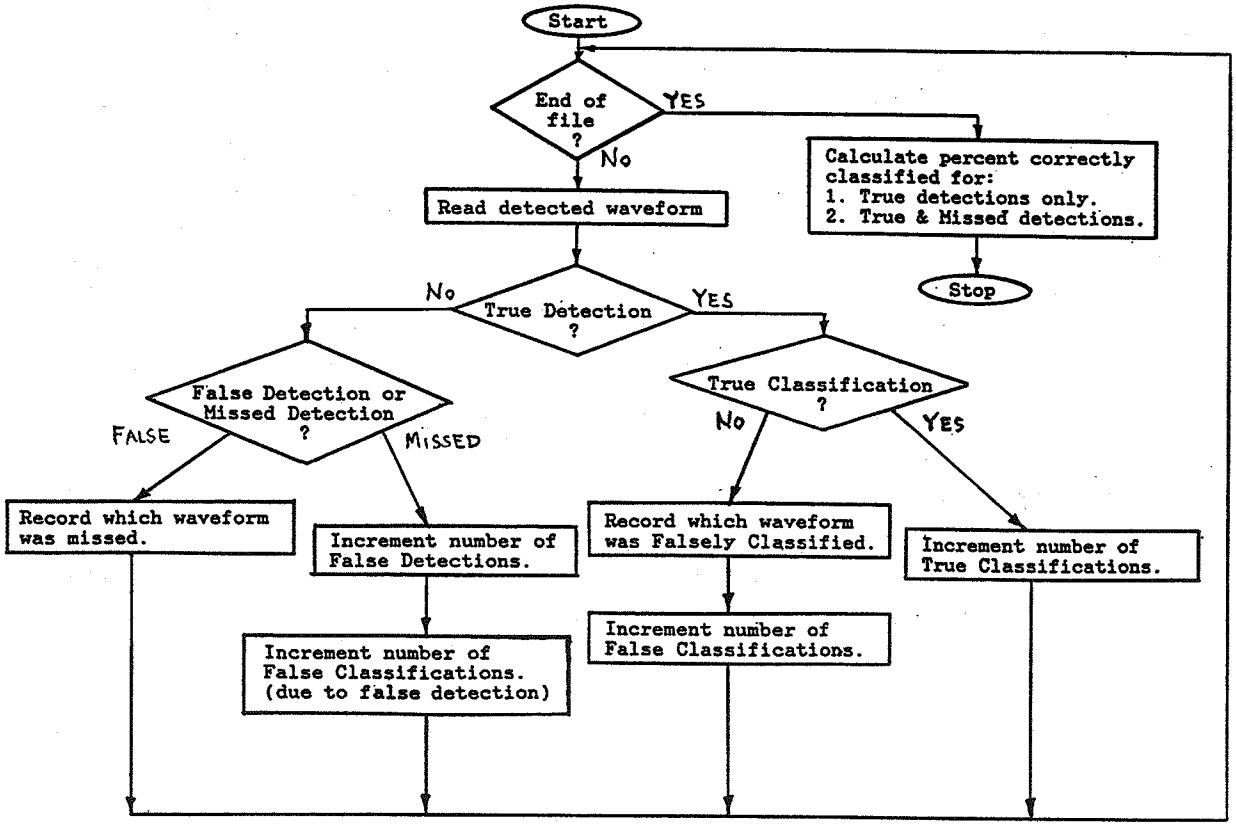


Figure 4.2. Flowchart of test program.

4.4 Investigation of the Cubic Spline Interpolation

Experiments investigating the effects of cubic spline interpolation (see Section 2.3) were done on the second data set for each of the five separation algorithms discussed above. Interpolation was not tested on the first data set since all of the waveforms inserted in the data set were all sampled the same. The results of such experiments can show which of the algorithms are most affected by interpolation.

4.5 Effects of Window Vector Size

The second data set was used to test the effects of changing the window vector size on the performance of the separation algorithms above. The Peak Classification algorithm was not tested here since it only uses one sample, the peak, for classification. The vector sizes tested were 17,11,and 5 samples in length. Two separate tests were conducted with and without the use of interpolation. A smaller window vector size means less computational time yet the information lost may inhibit performance. The information gained from such experiments and how inperpolation affects them can help one determine a reasonable window vector size best suited for the waveforms present in the data set.

4.6 Effects of Interpolation and Vector Size on Filtering

The final experiment investigated the effects of interpolation and window size on the performance of the RMS separation algorithm when using a first-order difference filter.

Window vector sizes tested were 17,11, and 5 samples in length. Some systems now use first-order difference filters to accentuate the rapidly rising edges of the waveform spikes [14]-[16]. Results of how interpolation and/or window vector size affect performance when using the filter can help one make probable decisions on window vector size and whether or not interpolation should be implemented.

CHAPTER 5

RESULTS

5.1 Separation Power

Separation matrices for the first data set were calculated for the RMS,PCA, and PCB feature sets described in Chapter 2. The three separation matrices are given in Figure 5.1. For each matrix the elements were scaled such that the two units showing the greatest separation equaled 100. The PCA method (principal component without normalization) yields an average separation of 67.46 units, with a minimum separation of 17.23. No other feature set showed better separation though units 1 and 2 have the smallest separation of all the features. RMS template matching yields the next highest separation average at 58.90 units with a minimum separation of 26.64. The PCB method (principal component with normalization) yields separation averaging 54.81 units, with a minimum separation of 17.86. Here, the units of smaller amplitude (units 1,2 and 2,3) yield slightly better separation than PCA without normalization at the expense of average separation. Usually, the minimum separation occurs between pairs of small amplitude units [28].

RMS Template Matching

	Unit 1	2	3
2	26.64		
3	46.10	37.73	
4	100.00	81.44	61.49

PCA (no normalization)

	Unit 1	2	3
2	17.23		
3	32.00	21.54	
4	100.00	97.54	77.54

PCB (with normalization)

	Unit 1	2	3
2	17.86		
3	40.77	25.00	
4	100.00	85.12	60.12

Figure 5.1. Separation Matrices were computed using the different classification features. The (i,j)th entry of each matrix is the scaled statistical difference between the ith and jth units. The elements of each matrix were scaled such that the greatest difference between two units equals 100.

5.2 Performance Results of Separation Algorithms

The first experiment described in Chapter 4 tests the separation performance of the five classification algorithms as a function of noise using the first data set (Figure 4.1a). Figures 5.2a and 5.2b show the percent of correct classifications for separation performance and overall performance, respectively. Figure 5.2a shows the percent correctly classified for waveforms which were true detections only and gives a better feel of the efficiency of the separation algorithms without considering detection performance. Figure 5.2b takes into consideration the amount of missed detections in the test file and shows the overall performance of the Learning Phase algorithms.

The results in Figure 5.2a generally confirm the predictions of the separation matrices. Note that the RMS and W-RMS methods performed generally the same as did the principal component methods PCA and PCB. The peak amplitude method performed well only for high SNRs (inverse percent noise). All the separation methods performed perfectly up to 10 percent noise. The RMS and W-RMS methods performed nearly perfect up to 30 percent noise by separating over 97 percent of the waveforms correctly. The PC methods remained perfect up to 20 percent noise and performed near perfect up to 40 percent noise by classifying over 98 percent of the waveforms correctly.

Figures 5.3 and 5.4 show cluster plots for the two principal component methods, PCA and PCB respectively, with 0, 40, 70, and 100 percent noise. At 40 percent noise, clusters corresponding to the four waveforms are still distinguishable for both methods.

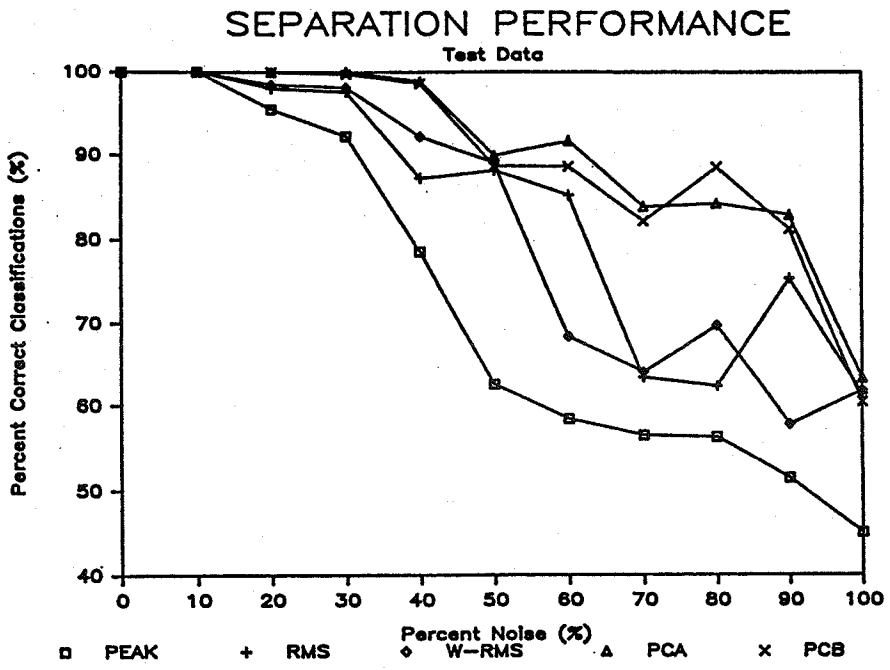


Figure 5.2a. Separation performance of the five algorithms as a function of percent noise.

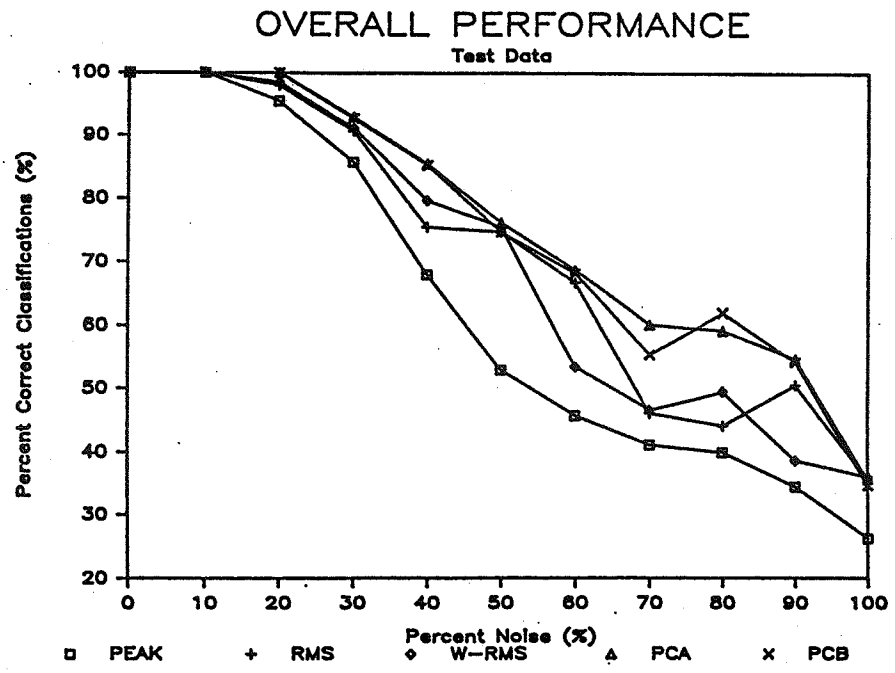


Figure 5.2b. Overall performance of the five algorithms as a function of percent noise.

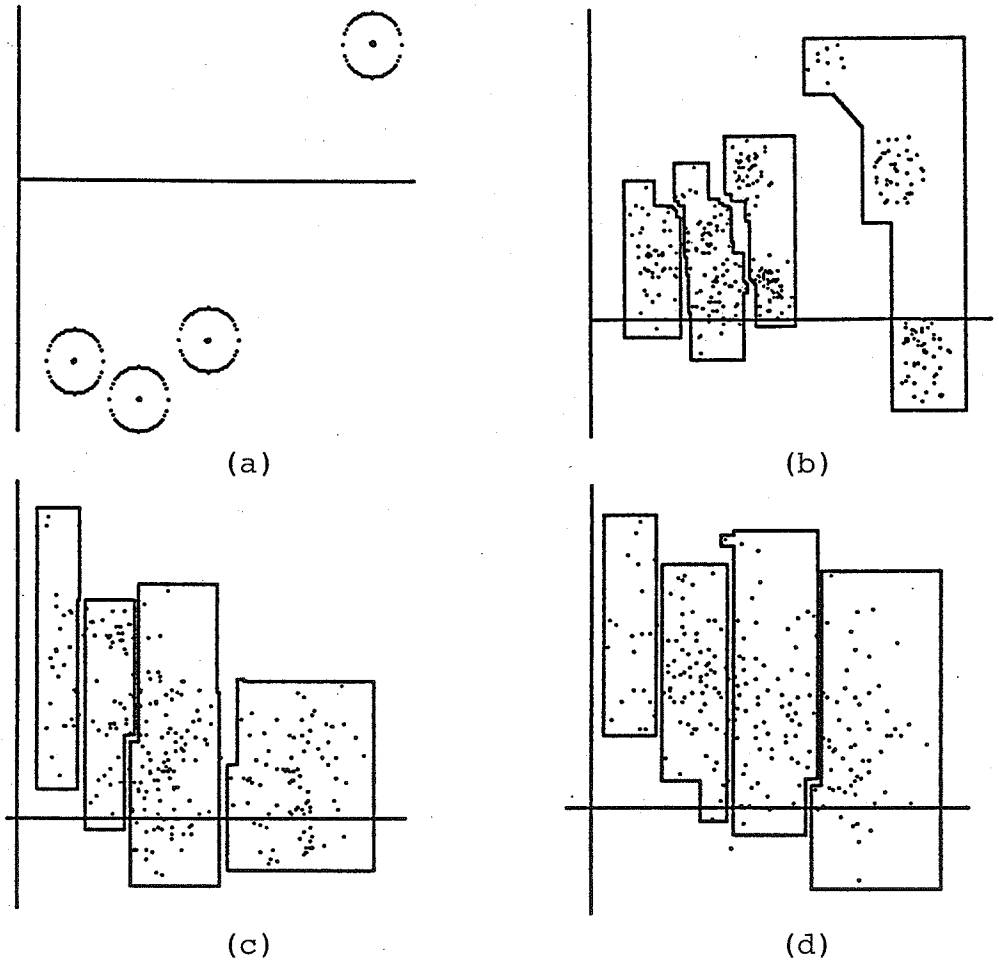


Figure 5.3. Cluster diagrams of method PCA (without normalization) for (a) 0 % noise, (b) 40 % noise, (c) 70 % noise and, (d) 100 % noise.

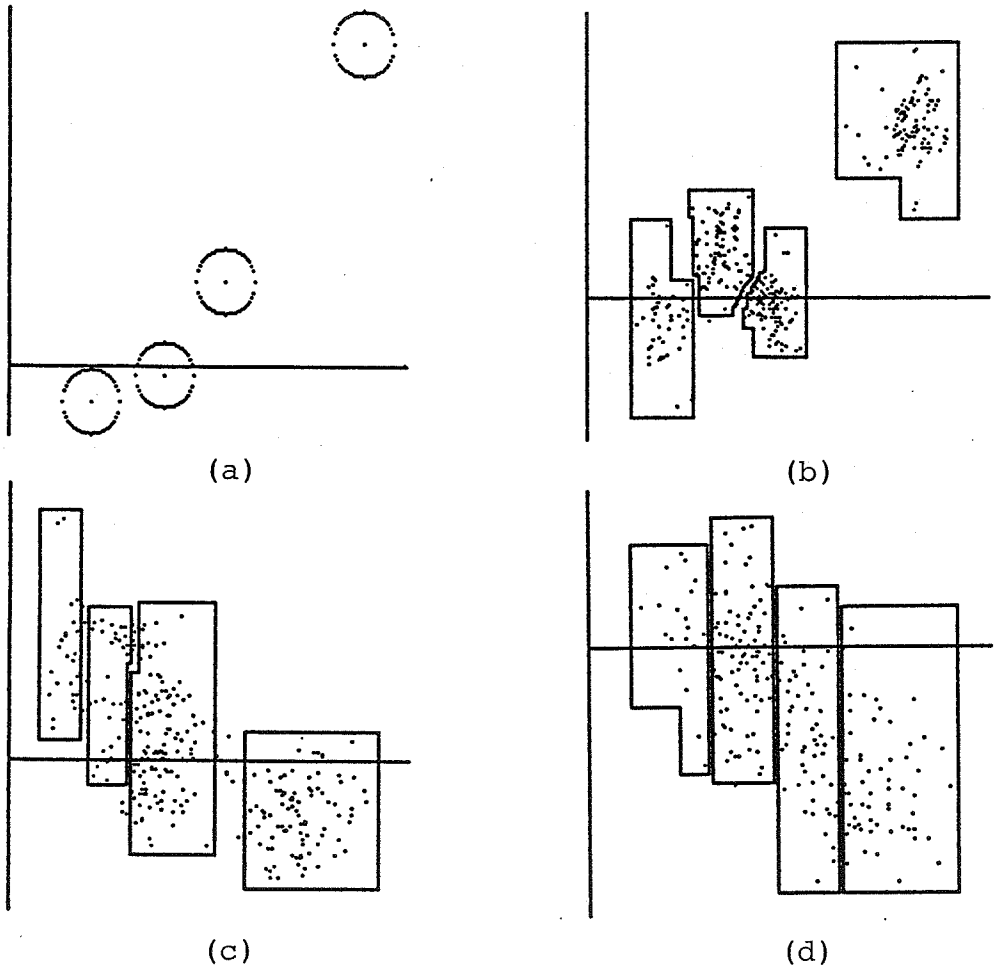


Figure 5.4. Cluster diagrams of method PCB (with normalization) for (a) 0 % noise, (b) 40 % noise, (c) 70 % noise and, (d) 100 % noise.

The larger amplitude waveform clusters for the nonnormalized case tend to spread out more than do the normalized case. The cluster plots also generally confirm the predictions of the separation matrices, showing slightly greater separation of smaller amplitude waveforms for the normalized case as compared to the nonnormalized method.

5.3 Results of the Cubic Spline Interpolation

In the second experiment conducted, the effects of the cubic spline interpolation on the second data set were investigated. Figure 5.5 shows that four of the five algorithms tested yielded improved results due to interpolation, with weighted template matching (W-RMS) showing the most improvement. Template matching (RMS) showed no improvement with interpolation with performance hampered by only 1.54 percent. Since the standard test file was selected by eye using the RMS Classification Display (Figure 3.8) without the help of interpolation, it suggests that the file was biased for slightly better performance without interpolation when using RMS classification. Though interpolation improved the performance of both principal component methods (PCA and PCB), the results yielded were less than expected. The reason for the poor performance of the techniques probably lies in the fact that not enough waveforms were present in the data set to allow sufficient clustering when classifing by eye.

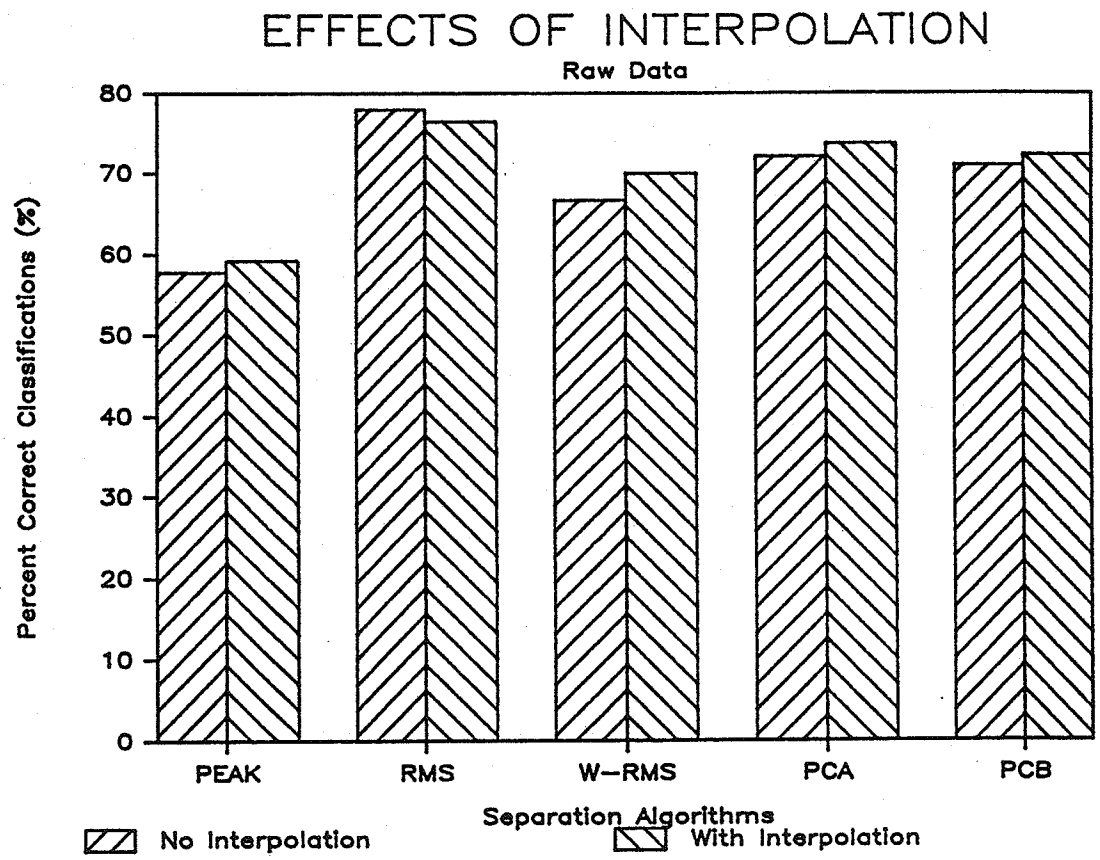


Figure 5.5. Performance of the five classification algorithms with and without cubic spline interpolation.

5.4 Results of Different Window Vector Sizes

Two experiments studying the effects of window vector sizes with and without the use of interpolation were conducted. Figure 5.6a shows the results without the use of interpolation using window vector sizes of 17,11, and 5 samples. RMS template matching showed no significant loss in performance when reducing the window vector size from 17 to 11 samples, dropping only 1.33 percent. W-RMS template matching and both PC methods dropped more than 4 percent with the PCA method showing the largest drop in performance with a 5.87 percent drop. When reducing the window vector size from 11 to 5 sample points, the W-RMS method in fact showed a slight improvement of 1.07 percent with RMS template matching yielding the largest drop in performance of 8.26 percent. Both PC methods dropped less than 2 percent in performance. Overall, the average drop in performance (percent correct classification) was 3.44 percent.

Figure 5.6b shows the results with the use of interpolation using window vector sizes of 17,11, and 5 samples. Both the RMS and W-RMS template matching methods showed less than 2.5 percent reduction in performance when lowering window vector size from 17 to 11 samples with W-RMS having the smallest drop of 2.34 percent. The largest drop of 5.62 percent in performance was yielded by the PCA method. Reducing the window vector size from 11 to 5 samples, the PCA method showed a slight improvement of 0.77 percent with the PCB method with the largest drop of 4.42 percent. In general, reductions in performance averaged 3.26 percent, with interpolation invoked showing a slight improvement

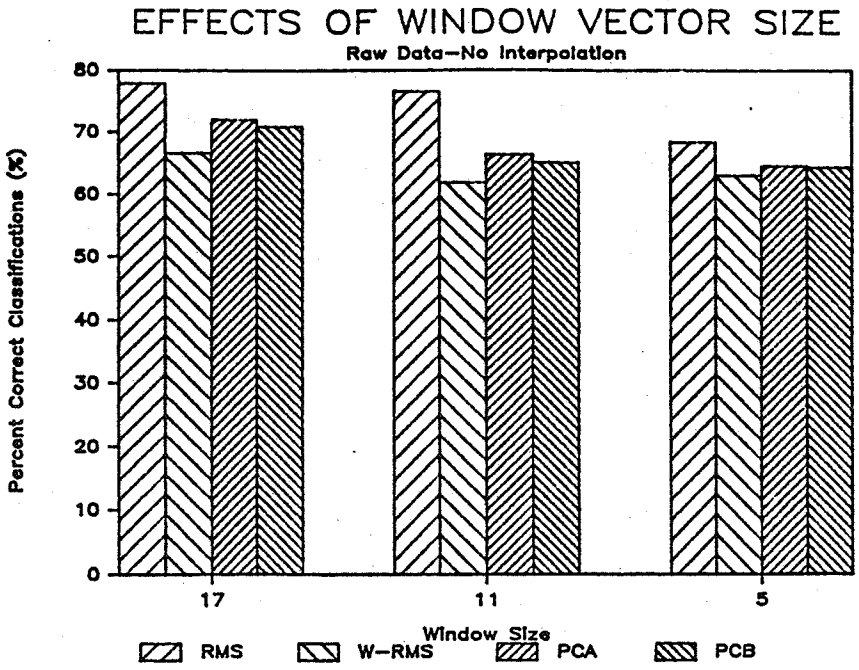


Figure 5.6a. Performance of four separation algorithms without the use of interpolation as a function of window vector size.

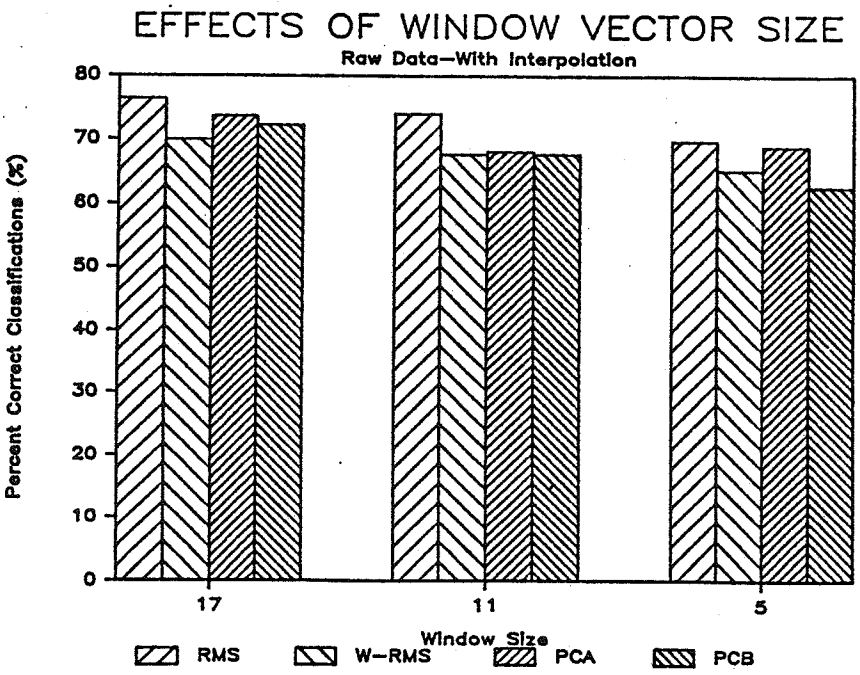


Figure 5.6b. Performance of four separation algorithms with the use of interpolation as a function of window vector size.

of 0.18 as compared to without using interpolation. Averaged over all algorithms and window vector sizes, a 1.46 percent improvement was found using interpolation.

Interpolation was found to be most effective when reducing the window vector size from 17 to 11 samples. For example, the RMS template matching would require 3*17=51 multiplications in order to compare the observed waveform to an existing template, i.e., the three groups of 17 multiplications are; 1. With observed waveform aligned at peak, 2. To left of peak, and 3. To right of peak, as compared to 3*11=33 multiplications with a window size of 11. If the maximum of 16 templates existed, the total amount of multiplications using 17 samples would be 816 compared to 528 with 11 samples. Altogether, the use of interpolation allows one to reduce the window vector size (17 to 11 samples) allowing a much faster operation time without a significant loss in classification performance.

5.5 Results of Interpolation and Window Vector Size on Filtering

The final experiment in Chapter 4 investigated the effects of interpolation and window vector size on the performance of the RMS template matching algorithm when using a first-order filter. Figure 5.7 shows an increase in performance (correct classifications) for all three window vector sizes when using interpolation. The results show an average increase of 4.63 percent in correct classifications when interpolation was implemented. Generally, the first-order and second-order filters lower the resolution of the spike waveforms (Figure 3.7). Though

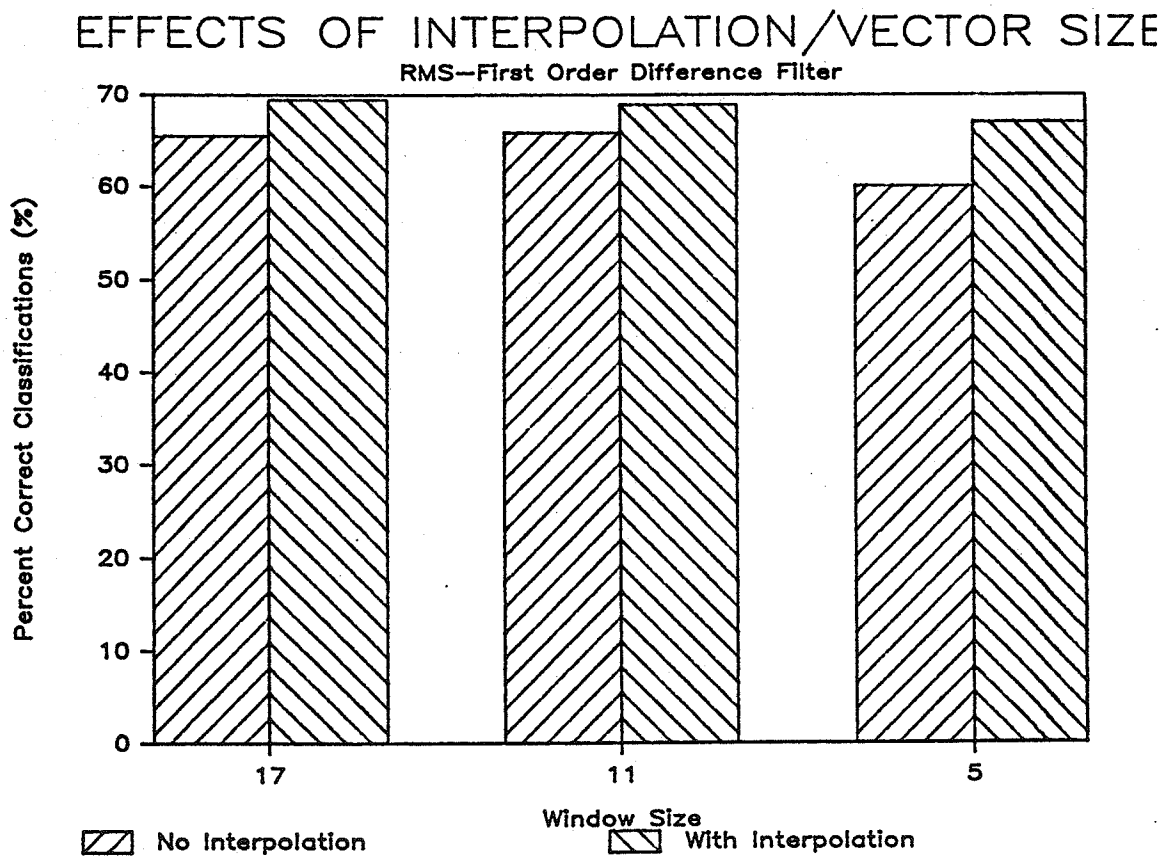


Figure 5.7. Performance of the RMS template matching method using a first-order filter. Effects on performance with and without interpolation as a function of window vector size are shown.

such filters make the spikes easier to detect, in essence they increase the frequency content of the samples making cubic spline interpolation much more effective. The results show the greatest increase in performance of 6.95 percent when using a window vector size of 5 sample points. The smallest increase was a significant 3.93 percent when using a window vector size of 17 sample points.

With the use of interpolation, one could use a window vector size of 5 samples and still get better performance than using a window vector size of 17 or 11 samples without interpolation (see Figure 5.7). As stated above, interpolation here would allow one to efficiently reduce the window vector size to 5 samples allowing much faster execution time with improved separation performance compared to using a larger time-consuming window vector size (17 or 11 samples) without interpolation.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

The results obtained show that the Automated Learning Phase system functioned as was desired. The investigation of several classification techniques indicates that it is possible to separate multiunit neuronal data with a fair amount of certainty even when corrupted with noise. As the test results show, more complex separation algorithms are needed for effective classification of the noisier action potential data. The mission of the Learning Phase is to download classification and dectection parameters to the Operational Phase. With the hindsight gained from the Learning Phase, the Operational Phase should operate more effectively. The experiments conducted in this thesis should assist others when deciding which technique is most feasible for their unique systems.

The investigation of the cubic spline interpolation suggests its usefulness in increasing the performance of the classification techniques. Though detection methods were not tested in this thesis, experiments on how interpolation affects (and possibly improves) detection methods would be interesting. Experiments have shown that interpolation can reduce the number of needed multiplications (Sections 5.4 and 5.5) thereby making it desirable for real-time Operational Phase systems.

Upon examination of the performance of the RMS template matching routine in which the data were filtered, using a simple first-order difference filter emphasized the need for cubic spline interpolation for classification. The filtering also showed that smaller window vector sizes are more reliable for separation when using interpolation. As stated earlier, smaller window vector sizes significantly decrease the amount of multiplications making all the algorithms computationally efficient. The cubic spline method of interpolation takes approximately 5 multiplications per sample point making it attractive for real-time applications.

Tests using real data show that the PC methods are less desirable than RMS template matching or W-RMS template matching. This is due to the fact that the small number of waveforms detected were not enough to show sufficient clusters, thereby making classification by eye more difficult. Since the Learning Phase inherently uses a small sample of the entire data, it will often be true that the sample of waveforms is too small for effective PC classification using clustering. Also, since the PC method calculates its two optimal PC vectors using the templates extracted from one of the other separation routines, execution time becomes almost doubled. With the capability of downloading the templates to the peripheral processor (see Section 3.1), it would generally seem more efficient that the PC algorithm be performed in the Operational Phase than in the Learning Phase. Here, the PC methods could analyze the whole data set as compared to a fraction of it. With the increase of waveforms to project

onto the two-dimensional PC space, it would be more likely to observe clusters thereby making classification boundaries much more apparent to the human eye.

With future advances in technology, computers will become faster and mass storage less expensive. Use of multiple electrode probes will allow the extraction of more information (features) for sophisticated real-time separation techniques. Such systems will constantly be redefining parameters from newly acquired data and recomputing past data in hopes of enhancing performance as a function of gained knowledge.

REFERENCES

[1] Abeles, M. and Goldstein, M.H., Jr., "Multispike train analysis," Proc. IEEE, vol. 65, pp. 762-773, 1977.

[2] Calvin, W.H., "Some simple spike separation techniques for simultaneously recorded neurons," Electroenceph. Clin. Neurophysiol., vol. 34, pp. 94-96, 1973.

[3] Camp, C. and Pinsker, H., "Computer separation of unitary spikes from whole-nerve recordings," Brain Res., vol. 169, pp. 455-479, 1979.

[4] Dahlquist, G. and Bjorck, A., Numerical Methods, (trans. N. Anderson). Prentice Hall, Englewood Cliffs, NJ, pp. 131-134, 1974.

[5] D'Hollander, E.H. and Orban, G.A., "Spike recognition and on-line classification by unsupervised learning system," IEEE Trans. Biomed. Eng., vol. BME-26, pp. 279-284, 1979.

[6] Dinning, G.J. and Sanderson, A.C., "Real-time classification of multiunit neural signals using reduced feature sets," IEEE Trans. Biomed. Eng., vol. BME-28, pp. 804-812, 1981.

[7] Drake 1986 "Signal Processing System for Multichannel Recording Electrodes," Phase I Final Report, NINCDS Contract #N43-NS-2385, National Institutes of Health, 1986.

[8] Fukunaga, K. Introduction to Statistical Pattern Recognition, New York, Academic Press, 1972.

[9] Gerstein, G.L., Bloom, M.J., Espinosa, I.E., and Turner, M.R., "Design of a laboratory for multineuron studies," IEEE Trans. Syst. Man, Cyber., vol SMC-13, pp. 668-676, 1982.

[10] Gerstein, G.L., and Clark, W.A., "Simultaneous studies of firing patterns in several neurons," Science, vol. 143, pp. 1325-1327, 1964.

[11] Glaser, E.M. "Separation of neuronal activity by waveform analysis," in Advances in Biomedical Engineering, Vol I, R.M. Kenedi (ed.), Academic Press, London, 1971.

[12] Heetderks, W.J., "Criteria for evaluating multiunit spike activity," Biol. Cybernetics, vol. 29, pp. 215-220, 1982.

[13] MacPhearson, L. and Lennon, W.J., "A digital technique for selecting action potential pulse of a specific height," Med. Biol. Eng., vol. 5, pp. 401-404, 1967.

[14] McGill K.C., Cummins K.L., and Dorfman L.J. 1984, "Automatic decomposition of the clinical electromyogram," IEEE Trans. Biomed. Eng., vol. BME-32, pp. 470-477, 1985.

[15] McGill K.C. and Dorfman L.J. 1983, "High-resolution alignment of sampled waveforms," IEEE Trans. Biomed. Eng., vol. BME-31, pp. 462-468, 1984.

[16] McGill, K.C and McMillan, K.L., "A smart trigger for real-time spike classification," Proc. 8th IEEE EMBS Conf., pp. 275-278, Fort Worth TX, 1986.

[17] Novak, J.L. and Wheeler, B.C., "Multisite hippocampal slice recording and stimulation using a 32 element microelectrode array," in press, J. Neurosci. Meth.

[18] O'Connell, R.J., Kocsis, W.A., and Schoenfeld, R.L., "Minicomputer identification and timing of nerve impulses mixed in a single recording channel," Proc. IEEE, vol. 61, pp. 1615-1621, 1973.

[19] Oppenheim, A.V. and Schafer, R.W., Digital Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1975.

[20] Prochazka, V.J. and Kornhuber, H.H., "On-line multi-unit sorting with resolution of superpositioned potentials," Electroenceph. Clin. Neurophysiol., vol. 34, pp. 91-93, 1973.

[21] Quint, S.R., Greenwood, R.S., Howard, J.F., and Gomez, J.V., "Automated data acquisition and analysis of neural evoked potentials," Comp. Meth. Prog. Biomedicine, vol. 20, pp. 35-44, 1985.

[22] Roberts, W.M. and Hartline, D.K., "Separation of multi-unit nerve impulse trains by a multi-channel linear filter algorithm," Brain Res., vol. 94, pp. 141-149, 1975.

[23] Schmidt, E.M. "Instruments for sorting neuroelectric data: a review," J. Neurosci. Meth., vol. 12, pp. 1-24, 1984.

[24] Schmidt, E.M. and Stromberg, M.W., "Computer dissection of peripheral nerve bundle activity," Computers Biomed. Res., vol. 2, pp. 446-455, 1969.

[25] Smith, S.R. "Performance of neural signal detection techniques for real-time data acquisition," M.S. Thesis, University of Illinois Electrical and Computer Engineering Department, Urbana IL, 1985.

[26] Smith, S.R. and Wheeler, B.C., "A real-time multiprocessor system for acquisiton of multichannel neural data," in preparation.

[27] Wheeler, B.C., "The evaluation of neural multi-unit separation techniques" Ph.D dissertation, Electrical Engineering Department, Cornell University, Ithaca, NY, 1981.

[28] Wheeler, B.C. and Heetderks, W.J., "Comparison of techniques for classification of multiple neural signals," IEEE Trans. Biomed. Eng., vol. BME-29, pp. 752-759, 1982.

[29] Wheeler, B.C. and Smith, S.R., "High resolution alignment of action potential waveforms with cubic splines," in press, J. Biomed. Eng.

[30] Wheeler, B.C. and Valesano, W.R., "Real-time digital-filter based data acquisition system for the detection of neural signals", Med. Biol. Eng. Comput., vol. 23, pp. 243-248, 1985.

[31] Wise, K.D., Najafi, K., and Drake, K.L., "A multichannel microprobe for intracortical single unit recording," Proc. IEEE/NSF Symp. Biosensors, pp. 87-89, 1987.