INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.

2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.
ZIED, ALI MOHAMED

FUZZY CLUSTERING IN A PARTITIONED KARHUNEN-LOEVE TRANSFORM DOMAIN-APPLICATION TO CHARACTERIZATION OF MULTIPLE-DIAGNOSIS VCG'S

The Ohio State University Ph.D. 1980

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106

18 Bedford Row, London WC1R 4EJ, England
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs _____
2. Colored illustrations _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Print shows through as there is text on both sides of page _____
6. Indistinct, broken or small print on several pages _____
7. Tightly bound copy with print lost in spine _____
8. Computer printout pages with indistinct print _____
9. Page(s) _____ lacking when material received, and not available from school or author
10. Page(s) 64 seem to be missing in numbering only as text follows
11. Poor carbon copy _____
12. Not original copy, several pages with blurred type _____
13. Appendix pages are poor copy _____
14. Original copy with light type _____
15. Curling and wrinkled pages _____
16. Other _____________________________________________
FUZZY CLUSTERING IN A PARTITIONED KARHUNEN-LOEVE
TRANSFORM DOMAIN—APPLICATION TO
CHARACTERIZATION OF MULTIPLE-DIAGNOSIS VCG'S

DISSERTATION
Presented in Partial Fulfillment of the Requirements
for the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Ali Mohamed Zied, B.Sc., M.Sc.

*********

The Ohio State University
1980

Reading Committee:

Herman Weed, M.Sc., P.E.
Richard Campbell, Ph.D.
Albert Kunz, M.D.
Said Koozekanani, Ph.D.

Approved By

Advisor

Department of
Electrical Engineering
ACKNOWLEDGEMENTS

This dissertation is dedicated to my wife, Soad, whose continuous support and sacrifice have helped me greatly. I am highly indebted to Dr. M. Edward Womble for his initial guidance, and to the USAF School of Aerospace Medicine for furnishing the computing facilities and the database. This work has been performed at the above site, and in direct support of an ongoing effort to develop a system for "Advanced Computerized VCG Interpretation". The advice, counsel, and assistance of my advisor, Professor Herman Weed, has been invaluable during this project.

These acknowledgements would not be complete without special thanks to Ms. Leticia Rivera who was very understanding and patient during the preparation and typing of this thesis.
VITA

February 2, 1944 ....... Born - Alexandria, Egypt
1964 ................. B.Sc.E.E., College of Engineering Alexandria University, Egypt
1965-1969 ............ Research Engineer, Nuclear Physics Laboratory Atomic Energy Agency - Cairo, Egypt
1969-1971 ............ Research Assistant, Nuclear Engineering Department, The Ohio State University, Columbus, Ohio
1971 ................. M.Sc., The Ohio State University, Columbus, Ohio
1971-1973 ............ Research Associate, Biomedical Engineering Center, The Ohio State University, Columbus, Ohio
1973-1974 ............ Assistant Professor, Ohio Institute of Technology, Columbus, Ohio
1974-1978 ............ Systems Analyst, Department of Physiology, The Ohio State University, Columbus, Ohio
1978-1980 ............ Biomedical Engineer, Clinical Sciences Division, USAF School of Aerospace Medicine, San Antonio, Texas

PRIMARY FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Biomedical Engineering. Professor Herman Weed.

Studies in Nuclear Engineering. Professor Donald Glower.

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Electrocardiography: Approximate or Optimal</td>
<td>1</td>
</tr>
<tr>
<td>The ECG Diagnostic Interpretation</td>
<td></td>
</tr>
<tr>
<td>Process and its Automation</td>
<td>6</td>
</tr>
<tr>
<td>Conventional Definitions and Definable Points</td>
<td>9</td>
</tr>
<tr>
<td>Outline of Thesis</td>
<td>13</td>
</tr>
<tr>
<td>II. A STATISTICAL APPROACH TO FEATURE EXTRACTION</td>
<td>15</td>
</tr>
<tr>
<td>Introduction</td>
<td>15</td>
</tr>
<tr>
<td>The Karhunen-Loeve Expansion</td>
<td>16</td>
</tr>
<tr>
<td>The Analog of Continuous Case</td>
<td>18</td>
</tr>
<tr>
<td>The Discrete Case</td>
<td>20</td>
</tr>
<tr>
<td>Optimality Properties</td>
<td>23</td>
</tr>
<tr>
<td>The K-L as a Filter</td>
<td>28</td>
</tr>
<tr>
<td>A Constraint</td>
<td>28</td>
</tr>
<tr>
<td>III. AN APPROACH TO CLASSIFIER DESIGN</td>
<td>30</td>
</tr>
<tr>
<td>Introduction</td>
<td>30</td>
</tr>
<tr>
<td>The Hard Partitioning Algorithm-ISODATA</td>
<td>32</td>
</tr>
<tr>
<td>Fuzzy ISODATA with Class-Dependent Fuzziness</td>
<td>35</td>
</tr>
<tr>
<td>Introduction</td>
<td>35</td>
</tr>
<tr>
<td>Mathematical Development</td>
<td>37</td>
</tr>
<tr>
<td>Concluding Remarks</td>
<td>41</td>
</tr>
<tr>
<td>IV. THE DATABASE</td>
<td>42</td>
</tr>
<tr>
<td>Introduction</td>
<td>42</td>
</tr>
<tr>
<td>Preprocessing</td>
<td>43</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Commercial Computer-Assisted Interpretation Systems</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Diagnostic Classes</td>
<td>44</td>
</tr>
<tr>
<td>5.1</td>
<td>Energy Fractions of the K-L in Each Partition</td>
<td>104</td>
</tr>
<tr>
<td>5.2</td>
<td>Class Population</td>
<td>124</td>
</tr>
<tr>
<td>5.3</td>
<td>Fuzziness Parameters</td>
<td>124</td>
</tr>
<tr>
<td>5.4</td>
<td>Disagreement Error Probabilities</td>
<td>124</td>
</tr>
<tr>
<td>5.5</td>
<td>Effect of Merging Class 29 into Classes 12, 13, on the Membership Functions</td>
<td>124</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1.1</td>
<td>The 12-lead Scalar System</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>The &quot;Weighted&quot; Frank Lead Set</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Vector Loop in the Left Sagittal and Transverse planes</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>The Classical Pattern Recognition Problem</td>
<td>10</td>
</tr>
<tr>
<td>1.5</td>
<td>Plots of X, Y, Z, and Vector Velocity</td>
<td>11</td>
</tr>
<tr>
<td>4.1</td>
<td>Tracing with Baseline Shift Knots Locations</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Noise Filtering Effect of the K-L</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>The P-Wave Reconstruction Problem</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Akant's Template</td>
<td>52</td>
</tr>
<tr>
<td>4.5</td>
<td>Mathematical Template Functions</td>
<td>55</td>
</tr>
<tr>
<td>4.6</td>
<td>The Multiple-Template Detection Scheme</td>
<td>56</td>
</tr>
<tr>
<td>4.7</td>
<td>Weighting Function</td>
<td>59</td>
</tr>
<tr>
<td>4.8</td>
<td>The Overall System Flow Structure</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>Original, Estimate, and Baseline-restored Tracing</td>
<td>65</td>
</tr>
<tr>
<td>5.2</td>
<td>Performance of the P-wave Detector</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Performance of the P-wave Detector (Continued)</td>
<td>67</td>
</tr>
<tr>
<td>5.4</td>
<td>P-wave Basis Vectors-Lead X</td>
<td>71</td>
</tr>
<tr>
<td>5.5</td>
<td>P-wave Basis Vectors-Lead X (Continued)</td>
<td>72</td>
</tr>
<tr>
<td>5.6</td>
<td>P-wave Basis Vectors-Lead Y</td>
<td>73</td>
</tr>
<tr>
<td>5.7</td>
<td>P-wave Basis Vectors-Lead Y (Continued)</td>
<td>74</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.28</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead X</td>
<td>95</td>
</tr>
<tr>
<td>5.29</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead X (Continued)</td>
<td>96</td>
</tr>
<tr>
<td>5.30</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead X (Continued)</td>
<td>97</td>
</tr>
<tr>
<td>5.31</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Y</td>
<td>98</td>
</tr>
<tr>
<td>5.32</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Y (Continued)</td>
<td>99</td>
</tr>
<tr>
<td>5.33</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Y (Continued)</td>
<td>100</td>
</tr>
<tr>
<td>5.34</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Z</td>
<td>101</td>
</tr>
<tr>
<td>5.35</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Z (Continued)</td>
<td>102</td>
</tr>
<tr>
<td>5.36</td>
<td>The &quot;Gross-Abnormal&quot; Basis Vectors-Lead Z (Continued)</td>
<td>103</td>
</tr>
<tr>
<td>5.37</td>
<td>A Test Beat and its Inverse K-L Using the &quot;All-Ensemble&quot; Partition-Example A</td>
<td>105</td>
</tr>
<tr>
<td>5.38</td>
<td>A Test Beat and its Inverse K-L Using the &quot;Gross-Abnormal&quot; Partition-Example A</td>
<td>106</td>
</tr>
<tr>
<td>5.39</td>
<td>A Test Beat and its Inverse K-L Using the &quot;All-Ensemble&quot; Partition-Example B</td>
<td>107</td>
</tr>
<tr>
<td>5.40</td>
<td>A Test Beat and its Inverse K-L Using the &quot;Gross-Abnormal&quot; Partition-Example B</td>
<td>108</td>
</tr>
<tr>
<td>5.41</td>
<td>A Single Lead Tracing and its Inverse K-L Using the &quot;All-Ensemble&quot; Partition (Top), and the &quot;Gross-Abnormal&quot; Partition (Bottom)-Example C</td>
<td>109</td>
</tr>
<tr>
<td>5.42</td>
<td>A Test Beat and its Inverse K-L Using the &quot;All-Ensemble&quot; Partition-Example D</td>
<td>110</td>
</tr>
<tr>
<td>5.43</td>
<td>A Test Beat and its Inverse K-L Using the &quot;QRS-Suppressed&quot; Partition-Example D</td>
<td>111</td>
</tr>
<tr>
<td>5.44</td>
<td>A Test Beat and its Inverse K-L Using the &quot;All-Ensemble Partition-Example E</td>
<td>112</td>
</tr>
<tr>
<td>5.45</td>
<td>A Test Beat and its Inverse K-L Using the &quot;QRS-Suppressed&quot; Partition-Example E</td>
<td>113</td>
</tr>
</tbody>
</table>
Figure

5.46 A Test Beat and its Inverse K-L Using the "All-Ensemble" Partition—Example F .............. 114

5.47 A Test Beat and its Inverse K-L Using the "QRS-Suppressed" Partition—Example F ............ 115

5.48 Feature Subspace of Components 1 and 2—Lead X .. 118

5.49 Feature Subspace of Components 2 and 3—Lead X .. 119

5.50 Feature Subspace of Components 1 and 2—Lead Y .. 120

5.51 Feature Subspace of Components 2 and 3—Lead Y .. 121

5.52 Feature Subspace of Components 1 and 2—Lead Z .. 122

5.53 Feature Subspace of Components 2 and 3—Lead Z .. 123
I. INTRODUCTION

1.1 Electrocardiography: Approximate or Optimal?

Electrocardiography generally deals with the potential differences resulting from the activity of the heart and recorded on the body surface. Extracting as much significant information as possible from the recording(s) is a function of:

   a. New information learned in electrophysiology
   b. Specificity and sensitivity of the information
   c. Identification of the optimal features or measurements
   d. The diagnostic criteria used in interpretation
   e. Measurements vs criteria errors [1]
   f. Ability to record and process information using certain minimum error criteria (e.g. minimum false positives).

Body surface potential distributions are the optimum recordings in predicting cardiac potentials and the anatomic and functional (physiologic or pathologic) changes in the heart that result in these potentials [2, 3]. This procedure is, however, clinically impractical due to lack of diagnostic criteria, complexity of the data, and the incomplete understanding of the heart as a signal generator. A common approximation has been the use of biophysical models from which two conventional lead systems have evolved: In the first model, the heart is considered to be a "volume" generator with certain leads being primarily sensitive to potentials generated in the myocardium nearest to the precordial electrodes.
This resulted in the conventional 12-lead scalar ECG system\(^1\) [see Fig. 1]. Although proponents of the scalar 12-lead model maintain that diagnostically significant information is gained by "local" leads (e.g. V1-V6), several researchers have concluded that redundancy and intercorrelation exist among the leads (in both the limb and the precordial subsets), and all leads can be synthesized from the upper orthogonal set via linear transformation \([4,5,6]\). The second biophysical model of the heart assumes a fixed-position free-rotating dipole (heart vector) in a homogeneous volume conductor. This resulted in the orthogonal lead system. Furthermore, to account for the anatomical structure of the torso and the eccentricity of the signal generator, a "weighted" orthogonal lead set has been introduced, known as the "Frank" set. This resulted in a system whereby simultaneous recordings of three pair of mutually orthogonal leads are generated, along with parametric plots (vector loops). Each loop is the planar projection of a 3-dimensional trajectory of what has been called the "heart vector", [See Figures 1.2-1.3].

Classically, it is these spatial-temporal display characteristics that gave vectorcardiography an "intermediate" position between scalar ECG and body surface mapping. It has also been noted that the dipole parameters (and hence the VCG leads) can be synthesized from body surface mappings by surface integration \([7]\).

Accuracy of the dipole model—as a first order approximation—has been challenged, when one tries to explain the multiple peaks in the isopotential surface maps. Computer modelling did show, however, that similar potential distributions can be generated by a rotating dipole in an inhomogeneous

---

1 The lead nomenclature recommended by the American Heart Association is: I,II,III,aVR,aVL,aVF,V1-V6.
Figure 1.1 The 12-lead Scalar System
\[ x = 0.6172A + 0.1707C - 0.8903I \]
\[ y = -H + \frac{2}{3}LL - \frac{1}{3}M \]
\[ z = 0.1999A - 0.3781C - 0.6421E - 0.4117I + 1.1331M \]

(RL is the reference electrode)

**Figure 1.2 The Weight" Frank Lead Set**
Figure 1.3 Vector Loops in The Left Sagittal and Transverse Planes
media. This suggests that the multiple peaks are "artifacts" introduced by the surrounding media, rather than a higher order pole cardiac generator [8]. To be conclusive, however, more work is needed to relate surface potentials to cardiac potentials and anatomy (cardiac and extracardiac) in order to provide the basis for accurate electrocardiography. Ultimately, direct cardiac mapping and/or intracardiac recordings—if proven clinically feasible—may prove to be the optimal approach to diagnosing cardiac abnormalities, insofar as maximizing both the sensitivity and specificity of the information.

1.2 The ECG Diagnostic Interpretation Process and its Automation:

Clinical features and feature measurements of the electrocardiogram have been empirically and/or semi-analytically associated with cardiac pathology for the past 60 years. For a cardiologist, the feature extraction problem is to identify and measure certain parameters (for example, wave amplitude, duration, segment slope, etc) that can be used to discriminate between different pathological classes (classification). An experienced electrocardiologist can recognize, however, each Bundle Branch Block tracing at a "glance", and without direct measurements. He applies an extremely sophisticated pattern recognition process to read a tracing, and only resorts to detailed threshold analysis of these measurements for more subtle and complex findings. This is why it is generally not possible to reverse the measurements process, i.e., to accurately and uniquely reconstruct an original cardiogram from its measurements. Only certain "features" have been reserved by the measurements. Since it is not known apriori which information is clinically significant and which is not,
large number of measurements have been defined in clinical cardiology. The classification problem is then a function of which subset of these measurements and which diagnostic criteria/thresholds the cardiologist uses. This is perhaps the reason behind repeatability (or lack of), both within and between cardiologists, and the variability in ability of different cardiologists insofar as the immeasurable pattern recognition part.

Computer-assisted electrocardiographic interpretation has evolved as a means of providing—among other things—the "uniform standard" [1]. Development of such interpretation systems began in the early 1960's, yielding today at least ten major programs for routine automated interpretation (see table 1). Yearly statistics on utilization of computerized interpretation systems are shown in the table, as reported by A.D. Little [1].

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>VOLUME</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>1,400,000</td>
<td>Proprietary</td>
</tr>
<tr>
<td>TELEMED</td>
<td>1,400,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>CROMED</td>
<td>500,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>MAYO</td>
<td>325,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>HP</td>
<td>500,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>PHONO-A-GRAM</td>
<td>140,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>SLEMED</td>
<td>150,000</td>
<td>Public Domain</td>
</tr>
<tr>
<td>ECAN</td>
<td>195,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>VA</td>
<td>160,000</td>
<td>&quot;</td>
</tr>
<tr>
<td>CEIS</td>
<td>110,000</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

All of the commercially available programs—with the exception of Pipberger's [10]—have taken the heuristic approach to analysis, and require overreading. Most algorithms are deterministic and attempt to
recognize the waveform using just amplitude and duration measurements, while ignoring contours and shapes. All front-end processing modules in the program have been developed to recognize waveform segments, and to measure time durations and amplitudes, or certain heuristic relationships derived thereof. The measurements are then entered into a deterministic using Boolean logic (Yes/No) and hierarchical decision tree structure. Threshold settings are used for discrimination and branching at each node in the decision tree. The branching logic is very complex and if a branch is falsely entered (borderline cases, for example), the logic never returns to the initial node. The decisions at all nodes are heavily thresholds and/or criteria dependent [10,11].

The above characteristics of the existing algorithms are problematic in that discontinuous decision thresholds are required within the framework of what is intuitively a continuum. The programs tend to be inherently sensitive to nondiagnostic morphological changes (noise, for example) and to lack reproducibility. This yields abnormally high error rates (false positives and/or negatives), even with 100% overreading, and makes it difficult to develop reliable comparative (or serial) comparison algorithms sensitive enough to detect subtle changes [1]. Recently, there have been systematic attempts, using statistical modeling techniques, to compute the most optimal features for classification, and to formulate probabilistic decision rules (or probability measures) in the classification process [12,13]. The approach centers on current efforts to mathematically model the morphology of a single beat of the VCG as a random stochastic process (or more accurately multiple processes). Each beat is then expanded in terms of the appropriate set of orthonormal functions. The model computes

\[ 2 \text{ Is a T-wave amplitude of } .2 \text{ mv normal but } .22 \text{ mv abnormal}.! \]
the coefficients of what is termed the "pattern vector". This feature extraction approach is a mapping (or a transform process as in case of the Fourier Transform) from an n-dimensional space (n=number of samples) to a lower m-dimension space (m=number of features or coefficients of the truncated transform. m≤n) using the Karhunen-Loeve expansion. The class separability of the distribution of the original data is the same as that of the features. Therefore, the mapping should not cause any significant loss of class separability. Pattern classification is then performed on the pattern vectors in an m-dimensional feature space using supervised learning techniques to partition the space into clusters (or other discriminants) which correspond to clinical diagnostic classes. The pattern recognition problem as depicted in Fig. 1.4 is an optimization problem. This is accomplished by choosing the most efficient feature extractor and the most tractable classifier.

1.3 Conventional Definitions and Definable Points:

The three orthogonal components of the VCG are conventionally labelled X, Y, and Z. One heart beat is considered a 3-component, time-varying, quasi-stationary vector process, and each component is the projection of the heart vector, h(t), in the orthogonal coordinate system. Each beat may be further segmented into two or more processes such as the P-processes and the QRST-processes (time segmentation or partitioning). The ensemble may be also partitioned into several subsets of processes such as the gross depolarization subset and the repolarization abnormalities subset. A CALCOMP plot of the classical 3-component vectorcardiogram and the magnitude of the heart vector incremental velocity is shown in Fig. 1.5.
Figure 1.4 The Classical Pattern Recognition Problem

n-sample Input  
n=200/lead

m-feature  
Pattern Vector  
m=20/lead

1-class Probabilities
n-sample Input
n=200/lead

m-feature
Pattern Vector
m=20/lead

1-class Probabilities

Figure 1.4 The Classical Pattern Recognition Problem
Figure 1.5 Plots of X, Y, Z, and Vector Velocity
A sampled heart vector \( h(k) \) may be expressed as:

\[
    h(K) = [x(K) \quad y(K) \quad z(K)]
\]

The incremental velocity vector is defined by

\[
    v(K) = [\delta x(K) \quad \delta y(K) \quad \delta z(K)]
\]

where

\[
    \delta x(K) = x(K) - x(K+1) \\
    \delta y(K) = y(K) - y(K+1) \\
    \delta z(K) = z(K) - z(K+1)
\]

then

\[
    |v(K)| = \left[ \left| \delta x(K) \right|^2 + \left| \delta y(K) \right|^2 + \left| \delta z(K) \right|^2 \right]^{1/2}
\]

The velocity plot is extremely useful in defining three points of interest:

1. **Fiducial Point (maximum activity):**
   
   This is defined as the location of maximum incremental velocity on the downslope of the R-wave. This is a highly accurate and consistent point of alignment for beat-to-beat comparison, correlation, and averaging.

2. **Reference Point or Baseline Knot (minimum activity):**
   
   This is defined as the location of minimum (or zero) velocity that lies within the P-R interval. High uncertainty is associated with the location of this point, and it hardly exists under certain pathological conditions (PVC for example). On the other hand, it has been classically found that for most VCG's it is the only definable point for baseline tracking and restoration, in preprocessing VCG's for computer analysis [14] and waveform segmentation [12].

3. **J—Junction Point:**
   
   This point characterizes the ending of the depolarization phase of the ventricular chambers and the onset of their repolarization phase[15].
With such loosely defined characteristic, no "robust" algorithm exist for its detection. If detectable, it will naturally lead to additional waveform segmentation for the feature extraction process [12].

It is the issues discussed above, originally motivated by the potential of applying statistical signal analysis techniques to VCG classification that led to the formulation and general analytical and numerical methods developed in this study.

1.3 Outline of Thesis

In Chapter 2, the feature extraction problem using the Karhunen-Loeve expansion is considered. The expansion is derived from its ensemble-global error minimizing property, for both single-component and three component stochastic vector process. An elegant proof of its optimality is derived, and its adaptability to serial comparative analysis is discussed. The noise filtering properties of the expansion are also presented.

In Chapter 3, different strategies for the classifier design, and partitioning of the feature space is presented. Classical supervised learning via hard and fuzzy clustering is discussed. A novel extension of the fuzzy partitioning algorithm to include class-dependent (or more naturally cluster-dependent) fuzziness is derived and implemented. Applicability of the latter approach to classification of multiple-diagnosis VCG's is discussed.

In Chapter 4, preprocessing and formulation of the database for training the clustering algorithm is discussed. This includes baseline restoration based on a "true" cubic spline, averaging, and time-
segmentation of each beat into multiple stochastic processes. A new algorithm for P-wave detection is derived, based on multiple-template matching for time segmentation. New concepts of ensemble-partitioning and process-partitioning are introduced, and a scheme for implementing a 2-partition problem is presented. Finally, an overall hierarchical scheme of successive time/ensemble partitioning is discussed.

In Chapter 5, the signal analysis techniques developed in this thesis are applied to the problem of characterization of single and multiple diagnosis VCG's. Extensive experimental results are presented for this problem.

Finally, conclusions are presented and other aspects of the overall problem that require additional analytical and statistical studies are suggested.
II. A STATISTICAL APPROACH TO FEATURE EXTRACTION

2.1 Introduction

A linear feature extraction/selection scheme for automated electrocardiographic interpretation will be presented in this chapter. The problem is concerned with eliminating redundancy in the raw data using what is mathematically termed "dimensionality reduction" or data compression techniques. Today, the automated interpretation process involves transmitting ECG/VCG's to a central computing facility via standard voice grade telephone lines. A computer program extracts the classical measurements (or features) and performs the interpretation. There is growing need for digital transmission of cardiograms as a result of the well-known noise problems associated with transmitting analog data in the form of FM signals over the telephone lines [16,17], and the advantages of maintaining a digital database for subsequent data analysis and serial comparison studies. The transmission problem is being completely disassociated from the feature extraction problem in the current systems. The main problem with digital transmission and storage of ECG/VCG's is the large volume of bits required to adequately represent the signals [18,19]. Proposed solutions in the literature are based on taking advantage of the large sample-to-sample and beat-to-beat redundancy associated with cardiograms. The approaches range from sending and storing samples only when a change exceeds a certain threshold through using time series to represent the beats, to using a transform which reduces quantization redundancy, and using Huffman coding techniques [20,22]. In addition,
combination of the procedures are being used. At the receiving site, the data is decoded for use by the interpretation system.

Based on the above, and realizing the limitations in using classical measurements to approximate the signals, an innovative approach is presented in this chapter to optimize the transmission/feature extraction process by performing it at the front-end. The approach is based on modern statistical pattern recognition techniques.

2.2 The Karhunen-Loeve Expansion

Earlier efforts to achieve data compression of ECG's involved developing a linear transformation of the standard scalar signals to the equivalent VCG orthogonal components, yielding therefore a significant 4:1 compression ratio [4,6]. In other work, additional data compression was obtained by transforming the signals from the Frank coordinate system to the eigen plane (or intrinsic) coordinate system [23]. The reduction is a result of the planar properties of the VCG, which if consistently true, would allow a 3:2 compression ratio via neglecting the out-of-plane component. Subsequent results have not borne out the optimism with the approach [12].

Modern signal analysis techniques (deterministic as well as statistical) present alternate tools for signal approximation via transformation or mapping [24]. The techniques are categorized into linear, non-linear, parametric, and non-parametric. Statistical linear feature extractors are of particular interest since they are analytically and computationally tractable, and are well suited for ensemble-based problems [24]. Although suboptimal [25], they are simpler to implement and more
reliable than non-linear feature extractors [26], and lead to simpler classifers especially when the number of features is high.

The fundamental mathematical tools for statistical decision-making adopt some kind of criterion, and choose the linear transformation that optimizes the criterion. Of interest to our problem is to minimize the ensemble—global mean—square approximation error $^3$, since it the generally accepted requirement to design the automated interpretation system with minimum overall misclassification error. Accordingly, one must seek the most efficient feature extractor, i.e. least number of features, that optimally preserve the salient global and local characteristics. The Karhunen—Loeve expansion (K—L) is a transformation technique that optimizes the criteria of interest [27,29]. For a distribution of non—periodic random processes, the K—L is a mapping algorithm from an original n—dimensional space (n=number of samples in a sample waveform) to an m—dimensional feature space, with $m \leq n$. The expansion is performed in terms of orthonormal basis functions that are derived from the second order statistics of the processes, i.e. the autocorrelation function. For analog processes, the basis functions are the eigen functions of the autocorrelation. For discrete processes (sampled—data), the basis vectors are the eigen vectors of the autocorrelation matrix. If the corresponding eigen values are indexed in a descending order, the first m—largest terms in the expansion carry the greatest energy of the process (will be proved later) and minimal entropy of the coefficients $^4$. The significance of the first property is that no other expansion yields lower approximation error (in the mean square sense), thus yielding the most

---

3 Corollary is the term ensemble-mean cost function
4 Entropy is a statistical means of uncertainty
efficient coefficients (or features) among linear feature extractors. The second property implies that the resulting pattern vectors (the coefficients of the expansion constitute its elements) are of minimal divergence. In other words, the resulting pattern vector for each sample waveform as it maps into the m-dimensional feature space and forms a cluster with similar classes of waveforms - the resulting clusters will have minimum intracluster dispersion. Accordingly, the resulting clusters are of finite boundaries and separable [27]; this is a very desirable property in order to minimize class overlapping in the feature space.

The following two sections present the mathematical basis of the K-L expansions for both continuous and discrete processed, as derived in the theoretical literature [2, 4, 29-31]. The derivation is then extended to the 3-component vector process. Finally, a novel and elegant proof of its error optimality properties is presented.

2.2.2 The Analog or Continuous Case:

Let \( \{\phi_\ell(t)\} \), \( \ell = 1, 2, \ldots, \infty \) denote a complete set of orthonormal functions in the interval \([0, T]\):

Then

\[
\int_0^T \phi_\ell(t) \phi_m(t) \, dt = \delta_{\ell m}
\]

where

\[
\delta_{\ell m} = 1 \quad \text{if} \quad \ell = m \quad \text{and} \quad \delta_{\ell m} = 0 \quad \text{if} \quad \ell \neq m
\]

Consider C-number of classes \( \gamma_1, \gamma_2, \ldots, \gamma_C \) of continuous random processes \( \{f(t)\} \)

where \( f \) is non-periodic in the interval \([0, T]\). \( f \) may be expanded in the form:

\[
f(t) = \sum \alpha_\ell \phi_\ell(t)
\]
The orthonormal functions \( \{ \phi_{k}(t) \} \) and the coefficients \( \{ \alpha_{k} \} \) are determined as follows:

Substituting 2.2 into 2.1 yields

\[
\begin{align*}
  f(t) &= \int_{0}^{T} f(s) \sum_{k} \phi_{k}^{*}(s) \phi_{k}(t) \, ds \\
  \Rightarrow R(t,s) &= \sum \mathcal{P}(\chi_{k}) \mathbb{E} \{ f_{k}(t) f_{k}^{*}(s) \}
\end{align*}
\]  

(2.3)

Where \( \mathcal{P}(\chi_{k}) \) is the apriori probability of classes \( \chi_{k} \). Substituting equation 2.2 into 2.3 yields

\[
R(t,s) = \sum \mathcal{P}(\chi_{k}) \sum_{m} \sum_{l} \phi_{l}(t) \phi_{m}^{*}(s) \mathbb{E} \{ \alpha_{l} \alpha_{m}^{*} \}
\]

Since the coefficients \( \{ \alpha_{k} \} \) are random, and by imposing the following condition of statistical independence:

\[
\sum \mathcal{P}(\chi_{k}) \mathbb{E} \{ \alpha_{l} \alpha_{m}^{*} \} = \begin{cases} 
\lambda_{l} & l=m \\
0 & l \neq m
\end{cases} \quad (E \equiv \text{expected value})
\]

Then \( R(t,s) \) may be written as:

\[
R(t,s) = \sum \lambda_{l} \phi_{l}(t) \phi_{l}^{*}(s) \quad (2.4)
\]

Multiplying 2.4 by \( \phi_{m}(s) \) and integrating over the interval \([0, T]\) yields:

\[
\int_{0}^{T} R(t,s) \phi_{m}(s) \, ds = \sum \lambda_{l} \phi_{l}(t) \int_{0}^{T} \phi_{l}(s) \phi_{m}(s) \, ds = \lambda_{m} \phi_{m}(t)
\]

or

\[
\int_{0}^{T} R(t,s) \phi_{m}(s) \, ds = \lambda_{m} \phi_{m}(t) \quad (2.5)
\]

Equation 2.5 is the general form of the eigen value problem. The expansion given in equation 2.1, using the eigen functions as determined by 2.5, is known as the Karhunen-Loeve Expansion.

\( \{ \phi_{m}(t) \} \) are the eigen functions of \( R(t,s) \), and \( \{ \lambda_{m} \} \) are the corresponding eigen values.
2.2.2 The Discrete Case

2.2.2.1 Single-component process

A discrete-time random process \( f_\ell \) in the form

\[
f_\ell = \begin{bmatrix} f_\ell(1) & f_\ell(2) & \cdots & f_\ell(n) \end{bmatrix}^T
\]

may be presented as:

\[
f_\ell = \sum_{m=1}^{n} \alpha_{\ell m} \phi_m
\]

where

\[
\phi_m = \begin{bmatrix} \phi_m(1) \\ \phi_m(2) \\ \vdots \\ \phi_m(n) \end{bmatrix}
\]

Define

\[
\alpha_\ell = \begin{bmatrix} \alpha_{\ell 1} \\ \alpha_{\ell 2} \\ \vdots \\ \alpha_{\ell n} \end{bmatrix}
\]

and

\[
\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix}
\]

Equation 2.6 may be written as:

\[
f_\ell = \phi \cdot \alpha_\ell
\]

The matrix \( \phi \) is made up of \( n \)-linearly independent column vectors called the "basis" vectors. If \( \phi_1, \phi_2, \ldots \) are chosen to be orthogonal, i.e.

\[
\sum \phi_i(t) \phi_j(t) = \delta_{ij}
\]

then

\[
\alpha_{\ell m} = \sum f_\ell(t) \phi_m(t)
\]

or

\[
\alpha_\ell = \phi \cdot f_\ell
\]

The discrete counterpart of the autocorrelation function \( R(t,s) \) is the autocorrelation matrix. This is defined as:
\[
R = \sum_k P(\xi_k) E \{ f_k f_k^T \} = \sum_k P(\xi_k) E \{ \phi \alpha_k \alpha_k^T \} = \phi \left( \sum_k P(\xi_k) E \{ \alpha_k \alpha_k^T \} \right) \phi^T
\]

If the coefficients vectors \( \{ \alpha_k \} \) are statistically uncorrelated in the sense

\[
\sum_k P(\xi_k) E \{ \alpha_k \alpha_k^T \} = \Lambda
\]

where \( \Lambda \) is a diagonal matrix in the form \( \text{diag}(\Lambda) = [\lambda_1 \ldots \lambda_n] \) then

\[
R = \phi \Lambda \phi^T
\]

or

\[
R \phi = \phi \lambda
\]

or \( R \phi_j = \phi_j \lambda_j \) for the \( \lambda_j \)th eigen value. (2.8)

Equation 2.8 is the K-L expansion in the matrix form.

2.2.2.2 Three-component Process (vector process)

It is relevant at this stage to extend the formulation of the eigenvalue problem to processes with 3 components such as the vector-cardiogram. A vector process \( f \) may be written as

\[
f = \begin{bmatrix} f_1 & \ldots & f_3 \end{bmatrix} \text{ 3n x 1 vector}
\]
The process \( f \) may be expanded in terms of the basis functions \( \{ \phi_j \} \) as follows:

Define

\[
\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \quad \mathbf{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}
\]

where

\[
\phi = [\phi_1 \quad \phi_2 \quad \ldots \quad \phi_n]
\]

then

\[
\mathbf{f} = \phi \mathbf{\alpha}
\]

The coefficients vector \( \mathbf{\alpha} \) of the expansion is of the form

\[
\mathbf{\alpha} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \vdots & \vdots & \vdots \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix}
\]

Since

\[
E(\mathbf{f} \mathbf{f}^T) = \begin{bmatrix} f_1 f_1^T & \cdots & f_1 f_3^T \\ \cdots & \cdots & \cdots \\ f_3 f_1^T & \cdots & f_3 f_3^T \end{bmatrix}
\]

then the autocorrelation function for the vector process \( \mathbf{f} \) may be written as

\[
\mathbf{R} = \begin{bmatrix} R_{11} & \cdots & R_{13} \\ \vdots & \ddots & \vdots \\ R_{31} & \cdots & R_{33} \end{bmatrix}
\]

(2.9)

where \( R_{ij} \) is the crosscorrelation function for the \( i^{\text{th}} \) and \( j^{\text{th}} \) components of the process.
Special Case:

Since the three components of the vectorcardiogram are mutually independent and statistically uncorrelated, then

\[ R_{ij} = \text{null matrix} \quad \text{for} \quad i \neq j \]

Accordingly

\[ R = \begin{bmatrix} R_{11} & 0 & 0 \\ 0 & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \]

Following the derivation in 2.2.2 above, we get

\[ R \cdot \phi_j = \lambda_j \phi_j \] (2.10)

2.2.3 Optimality Properties

The following are the most important properties of the K-L from information-theoretic considerations:

1. If a coefficient, say \( \alpha_i \), is deleted from the expansion, then the mean square magnitude of truncation error \( \overline{E^2(m)} \) increases by the value of \( \alpha_i \), i.e.

\[ \overline{E^2(m)} = E \left\{ \| f - \hat{f}(m) \| ^2 \right\} \]

where

\[ f = \text{the random process} \]

and

\[ \hat{f} = \text{estimate of} \quad \text{after truncating} \ n-m \text{ terms} \]

If the eigen values are arranged such that

\[ \lambda_1 \geq \lambda_2 \ldots \geq \lambda_m \]

\[ \text{E is the expected value operator} \]
then the coefficients should be arranged in the same way.

2. The coefficients of the expansion are mutually uncorrelated

3. Over all choices of orthogonal basis functions, the eigen vectors of the autocorrelation $R$ minimizes $\overline{E^2(C_m)}$. In other words, $\overline{E^2(C_m)}$ is bounded form below by $\overline{E^2(C_m)}$ K-L (ensemble-average). Proof of this important property is presented later in this chapter.

Interestingly, it has been shown that for a periodic non-stationary process ($R(t,s) = R(t,s)$; $E(f) = 0$ or const.), the autocorrelation matrix (or covariance matrix in the case of $E(f)=0$) is symmetric with 1's diagonal elements, and the K-L expansion reduces to the Fourier Transform [32]. The eigen values are the DFT of the covariance matrix. Since the K-L is the optimal expansion yielding the smallest mean square approximation error, it is more efficient than the Fourier Transform.

4. The entropy function $H_i$ of any set of eigen values $\{\lambda_i\}$ is defined as

$$H_i = -\sum \lambda_i \log \lambda_i$$

If $\{\lambda_i\}$ are the eigen values of

$$R \phi_j = \lambda_j \phi_j$$

over all the basis functions, the K-L's yield the lowest entropy.

2.2.3.1 Lower Bound of $\overline{E^2(C_m)}$

For a linear expansion in terms of an orthonormal set $\{\phi_j\}$, $\overline{E^2(C_m)}$ may be written in the form

$$\overline{E^2(C_m)} = \sum_{i=m+1}^{n} E \left\{ |\alpha_i|^2 \right\}$$

(2.11)

---

4 Assume for simplicity that $p(\alpha) = \text{Const.}$
Equation 2.11 is the general form for $E^2_{cm}$ after truncating $m$-terms

**Special Case:**

If $\{\phi_j\}$ is a set of eigen vectors that satisfy

$$A \phi_j = \lambda_j \phi_j$$

i.e.

$$\{\phi_i\} = \{\varphi_j\}$$

for all $j$'s, then equation 2.11 reduces to

$$E^2_{cm} = \sum_{j=m+1}^{n} \phi_j^T \lambda_j \phi_j = \sum_{j=m+1}^{n} \lambda_j$$

(2.12)

Define

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_4 \end{bmatrix}$$

where $\Lambda_1$ is an $m \times m$ diagonal submatrix

and $\Lambda_4 = n-m \times n-m$ diagonal submatrix

Equation 2.12 may be written as

$$E^2_{kl}(cm) = \text{tr} (\Lambda_4)$$

**General Case**

For the orthonormal set $\{\phi_i\}$, equation 2.11 may be reduced as follows

$$E^2_{cm} = \sum \phi_i^T R \phi_i$$

$$= \sum \phi_i^T (R \phi_i^T) \phi_i$$
\[
\begin{align*}
&= \sum (\phi_i^T \phi_i) \wedge (\phi_i^T \phi_i) \\
&= \sum (\phi_i^T \phi_i) \wedge (\phi_i^T \phi_i)^T \\
&= \sum \psi_i \wedge \psi_i^T
\end{align*}
\]

where
\[
\psi_i = \phi_i^T \phi_i
\]
The term \(\psi_i \wedge \psi_i^T\) may be written as
\[
\begin{bmatrix}
\psi_1 & \psi_2 \\
\psi_3 & \psi_4
\end{bmatrix}
\begin{bmatrix}
\Lambda_1 & 0 \\
0 & \Lambda_4
\end{bmatrix}
\begin{bmatrix}
\psi_1^T & \psi_3^T \\
\psi_2^T & \psi_4^T
\end{bmatrix}
\]
\[
\begin{bmatrix}
\psi_1 \wedge \psi_1^T + \psi_2 \wedge \psi_2^T & \cdots \\
\cdots & \psi_3 \wedge \psi_3^T + \psi_4 \wedge \psi_4^T
\end{bmatrix}
\]

then
\[
E^{-2}c_m) = tr (\psi_3 \wedge \psi_3^T + \psi_4 \wedge \psi_4^T)
\]

Since
\[
\psi_i \psi_i^T = (\phi_i^T \phi_i)(\phi_i^T \phi_i)^T = (\phi_i^T \phi_i)^T = I
\]
then \(\{\psi_i\}\) is orthonormal

or
\[
\psi_i \psi_i^T = \psi_i^T \psi_i = I
\]

Starting from the product
\[
\begin{bmatrix}
\psi_1 & \psi_2 \\
\psi_3 & \psi_4
\end{bmatrix}
\begin{bmatrix}
\psi_1^T & \psi_3^T \\
\psi_2^T & \psi_4^T
\end{bmatrix}
\begin{bmatrix}
\psi_1 \wedge \psi_1^T + \psi_2 \wedge \psi_2^T & \cdots \\
\cdots & \psi_3 \wedge \psi_3^T + \psi_4 \wedge \psi_4^T
\end{bmatrix} = I
\]
then
\[ \psi_1 \psi_1^T + \psi_2 \psi_2^T = I \]
\[ \psi_3 \psi_3^T + \psi_4 \psi_4^T = I \]

Similarly
\[ \psi_1^T \psi_1 + \psi_3^T \psi_3 = I \]
\[ \psi_2^T \psi_2 + \psi_4^T \psi_4 = I \]  \hspace{1cm} (2.16)

Using equations 2.16 in 2.15 yields
\[
\bar{E}^2(m) = tr (\lambda_1 \psi_3^T \psi_3 + \lambda_4 \psi_4^T \psi_4)
\]
\[
= tr (\lambda_1 \psi_3^T \psi_3 + \lambda_4 - \lambda_4 \psi_2^T \psi_2)
\]
\[
= tr \lambda_4 + tr (\lambda_1 \psi_3^T \psi_3) - tr (\lambda_4 \psi_3^T \psi_3)
\]

Since $tr(AB) \leq trA trB$ and $tr(CB) = tr(BA)$

then
\[
\bar{E}^2(m) = tr \lambda_4 + tr (\lambda_1 \psi_3^T \psi_3) - tr (\lambda_4 \psi_3^T \psi_3)
\]
\[
\geq tr \lambda_4 + tr (\lambda_1 \psi_3^T \psi_3) - tr \lambda_4 tr (\psi_2^T \psi_2)
\]
\[
= tr \lambda_4 + tr \lambda_1 tr \lambda_4 tr \lambda_4 tr (\psi_1^T \psi_1) - tr \lambda_4 tr (\psi_1^T \psi_1)
\]

or
\[
\bar{E}^2(m) \geq tr \lambda_4 + (tr \lambda_1 tr \lambda_4) + tr \lambda_4 tr (\psi_1^T \psi_1 - tr \lambda_4 tr (\psi_1^T \psi_1)
\]
\[
= tr \lambda_4 + (tr \lambda_1 tr \lambda_4) tr (\psi_1^T \psi_1)
\]

Since $\psi_1^T \psi_1$ is positive, and $tr \lambda_1 \geq tr \lambda_4$ (i.e. the eigenvalues are ordered in a descending order) then
\[
\bar{E}^2(m) \geq tr \lambda_4
\]
2.2.4 The K-L as a Filter

A combined effect of the properties described above is a striking characteristic of the K-L as a filter for certain types of stochastic processes. In computing the basis functions from the covariance matrix, the individual waveforms are aligned apriori in the time domain, creating essentially a set of non-stationary processes, for which the expansion is most efficient. As a result, all non-coherent data (noise) form a set of stationary processes. This noise gets represented by the lowest magnitude eigen values and functions. By truncating the expansion, noise is eliminated. This is not the case with DFT for one type of noise encountered in ECG/VCG recordings, mainly muscle tremer noise. It is straightforward to filter out power line noise in the frequency domain using Fourier expansion. This is not the case with muscle tremer since its frequency spectrum overlaps the spectrum of the data. This will be experimentally demonstrated in chapter 5.

2.3 A Constraint

The mathematical formulation of a sub-optimal feature extractor has been presented. The structure of any optimization problem is data-dependent. In all cases, a trade-off is introduced. Perhaps the major limitation of the K-L is that the expansion is a compromise in which
representation accuracy in specific local segments of the waveform is compromised for ensemble-global or waveform-global minimum error considerations. In other words, to detect subtle changes in a limited segment of a waveform, a large number of expansion coefficients may be needed, and the approach would be very inefficient since a much better feature extractor—in this case—could be devised by taking as the single feature the value of the waveform at the segment location in question [28].

Whether the K-L is the most suitable feature extractor to devise a technique for serial comparative analysis—based on the K-L features for highly localized subtle changes needs further analysis from both the information-theoretic and the data base structure considerations. This is beyond the scope of this work.
III. AN APPROACH TO CLASSIFIER DESIGN

3.1 Introduction

The pattern recognition problem is classically accomplished in two sequential procedures: 1. Feature extraction, 2. Classification. Attention is devoted in this chapter to classifier design.

Several techniques—parametric and nonparametric—have been developed in the theoretical literature [24,27] that vary in complexity, depending on whether or not the distribution of the random pattern vectors to be classified is known (or can be estimated). Theoreticians argue that the real solution to the statistical pattern recognition problem lies in the nonparametric (or distribution-free) statistical methods because assumption of parameter statistics is often unjustified [27]. Parametric techniques, however, are simpler to implement.

The theoretical approach to design a statistical classifier is to formulate an optimum decision rule. This requires apriori analysis of the data structure in the feature space, \( \mathcal{X} \), to seek the best partitioning of the space into classes by either linear or nonlinear decision boundaries. The optimum boundaries (or discriminant functions) are based on Bayes Decision Theory. The Bayes classifier is an optimal approach in that it minimizes the probability of classification error. For a two-class feature space, the technique provides a decision rule for a Yes/No decision (is the pattern vector normal or abnormal?) by comparing the a posteriori probability \( P(\mathcal{X}_1|x) \) that vector \( x \) belongs to class \( \mathcal{X}_1 \) to the similar probability \( P(\mathcal{X}_2|x) \) that it belongs to class \( \mathcal{X}_2 \), or
From Bayes rule
\[
\frac{p(\xi_1|\mathbf{x})}{p(\xi_2|\mathbf{x})} = \frac{p(\mathbf{x}|\xi_1)p(\xi_1)}{p(\mathbf{x})},
\]
\[
p(\mathbf{x}|\xi_i) = \frac{p(\mathbf{x}|\xi_1)p(\xi_1)}{p(\mathbf{x})},
\]
Where
\[
p(\mathbf{x}|\xi_i) = \text{conditional probability function of } \mathbf{x}
\]
\[
= \text{probability that } \mathbf{x} \text{ is assigned to } \xi_i \text{ while } \xi_i
\]
is the true class,
\[
p(\xi_i) = \text{apriori probability of } \xi_i
\]
\[
p(\mathbf{x}) = \text{apriori probability of } \mathbf{x}.
\]
Once the conditional probability density \( p(\mathbf{x}|\xi_i) \) for the pattern vectors distribution is known (or estimate thereof), the classifier is completely defined for the 2-class problem. The technique may be extended for multiclass feature space, and optimum discrimination is obtained by using a multivariate Gaussian distribution for \( p(\mathbf{x}|\xi_i) \) of the following form
\[
p(\mathbf{x}|\xi_i) \propto \exp \left[ -\frac{1}{2} (\mathbf{x}-\mathbf{m}_j)^T \mathbf{R}_j^{-1} (\mathbf{x}-\mathbf{m}_j) \right]
\]
where
\[
\mathbf{m}_j, \mathbf{R}_j = \text{mean and covariance of the } j^\text{th} \text{ class}
\]
Estimates of the class parameters \( \mathbf{m}_j, \mathbf{R}_j \) are based on the statistics of the data, and hence the approach is called "supervised learning" [24].

Several studies used the Bayes classifier to distinguish between normals/abnormals (screening) [6]. The \( p(\mathbf{x}|\xi_i) \)'s were estimated from the database using a procedure suggested by Specht [39]. The overall accuracy was 80%.
Clustering techniques have been successfully used for class parameters estimation, and the screening accuracy has been improved over that of [6] using only two K-L coefficients as features [13].

As a mathematical procedure, clustering is, in general, made up of a criterion and an algorithm. When the criterion is defined in terms of certain statistical parameters, it is termed "parametric clustering". Several criteria have been formulated in terms of performance indices, such as within-class and between-class scatter measure (scatter matrix), between-class distance (Battacharyya's distance), and total within-class square error [13]. The latter has been found to be suitable for the medical diagnosis problem [35]. The algorithm is then the optimal technique that groups samples to maximize class separability and hence minimizes probability of misclassification. The algorithm may be supervised in a sense that initial seeding is introduced such that the final classes represent on a one-to-one basis clinically accepted diagnostic classes. Two procedures are of interest here: 1. The hard ISODATA, 2. Fuzzy clustering.

In the next two sections, a review of both procedures as reported in the theoretical literature is presented. An innovative extension of the fuzzy clustering algorithm is introduced, and its applicability to reduction of class overlapping and characterization of multiple-class VCG's (tracings with mixed diagnosis) is discussed.

3.2 The Hard Partitioning Algorithm—ISODATA

The criterion for this algorithm is

$$\min J = \min \left( \sum_{j=1}^{C} \sum_{x \in \delta_j} || x - m_j ||^2 \right)$$

1 Iterative Self-Organizing Data Analysis Technique A
where

\[
C = \# \text{ of classes}
\]

\[
m_j = j^{th} \text{ class center}
\]

\(m_j\) is given by

\[
m_j = \frac{1}{N_j} \sum_{x \in \gamma_j} x
\]

The measure \(\|x - m_j\|\) is the Euclidean distance between \(x\) and \(m_j\) (serves as a measure of their similarity). ISODATA is a clusters seeking algorithm using successive iterations [36]. It involves initial seeding of clusters centers, iterative computation thereof, clusters splitting and merging based on predefined parameters, and minimization of the error index in each iteration. As the algorithm seeks to minimize it adds on a fairly comprehensive set of heuristic procedures. Since it is the basic of fuzzy IOSDATA, it will be summarized here. The following outlines its steps and the associated heuristic conditions:

1. Select initial cluster seeds \(\{m_j\}; 1 \leq j \leq C\) for the cluster set \(\{\gamma_j\}\) in the feature space

2. Assign the \(N\)-samples in the features space to \(C\) clusters using

\[x \in \gamma_j \quad \text{if} \quad \|x - m_j\| \leq \|x - m_l\|, \quad l \neq j ; \quad 1 \leq l \leq N\]

3. Eliminate clusters with fewer than a predefined minimum and update the value of \(C\) accordingly (heuristic)

4. Update each cluster center \(m_j\) by setting

\[
m_j = \frac{\sum_j w_j(x) \cdot x}{\sum_j w_j(x)} , \quad x \in \gamma_j , \quad 1 \leq j \leq C
\]

where \(w_j(x)\) is known as the membership function

\[
\sum_j w_j(x) = 1
\]

(3.3)
\[ W_j(x) = 0 \quad \text{or} \quad 1 \]

i.e., \( x \) is one and only one class. Equation 3.2 reduces to

\[ m_j = \frac{1}{N_j} \sum_{x \in Y_j} x \quad \text{(3.4)} \]

5. Compute the average Euclidean distance of samples in cluster \( Y_j \)
from the corresponding center \( m_j \) by

\[ \bar{d}_j = \frac{1}{N_j} \sum_{x \in Y_j} \| x - m_j \| \]

\[ = \frac{1}{N_j} \sum_{x \in Y_j} \| x - x_j \| \quad \text{(3.5)} \]

6. Average all \( d_j \) over all clusters

\[ d = \frac{1}{N} \sum_{j=1}^{C} N_j \bar{d}_j \quad \text{(3.6)} \]

where

\[ N = \sum_j N_j \]

7. Decision point...! (heuristic)

a. If this is the last iteration, set threshold for minimum between class distance to zero, then go to step 9

b. If \( C \geq (\text{the number of desired clusters})/2 \), go to step 8

c. If this is an even numbered iteration, or if \( C \geq \text{twice the number of desired clusters} \), then go to step 9

d. Otherwise continue

8. Compute the maximum component of the standard deviation vector \( \sigma_j \)
for each cluster \( Y_j \), where

\[ \sigma_j = [\sigma_{j1} \sigma_{j2} \ldots \sigma_{jn_j}] \]

\[ \sigma_{j}^{\max} = \max \left[ \frac{1}{N_j} \sum_k (x_{ik} - m_{ij})^2 \right]^{1/2} \quad \text{(3.7)} \]

\( n = \text{dimension of pattern vector} \)
\( x_{ik} \) = \( i \) th component of the \( k \) th pattern vector in \( \mathbf{X}_j \)
\( m_{ij} \) = \( i \) th component of the center of the \( \mathbf{X}_j \) class

If for any cluster \( \mathbf{X}_j \), \( \sigma_j \not\leq \) a pre-defined threshold (heuristic), and

a. \( \overline{d}_j > \overline{d}_{j_{\text{max}}} \) and \( N_j \geq 2 \) (minimum desired samples/cluster+1)
or

b. \( C \leq \) (number of desired clusters/2)

then split the cluster, then go to step 2

c. Otherwise continue

9. Compute the between-class Euclidean distance \( d_{ij} \)

where

\[
d_{ij} = \| m_{l} - m_{j} \| ; \quad 1 \leq l \leq C-1, l+1 \leq j \leq C
\]

(3.8)

Merge those cluster pairs whose distances are less than a pre-defined threshold (heuristic).

10. If this is the last iteration, stop, otherwise go to step 2.

3.3 Fuzzy ISODATA with Class-Dependent Fuzziness

3.3.1 Introduction

Close examination of equations 3.2 and 3.4 shows that assigning a 0 or 1 to the membership functions constitutes a constraint that clearly underlies the hard clustering algorithm. It implies that each pattern vector belongs to one and only one cluster. This is unrealistic in dealing with the medical diagnosis process in general and electrocardiographic interpretation in particular. Boundaries between "distinct" medical diagnostic classes are not "sharply" defined. The implication is that some class boundaries are not hard but rather are fuzzy [35]. The problem may
be traced to any of the following factors (or a combination thereof):

1. Rather than asking whether or not a patient belongs to a particular class, it is more appropriate to ask: "how much of this class has he..?"

2. Relatively large number of sample cases in a sick population may have the characteristics of several classes (for example RBB w/RAD or ST w/T). It is therefore more natural to assign to each case a set of probabilities or membership functions.

3. Limited specificity of the electrocardiogram is perhaps the most important factor after all. Clinically, there is inherent uncertainty in labelling certain categories of classes (for example LVH) since the ECG provides information on the electrical processes in the heart, and information as to the anatomical, physiological, and biochemical processes is generated by nonelectrocardiographic means (for example, the echocardiogram is more informative). The limited availability of a "complete" set of these clinical "cofactors" adds to the misclassification error, and training any proper classifier with certain categories of cardiograms may artificially create loosely defined boundaries in the feature space [35].

Algorithms for fuzzy clustering have been developed by several researchers [37,38], and several classification schemes exist for interpreting EEG patterns [33] and other medical diagnosis problems [34]. The algorithm proposed by Gustafson [35] is being extended in the following section by introducing the concept of class-dependent fuzziness. Fixed fuzziness, as proposed in [35], assumes that all clusters are equally loosely bounded and close. Results obtained by the Author do not justify the assumption of fixed or equal fuzziness.
3.3.2 Mathematical Development

Equation 3.3 may be written as

\[ 0 \leq w_j(x) \leq 1 \quad \forall x_i \in \mathcal{S}_j, \quad 1 \leq i \leq N, \quad 1 \leq j \leq C \]  

(3.9)

so as to minimize

\[ J = \sum_i \sum_j w_{ij} \alpha_j d_{ij} \quad ; \quad \alpha_j \geq 1 \]  

(3.10)

with the constraint

\[ \sum w_{ij}(x) = 1 \quad \forall x_i \in \mathcal{S} \]  

(3.11)

\( d_{ij} \) is a specialized norm distance metric expressed as

\[ d_{ij} = d(x_i, m_j) \]

where

\[ m_j = \text{\( j \) th cluster center} \]

\[ \alpha_j = \text{fuzziness parameter for the \( j \) th cluster} \]

Optimization of 3.10 with constraint 3.11 is a classically-known problem in Variational Calculus. By properly defining \( d_{ij} \) and preselecting a set of \( \alpha_j \)'s, the optimization problem is completely formulated.

Define

\[ d_{ij}(m_j) = (x_i - m_j)^T M_j (x_i - m_j) \]  

(3.12)

where \( M_j \) is symmetric positive-definite matrix with

\[ |M_j| = \text{constant}. \]

The first part of the optimization problem is to find the optimal \( m_j, M_j \).

Gustafson solved the problem for fixed-\( \alpha_j \) and obtained the following results

\[ m_{j,\text{opt}} = \frac{\sum_i w_{ij} \alpha x_i}{\sum_i w_{ij}} = m_f \]  

(3.13)
where

\[ m_{fj} = \text{fuzzy mean of class} \]
\[ n = \text{dimension of the feature space} \]
\[ P_{fj} = \text{fuzzy covariance matrix for the } j \text{ th class, defined as} \]
\[
P_{fj} = \frac{\sum_{i} w_{ij}^{\alpha} (x_i - m_{fj}) (x_i - m_{fj})^T}{\sum_{i} w_{ij}^{\alpha}} \tag{3.15} \]

The second part of the optimization problem is to find the optimal membership functions \( \{w_{ij}\} \) using a class-dependent set of fuzziness parameters \( \{\alpha_j\} \). This new concept for this portion of the optimization problem may be incorporated as follows:

Equation 3.11 may be written as

\[ J = \sum_i J_i(w) \]

where

\[ J_i(w) = \sum_j w_{ij}^{\alpha} d_{ij} \]

To find the extremum of \( J_i(w) \) with the constraint in 3.12, define the functional (in matrix form)

\[ J_i(w) = \sum_j w_{ij}^{\alpha} d_{ij} + \lambda_i \sum_j w_{ij} \]

where \( \lambda_i \) is a Lagrangian multiplier.

For a minimum

\[ \frac{\partial J_i(w)}{\partial w_{ij}} = 0 \]

and
\[
\frac{\partial w_{ij}}{\partial w_{ij}} > 0
\]

then
\[
\alpha_j w_{ij_{opt}}^{\alpha_j-1} d_{ij} + \lambda_i = 0
\]

or
\[
w_{ij_{opt}} = \left( \frac{\lambda_i_{opt}}{\alpha_j d_{ij}} \right)^{\frac{1}{\alpha_j-1}} \tag{3.16}
\]

Similarly
\[
\frac{\partial^2 J_i(w)}{\partial w_{ij}^2} = \alpha_j (\alpha_j-1) w_{ij}^{\alpha_j-2}
\]

If
\[
d_{ij} \geq 0 ; \quad w_{ij} > 0
\]

and for
\[
\frac{\partial^2 J_i(w)}{\partial w_{ij}^2} \quad \text{to be positive}
\]

then \( \alpha_j \) must be 1

Using 3.16 in 3.11 yields
\[
\sum_j \left( \frac{-\lambda_i_{opt}}{\alpha_j d_{ij}} \right)^{\frac{1}{\alpha_j-1}} = 1 \tag{3.17}
\]

By solving equation 3.17 for \( \lambda_i_{opt} \) and substituting the result in 3.16 yield the desired expression for \( w_{ij_{opt}} \). Although there is no closed-form solution for 3.17, it may be solved numerically. It is of interest, however, to consider the following special case:

Define
\[
\alpha_j = \alpha_1 \text{-fixed for the subset } S
\]
\[
\alpha_2 \text{-fixed for the remaining } C-S \text{ classes}
\]
Where

\( C \) = the total number of classes in the feature space

The first subset may be the ensemble of the well-separated classes, while the second subset includes the closely-separated and overlapping ones.

Further, if \( \alpha_1 \) is chosen to be equal 2 and \( \alpha_2 \) to be equal 3, a closed-form solution is reachable, as follows:

Equation 3.17 reduces to

\[
\left( -\frac{\lambda_i}{2} \right) \sum_{j=1}^{S} \frac{1}{d_{ij}} + \left( -\frac{\lambda_i}{3} \right) \sum_{j=S+1}^{C} \left( \frac{1}{d_{ij}} \right)^{\frac{1}{2}} = 1
\]

This equation is in the form

\[
a Z + b Z^{\frac{1}{2}} = 1
\]

and its solution may be written as (after simple manipulations)

\[
Z_{opt} = \left( -\lambda_{i, opt} \right) = \frac{1}{a} + \frac{b}{2a^2} \left( b - (b^2 + 4a)^{\frac{1}{2}} \right)
\]

where

\[
a = \sum_{l=1}^{S} \frac{1}{2d_{il}}
\]

and

\[
b = \sum_{l=S+1}^{C} \left( \frac{1}{3d_{il}} \right)^{\frac{1}{2}}
\]

Hence

\[
W_{ij, opt} = \frac{1}{2d_{ij}} \left( \frac{1}{a} + \frac{b}{2a^2} \left( b - (b^2 + 4a)^{\frac{1}{2}} \right) \right); 1 \leq j \leq S
\]

\[
= \frac{1}{3d_{ij}} \left( -\frac{b}{2a} + \frac{1}{2a} \left( b^2 + 4a \right)^{\frac{1}{2}} \right); S+1 \leq j \leq C
\]

(3.18)

The following iterative algorithm has been suggested by Gustafson [35]. It is being generalized and implemented in this work to include cluster-dependent fuzziness.

1. Given samples of pattern vectors \( \{X_i\} \), initial partitions \( \{m_{ij}^{0}, P_{ij}^{0}\} \) and fuzziness parameters \( \alpha_1 = 2 \), \( \alpha_2 = 3 \)

2. Compute \( \{d_{ij}(m_{ij}^{(k)}, P_{ij}^{(k)})\} \) using 3.12, where \( k \) is the order of
iteration

3. Compute \( \{ W_{ij}^{(k)} \} \) using 3.18. If \( d_{ij} = 0 \), set \( W_{ij} = 1 \) and set \( W_{il} = 0 \) \( \forall \ l \neq j \).

4. Compute new partitions \( \{ m_{ij}^{(k+1)} \}, \{ \alpha_{ij}^{(k+1)} \} \) using 3.13, 3.14.

5. Go to step 2 for the next iteration. Stop if certain convergence criterion is reached (heuristic).

3.4 Concluding Remarks

In applying "supervised" classification techniques to the multi-class problem, a need was identified for developing a parametric clusterer based on a generalized fuzzy clustering algorithm with class-dependent fuzziness. By developing the classification procedure, the second half of the pattern recognition problem is completely formulated and solved. The underlying philosophy in this statistical approach is not only to train the algorithms using an ensemble of VCG's, but to investigate different strategies for structuring the ensemble or "database", in order to optimize the feature extractor and to reduce the complexity of the features distribution in the feature space. In the next chapter, the database will be studied.
IV. THE DATABASE

4.1 Introduction

The task of developing a pre-diagnosed training set for the supervised clustering algorithm is not an easy one. In pursuit of a fully automated system, one must adopt authoritative standards for medical diagnosis—the so called "Gold" standards—to ultimately achieve the minimal misclassification error using statistical pattern recognition techniques. Ironically, significant variability in electrocardiographic interpretation exist between physicians. This is due to differences in measuring techniques (e.g. onset and offset points of deflection in a record...), use of imprecise terminology in some cases (e.g. wandering pacemaker, hemiblock, extrasystole...etc.), and selection of diagnostic criteria. In the final analysis, defining a complete set of valid diagnostic criteria is the most critical of all. Although it is possible to validate the criteria in certain types of interpretations (type -A\textsuperscript{1} cardiograms), a number of uncertainties exist. These include physiologic variations, both within and between recordings, variations due to equipment technique, variations due to age, sex, and other similar factors, and statistical variations between population subsets. The task of finding and validating an optimal set of diagnostic criteria is beyond the scope of this work. A "fixed" set of diagnostic classes has been compiled and defined by USAFSAM Cardiologists, and is being

---

\textsuperscript{1} Data whose interpretation may be validated against confirmatory non-electrocardiographic data (or cofactors). Examples are anatomic or pathologic abnormalities.
presented here (see table 4.1). These classes represent a set of relatively pure abnormalities (Single-disease states) except for classes 5, 6, 29, and 99. The latter include all cases of mixed abnormalities (or multiple-disease states).

4.2 Preprocessing

Tracings of raw data are composite waveforms of the original signal, baseline fluctuation, and noise. Techniques for signal extraction have been extensively investigated in the literature [14,17]. The next two sections outline the approach used in this work using existing algorithms with innovative modifications.

4.2.1 Baseline Restoration

Underlying baseline variations in a VCG record are generally attributed to respiration and electrode-skin contact potential resistance modulation. Some techniques have tried to correct baseline drifting by high pass filtering, but this approach influences diagnostically-sensitive low frequency components in the ST-segment and U-wave of the electrocardiogram. Other techniques use estimator polynomials of various degrees for best fit. The key to baseline estimation via curve fitting is to preidentify zero-reference points (or knots) through which the estimator must pass. Meyer [14] developed a technique for identifying the isoelectric points in the PR segment of each beat and used them for zero-reference. He then developed a crude algorithm for baseline restoration based on third-order polynomials (cubic-spline) with stable solution. In applying the technique to the VCG database, it was
Table 4.1 Diagnostic Classes

<table>
<thead>
<tr>
<th>NOTE</th>
<th>CLASS#</th>
<th>DESCRIPTION</th>
<th>ABBREV.</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Normal</td>
<td></td>
<td>Norm</td>
</tr>
<tr>
<td>02</td>
<td>Atrioventricular Block (1st, 2nd, 3rd degree</td>
<td>AVB</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Pre-excitation short PR with normal or abnormal QRS complexes</td>
<td>PREE</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Right Bundle Branch Block</td>
<td>RBBB</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>RBBB with Left Axis Deviation</td>
<td>RBBL</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>RBBB with Right Axis Deviation</td>
<td>RBBR</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Left Bundle Branch Block</td>
<td>LBBB</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>Terminal (40msec) Intraventricular Conduction Delay</td>
<td>IVCD</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>LAD (-30°) Left Axis Deviation</td>
<td>LAD</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>RAD (+100°) Right Axis Deviation</td>
<td>RAD</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>S1 S2 S3 S1, 2, 3 Pattern</td>
<td>S123</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>T wave abnormalities (T wave amplitude 0.2 mv in frontal plane, wide QRST angle, and abnormal L/W ratio)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ST segment abnormalities (Straightening or depression)</td>
<td>ST</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Early repolarization (ST +0.10 mv in frontal plane)</td>
<td>REP</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Left ventricular Hypertrophy (with/without repolarization abnormalities but QRS amplitudes: 2.0 mv in FP &amp; TP)</td>
<td>LVH</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Right ventricular Hypertrophy</td>
<td>RVH</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ANT/ANTSEPTAL MI Anterior/Anteseptal Myocardial Infarction</td>
<td>ANT</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Inferior MI Inferior Myocardial Infarction</td>
<td>INMI</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Posterior MI Posterior Myocardial Infarction</td>
<td>POMI</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Atrial Axis Abnormality</td>
<td>AAXA</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Left Atrial abnormality</td>
<td>LAA</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Right Atrial Abnormality</td>
<td>RAA</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Increased ANT forces (R/S 1 in V1 &amp; V2 and or ANT forces 0.5 mv</td>
<td>IANT</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Decrease ANT forces (poor R-wave progression including total time anteriorly less than 30 msec</td>
<td>DANT</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Increased initial forces superiorly (25 msec duration</td>
<td>IIFS</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Septal Hypertrophy (VCG codes 740, 471, &amp; 728 (Q 0.3 mv in V5</td>
<td>SHYP</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Biventricular Hypertrophy</td>
<td>BHYP</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>ST and T wave changes</td>
<td>STT</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Probably normal - use for test set (pure-not associated with a pure normal class (mixed or partial)</td>
<td>STT</td>
<td></td>
</tr>
</tbody>
</table>

(1) Pure (2) Mixed
noticed that the algorithm yields successive segments of overestimation and underestimation and PR-segment morphology changes, especially under high R-R variability conditions and large baseline fluctuations. The technique is reformulated in this work to correct this problem. This is accomplished by fitting an exact cubic spline to each successive four knots, with derived updates for all derivative at each knot as will be explained later. This is in direct contrast to the approach in [14], whereby only the second and third derivatives are being updated and the first derivative being approximated and not computed from the spline. Only the first segment of the spline is kept, and the procedure is repeated across the tracing. In addition, a test for spike-type and PVC outliers is applied to the data prior to baseline restoration. A complete mathematical derivation of the baseline estimator is presented here with particular emphasis on deriving the constraints at each knot.

Let \( Y(t) = Y_0 + A_1 t + A_2 t^2 + A_3 t^3 \) (4.1) be the estimator through the knots \((Y_0, T_0), (Y_1, T_1), (Y_2, T_2)\) (knot-finding algorithm is described elsewhere [14]).

Substituting the knots coordinates in 4.1 yields

\[
\begin{bmatrix}
Y_1 - Y_0 \\
Y_2 - Y_0 \\
Y_3 - Y_0
\end{bmatrix} =
\begin{bmatrix}
T_1 & T_1^2 & T_1^3 \\
T_2 & T_2^2 & T_2^3 \\
T_3 & T_3^2 & T_3^3
\end{bmatrix}
\begin{bmatrix}
A_1 \\
A_2 \\
A_3
\end{bmatrix}
\]

or in the matrix form

\[
Y = T \cdot A
\] (4.2)

These points are placed at least 23 msec prior to the R-wave fiducial point and within the PR interval.
Figure 4.1 Tracing with Baseline Shift-Knots Locations
\[ A = T^{-1} \cdot Y \]

\( Y^{-1} \) may be written as

\[
T^{-1} = \begin{bmatrix}
T_{11} & T_{12} & T_{13} \\
T_{21} & T_{22} & T_{23} \\
T_{31} & T_{32} & T_{33}
\end{bmatrix}
\]

then

\[
A_1 = T_{11} (Y_1 - Y_0) + T_{12} (Y_2 - Y_0) + T_{13} (Y_3 - Y_0)
\]

\[
A_2 = T_{21} (Y_1 - Y_0) + T_{22} (Y_2 - Y_0) + T_{23} (Y_3 - Y_0)
\]

\[
A_3 = T_{31} (Y_1 - Y_0) + T_{32} (Y_2 - Y_0) + T_{33} (Y_3 - Y_0)
\]

Evaluation of the first derivative of 4.1 at \( t=T1 \) and \( t=0 \) yields

\[
\gamma'(t_i) = (T_{11} + 2T_1T_{21} + 3T_1^2T_{31})(Y_1 - Y_0)
\]

\[
+ (T_{12} + 2T_1T_{22} + 3T_1^2T_{32})(Y_2 - Y_0)
\]

\[
+ (T_{13} + 2T_1T_{23} + 3T_1^2T_{33})(Y_3 - Y_0)
\]

(4.3)

\[
\gamma'(0) = T_{11} (Y_1 - Y_0)
\]

\[
+ T_{12} (Y_2 - Y_0)
\]

\[
+ T_{13} (Y_3 - Y_0)
\]

(4.4)

Denote \( \gamma'(0) = \gamma_0' \) and \( \gamma'(T_i) = \gamma'_i \)

Equation 4.1 may be written as

\[
Y_i = \gamma'''(0) T_i^3 / 6 + \gamma''(0) T_i^2 / 2 + \gamma'(0) T_i + Y_0
\]

(4.5)

then

\[
\gamma'_1 = \gamma'''(0) T_i^2 / 2 + \gamma''(0) T_i + \gamma'(0)
\]

(4.6)

Solving this pair of equations for \( \gamma''(0) \) and \( \gamma'''(0) \) yields

\[
\gamma''(0) = -6 (Y_0 - Y_i) / T_i^2 - 2 (2 Y_0 + \gamma'_1) / T_i
\]

\[
\gamma'''(0) = 12 (Y_0 - Y_i) / T_i^3 + 6 (Y_0 + \gamma'_1) / T_i^2
\]
The baseline estimate may then proceed as described by Meyer [14] using the following iteration formula

\[
\begin{bmatrix}
Y(n) \\
Y'(n) \\
Y''(n) \\
Y'''(n)
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & \frac{1}{2} & \frac{1}{6} \\
0 & 1 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
Y(n-1) \\
Y'(n-1) \\
Y''(n-1) \\
Y'''(n-1)
\end{bmatrix}
\]

(4.7)

where N= order of iteration.

Starting with N=1, equation 4.7 is iteratively applied for all values of \(Y(t)\) in the \([\phi, T,]\) interval, using the initial values \(Y(\phi)\) in 4.4, \(Y'(\phi)\) in 4.5 and \(Y''(\phi)\) in 4.6.

At each new PR-knot, new \(Y'(\phi)\) and \(Y''(\phi)\) are computed using 4.5, 4.6, keeping \(Y'(t)\) continuous at the knot using equation 4.3 (for each lead).

In detecting the knot location, the R-wave fiducial point location is first identified from the spatial wave velocity. The spline than passes through each and every knot unless an outlier is detected. An outlier is defined as a segment of the waveform that is grossly different in morphology from the rest of it. Mathematically, an outlier is detected by selecting the spatial vector velocity of the QRS of the first beat and subsequently computing the correlation coefficient, \(P\), with each successive QRS-segment. A knot is declared valid if the coefficient exceeds a certain threshold, i.e:

\[
P = \frac{\text{1st beat spatial vector velocity}}{\text{k th beat spatial vector velocity}} \geq \beta
\]

\[\beta = 0.9 \quad \text{(threshold)}\]

and where \(\text{1st beat spatial vector velocity}\)

\(\text{k th beat spatial vector velocity}\)

Proper selection of \(\beta\) will easily detect outliers such as PVC's and
spikes, experimental results using this technique are presented in Chapter 5.

4.2.2 Removal of Non-Coherent Noise

It was shown mathematically [32] that the K-L expansion reduces to the Discrete Fourier Transform for stationary purposes. The 60-HZ noise and similar non-coherent interference are examples of stationary processes added to the original signal. During the eigen vector transformation process using the K-L, only the vectors associated with the largest eigen values are selected; this has the effect of filtering. This is clearly demonstrated in Figure 4.2. Signal conditioning for other common noise factors, and signal baud-limiting is done at the analog front-end of the signal acquisition using a fourth order Butterworth filter with 100-HZ cut off. A/D conversion is limited to 500 samples/sec, 10-bit resolution as specified by the AHA in [18]. In addition, individual beats within a tracing are being cross correlated and those with unnormalized correlations coefficients above a certain threshold (.95 in this case) are averaged. Non-coherence data (with the signal) get zero-averaged using this technique.

4.3 Partitioning

It has been shown that the K-L expansion over the whole beat (time-global) and using the entire database (ensemble-global) has certain limitations that are attributable to the global optimality criterion [40]. The criterion implies that this type of feature extraction minimizes the ensemble-global error, and it is most efficient for
Figure 4.2 Noise Filtering Effect of the K-L
non-stationary processes. This is clearly a compromise in which accuracy of representation in specific segments of the waveform (local error) is traded off for the simplicity of the feature extraction. Figure 4.3 shows a raw beat and its reconstruction from a 20th order expansion (20/lead) with local reconstruction error in the P-segment.

Solution to the local error problem depends on the location of the error. This has been accomplished by segmentation of the waveform (time partitioning), the ensemble (ensemble-partitioning), the process (process partitioning), and a combination thereof, in a hierarchical feature extraction scheme.

4.3.1 Time Partitioning

The P-wave process originates from the atria, while the QRST-Complex originates from the ventricules (with slight overlapping during the QRS). The underlying differences in their respective morphology, amplitude, and timing distributions make it quite reasonable to treat them as two separate but loosely connected processes. In performing the K-L expansion, one tends to optimize the coefficients by aligning the data (beats) on their R-Wave fiducial points. An aligned beat is then a sample in an ensemble of high energy non-stationary processes (QRST's) and low energy quasi-stationary processes (P-waves). This yields an expansion (and hence pattern vectors) that is heavily weighted by the QRST and more efficient in representing it than the P-wave. Figure 4.3 depicts this phenomena. Akant investigated a modified K-L expansion scheme [28] whereby separate K-L expansion is computed for each process. Alignment of the P-wave was done by cross-correlating
Figure 4.3 The P-wave Reconstruction Problem

Figure 4.4 Akant's Template
a template of predefined width and shape (see Figure 4.4) with a back­ward moving window of equal width along the waveform spatial vector velocity, and starting at the PR-knot. A P-wave is declared present of the maximum unnormalized correlation coefficient, $p_{VT}$, exceeds a pre­defined threshold, and a slightly eccentric point within the window is the P-wave "fiducial point".

i.e.

$$P_{VT} = \frac{V}{T} \geq \beta_P$$

$v =$ spatial vector velocity of raw data prior to Pr-knot

$T =$ template

$\beta_P =$ threshold

Apparently the only criterion used in selecting the template was choos­ing it with a width equal to the average width of an ensemble of P-waves in the database. Clearly, the template morphology is not of adequate measure of real "likeness" or similarity to actual P-wave vector veloc­ities. Hence, the technique misses low amplitude P-waves, and mislabels U-waves and low amplitude T-waves as P-waves. To reduce the miss rate, the normalized Correlation Coefficient, $p_{VT_{norm}}$, was used where

$$P_{VT_{norm}} = \frac{\sqrt{V}}{\sqrt{\sigma_V \sigma_T}}$$

$\sigma_V =$ standard deviation of signal

$\sigma_T =$ standard deviation of template

It is proposed in this work to use a scheme of multiple-template match­ing using a set $\{T_j\}$ of P-wave velocity morphologies. Intensive research through the database indicated that most morphologies fall into at least one of three templates. Three mathematical functions fit those
templates, namely the symmetrical normal distribution, left-skewed, and right-skewed distribution functions, as shown in Figure 4.4. The templates are chosen not only to resemble actual data, but to have the same mean and standard deviation (in the amplitude sense). The latter is mathematically desirable to use the normalized correlation coefficient for detection. The detection criterion is

$$\max \ P_{VT \norm} = \max \left( \frac{\sqrt{T_j}}{\sigma \sqrt{T_j}} \right) \geq \beta \ p, \ 1 \leq j \leq s \ (4.9)$$

$s = \text{number of templates (heuristically chosen to be 3)}$.

A scheme based on 4.9 is depicted in Figure 4.5. This approach is essentially a modelling approach to the P-wave problem in terms of a set of parallel models, and as more models are learned from the database, additional ones may be included.

4.3.2 Ensemble Partitioning

The approximation power of the K-L relies on precise apriori estimates of class distribution probabilities $p(\xi_j)$ within the database. It is obvious that to obtain the most accurate representation of an ensemble of signal (global accuracy) using a fixed number of basic functions, one should have the set that span the space which contains on average the greatest proportion of the energy in the process. What matters here is the space spanned by the basis, not the basis itself. This implies that the covariance matrix, $R$, should be computed for an infinite-size database. Finiteness of the ensemble size in this study is expected to yield suboptimal results, since $p(\xi_j)$ is not known apriori. Two observations may be asserted here:
Figure 4.5 Mathematical Template Functions

\[ Y_L = Y_0 t e^{-\frac{1}{2}t} \]
\[ Y_S = \frac{Y_0}{\sqrt{2\pi}} e^ {-\frac{1}{2}(t-30)^2} \]
\[ Y_R = Y_0 \left(1-t e^{-\frac{1}{2}t}\right) \]
Figure 4.6 The Multi-template Detection Scheme
Figure 4.6 The Multi-template Detection Scheme
2. Compute the normalized correlation coefficient, $p_n$, between $V_m$ and the vector velocity, $V_m$ of each sample in the database, i.e.

$$p_n = \frac{V_m \cdot V_x}{\sigma_m \sigma_x}$$

for all $x$'s

where $\sigma_m$, $\sigma_x$ are the standard deviations of $V_m$, $X$ respectively.

3. Assign the sample to one partition if $p_n$ exceeds certain threshold, otherwise assign it to the other partition.

4.3.3 Process Partitioning

In working with finite-size database, investigators have shown that feature selection procedure renders itself statistically significant results if the number of samples per class in the training set is at least equal to the number of features needed [6,28,41]. As one proceeds with the clustering analysis in the feature space, certain assumptions are being made concerning the morphologies of the clusters themselves. A fundamental assumption here is that all classes are Gaussian and hyperelliptic. Investigation of the validity of this assumption is beyond the scope of his work, since a database of at least 5,000 cardiograms is needed to verify it (or deny it) [41]. Gustafson has found, however, that certain classes overlap severely [35]. For example, most repolarization abnormality classes do overlap with the class of normals. This may be attributed to:

1. Inaccuracies in the thresholds within the diagnostic criteria that the Cardiologist applies to generate the training set, and

2. The relatively low energy contents of the ST, T, portions compared to the QRS portion in the QRST process.
The first attribute is an ever-continuing effort in Cardiology to estab-
lish the "Gold" standards for diagnosis. It is the second attribute 
that is being investigated in this work. Optimally, one should segment 
the QRST process into two separate processes, similar to what is done 
to partition the waveform to P-and QRST, if a junction-knot for seg-
mentation can be identified. In the original time-partitioning proce-
dure, the PR-knot is relatively easy to identify since it signifies a 
transition between the end of the atrial depolarization and the onset of 
ventricular depolarization, with minimum activity (in the velocity 
sense). On the other hand, the QRST is a two phase process: a depolar-
ization phase (the QRS-complex) followed by a repolarization phase (the 
ST-segment and T-wave). The two phases are connected at what is classi-
cally termed the J-junction. Technical discussions with the USAFSAM 
Cardiologists clearly indicated that no robust criteria exist to detect 
such a junction, mainly because the repolarization phase may very well 
start early in the process within the time segment of the QRS complex.

A suboptimal approach is investigated in this work, whereby the 
repolarization abnormal subset of the ensemble is selected. The QRST in 
each sample of the subset is time-weighted prior to computing the covar-
iance matrix. The weighting function is chosen such that it emphasizes 
the ST and T portion, and it suppresses the QRS complex. (see Fig. 4.6).

![Figure 4.7 Weighting Function](image)
The following outlines the process partitioning scheme:

1. Align the beat with the weighting function, with the A point coincident with the R-wave fiducial points.

2. Cross multiply the beat with the weighting function only in the intervals.

3. Replace the segment of the resulting wave in the $A^1 A^2$ interval by a straight line segment with continuous Y values at $A^1$ and $A^2$.


4.4 The Overall System

A functional flow diagram is the overall partitioning/classification hierarchy is shown in Figure 4.7. Experimental results at different levels of the decision logic will be presented in the next chapter.

4.5 Concluding Remarks

The underlying strategies in this Chapter were to investigate different techniques of partitioning of the database for extracting the most efficient features. It has been established that a single "universal" set of the basis functions for all categories of VCG's may not be desirable. Rather, at least two sets of K-L basis functions should yield more efficient features for two commonly known populations, namely gross abnormals and repolarization abnormals. This implies hierarchy in designing the overall system. Although this is a tree-type structure, it presents a compromise between decision tree logic, and the open-ended probabilistic approach in [41] using a universal set of functions.
Figure 4.8 The Overall System Flow Structure
V. EXPERIMENTAL STUDIES

5.1 Introduction

Previous chapters have considered the analytical basis for the pattern recognition procedure. First, some information concerning the suboptimal characteristics of the K-L feature extractor and its limitations have been derived. Then, techniques for discriminant analysis in the feature space, with particular emphasis on the fuzzy clustering procedure and class-dependent fuzziness, have been discussed. Finally, the partitioning procedures for the finite-size database have been presented in order to seek the most efficient features using the K-L.

In section 5.2, numerical results on the preprocessing and partitioning of the database will be presented. This includes baseline restoration, time, ensemble, and process partitioning.

Since 5.3 presents sets of K-L basis vectors for each ensemble partition and randomly selected reconstruction from the original data (inverse K-L) using each set of the K-L. Approximation errors of data common to each ensemble partition will be compared.

In section 5.4, the fuzzy clustering algorithm with class-dependent fuzziness will be tested on a subset of prediagnosed classes of the ensemble (classes with large enough members to yield statistically significant discrimination results). Two strategies for mapping multiple-diagnosis cases are presented, using the concept of "membership" functions, to signify how "much" of one or more diseases each case has.
Finally, section 5.5 presents concluding remarks and suggestions for future efforts.

5.2 Preprocessing and Partitioning

An ensemble of 676 clinically pre-diagnosed VCG's has been recorded on 8-track, 800 bpi, IBM-compatible tapes. The data was previously digitized with 10-bit sample resolution, 500 samples/sec, 12 seconds/record, and 100 records/tape of simultaneous X,Y, and Z tracings. The signals were band-limited at acquisition time using a front-end analog Butterworth filter with 100-Hz cut-off frequency.

The first phase of preprocessing involves baseline restoration of all 3 leads using the technique discussed in Chapter 4. Figure 5.1 shows a three second segment of a single lead of the raw data, its vector velocity, the estimated baseline, and the restored data. The magnitude of the local residual variations in the restored record was found to be statistically less than 8 percent (amplitude-wise) in the ST and T segments of the waveform, and no significant change in the PR segment slopes and morphologies of the Q-wave.

The restored record is then reduced by averaging all successive beats that crosscorrelate at better than .95 (1-2 averages/beat). The average beats of all records are then positioned in a fixed 800 millisecond window, with all fiducial points coincident with an a priori fixed location in the window. The location within the window has been heuristically chosen such that 1/3 of the beat lies to the left of the fiducial point and the other 2/3 to its right. The averaged beats are further reduced by selecting every other sample within each beat, effectively reducing the sampling rate from 500 to 250 samples/sec and the dimensionality of
Figure 5.1 Original, Estimate, and Baseline-restored Tracing
each record from 400 to 200. The ensemble of the 200 samples/lead, 3 leads/beat, and 1-2 beats/patient files constitutes the database, later referred to as the "All-Ensemble". The second phase of preprocessing involves partitioning each beat into two separate processes, namely the P's and the QRST's. This is accomplished by applying the multi-template P-wave detection procedure discussed in Chapter 4. Figures 5.2 and 5.3 demonstrates the performance of the algorithm on randomly selected samples from the ensemble. Multi-template matching is performed on the vector velocity of a backward moving 200-millisecond window, starting at the PR-knot. Since the algorithm is very sensitive to any underlying baseline, pathologic or artifact (such as P on T, P on U, or residual baseline after cubic spline restoration), the base of each correlated segment is first restored to a zero reference by vertical subtraction of a straight line base from the segment (see sketch below).

The detection threshold is set at .85 (heuristic), and at least one of the templates captured each P-wave in the ensemble. S, L, R in figures 5.2 and 5.3 denote the maximum value of the corresponding template correlation coefficient, and, PRS, PRL, PRR denote the locations of the maxima. The detector also marked their corresponding locations.
Figure 5.3 Performance of the P-wave Detector (Continued)
Figure 5.2 Performance of the P-wave Detector
The locations of the detected P-waves are then saved in a computer disk file for use in alignment of the P-waves prior to computing their K–L basis vectors. It is worth-mentioning here that no digital filtering is performed on the raw data for noise suppression and removal of muscle artifact, since the truncated K–L is a filter in itself. Only the vector velocity is digitally filtered for the purpose for detecting the R-wave fiducial point.

The last phase of preprocessing involves both ensemble- and process-partitioning, as outlined in Chapter 4. The All-Ensemble is partitioned into two: the first partition includes all patients with gross abnormalities such as RBBB's, LBBB's, LVH's, while the other partition is the rest of the ensemble. The first will be referred to as the Gross-Abnormal, and the second as the All-But-Gross. The All-Ensemble partition remains intact, in addition to the newly generated ones. Next, each beat in the All-But-Gross partition is suppressed as explained in Chapter 4 (i.e., QRS removed), yielding the "QRS-Suppressed" partition. The objective here is to generate multiple sets of K–L basis vectors and to compare the efficiency of the feature extractor using the All-Ensemble set to its efficiency using the other sets.

5.3 The Basis Vectors

Each of the three partitions (All-Ensemble, Gross-Abnormal, and QRS-Suppressed) is composed of a dataset of 200-sample beats. In addition, the P-wave locations are used to assemble a fourth partition whose elements consist of 200-millisecond segments (or 50 samples) of the original beats, with each P-wave location coincident with the center
of the corresponding segment. The K-L is then performed on the resulting "P-Partition". Plots of the mean and the 5 principal K-L vectors are shown in figures 5.4-5.9 (5 vectors per lead).

Three sets of K-L basis vectors are then computed, one for each of the three ensemble partitions. The mean and the 8 principal K-L vectors are plotted in figures 5.10-5.18 for the All-Ensemble partition, in figures 5.19-5.27 for the QRS-Suppressed partition, and in figures 5.28-5.36 for the Gross-Abnormal partition. Cross and close examination of the three sets clearly shows how the characteristic morphologies of the components resemble the dominant stochastic processes in the partition. Specifically, the first component of the K-L set from the Gross-Abnormal partition carries more energy in the QRS complex, and the QRS-Suppressed set reflects more accurately the morphologies of the ST and T segments, when compared with the All-Ensemble set. Table 5.1 shows the cumulative energy (sum of principal eigen values of the covariance matrix) for lead X in each set. The All-Ensemble set captures 98% of the energy in the expansion using 14 coefficients, while the other two sets capture the same amount of energy with only 10-11 coefficients. To demonstrate further, the K-L expansion is performed on 3 cases of gross abnormalities, using the All-Ensemble set and the Gross-Abnormal set separately. The results are plotted in figures 5.37-5.41. Each plot is an overlay of the inverse K-L (or reconstruction) of each case on the original beat, using 5 K-L vectors for the P-wave's and 15 K-L vectors for the QRST's. Reconstruction errors for each beat are computed using a cost function (integrated square of the difference between the original and its reconstruction), and is printed with each plot using the Calcomp plotting
Figure 5.4 P-wave Basis Vectors—Lead X

Vertical: .6 mv/unit
Horizontal: 60 ms/unit
Figure 5.5 P-wave Basis Vectors—Lead X (Continued)

Vertical: .6 mv/unit
Horizontal: 60 ms/unit
Figure 5.6 P-wave Basis Vectors-Lead Y

Vertical .6 mV/unit
Horizontal: 60 ms/unit
Figure 5.7 P-wave Basis Vectors—Lead Y (Continued)
Figure 5.8 P-wave Basis Vectors—Lead Z

Vertical: .6 mv/unit
Horizontal: 60 ms/unit
Figure 5.9 P-wave Basis Vectors—Lead $\zeta$ (Continued)
Vertical .6 mv/unit
Horizontal: 240 ms/unit

Figure 5.10 The "All-Ensemble" Basis Vectors—Lead X
Vertical: .6 mv/unit
Horizontal: 240 ms/unit

Figure 5.11 The "All-Ensemble" Basis Vectors—Lead X (Continued)
Vertical: .6 mv/unit
Horizontal: 240 ms/unit

Figure 5.12 The "All-Ensemble" Basis Vectors—Lead X (Continued)
Figure 5.13 The "All-Ensemble" Basis Vectors—Lead Y

Vertical: .6 mv/unit
Horizontal: 240 ms/unit
Figure 5.14 The "All-Ensemble" Basis Vectors–Lead Y (Continued)
Figure 5.15 The "All-Ensemble" Basis Vectors—Lead Y (Continued)
Figure 5.16 The "All-Ensemble" Basis Vectors- Lead Z

Vertical: .6 mV/unit
Horizontal: 240 ms/unit
Figure 5.17 The "All-Ensemble" Basis Vectors—Lead Z (Continued)
Figure 5.18 The "All-Ensemble" Basis Vectors—Lead Z (Continued)
Vertical: .6 mv/unit
Horizontal: 240 ms/unit

Figure 5.19 The "QRS-Suppressed" Basis Vectors—Lead X
Figure 5.20 The "QRS-Suppressed" Basis Vectors—Lead X (Continued)
5.21 The "QRS-Suppressed" Basis Vectors—Lead X (Continued)

Figure 5.21 The "QRS-Suppressed" Basis Vectors—Lead X (Continued)
Figure 5.22 The "QRS-Suppressed" Basis Vectors—Lead Y

Vertical: .6 mv/unit
Horizontal: 240 ms/unit
Figure 5.23 The "QRS-Suppressed" Basis Vectors—Lead Y (Continued)
Figure 5.24 The "QRS-Suppressed" Basis Vectors—Lead Y (Continued)
Figure 5.25 The "QRS-Suppressed" Basis Vectors—Lead Z

Vertical : .6 mv/unit
Horizontal: 240 ms/unit
Vertical: .6 mv/unit
Horizontal: 240 ms/unit

Figure 5.26 The "QRS-Suppressed" Basis Vectors—Lead Z (Continued)
Figure 5.27 The "QRS-Suppressed" Basis Vectors—Lead Z (Continued)
Figure 5.28 The "Gross–Abnormal" Basis Vectors–Lead X
Figure 5.29 The "Gross-Abnormal" Basis Vectors—Lead X (Continued)
Figure 5.30 The "Gross-Abnormal" Basis Vectors- Lead X (Continued)
Figure 5.31 The "Gross-Abnormal" Basis Vectors—Lead Y

Vertical: .6 mv/unit
Horizontal: 240 ms/unit
Figure 5.32 The "Gross-Abnormal" Basis Vectors—Lead Y (Continued)
Figure 5.33 The "Gross-Abnormal" Basis Vectors - Lead Y (Continued)

Vertical: .6 mv/unit
Horizontal: 240 ms/unit
Figure 5.34 The "Gross-Abnormal" Basis Vectors—Lead Z
Figure 5.35 The "Gross-Abnormal" Basis Vectors—Lead Z (Continued)
Figure 5.36 The Gross-Abnormal" Basis Vectors- Lead Z (Continued)

Vertical: .6 mv/unit
Horizontal: 240 ms/unit
Table 5.1. Energy Fractions Of The K-L in Each Partition

Lead X

<table>
<thead>
<tr>
<th>Eigen Vector</th>
<th>Fraction Of Energy</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>All-Ensemble</td>
<td>Gross-Abnormal</td>
<td>QRS-Suppressed</td>
</tr>
<tr>
<td>20</td>
<td>0.474792E+00</td>
<td>0.479136E+00</td>
<td>0.428320E+00</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.648826E+00</td>
<td>0.739738E+00</td>
<td>0.680204E+00</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.762025E+00</td>
<td>0.847361E+00</td>
<td>0.853621E+00</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.821309E+00</td>
<td>0.905073E+00</td>
<td>0.909284E+00</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.867052E+00</td>
<td>0.940203E+00</td>
<td>0.904442E+00</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.907094E+00</td>
<td>0.961323E+00</td>
<td>0.959827E+00</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.935625E+00</td>
<td>0.974667E+00</td>
<td>0.972946E+00</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.952918E+00</td>
<td>0.982711E+00</td>
<td>0.981656E+00</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.965550E+00</td>
<td>0.987213E+00</td>
<td>0.986615E+00</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.976105E+00</td>
<td>0.990640E+00</td>
<td>0.989785E+00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.980175E+00</td>
<td>0.992956E+00</td>
<td>0.991715E+00</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.983883E+00</td>
<td>0.994777E+00</td>
<td>0.993383E+00</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.986806E+00</td>
<td>0.996070E+00</td>
<td>0.994697E+00</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.989226E+00</td>
<td>0.996902E+00</td>
<td>0.995712E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.990883E+00</td>
<td>0.997542E+00</td>
<td>0.996485E+00</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.992378E+00</td>
<td>0.998108E+00</td>
<td>0.997211E+00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.993634E+00</td>
<td>0.998459E+00</td>
<td>0.997825E+00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.994597E+00</td>
<td>0.998741E+00</td>
<td>0.998210E+00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.995415E+00</td>
<td>0.998960E+00</td>
<td>0.998549E+00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.996146E+00</td>
<td>0.999168E+00</td>
<td>0.998825E+00</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.37 A Test Beat and Its Inverse K-L Using the All-Ensemble Partition—Example A
Figure 5.38 A Test Beat and its Inverse K-L using the Gross-Abnormal Partition—Example A
Figure 5.39 A Test Beat and its Inverse K-L Using the All-Ensemble Partition—Example B
Figure 5.40 A Test Beat and its Inverse K-L using the Gross-Abnormal Partition—Example B
Figure 5.41 A Single Lead Tracing and its Inverse K-L Using the All-Ensemble Partition (Top), and the Gross-Abnormal Partition (Bottom)—Example C
Figure 5.42 A Test Beat and its Inverse K-L Using the All-Ensemble Partition- Example D
Figure 5.43 A Test Beat and its Inverse K–L Using the QRS-Suppressed Partition—Example D
Figure 5.44 A Test Beat and its Inverse K-L Using the All-Ensemble Partition—Example E
Figure 5.45 A Test Beat and its Inverse K-L Using the QRS-Suppressed Partition- Example E
Figure 5.46 A Test Beat and its Inverse K-L Using the All-Ensemble Partition—Example F
Figure 5.47 A Test Beat and its Inverse K-L Using the QRS-Suppressed Partition—Example F
package. The plots demonstrate the significant improvement in reconstruction accuracy by using the Gross-Abnormal set instead of the All-Ensemble set.

Similarly, figures 5.42-5.47 demonstrate the same trend for 3 cases of repolarization abnormality cases using the QRS-Suppressed set. The latter is of particular interest for detecting subtle changes in case of asymptomatic-population-based studies.

5.4 Clustering Studies

The fuzzy clustering procedure (FC) developed in Chapter 4 was implemented and tested on a subset of the database, using an IBM-360 computer. The pattern vectors for the subset were computed using the All-Ensemble basis vectors, and were selected according to the following criteria:

1. The original waveforms belong to pure and/or multiple-diagnosis classes, with the clinical diagnosis known apriori

2. Each class to be studied must contain a number of VCG's at least equal to the dimension of the feature space being studied.

Estimation of the optimal dimensionality of the feature space is beyond the scope of this work, since it requires a "large" database. Starting with a very limited database of 676 VCG's, only 168 cases (occupying 10 classes) satisfied the first criterion. Table 5.2 summarizes the patient populations per class in the subset. The first 9 classes in the table are pure, while the last one is a mixed class.

1 Gustafson [35] estimated that at least a 60th order K-L (20/lead) is needed to capture all subtleties in the morphology of a VCG.
Also, a maximum of 8-10 dimensions is the suboptimal choice for the second criterion (not all classes have enough data for their analysis). Further, to examine the complexity of the feature space, the classes separability, and to select, accordingly, the 10 best features for clustering, the components of the subset pattern vectors are mapped and plotted into a series of two dimensional subspaces as shown in figures 5.48-5.49 for the X-lead, in figures 5.50-5.51 for the Y-lead, and in figures 5.52-5.53 for the Z-lead. Two observations are evident from the plots:

1. Class separability (or more accurately, the degree of non overlapping) is better in the X- and Z-lead subspaces than in the Y-lead subspaces

2. Gross abnormal classes classes such as RBBB (#4), LBBB( #7), and LVH (#11 or "B") are relatively widely separated and non overlapping.

Based on the above, and in order to keep the memory requirement and CPU time to reasonable level, The FC procedure was tested in a 8 -dimensional feature using the principal 8 components of the X-lead pattern vector for each sample wave form in the subset. The computational experience, with this approach, however, indicates that using up to 60 components (20/lead) is entirely feasible. The underlying concept in this work is to investigate whether class-dependent fuzziness improves clustering or not, using appropriate probability error measure, and whether patients in multiple diagnosis classes (such as ST+T)
Figure 5.48 Feature Subspace of Components 1 and 2—Lead X
Figure 5.49 Feature Subspace of Components 2 and 3—Lead X
Figure 5.50 Feature Subspace of Components 1 and 2—Lead Y
Figure 5.51 Feature Subspace of Components 2 and 3—Lead Y
Figure 5.52 Feature Subspace of Components 1 and 2—Lead Z
Figure 5.53 Feature Subspace of Components 2 and 3-
### Table 5.2 Class Population

<table>
<thead>
<tr>
<th>NOTE</th>
<th>CLASS</th>
<th>ABBREVIATION</th>
<th># CASES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURE 1</td>
<td>NORM</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>PURE 4</td>
<td>RBBB</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>PURE 7</td>
<td>LBBB</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>PURE 8</td>
<td>IVCD</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>PURE 12(C)</td>
<td>T</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>PURE 13(D)</td>
<td>ST</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>PURE 15(F)</td>
<td>LVH</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>PURE 23(G)</td>
<td>IANT</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>PURE 25(H)</td>
<td>IFFS</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>MIXED 29(I)</td>
<td>ST + T</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.3 Fuzziness Parameters

<table>
<thead>
<tr>
<th>CLASS#</th>
<th>RUN 1</th>
<th>RUN 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>29</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 5.4 Disagreement Error Probabilities

<table>
<thead>
<tr>
<th>CLASS</th>
<th>ABBREVIATION</th>
<th>ERROR PROB. FIXED</th>
<th>ERROR PROB. VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1</td>
<td>NORM</td>
<td>.84</td>
<td>.78</td>
</tr>
<tr>
<td>4</td>
<td>RBBB</td>
<td>.22</td>
<td>.14</td>
</tr>
<tr>
<td>7</td>
<td>LBBB</td>
<td>.23</td>
<td>.19</td>
</tr>
<tr>
<td>*8</td>
<td>IVCD</td>
<td>.72</td>
<td>.67</td>
</tr>
<tr>
<td>*12</td>
<td>T</td>
<td>.59</td>
<td>.47</td>
</tr>
<tr>
<td>*13</td>
<td>ST</td>
<td>.79</td>
<td>.79</td>
</tr>
<tr>
<td>15</td>
<td>LVH</td>
<td>.21</td>
<td>.25</td>
</tr>
<tr>
<td>*23</td>
<td>IANT</td>
<td>.87</td>
<td>.54</td>
</tr>
<tr>
<td>*25</td>
<td>IFFS</td>
<td>.91</td>
<td>.72</td>
</tr>
<tr>
<td>*29</td>
<td>ST + T</td>
<td>.57</td>
<td>.24</td>
</tr>
</tbody>
</table>

### Table 5.5 Effect Of Merging Class 29 Into Classes 12,13 On The Membership Functions

<table>
<thead>
<tr>
<th>AV MEMBERSHIP FUNCTION</th>
<th>CLASS 12</th>
<th>CLASS 13</th>
<th>CLASS 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN 1</td>
<td>.0326</td>
<td>.0599</td>
<td>.0472</td>
</tr>
<tr>
<td>RUN 2</td>
<td>.1025</td>
<td>.0935</td>
<td></td>
</tr>
</tbody>
</table>
are best analyzed as members of a separate entity class in itself, or as members of two or more "parent" classes (such as ST and T).

In performing the clustering procedure, three important tasks need to be formulated:

1. Initial selection of each class's parameters (means and covariances). The underlying assumption here is that all class morphologies are Gaussian in the feature space (validation of this assumption is beyond the scope of this work). The data in the subset is first sorted by Cardiologist's class (each patient pattern vector file is allocated 60 words preceded by his ID) with a null file between classes. The initial means for each class are the averages of all patients in the corresponding Cardiologist's class. The initial covariances are chosen to be diagonal, with all diagonal elements set to 1.

2. Computation of interclass distances. Gustafson [35] proposed the Battacharyya distance as a distance measure, defined as follows:

\[
\text{BAT} = \frac{1}{4} \left( m_i - m_j \right)^T (p_i + p_j)^{-1} \left( m_i - m_j \right) + \frac{1}{2} \ln \frac{\left| (p_i + p_j) / 2 \right|}{\left| p_i \right|^{1/2} \left| p_j \right|^{1/2}}
\]

where \( m_i, p_i, m_j, \) and \( p_j \) are the means and covariances for clusters \( i, j \)

3. Computation of "discrepancy" error probability. Gustafson [35] proposed a distance measure between the natural class as determined by the FC algorithm, and the Cardiologist's determined class. This is defined as follows:
Denote
\[ w_{ij} = w_j(x_i) \text{= membership function of } x_i \text{ in class } \mathcal{G}_j \]

Let
\[ w_i = \max_j (w_{ij}) \]

and define
\[ e_{ii} = w_i - w_{ik}, \quad i \neq k; \quad e_{ii} = 0, \quad i = k \]

D iscrepancy error probability = average \( \left( \frac{I(e_{ii})}{N_K} \right) \) for class \( \mathcal{G}_K \)

where
\[ I = \text{Indicating function} \]
\[ = \begin{cases} 1 & e_{ii} > 0 \\ 0 & e_{ii} = 0 \end{cases} \]

and
\[ N_K = \# \text{ of samples in class} \]

Two 3-pass computer runs were performed on the data subset using the set of fuzziness parameters listed in table 5.3. The first run assumes equal or fixed fuzziness, while the other assigns low fuzziness to RBBB, LBBB, LVH, and high fuzziness to the rest. The discrepancy probabilities are summarized in table 5.4 for both runs. The results clearly indicate that the variable-\( \alpha \) run reduces the error probability, compared with the fixed-\( \alpha \) run, for all classes except the NORM, ST, and IIFS. Failure of the algorithm in these cases is perhaps attributable to the complexity and severe overlapping of those classes in the feature space.

Next, two computer runs were executed, one with class #29 as a separate entity in itself, and the other with members of the class equally distributed (or pregrouped) between classes #12 and #13 respectfully. Table 5.5 summarizes the average membership functions for all
members of class #29 as computed form the two runs. With class #29 intact, the membership functions depict equal belonging—on the average—to each one of the three classes #12, #13, and #29. The second run assigns significantly higher membership functions for the same data with respect to class #12 and #13 if the data were preassigned to the two classes. The latter suggests that VCG's with multiple diagnosis are best looked at as multiple members (with an appropriate membership functions) of the nearest set of pure classes, rather than being a class of their own.

5.5 Conclusions and Possible Future Work

In this work, new techniques, the optimally partitioned K-L expansions for the characterization of the cardiograms as stochastic processes, were developed. The approach attempted to compute the most efficient sets of K-L basis vectors by time, ensemble, and process partitioning. Furthermore, classes separability in the feature space was investigated using a fuzzy clustering algorithm with class-dependent fuzziness, to characterize various degrees of overlapping in the feature space. The same scheme was tested on a subset of the database which includes relatively nonempty sets of pure classes and one multiple diagnosis class. Members of the multiple diagnosis classes were found to be best characterized as being cases with multiple membership functions to the adjacent pure classes, rather than being a class of their own.

Some suggestions for future research in this field are listed below:

1. Extensive statistical analysis to determine lead(s) specificity
to certain classes (or vice versa), in order to reduce the dimensionality of the feature space.

2. Comparative sensitivity analysis of the K-L as a linear feature extractor and nonlinear feature extractors, in order to identify the most efficient feature extractor for serial comparison.

3. Perhaps the most difficult problem is determination of a proper discriminant function to separate or characterize members of the severely overlapping classes in the feature space, where the algorithm in this work failed. Parametrization of these classes should be carefully developed from both an information-theoretic point of view and study of the database.

4. The partitioned K-L approach and class-dependent fuzziness clustering algorithm may find applications in other areas of medical data analysis such as brain waves pattern recognition, ocular and evoked potentials pattern classification, speech, and cardiac sounds recognition.
APPENDIX A
SOFTWARE MODULES

The following modules use standard IBM-360 Fortran G compiler with Cal-comp Plotting Library as specified by "PLOT GCLG" in the JCL. Every JCL procedure for file(s) access—input and output—is attached to each listing.

1. Module HBZ33BLZ (and H)

This program reads in digitized X,Y,Z data (500 samples/sec. 10-bit 2's complement in a 2-byte word right-justified) from a magnetic tape (7200 byte/record 1 record/block, 1 block/file, 1 file/patient, and up till 100 patients/tape). The programs calibrate the data, computes the spatial vector velocity, the R-wave fiducial points, the PR-knots, then restores the baseline. The program then averages all beats in the file and stores the baseline-restored data and average beats on a temporary disk file (1 disk file per patient file) labelled BLR.TEMP. This program is an extended version of Keiser's [14] to include true cubic spline restoration and outlier detection/rejection.

2. Module HBZ33MRH

This program merges all disk temporary files... BLR.TEMP into an archive file with only one EOF at the end of the last patient. The resulting file is stored on disk and is labelled ...MRG.DRAP.

3. Module HBZ33XYZ

This program extracts all the average beats (1 beat per patient) from ...MRG.DRAP, compresses the data by 2:1, sets a 200-point fixed window for all averaged beats with the fiducial point at location 83 in the
window. The resulting "ensemble" file is labelled XZY.ALL.

4. Module HBZ33XXXX (where XXX is GRS or STT)

These two programs are quite similar and are used for ensemble partitioning. The ensemble is sorted and two subsets are stored in two separate files.

a. File XYZ.GRS contains only gross abnormalities (RBBB, LBBB, LVH...).

b. File XYZ.STT contains all pure classes excluding gross abnormalities, and with the QRS complex suppressed.

c. File XYZ.ALL is still intact.

5. Modules HBZ33PA* (where* =A, G, or S)

These are actually the same program with 3 different endings in the JOB name and different data subsets. The * character designates which subset of the ensemble is processing (XYZ.ALL, XYZ.GRS, or XYZ.STT respectively). The program implements the 3-template P-wave detection algorithm. The resulting P-wave locations are stored in the files .PAT*** (where XXX is ALL, GRS and STT respectively).

6. Modules HBZ33 KL* (where* =A, G, or S)

This program computes the 60 K-L basis vectors (vector size =200) for the data subsets .XYZ.ALL, .XYZ.GRS, and .XYZ.STT. The resulting basis vectors are stored in the disk files .KLW.*** (where *** is ALL, GRS, and STT respectively). The "BASIS" subroutine has been developed by Halliday [32].

7. Modules HBZ33 PK* (where * =A, G, or S)

This program is very similar to the .KL* program above except for vector size of 50. It computes the basic vectors for the P-wave
segments at the locations stored in .PAT***. The resulting vectors are stored in the .PKL*** files (where *** files is ALL, GRS, and STT) respectively).

8. Modules HBZ33RE* (where * is A, G, or S)

The program computes the pattern vectors for each waveform in the subsets .XYZ.ALL, .XYZ.GRS, and .XYZ.STT. It computes a cost function as an integrated square error between the original waveform and reconstructed on (or inverse transform). It uses as input the files .XYZ.***, .KLW.***, and .PKL.***. The computed pattern vectors are then stored in the disk files .REC.XXX (where .XXX is .ALL, .GRS, and .STT).

9. This program sorts the pattern vector, file .REC*** such that all patients with pure classes are grouped, with a separation character between groups, and rejects all "99" classes except for few ones randomly assigned to each of the pure classes. The resulting disk file is .SOR.ALL.

10. This program is an extension of Gustafson's [41] to include class dependent fuzziness. The program computes and stores membership functions after each pass in multiclass run, using fuzzy parameters for all classes in the subset .SOR.ALL.
APPENDIX B

PROGRAM LISTING
PIPTI=HB233BLz.360;6
//HB233BLZ JOB (HM1,BO20,15,5,1,,Y,00),'HB2042091CR ZIED'
*/JOPARM L=5,E=10,RESTART
//TRDPRO EXEC PLOTGCLG,REGION,GO=160K,TIME=8
//FORT.SYSIN DD *
EQUIVALENCE (X(1),AVG(1,1,1))
DIMENSION AVG(600,4,3),C(600),BEATS(3),LFID(50),COEF(99)
DIMENSION PR(50,3),XX(4,3),X(6000),SCF(3)
DIMENSION JFID(50),A(4),IFID(50),TX(2),TY(2)
INTEGER*2 CASE(5),IX(6000,4),JX,ICT(52),JY,RCASE(6),ISP
INTEGER*2 SSAN(5),DOB(3),NAME(14),NCASE(3)
DATA A/8.0,5.0,3.0,0.0/
DATA ISP/' ' /
NT=0
C INPUT CONTROL PARAMETERS
READ (5,9) NC,MTL,NPL
9 FORMAT (10I2)
C INPUT COEFFICIENTS FOR THE DIGITAL FILTER FOR
IF(NC.GT.0) READ (5,8) (COEF(I),I=1,NC)
8 FORMAT (8E10.7)
C INPUT CASE NUMBER, AGE, HEIGHT, WEIGHT, AND CALIBRATION DATA
91 READ (5,1) CASE,IAG,IHT,IWT,((IX(I,J),J=1,750),I=1,1,750)
1 FORMAT (GA1,3A2,250A2))
C CONVERT CASE NUMBER FROM ASCII TO EBCDIC
CALL ASCBEC (CASE,CASE,6)
READ (5,7) RCASE,NCASE,SSAN,DOB,NAM
7 FORMAT (6A1,IX,3A2,IX,5A2,3A2,IX,14A2)
DO 100 I=1,6
IF(RCASE(I).EQ.ISP) GO TO 100
IF(RCASE(I).EQ.CASE(I)) GO TO 100
WRITE(6,11) RCASE,CASE
11 FORMAT(1H1,GA1,' REQUESTED','5X,GA1,' FOUND')
STOP
100 CONTINUE
WRITE (5,10) CASE,NCASE,SSAN,DOB,NAM
10 FORMAT ('1VCG NUMBER: ',3A2,' CASE NUMBER: ',3A2,' SSAN: ',5A2,
1' DOB: ',3A2,' NAME: ',14A2)
WRITE (10) NCASE,SSAN,DOB,NAM,IAG,IHT,INT
C DETERMINE SCALE FACTORS FROM CALIBRATION PULSES
DO 270 LD=1,3
C FIND MAXIMUM AND MINIMUM VALUE FOR EACH LEAD
JX=IX(1,LD)
JY=JX
DO 210 I=2,750
IF(IX(I,LD).GT.JX) JX=IX(I,LD)
IF(IX(I,LD).LT.JY) JY=IX(I,LD)
210 CONTINUE
C DETERMINE MIDRANGE
JZ=(JX+JY)/2
SUM=0.
C=0.
INC=(JX-JZ)/2
DO 220 I=1,32
ICT(I) = 0
C SUM ALL VALUES BELOW MIDRANGE AND A FREQUENCY DISTRIBUTION OF
C THOSE ABOVE MIDRANGE
    DO 250 I = 1, 750
        IF (IX(I,LD) .GE. J2) 230, 240, 240
    230  SUM = SUM + IX(I,LD)
        CNT = CNT + 1.
        GO TO 250
    240  IND = (IX(I,LD) - J2) / INCR + 1
        ICT(IND) = ICT(IND) + 1
    250 CONTINUE

    JY = 0
C SCALE FACTOR EQUALS MODE OF DISTRIBUTION OF VALUES ABOVE MIDRANGE
C MINUS MEAN OF ALL VALUES LESS THAN MIDRANGE
    DO 260 I = 1, 49
        IS = ICT(I)
        IF (IS .LT. JY) GO TO 260
        JY = IS
        JJX = I - 1
    260 CONTINUE
    I = (2 * ICT(JJX + 1) - ICT(JJX) - ICT(JJX + 2))
    FMAX = FLOAT (ICT(JJX + 1) - ICT(JJX)) / I
    SCF(LD) = 2.0 * (FMAX - (SUM / CNT))

WRITE (6, 9998) SCF
9998 FORMAT (1H0, 3F10.3)

I2 = 0
C READ DATA
    DO 330 NBLKS = 1, 5
        I1 = I2 + 1
        I2 = I2 + 1200
        READ ( 9, 2 ) ((IX(I,J), J = 1, 3), I = I1, I2)
            2 FORMAT (15(250A2))
    330 CONTINUE
C INTRODUCE AN ARTIFICIAL BASELINE SHIFT
    DO 337 LD = 1, 3
        DO 337 I = 1, 5000
            IBET = .25 * SCF(LD) * SIN (6.28 * .002 * I/4) + .25 * SCF(LD) * SIN (3.14 * .002 * I/4)
            IX(I, LD) = IX(I, LD) + IBET
    337 CONTINUE
    DO 930 I = 1, 6000

X(I) = 0.
N3 = NC/2
N1 = N3 + 1
N2 = 5000 - N3
    DO 950 LD = 1, 3
        Y2 = 0.
            Y1 = 0.
            DO 950 I = N1, N2
                J1 = I - N3
                J2 = I + N3
                J3 = 0
                DO 950 J = 1, J3
                    950 CONTINUE
            950 CONTINUE
    930 CONTINUE

W R I T E ( 6 , 9998 ) SCF
9998 FORMAT (1H0, 3F10.3)
Y3=0.
C APPLY DIGITAL FILTER TO EACH LEAD
DO 940 J=J1,J2
   J3=J3+1
   XI=IX(J,LD)/SCF(LD)
940 Y3=Y3+COEF(J3)*XI
   XD=Y3-Y1
   Y1=Y2
   Y2=Y3
C ACCUMULATE
   IF(XD.LT.0.) XD=-XD
250 X(I)=X(I)+XD
   DO 955 I=1,NC
   X(I)=0.
955 X(6001-I)=0.
C DETERMINE MAXIMUM
   VMAX=0
   DO 960 I=N1,N2
   IF(X(I).GT.VMAX) VMAX=X(I)
C NORMALIZE
   DO 991 I=1,6000
      X(I)=X(I)/VMAX
C SCALE TO MICROVOLTS
991 IX(I)=1000.*X(I)
C LOCATE FIDUCIALS -- FIRST PEAK OF SPEED GREATER THAN 20 PERCENT
C OF THE MAXIMUM FOLLOWED BY A VALLEY OF
C 10 PERCENT OF THE MAXIMUM
IFID=0
I=N1+2
IFSW=0
970 I=I+1
   IF(IFSW.NE.0) GO TO 973
   IF(X(I).LT.0.30) GO TO 975
   IF(X(I+1).GT.X(I)) GO TO 975
   IFSW=I
   GO TO 975
973 IF(X(I+1).LT.X(I)) GO TO 975
   IF(X(IFSW)-X(I).LT.0.1) GO TO 974
   NFID=NFID+1
   IFID(NFID)=IFSW
   I=I+100
974 IFSW=0
975 IF(X(I).LT.N2 ) GO TO 970
C SAVE NUMBER AND LOCATIONS OF FIDUCIALS
WRITE (10) NFID,(IFID(I),I=1,NFID)
IMTRR=(IFID(NFID)-IFID(1))/(NFID-1)
IF(NPL.NE.0) GO TO 992
C PLOT
990 CALL EQSTYP (0,-1,CASE,0,1)
   CALL HEUPEN(1)
   CALL EQMTTL (CASE,1.021,1.,10.,0.)
   CALL EQSCL (1.0,12000.,0.,1.,0.3,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.)
   CALL BPLOT (1,6000,1.2,X,1.2,0,CASE,0.07)
   CALL BOVPLT (1)
C EXAMINE FID. FOR PREMAT. BEATS*OUTLIERS FOR DELETION IN BASELINE
C NOISE ESTIMATION AND REMOVAL

992  RR=INTRR
    I=0
421  I=I+1
     IF(IFID(I).LT.N1+60) GO TO 421
     JFID(1)=IFID(I)
     LFID(1)=JFID(1)
     IND=1
     I1=I+1
     SM=0.
     FCNT=0.
     NFID=1
     DO 425 I=I1,NFID
     DIF=IFID(I)-IFID(I-1)
     FCNT=FCNT+1.
     SH=SH+DIF
     ESTRR=SH/FCNT
     IF(RATIO.LT.0.8) GO TO 422
     HFRD=HFRD+1
     LFID(NFID)=IFID(I)
422  CONTINUE
     RATIO=DIF/RR
     IF(RATIO.LT.0.8) GO TO 425
     X1=IFID(2)-40
     KI=IFID(I)-40
     CALL CORCOP(X(K1),X(KI),100,R)
     IF(R.LT.0.8) GO TO 425
     IND=IND+1
     JFID(INDF)=IFID(I)
425  CONTINUE
     IF(NPL.NE.0) GO TO 996

C HARF FIDUCIALS USED FOR BASELINE NOISE ESTIMATION AND REMOVAL
     CALL NEWPEN (2)
     TY(1)=0.
     TY(2)=8.
     DO 995 I=1,INDF
     TX(1)=JFID(I)-1
     TX(1)=2.*TX(1)
     TX(2)=TX(1)
     CALL BPLOT (1,2,1,TX,TY,1,0.,0,CASE,0.07)
     TEMP=TY(1)
     TY(1)=TY(2)
     TY(2)=TEMP
994  CALL BSCPLT (1)
995  CONTINUE

C DEFINE PR SEGMENT AS A 16 MS SEGMENT OF HT'INT!
C LOCATED IN THE REGION 20 TO 120 MS BEFORE THE FIDUCIAL
996  DO 430 I=1,INDF
     J1=JFID(I)-50
     J2=J1+0
     VMAX=0.
     DO 429 J=J1,J2
429 \[ V_{MAX} = V_{MAX} + X(J) \]
\[ J2 = J1 + 40 \]
\[ VSUM = V_{MAX} \]
\[ JFID(I) = J1 \]
\[ DO 430 J = J1, J2 \]
\[ VSUM = VSUM - X(J) + X(J+9) \]
\[ IF(VSUM.GT.VMAX) GO TO 430 \]
\[ VMAX = VSUM \]
\[ JFID(I) = J1 + 1 \]

430 CONTINUE

C MARK PR SEGMENTS ON THE PLOT

TY(1) = 0.
TY(2) = 0.1
DO 440 I = 1, INDF
II = JFID(I)
IF(NPL.NE.0) GO TO 433
TX(1) = 2.*FLOAT(I1) - 2.
TX(2) = T X(1)
CALL B P L O T (1, 2, 1, TX, TY, 1, 0., 0., CASE, 0.07)
CALL BOVPLOT (1)
TX(1) = TX(1) + 16.
TX(2) = TX(1)
CALL B P L O T (1, 2, 1, TX, TY, 1, 0., 0., CASE, 0.07)
CALL BOVPLOT (1)

433 I2 = I1 + 8

C COMPUTE THE MEAN VALUE OF THE PR SEGMENT

DO 440 LD = 1, 3
PR(I, LD) = 0.
DO 435 J = I1, I2
435 PR(I, LD) = PR(I, LD) + IX(J, LD)
440 PR(I, LD) = PR(I, LD) / (9.*SCF(LD))
IF(NPL.NE.0) GO TO 334

C PLOT THE RAW DATA

CALL PLOT (1, 2, 1, TX, X(1), 1, 0., 0., CASE, 0.07)
CALL BOVPLOT (1)

C INITIALIZE BASELINE ESTIMATOR COEFFICIENTS

334 T1 = FLOAT(JFID(2) - JFID(1))
T2 = FLOAT(JFID(3) - JFID(1))
T3 = FLOAT(JFID(4) - JFID(1))
DEL1 = T1*(T2*T2*T3*T3-T3*T2*T2)
DO 1111 LD = 1, 3
1111 XK(4, LD) = PR(I, LD)
DEL1 = T1*T1*(T2*T2*T3*T3-T3*T2*T2)
DE L2 = DEL1 - DEL2 + DEL2
T11 = (T2*T2*T3*T3*T3*T3-T3*T2*T2)/DEL
T12 = (T1*T1*T1*T1*T1*T1-T2*T2*T2)/DEL
T13 = (T1*T1*T1*T1*T1*T1-T3*T3*T3)/DEL
T21 = (T1*T1*T1*T1*T1*T1-T2*T2*T2)/DEL
T22 = (T1*T1*T1*T1*T1*T1-T2*T2*T2)/DEL
T23=(T2*T1*T1-T1*T2*T2*T2)/DEL
T31=(T2*T3-T3*T2)/DEL
T32=(T3*T1*T1-T1*T3*T3)/DEL
T33=(T1*T2-T2*T1*T1)/DEL
DO 500 LD=1,3
XK(1,LD)=12.*(PR(1,LD)-PR(2,LD))/(T1*T1)
B1=(PR(2,LD)-PR(1,LD))*(2.*T1+2.*T3+3.*T1*T1*T1)
B3=(PR(4,LD)-PR(1,LD))*(2.*T3+2.*T1*T3+3.*T1*T1*T1)
XK(1,LD)=XK(1,LD)+6.*(B1+B2+B3)/(T1*T1)
C1=(PR(2,LD)-PR(1,LD))*T1
C2=(PR(3,LD)-PR(1,LD))*T1
C3=(PR(4,LD)-PR(1,LD))*T1
XK(2,LD)=C1+C2+C3
500 XK(2,LD)=XK(2,LD)-2.*(B1+B2+B3+C1+C2+C3)/T1
C REMOVE BASELINE AND AVERAGE 3 LEADS
CALL NEWPEN (1)
601 DO 700 LD=1,3
DO 640 I=2,INDF
I1=JFID(I-1)
I2=JFID(I-2)
C REMOVE BASELINE BETWEEN NODES
DO 630 J=1,12
X(J)=X(J,LD)/SCF(LD)-XK(4,LD)
E4=XK(1,LD)/6.+XK(2,LD)/2.+XK(3,LD)+XK(4,LD)
E3=XK(1,LD)/2.+XK(2,LD)+XK(3,LD)
E2=XK(1,LD)+XK(2,LD)
XK(2,LD)=E2
XK(3,LD)=E3
630 XK(4,LD)=E4
C RECOMPUTE COEFFICIENTS FOR NEXT PAIR OF NODES
IF(I.EQ.INDF) GO TO 640
T1=FLOAT(JFID(I+1)-JFID(I))
IF(I.GE.INDF-2) GO TO 632
T2=FLOAT(JFID(I+2)-JFID(I))
T3=FLOAT(JFID(I+3)-JFID(I))
DEL1=T1*(T2*T3*T3*T3-T2*T2*T2)
DEL2=T1*T1*(T2*T3*T3*T3-T2*T2*T2)
DEL3=T1*T1*T1*(T2*T3*T3-T3*T3)
DEL=DEL1-DEL2+DEL3
T11=(T2*T3*T3-T3*T3)*T2/DEL
T12=(T2*T3-T3*T3)*T1/DEL
T13=(T1*T3*T3-T3*T3)*T1/DEL
T21=(T2*T3-T3*T3)*T1/DEL
T22=(T2*T3-T3*T3)*T1/DEL
T23=(T3*T1-T1*T3-T3)*T1/DEL
T31=(T2*T3-T3*T3)*T1/DEL
T32=(T3*T1-T1*T3-T3)*T1/DEL
T33=(T3*T1-T1*T3-T3)*T1/DEL
IF(I.NE.INDF-1) GO TO 633
S32 XK(1,LD)=12.*(XK(4,LD)-PR(I+1,LD))/(T1*T1)
XK(1,LD)=XK(1,LD)+6.*(XK(3,LD)+DERV)/(T1*T1)
XK(2,LD)=-6.*(XK(4,LD)-PR(I+1,LD))/(T1*T1)
XK(2,LD)=XK(2,LD)-2.*(2.*XK(3,LD)+DERV)/T1
GO TO 640
633 XK(4,LD)=PR(I,LD)
B1=(PR((I+1),LD)-PR(I,LD))*((T11+2.*T1*T21+3.*T1*T1*T31)
B2=(PR((I+2),LD)-PR(I,LD))*((T12+2.*T1*T22+3.*T1*T1*T32)
B3=(PR((I+3),LD)-PR(I,LD))*((T13+2.*T1*T23+3.*T1*T1*T33)
DERV=B1+B2+B3
GO TO 632
640 CONTINUE
C ZERO INVALID DATA AFTER BASELINE NOISE REMOVAL
J1=JPID(1)
DO 650 I=1,J1
650 X(I)=0.
J1=JPID(JDP)
DO 660 I=J1,J10000
660 X(I)=0.
C SCALE DATA TO MICROVOLT
DO 690 I=1,J10000
690 IX(I,LD)=1000.*X(I)
IF(NPL.NE.0) GO TO 680
C PLOT BASELINE NOISE REMOVED DATA
CALL NEWPEN(2)
DO 691 I=1,J10000
691 X(I)=X(I)+A(LD)
CALL BPLOT(1,3600,1,2,X,1,2,,0,CASE,0.07)
CALL BOVPLT(1)
CALL NEWPEN(1)
DO 697 I=1,J10000
697 X(I)=A(LD)
CALL BPLOT(1,3600,1,2,X,1,2,,0,CASE,0.07)
CALL BOVPLT(1)
CALL NEWPEN(2)
680 CONTINUE
DO 6970 I=1,J10000
6970 X(I)=X(I)+.25*SIN(.28*SIN(.002*I/4)+.25*SIN(3.14*.002*I/4)
CALL BPLOT(1,3600,1,2,X,1,2,,0,CASE,0.07)
CALL BOVPLT(1)
700 CONTINUE
CALL ENDPIT(0)
DO 710 I=1,J10000
710 WRITES (10) (IX(I,LD),LD=1,4)
678 READ (9,1, END=6800) CASE
0 GO TO 670
6600 CONTINUE
C LOCATE VALID DATA FOR AVERAGING
II=0
WRITES (6,13) NFID,(IFID(I),I=1,NPID)
13 FORMAT ('0',16,' BEATS WITH PUDICIALS AT',16(IS,''))
1200 II=II+1
II=IFID(II)-271
IF(II.LT.IFID(1)) GO TO 1200
NF=NFID
1210 NF=NF-1
   IL=IFID(NF)+330
   IF(IL.GT.IFID(NF)) GO TO 1210
C CLEAR ARRAYS USED TO CONTROL AND COUNT ACCUMULATION OF BEATS
DO 1225 I=1,3
1225 BEATS(I)=0.
C BEGIN CORRELATION AND AVERAGING
   NAVG=0
   DO 1260 I=II,NF
      J1=IFID(I)-270
      J2=J1+599
      DO 1255 IAVG=1,3
         IF(BEATS(IAVG).NE.0.) GO TO 1253
      C AVERAGE PRESENT TO CORRELATE THIS BEAT WITH
      C NO: STORE THIS BEAT FOR A NEW TEMPLATE
      NAVG=NAVG+1
      WRITE (6,1225)I,J1,J2,IAVG
   1225 FORMAT ('BEAT ','I2,5X,'FROM ','I5,' TO ','I5,5X,'TEMPLATE ','I2)
   DO 1220 LD=1,4
      JJ=0
      DO 1220 J=J1,J2
         JJ=JJ+1
   1220 AVG(JJ,LD,IAVG)=IX(J,LD)/1000.
   BEATS(IAVG)=1.
   IF(NPL.GT.0) GO TO 1260
C PLOT THE NEW TEMPLATE
   CALL BGNSTP (0,-1,CASE,0,0,1)
   CALL BGENTL (CASE,3,0,21,1.,10.,0.,0.)
   CALL BGENSL (1.,0.,1200.,0.,1.,0.,0.,3.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.)
   DO 850 LD=1,4
   DO 800 IT=1,600
   850 C(IT)=AVG(IT,LD,IAVG)+A(LD)
   CALL BPNLT (1)
   CALL BPLT (0)
   GO TO 1250
C YES: DETERMINE MAXIMUM CORRELATION OF A 200 MS REGION OF
C MOVING PLUS AND MINUS 10 MS FROM PIGDUCIAL ALIGNMENT
1253 RK=0.
   RR=-10
   JJ=0
   DO 1230 J=J1,J2
      JJ=JJ+1
   1230 C(JJ)=IX(J,4)/1000.
   DO 1240 K=1,21
      J=J2+K
      CALL CONCOP(AVG(240,4,IAVG),C(JJ),100,R)
      IF(R.RT.RK) GO TO 1240
      RK=R
      JSAVE=IFID(J)+R
   1240 ZR=ZR+1
      RR=JSAVE-IFID(J)
   WRITE (6,1251) I,IAVG,RR,RK
1251 FORMAT ('BEAT ','I2,' CORRELATES WITH TEMPLATE ','I2,' OFFSET BY ',)
1  I2, ' POINTS AS ', F8.5)
C  IF MAXIMUM CORRELATION IS LESS THAN 0.95 TRY ANOTHER TEMPLATE
   IF (EK.LT.0.95) GO TO 1255
C  ACCUMULATE BEAT ALIGNED FOR MAXIMUM CORRELATION
1256 J1=JSAVE=270
   J2=J1+599
   DO 1250 LD=1,3
      JJ=0
      DO 1250 J=J1,J2
         JJ=JJ+1
      1250 AVG(JJ,LD,IAVG)=AVG(JJ,LD,IAVG)+IX(J,LD)/1000.
      BEATS(IAVG)=BEATS(IAVG)+1.
   GO TO 1260
1255 CONTINUE
1250 CONTINUE
   WRITE (10) NAVG,(BEATS(I),I=1,NAVG)
   DO 1980 J=1,NAVG
5  C  DIVIDE BY THE NUMBER OF BEATS ACCUMULATED
   DO 1270 LD=1,3
   DO 1270 I=1,600
      1270 AVG(I,LD,J)=AVG(I,LD,J)/BEATS(J)
   IF (NAV.GT.0) GO TO 1955
5  C  PLOT AVERAGED BEATS
   CALL BGPSTP (0,-1,CASE,3,0,1)
   CALL BGNTTL (CASE,3,0.21,1.,10.,0.,0.)
   CALL BGNSCL (1.,0.,1200.,0.,1.,0,0,3.,1.,1.,0.,0,0.,0.,0.,0.,0,0,1.4)
   DO 1950 LD=1,4
   DO 1900 I=1,600
      1900 C(I)=AVG(I,LD,J)+A(LD)
   CALL BHPLOT (1,600,1,2,C,1,2.,0,CASE,0.07)
1950 CALL BGPPLT (1)
   CALL BNDPLT (0)
1955 CONTINUE
   DO 1956 I=1,600
      1956 WRITE (10) (AVG(I,LD,J),LD=1,4)
1950 CONTINUE
ENDFILE 10
C  CHECK IF ANOTHER PATIENT TO PROCESS
   MT=MT+1
   IF (MT.LT.NT) GO TO 91
   REWIND 9
   IF (NAV.LE.0) CALL ENDPLT (1)
STOP
END
SUBROUTINE ASC2EC (INPT,OUTPT,HUM)
C  SUBROUTINE TO CONVERT ASCII CHARACTERS TO EBCDIC CHARACTERS
C  INPT - AN INTEGER * 2 ARRAY CONTAINING 1 ASCII CHAR. PER ENTRY
C  FOR TRANSLATION
C  OUTPT - AN INTEGER * 2 ARRAY CONTAINING 1 EBCDIC CHAR. PER ENTRY
C  AFTER TRANSLATION
C  HUM - THE NUMBER OF ENTRIES TO BE TRANSLATED
   INTEGER*2 TABLE(5?), INPT(HUM), OUTPT(HUM)
   DATA TABLE '/ ' $ ' $ ' $ ) + , . / 0 1 2 3 4 5 6 7 8 9 :

DO 100 I=1,NUM
C CONVERT THE CHARACTER TO AN INDEX
IND=INPT(I)/256-31
C CHECK FOR A VALID CHARACTER AND SUBSTITUTE A SPACE IF NOT VALID
IF(IND.LT.1) IND=1
IF(IND.GT.59) IND=1
100 IOUTPT(I)=TABLE(IND)
RETURN
END
SUBROUTINE CORCOF(X,Y,N,R)
DIMENSION X(N),Y(N)
SXY=0.
SXX=0.
SYY=0.
SX=0.
SY=0.
DO 100 I=1,N
SX=SX+X(I)
SY=SY+Y(I)
SXX=SXX+X(I)*X(I)
SYY=SYY+Y(I)*Y(I)
100 SXY=SXY+X(I)*Y(I)
FN=FLOAT(N)
SXY=SXY-SX*SY/FN
SXX=SXX-SX*SX/FN
SYY=SYY-SY*SY/FN
R=SXY/(SQRT(SXX*SYY))
RETURN
END
/*
 //LKED.SYSLMOD DD DSN=SYS1.TESTLIB(HBZ33BLZ),DISP=SHR
 //GO.PT09F001 DD DCB=(LRECL=7200,BLKSIZE=7200,RECFM=F,BUFNO=1),
 // VOL=REF=HB.H3H01.755C016.HB9540.DR3808,
 // DSN=HB.H3H01.PIPE20001,
 // DISP=OLD,LABEL=(001,SL,,IN)
 //GO.PT10F001 DD DCB=(LRECL=212,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
 // UNIT=TAPE,Vol=SER=SCRTCH,DISP=(NEW,DELETE,DELETE),
 // DSN=HB.H3H01.Z332042.BLZ.NEW
 //GO.SYSIN DD *
C
C THIS PROGRAM CONCATENATES FILES FROM 10 INPUTS INTO 1 OUTPUT
C
DIMENSION IFID(50),AVG(200,4),BEATS(3),AV(4)
INTEGER*2 IX(4),CASE(25),IAHW(3)
   NS=0
   NS=4
90 READ(4) CASE,IAHW
   WRITE(10) CASE,IAHW
   WRITE(6,5)CASE(1),CASE(2),CASE(3)
5 FORMAT(' ',3A2,'HAS BEEN MERGED')
   READ(4)IFID,(IFID(I),I=1,NS)
   WRITE(10)INP,(AV(I),I=1,4)
   READ(4)IX
160 WRITE(10)IX
   READ(4)NAVG,(BEATS(I),J=1,NAVG)
   WRITE(10)NAVG,(BEATS(I),J=1,NAVG)
   DO 810 N=1,NAVG
   DO 910 J=1,500
   READ(4)AV
   WRITE(10)AV
810 CONTINUE
870 READ(4,END=880)AV
   GO TO 870
880 NS=NS+1
   IF(NS.LT.NS)GO TO 90
ENDFILE 10
STOP
END

*/*EDIT.STEPLIB DD DSN=SYS1.TESTLIB,DISP=SHR
GO.TO 4 FORMAT DD DBC=(LRECL=122,BLKSIZE=4096,RECFM=VBS,BUFSIZE=1),
   DISP=OLD,UNIT=3330,VOL=SER=000062,
   SPACE=(TRK,2,1,RLSE),
   DSN=HR.3301.2332042.BLK.TEMP41
-.-.
*/*GO.FTG4010 DD DBC=(LRECL=212,CLKSIZE=4096,RECFM=VBS,BUFSIZE=1),
   DISP=OLD,UNIT=3330,VOL=SER=000062,
   SPACE=(TRK,2,1,RLSE),
   DSN=HR.3301.3332042.BLK.TEMP50
GO.TO 1 FORMAT DD DBC=(LRECL=96,CLKSIZE=2048,RECFM=VBS,BUFSIZE=1),LABEL=(001,CL),
   DSN=HR.3301.3332042.HRG.BRMP01
*/
C THIS PROGRAM EXTRACTS 200 OUT OF 400 POINTS OF AVERAGED C WAVEFORMS EFFECTIVELY INCREASING SAMPLING RATE TO 1/ (4 MILLISECONDS)
DIMENSION X(200),IFID(50),A(4),AVG(200,4),BEATS(3),AV1(200,4)
DIMENSION ROT(3),LAB(25),AV(4)
INTEGER*2 IX(4),CASE(25),IAHW(3)
DATA LAB/'S-1','S-2','S-3','S-4','S-5','S-6','S-7','S-8','
1'S-9','S-10','S-11','S-12','S-13','S-14','S-15','S-16','S-17','
DATA M5,3.5,2.,0./
DATA TTY'SER1'/
READ(5,3)ISRALT,MS
3 FORMAT(2I4)
MS=0
90 READ(9,END=683)CASE,IAHW
READ(9)NFID,(IFID(I),I=1,NFID)
WRITE(6,999)
999 FORMAT(1X0,'START NEXT RUN WITH THE LAST SER NO OF THIS ONE')
WRITE(5,1)CASE,IAHW
1 FORMAT('CASE NUMBER :',J142,' SSAN :',5A2,' DOB :',5A2,' NAME :'
1,14A2,3I5)
INTRR=IFID(NFID)-IFID(1)/(NFID-1)
MR=1/(INTRR*.002)
JS=0
JS=JS+1
IF(JS.GT.6000)GO TO 170
DO 160 J=JS,6000
160 READ(9)IX
170 READ(9)BEATS(I),I=1,NBEATS
DO 360 X=1,NBEATS
ISRALT=ISRALT+1
WRITE(10)ISRALT,HR,(CASE(I),I=1,3),X
WRITE(6,5)ISRALT,HR,K
5 FORMAT(1X,'SER #',IS,8X,'HEART RATE',F20.10,' AVG #',6I2)
DO 810 J=1,106
810 READ(5)AV
ISW=-1
JJ=0
DO 820 J=1,400
READ(5)AV
ISW=ISW
IF(ISW.LT.0)GO TO 820
JJ=JJ+1
DO 320 LD=1,4
DO 830 J=1,94
830 READ(9)AV
DATA CASE(4)'/
CALL BGNSCL (1, FI, FL, FL, 2, 2, 1, 1, 1, 1, 2, 5, 1, 2, 5, 1, 14)
CALL BGNTTL (LAM(K+20), 1, 0, 14, 3, 10, 0, 0, 0)
CALL BGNSCL (1, 212, 1012, 0, 1, 0, 0, 3, 1, 496, 7, 1474, 1, 7874, 1, 5748, 7874, 10, 0, 0, 0, 14)
DO 410 LD=1, 4
DO 400 I=1, 200
400 Y(I) = AVG(Y, LD) + A(LD)
CALL BPLT (1, 200, 1, 2, 1, 4, 0, 0, 0, 0, 0, 1)
CALL BOVPLT (1)
410 CONTINUE
960 CONTINUE
880 NS=MS+1
IF (NS.LT.NS) GO TO 90
388 CALL BNDPLT (1)
STOP
END

S T O P
INTEGER*2 CASE(3)
DIMENSION AVG(200,3)

L=0
5 READ (5,10) IS
10 FORMAT (I3)
20 READ(9,END=999) IK, HR, (CASE(I), I=1,3), K
READ (5) ((AVG(I,J),I=1,200),J=1,3)
IF (IK .NE. IS) GO TO 20
L=L+1
WRITE(10) L, HR, (CASE(I), I=1,3), K
WRITE(6,150) L, HR, K
150 FORMAT (1H,'SER ',5X,'HEART RATE ',F20.10,'AVG ',12)
DO 16 M=1,3
16 CONTINUE
DO 15 I=65,101
AVG(I,J)=AVG(65,J)+AVG(101,J)-AVG(65,J)* (I-65)/36.
15 CONTINUE
16 CONTINUE
WRITE(10) ((AVG(I,J),I=1,200), J=1,3)
GO TO 5
999 ENDFILE 10
END

/*
//ISKED.SYSLIB DD DSN=SYSLIB.HEZ33STT, DISP=SHR
//GO.FT001 DD DISP=OLD,
// DCB=(LRECL=080, BLKSIZE=2048, RECFM=VES, SUPMOD=1),
// DSN=HB.H3H01.2332042.XYZ.DRAP
//GO.FT10001 DD DISP=(NEW, CATLG, DELETE), VOL=SER=000033,UNIT=3330,
// DCB=(LRECL=888, BLKSIZE=2048, RECFM=VES, SUPMOD=1),
// SPACE=(TRK, (50,5)), DSNNAME=HB.3301.3333042.XYZ.STT
//GO.SYSIN DD *
004
006
007
010
016
017
021
022
023
024
026
027
033
BEGIN

//HBS33GRS JOB (3H01,13020,15,5,1,,Y,00),'HBS0240107R ZIED'
/*JOBPARM L=5,CARDS=1000,2=15,RESTART
//GRS EXEC PLOTGCLG,REGION.GO=72K,TIM=12
//FORT.SYSIN DD *
INTEGR+2 CASE(3)
DIMENSION AVG(200,3)
L=0
5 READ(5,10)IS
10 FORMAT(I3)
20 READ (I,END=999) IK,HR,(CASE(I),I=1,3),K
READ(I) ((AVG(I,N),I=1,200),N=1,3)
IF (IK.NE.IS) GO TO 30
L=L+1
WRITE(10) L,HR,(CASE(I),I=1,3),K
WRITE(6,150) L,HR,
150 FORMAT(IH,'SER 'I5,'X',HEART RATE ','P20.10,'AVG 'I2)
WRITE (10) ( (AVG(I,A),I=1,200),N=1,3)
GO TO 5
999 ENDFILE 10
END
*/

//LKED.SYSLIB DD DSN=SYS1.TESTLIB(HBS33GRS),DISP=SHR
//GO.FT09F001 DD DISP=OLD,
//DCB=(LRECL=200,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
//VOL=SER=000038,UNIT=3330,
//DSN=HB.H3H01.33320042.XYZ.DRAP
//GO.FT10F001 DD DISP=(OLD),VOL=SER=000038,UNIT=3330,
//DCB=(LRECL=200,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
//VOL=SER=HB.H3H01.33320042.XYZ.GRS,
//SPACE=(TMR,(15,2))
//GO.SYSIN DD *
013
014
030
057
060
117
120
126
140
150
154
155
157
179
//HBZ33PAG JOB (3H01.B020), 'HBZ2042810R ZIED'
//JOBPARM L=5, E=5, RESTPART
//PLOT EXEC PLOTCLG,REGION.GO=64K,TIME=3
//FORT.SYSIN DD *
COMMON /SWITCH/ IPATSW, IPAT(7), IPATSW, IPAT(7), LPATSW, LIPUSW
COMMON /PATREC/ UVW(201,3), ID, RATE, ROT(3), JUNK, IREC, ICHR, MORE
COMMON /SUN/SUN(3)
COMMON /SPX/ SPVEG(100), XCOR(65,3), TEN(31,3), THEAN(3), TVAR(3),
1 COR(3), NNM, NMM, INFO, LIM1(3), LIM2(3), LOC(3), IFLAG(3), PWAMP(3)
COMMON /WINDOW/HPEDIT
COMMON /BASEC/ UVWVEG(201,3), GAMMA, ISTART, ISTOP, X(201), IDSTOP
CALL PLOTS(0,0,8)
CALL INIT

100 CONTINUE
CALL GETPAT
IF (IOP.EQ.0) GO TO 100
CALL PIZTL
GO TO 100
CALL MAG
CALL CORR
DO 109 I=1,3
CALL SEARCH(I)
109 IF (IFLAG(1).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9000) ID, IREC, LOC(1),
1 COR(1)
IF (IFLAG(2).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9010) ID, IREC, LOC(2),
1 COR(2)
IF (IFLAG(3).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9100) ID, IREC, LOC(3),
1 COR(3)
IF (IFLAG(1).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9005) ID, IREC, LOC(1),
1 COR(1)
IF (IFLAG(2).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9015) ID, IREC, LOC(2),
1 COR(2)
IF (IFLAG(3).EQ.0.AND. LPATSW.EQ.0) WRITE (6,9205) ID, IREC, LOC(3),
1 COR(3)
IF (IFLAG(1).EQ.-1.AND. LPATSW.EQ.0) WRITE (5,2002) ID, IREC
IF (IFLAG(2).EQ.-1.AND. LPATSW.EQ.0) WRITE (5,2912) ID, IREC
IF (IFLAG(3).EQ.-1.AND. LPATSW.EQ.0) WRITE (5,9302) ID, IREC
IF (COR(1).GE.COR(2)) GO TO 99
IF (COR(2).GE.COR(3)) GO TO 105
29 IF (COR(1).GE.COR(3)) GO TO 103
101 M=3
GO TO 111
103 M=1
104 GO TO 111
102 M=2
111 L=LOC(M)
IF (LIPUSW.EQ.1) WRITE(12,9100) ID, IREC, IFLAG(M), LOC(M), COR(M), XCOR(L,H)
CALL PLTCON
GO TO 100
9000 FORMAT (15H0PATIENT NUMBER,110,2X,13HRECORD NUMBER,110,5X,
1 11HTEMPLATE 1 ,3X,
1 16M WAVE LOCATION=14,5X,22HCORRELATION AMPLITUDE=,F10.5)
DATA ITT, INFO/31, 31/
  MEAN = THE AVERAGE OF T(I)
TVAR = THE STANDARD DEVIATION OF T(I)
DATA TMEAN/0.161024, 0.161024, 0.161024/  
DATA TVAR/0.142283, 0.127895, 0.127896/
DATA UMIN/100/
DATA RHEDEPT/93/  
DATA IREC, MORE/0, 1/
DATA IFIRST/0/  
DATA IXINC, IYINC, BOTMIN, TOPMIN, TITLE, WT/3.5, 3*1.0, 0.20, 11.0/
END
SUBROUTINE INIT
COMMON /SWITCH/, IPLTSW, IPATSW, IPAT(7), LPSW, IFUNSW
COMMON /PWAVEC/, SPVEC(100), XCOR(85, 3), TET(31, 3), THEAN(3), TVAR(3),
1  COR(3), HMIN, ITT, INFO, LIML(3), LIM2(3), LOC(3), IFLAG(3), PWAMP(3)
COMMON /BASESH/, ISAVELP
COMMON /ZSUM/ZSUM(3)

CARD 1 PLOT CONTROL:  0 = NO PLOTS
1 = PLOT ALL PATIENTS
2 = PLOT ONLY THE PATIENTS WITH THE FOLLOWING ID
   TWO NUMBERS ON THE CARD
   1 = PATIENT ID NUMBERS BETWEEN THE NEXT TWO
   2 = SPECIFIC PATIENT ID NUMBERS LISTED
CARD 2 PATIENT CONTROL:  0 = PATIENT RECORD NUMBERS BETWEEN THE NEXT
   TWO NUMBERS ON THE CARD
   1 = PATIENT ID NUMBERS BETWEEN THE NEXT TWO
   2 = PATIENT RECORD NUMBERS BETWEEN THE NEXT TWO
CARD 3 LINE PRINTER CONTROL:  0 = NO CORRELATION VECTOR
   1 = CORRELATION VECTOR
CARD 4 SAVE CONTROL:  0 = NO PWAVE LOCATION SAVE
   1 = ALL PWAVE LOCATIONS SAVED
CARD 5 PWAMP(1):  MIN ALLOWED CORRELATION FOR SYMM TEMPLATE
CARD 5 PWAMP(2):  MIN ALLOWED CORRELATION FOR L SKEWED TEMPLATE
CARD 5 PWAMP(3):  MIN ALLOWED CORRELATION FOR R SKEWED TEMPLATE
CARD 6 LIML1(1), LIM2(1):  LIMITS ON PWAVE SEARCH FOR SYMM.
CARD 6 LIML1(2), LIM2(2):  LIMITS ON PWAVE SEARCH FOR L SKEWED
CARD 6 LIML1(3), LIM2(3):  LIMITS ON PWAVE SEARCH FOR R SKEWED
CARD 9 LINE PRINTER CONTROL FOR BASELINE:  0 = NO PRINT
   1 = PRINT

WRITE (6, 9010)
READ (5, 9000) IPLTSW, (IPLT(J), J=1, 7)
WRITE (6, 9010) IPLTSW, (IPLT(J), J=1, 7)
READ (5, 9000) IPATSW, (IPAT(J), J=1, 7)
WRITE (6, 9011) IPATSW, (IPAT(J), J=1, 7)
READ (5, 9001) LPSW
WRITE (6, 9012) LPSW
READ (5, 9001) IFUNSW
WRITE (6, 9013) IFUNSW
READ (5, 9002) PWAMP(1)
WRITE (6, 9014) PWAMP(1)
READ (5, 9002) PWAMP(2)
WRITE (6, 9017) PWAMP(2)
READ (5, 9002) PWAMP(3)
WRITE (6, 9027) PWAMP(3)
READ (5,9003) LIM1(1),LIM2(1)
WRITE (6,9016) LIM1(1),LIM2(1)
READ (5,9003) LIM1(2),LIM2(2)
WRITE (6,9018) LIM1(2),LIM2(2)
READ (5,9003) LIM1(3),LIM2(3)
WRITE(6,9037) LIM1(3),LIM2(3)
READ (5,9001) IBA SLP
WRITE (6,9020) IBA SLP
WRITE (6,9015)
RETURN

9000 FORMAT (I2,8X,7I10)
9001 FORMAT (I2)
9002 FORMAT (F10.6)
9003 FORMAT (2I3)
2010 FORMAT (8H CARD 1:,I3,14H PLOT CONTROL/,12X,
  1 18H= PLOT NO PATIENTS/,12X,19H= PLOT ALL PATIENTS/,12X,
  2 48H= PLOT ONLY THE PATIENTS WITH THE FOLLOWING ID S/,12X,
  3 12X,7I10)
9011 FORMAT (8H CARD 2:,I3,17H PATIENT CONTROL/,12X,
  1 54H= PATIENT RECORD NUMBERS BETWEEN THE FOLLOWING NUMBERS/,12X,
  2 12X,50H= PATIENT ID NUMBERS BETWEEN THE FOLLOWING NUMBERS/,12X,
  3 12X,44H= PATIENT ID NUMBERS FROM THE FOLLOWING LIST/,12X,
  4 7I10)
2012 FORMAT (8H CARD 3:,I3,22H LINE PRINTER CONTROL/,12X,
  1 33H= DO NOT PRINT CORRELATION VECTOR/,12X,
  2 26H= PRINT CORRELATION VECTOR)
9013 FORMAT (8H CARD 4:,I3,15H SAVE CONTROL/,12X,
  1 31H= DO NOT SAVE P WAVE LOCATIONS/,12X,
  2 24H= SAVE P WAVE LOCATIONS)
9014 FORMAT (8H CARD 5:2X,F10.6,2X,
  1 53H= MINIMUM THRESHOLD ON CORRELATION AMPLITUDE FOR P WAVE
  2 46H= DETECTION WITH TEMPLATE # 1 CROSS CORRELATION)
9015 FORMAT (1H1)
9016 FORMAT (8H CARD 7:,2I3,
  1 25H= LIMITS ON P WAVE SEARCH,
  2 35H= FOR TEMPLATE # 1 CROSS CORRELATION)
9017 FORMAT (8H CARD 6:2X,F10.6,2X,
  1 53H= MINIMUM THRESHOLD ON CORRELATION AMPLITUDE FOR P WAVE
  2 44H= DETECTION WITH TEMPLATE # 2 CROSS CORRELATION)
9027 FORMAT (8H CARD 6:2X,F10.6,2X,
  1 53H= MINIMUM THRESHOLD ON CORRELATION AMPLITUDE FOR P WAVE
  2 44H= DETECTION WITH TEMPLATE # 3 CROSS CORRELATION)
9018 FORMAT (8H CARD 6:,2I3,
  1 25H= LIMITS ON P WAVE SEARCH,
  2 33H= FOR TEMPLATE # 2 CROSS CORRELATION)
9019 FORMAT (1H1,24X,50H DPAT,/,12X,
  1 45H= HUCI TEMPLATE CROSS CORRELATION/,12X,
  2 12X,10H= NO PRINT/,12X,7H1= PRINT)
END
SUBROUTINE GSTPAT
COMMON /PATREC/ UVW(201,3),ID,PATE,ROT(3),JUNK,IREC,IOK,MORE
COMMON /SWITCH/ IPATSW,IPAT(7),IPATSW,IPAT(7),LPWSW,IPUNSW
C THE CURRENT TAPE HAS NO EOF MARK. INSTEAD THE ID OF THE LAST PATIENT
C IS KNOWN-LAST. THEREFORE, TO AVOID CRASHING OFF THE MACHINE AND LOSING
C THE PLOT, GETPAT CHECKS TO SEE IF THE CURRENT PATIENT IS THE LAST ONE
C ON THE TAPE AND IF SO SETS MORE=0.
LAST=9890
IF (MORE.EQ.0) GO TO 500
C READ A RECORD AND THEN DECIDE WHETHER TO USE IT
C IF THE PATIENT IS TO BE PROCESSED IOK=1
C INCREMENT THE RECORD COUNTER
IREC=IREC + 1
C READ THE IREC TH RECORD
CALL READPT
IOK=0
C PROCESSING BY RECORD LOCATION: IS THE CURRENT RECORD BEYOND THE
C UPPER LIMIT.
IF (IPATSW.EQ.0 .AND. IREC.GT.IPAT(2)) GO TO 500
C PROCESSING BY RECORD LOCATION: IS THE CURRENT RECORD BETWEEN THE LIMIT
IF (IPATSW.EQ.0 .AND. (IREC.GE.IPAT(1) .AND. IREC.LE.IPAT(2))) IOK=1
1  IOK=1
C PROCESSING BY PATIENT ID: IS THE CURRENT ID BETWEEN THE LIMITS.
C NO TEST FOR WHETHER THE CURRENT ID IS BEYOND THE UPPER LIMIT BECAUSE
C THE ID S MIGHT NOT BE ARRANGED IN SEQUENCE ON THE TAPE
IF (IPATSW.EQ.1 .AND. (ID.GE.IPAT(1) .AND. ID.LE.IPAT(2))) IOK=1
IF (ID.EQ.LAST) MORE=0
IF (IPATSW.NE.2) RETURN
C PROCESSING BY SPECIFIC PATIENT ID S. CHECK FOR ADDITIONAL PATIENTS
C (IF YES SET MORE=1) ONLY WHEN A DESIRED PATIENT IS FOUND.
DO 200 I=1,7
IF (ID.EQ.IPAT(I)) GO TO 300
200 CONTINUE
IF (ID.EQ.LAST) MORE=0
RETURN
300 CONTINUE
IOK=1
IPAT(I)=0
MORE=0
DO 400 I=1,7
IF (IPAT(I).NE.0) MORE=1
400 CONTINUE
IF (ID.EQ.LAST) MORE=0
RETURN
500 CONTINUE
CALL PLOT(5.,0.,3)
CALL WHERE(XX,YY,FAC)
CALL PLOT(XX,YY,999)
CALL EXIT
END
SUBROUTINE READPT
COMMON /PATREC/ UVW(201,3),ID,PATE,ROT(3),JUNK,IREC,IOK,MORE
CALL BUF(6)
EQUIVALENCE (ID,BUF(1))
INTEGER*2 CASE(3)
READ(9,END=100) ID,RATE,(CASE(I),I=1,3),K
C BUFFER IN WAS OK OR HAD PARITY ERROR
10 CONTINUE
READ(9,END=100) ((UVW(J,I),J=1,200),I=1,3)
C BUFFER IN WAS OK OR HAD PARITY ERROR
20 CONTINUE
RETURN
C READ EOF
100 CONTINUE
CALL PLOT(5.,0.,3)
CALL WHERE(XX,YY,FACT)
CALL PLOT(XX,YY,999)
ENDFILE 10
CALL EXIT
END
SUBROUTINE MAG
COMMON /PWAVEC/ SPVEC(100),XCOR(S5,3),TEN(31,3),TMEAN(2),TVAR(3),
1 COR(3),NMIN,ITTT,INFO,LIM1(3),LIM2(3),LOC(3),IFLAG(3),PWAMP(3)
COMMON /SUM/SUM(3)
COMMON /PATREC/ UVW(201,3),ID,RATE,ROT(3),JUNK,IREC,IRK,MORE
COMMON /WINDOW/MFEDPT
N=MFEDPT+(INFO-1)/2
DO 10 I=1,N
SPVEC(I)=SQRT( UVW(I,1)**2 + UVW(I,2)**2 + UVW(I,3)**2)
10 CONTINUE
RETURN
END
SUBROUTINE CORR
COMMON /PWAVEC/ SPVEC(100),XCOR(S5,3),TEN(31,3),TMEAN(3),TVAR(2),
1 COR(3),NMIN,ITTT,INFO,LIM1(3),LIM2(3),LOC(3),IFLAG(3),PWAMP(3)
COMMON /ZSUM/ZSUM(3)
COMMON /WINDOW/MFEDPT
DO 10 I=1,MFEDPT
XCOR(I,1)=0.0
XCOR(I,2)=0.0
XCOR(I,3)=0.0
10 CONTINUE
C WRITE (6,9001) SPVEC
C WRITE (6,9003) TEN
LENGTH=(INFO-1)/2
IPSEG=1+LENGTH
IPEND=MFEDPT
LEN=LENGTH+1
ISUM=LEN+1
ALPHA=1.0/FLOAT(INFO-1)
BETA=ALPHA/FLOAT(INFO)
C WRITE (6,9000) LENGTH,IPSEG,IPEND,LEN,ISUM,ALPHA,BETA,TMEAN,TVAR
J=IPSEG-1
C INITIALIZE SUM OF DATA ARRAY
CLCP=(SPVEC(31)-SPVEC(1))/30.
SUM=SPVEC(IPSEG)-(SPVEC(1)+CLCP*15.)
SUMDO=SUM**2
300 CONTINUE
DO 200 L=1,LENGTH
X=SPVEC(IPBEG-L)-(SPVEC(IPBEG-LENGTH)+SLOP*(LENGTH-L))
Y=SPVEC(IPBEG+L)-(SPVEC(IPBEG-LENGTH)+SLOP*(LENGTH+L))
SUM=SUMD+X+Y
SUMDSQ=SUMDSQ+X**2+Y**2
200 CONTINUE

C J IS A POINTER TO THE LOCATION OF THE SEGMENT (MIDDLE POINT)
J=J+1
SLOP=(SPVEC(J+15)-SPVEC(J-15))/30.
DO 401 K=1,3
401 SUM(K)=TEM(LEN,K)*(SPVEC(J)-(SPVEC(J-LENGTH)+SLOP*(LENGTH)))
DO 450 X=1,LENGTH
SLOP=(SPVEC(J+LENGTH)-SPVEC(J-LENGTH))/(2.*LENGTH)
X=SPVEC(J-I)-(SPVEC(J-15)+SLOP*(15-I-1))
Y=SPVEC(J+I)-(SPVEC(J+15)+SLOP*(15+I-1))
SUM(K)=SUM(K)+TEM((LEN-I),K)*X+TEM((LEN+I),K)*Y
450 CONTINUE

C WRITE(6,9002) SUM,SUNH,SUMDSQ
DO 950 K=1,3
XCOR(J,K)=ALPHA*(SUM(K)-THEAN(K)*SUMD)
XCOR(J,K)=XCOR(J,K)/(TVAR(K)*SQRT(ALPHA*SUMDSQ-BETA*SUMD**2))
950 CONTINUE
IF (J.EQ.IPEND) RETURN
IPBEG=IPBEG+1
SLOP=SPVEC(IPBEG+LENGTH)-SPVEC(IPBEG-LENGTH)
SLOP=SLOP/(1.*LENGTH)
SUMDSQ=SUMDSQ+X**2+Y**2
X=SPVEC(IPBEG+LENGTH)-SPVEC(IPBEG-LENGTH)
SLOP=SLOP/(2.*LENGTH)
SUM=SUMD+X+Y
GO TO 300

900 FORMAT (1H,9H LENGTH=,110,8H IPBEG=,110,8H IPEND=,110,
1 6H LEN=,110,9H ISUMP=,110,7H ALPHA=,E12.5,
2 7H BETA=,E12.5,8H TMEN=,E12.5,7H TVAR=,E12.5,/,)
9001 FORMAT (1H,8H SPVEC=,/,11(10F10.6,/,))
9002 FORMAT (1H,6H SUN=,E12.5,7H SUNH=,E12.5,9H SUMDSQ=,E12.5)
9003 FORMAT (1H,5H TENV=,/,11(10F10.6,/,))
END

SUBROUTINE SEARCH(K)
COMMON /PAVEC/ SPVEC(100),XCOR(85,3),TEM(31,3),THEAN(3),TVAR(3),
1 COR(3),NMIN,I MINT,ILMIN(3),ILMIN(3),LCM(3),ITRAN(3),IFLAG(3),PIAH(3)
COMMON /ZSUM/ZSUM(3)
COMMON /WINDOW/ MPEDP D
I=(INFO - 1)/2 + 1
ISTART=1 + IMIN(K)
IF (ISTART.LT.1) ISTART=1
ISTOP=MPEDP D - LIM2(K)
200 CONTINUE
COR(K)=0.0
LOC(K)=ISTART
IF (ISTART.GT.ISTOP) GO TO 1200
DO 300 I=ISTART,ISTOP
IF (XCOR(I,K).LT.COR(K)) GC TO 300
COR(K)=XCOR(I,K)
300 CONTINUE
LOC(I) = 1
CONTINUE
IF (LOC(K), NE, ISTART) GO TO 400
ISTART = ISTART + 1
GO TO 300

400 CONTINUE
IF (LOC(K), NE, ISTOP) GO TO 500
ISTOP = ISTOP - 1
GO TO 200

500 CONTINUE
IF (COR(K), LT, PWAMP(K)) IFLAG(K) = 0
IF (COR(K), GT, PWAMP(K)) IFLAG(K) = 1
RETURN

1200 CONTINUE
IFLAG(K) = -1
LOC(K) = 0
RETURN

END

SUBROUTINE PLTCON
COMMON /PHAVEC/ SPVEC(100), XCOR(85,3), TEM(31,3), THEAN(3), TVAR(3),
1 COR(3), HMM, ITIT, INFO, LIM1(3), LIM2(3), LOC(3), IFLAG(3), PWAMP(3)
COMMON /SSUM/ SUM(3)
COMMON /IPREC/ UUV(201,3), ID, RATE, ROT(3), JUNK, IREC, IOK, MORE
COMMON /PLOT/ T(35), XINC, YINC, BOTHM, TOPM, TITLE, H, YRESET,
1 NUMPLT, COUNT, IFIRST, LENPLT
COMMON /WINDOW/ NPLT(3)
IF (IPLTSW, EQ, O) RETURN
IF (IFIRST, NE, 0) GO TO 100
IFIRST = 1
NUMPLT = (HT - TOPMIN - BOTMIN - 1.0)/YINC
REAL * INTEGER GIVES REAL
YRESET = YINC * (NUMPLT - 1)
COUNT = NUMPLT
LENPLT = 75
IF (LENPLT, GT, NPLT) LENPLT = NPLT
DO 10 I = 1, LENPLT
T(I) = 0.004 * I + TITLE
CONTINUE
T(LENPLT + 1) = 0.0
T(LENPLT + 2) = 0.2
CALL PLOT (2.0, -11.0, -3)
CALL PLOT (0.0, 0.75, -3)
CALL PLOT (0.0, HT - 0.75 - TOPMIN - 1.0 - YINC/2.0, -3)
Y = YINC/2.0 + 0.5
CALL SYMBOL (0.0, Y, 0.1, 19, NEWPAT, CORRELATION, 0.0, 47)
Y = Y - 0.15
CALL SYMBOL (0.0, Y, 0.1, 8, HLLIM(1), =, 0.0, 8)
CALL SYMBOL (0.0, Y, 0.1, 8, LIM(1))
CALL NUMBER (0.9, Y, 0.1, 8, LIM, 0.0, -1)
CALL SYMBOL (1.5, Y, 0.1, 8, HLLIM(2), =, 0.0, 5)
CALL SYMBOL (3.0, Y, 0.1, 8, LIM, 0.0, -1)
CALL SYMBOL (3.0, Y, 0.1, 8, PWAMP(1), =, 0.0, 9)
\[ X = \frac{0.004 \times (MFEDPT - LOC(3))}{(LENPLT + 2)} \]

CALL NUMBER (0.4, -0.75, 0.1, XP, 0.0, 3)
CALL LINE (T(2), SPVEC(2), LENPLT=1, 1, 0, 0)
C
DRAW ZERO LINE
XP=TITLE/T(LENPLT + 2)
CALL PLOT (XP, 0.0, 3)
CALL PLOT (XP + 1.00, 0.0, 2)
C
MARK WAVES
IF (IFLAG(1).EQ.-1) GO TO 1000
IPLOC=LOC(1)
XP=T(IPLOC)/T(LENPLT + 2)
CALL PLOT (XP, 0.25, 2)
IP (IFLAG(1).EQ.0) CALL SYMBOL (XP=0.25, 0.25, 0.1, 2HS*, 0.0, 2)
IF (IFLAG(1).EQ.1) CALL SYMBOL (XP=0.15, 0.25, 0.1, 1HS, 0.0, 1)
1000 CONTINUE
IF (IFLAG(2).EQ.-1) GO TO 2000
IPLOC=LOC(2)
XP=T(IPLOC)/T(LENPLT + 2)
CALL PLOT (XP, 0.25, 2)
IP (IFLAG(2).EQ.0) CALL SYMBOL (XP=0.25, 0.25, 0.1, 2HL*, 0.0, 2)
IF (IFLAG(2).EQ.1) CALL SYMBOL (XP=0.15, 0.25, 0.1, 1HL, 0.0, 1)
2000 CONTINUE
IF (IFLAG(2).EQ.-1) GO TO 3000
IPLOC=LOC(3)
XP=T(IPLOC)/T(LENPLT + 2)
CALL PLOT (XP, 0.375, 3)
CALL PLOT (XP, 0.000, 2)
IP (IFLAG(3).EQ.0) CALL SYMBOL (XP=0.25, 0.25, 0.1, 2HR*, 0.0, 2)
IF (IFLAG(3).EQ.1) CALL SYMBOL (XP=0.15, 0.25, 0.1, 1HR, 0.0, 1)
3000 CONTINUE
C
DRAW TICK MARK
XP=T(MFEDPT - 20)/T(LENPLT + 2)
CALL PLOT (XP, -0.1, 3)
CALL PLOT (XP, 0.1, 2)
KOUNT=KOUNT - 1
IF (KOUNT.NE.0) CALL PLOT (0.0, -YINCR, -3)
IF (KOUNT.NE.0) RETURN
CALL PLOT (XIINCR, YRESET, -3)
KOUNT=NUMPLT
RETURN
CEND

/*
//LHED.SYS1.HOD DD DSN=SYS1.TESTLIB(HB233PAT),DISP=CHR
//GO.FTC9PC01 DD DISP=OLD,
// DCB=(LRECL=808,BLKSIZE=2048,RECFM=VBS,BUFNG=1),
// VOL=SER=0000088,UNIT=33230,
// DSNAME=H3H01.Z332042.XYZ.GRS
//GO.F10F001 DD DISP=(OLD),VOL=SER=000006,UNIT=3331C,
// DCB=(LRECL=032,BLKSIZE=1024,RECFM=VBS,BUFNC=1),
// SPACE=(TRK,(5,1)),DSNAME=H3H01.Z332042.PAT.GRS
//GO.SYSIN DD *
C1
01 00 01
0.0 0.0 0.0
000015 000015 000015
/*
* */
PIP>
C PROGRAM TO GENERATE 20 KL VECTORS AND COEFFICIENTS
C ISW CONTAINS RANGES OF SERIAL NUMBERS TO BE PROCESSED

REAL VECl(200), VEC2(200), VEC3(200), X(200,3)
REAL*8 RT(60), VEC(200,60)
REAL SH(200,3), SS(20100,3), SVEC(200)
INTEGER*2 CASE(26)
DIMENSION ISW(500), KLVEC(3), LDV(20), TX(2), TX(2), LAV(2)
DIMENSION ALFABT(26)
DIMENSION KLD(3)
DATA KLD/' l', 'L* 2', 'L* 3'/
DATA ALFABT/'M-A', 'M-B', 'M-C', 'M-D', 'M-E', 'M-F', 'M-G', 'M-H', 'M-I',
1 'M-J', 'M-K', 'M-L', 'M-M', 'M-N', 'M-O', 'M-P', 'M-Q', 'M-R', 'M-S',
2 'M-T', 'M-U', 'M-V', 'M-W', 'M-X', 'M-Y', 'M-Z'/
DATA KLVEC/'K-L', 'VECT', 'ORS'/
DATA LAV/'AVG', 'VECT'/
DATA LDV/'1', '2', '3', '4', '5', '6', '7',
1 '8', '9', '10', '11', '12', '13', '14', '15',
2 '16', '17', '18', '19', '20'/
ICHECK = 0

FORMAT('0 CHECK', '2X', 'I4', ' AT CHECKPOINT', '2X', 'A4')
LEN=200
LIT=(LEN*(LEN+1))/2
IK=0

DO 7 I=1,500
  ISW(I)=1
DO 12 I=1,500,2
  READ(5,11)(ISW(I+J-1), J=1,2)
  IF(ISW(I).LE.0) GO TO 13
  IK=IK+1
  K=I
  FORMAT(2I6)
CONTINUE
13 IFP=ISW(K+1)
C READ IN LIMITS ON HEART RATE IN /MIN
  READ(3,16) PATL, PATH
16 FORMAT(2F6,1)
  RATL=PATL/60.
  RATH=PATH/60.
C INITIALIZE RUNNING TOTAL
  DO 60 J=1,3
    DO 70 I=1,LEN
    SM(I,J)=0.
  DO 50 I=1,LNT
    SS(I,J)=0.
50 IADD=0
  READ(5,END=100) IDF, RATE, (CASE(I), I=1,3), X
  READ(2) (X(I, J), I=1,200), J=1,3
  IF2=IK*2
  DO 72 X=1,IK2,2
    IF ((RATE.LT.RATL).OR.(RATX.GT.RATH)) GO TO 30
12 CONTINUE
IF(ISW(K).LE.0)GO TO 80
IF((IDF.GE.ISW(K)).AND.(IDF.LE.ISW(K+1)))GO TO 72
10 CONTINUE
GO TO 80
72 IADD=IADD+1
WRITE(6,999)IDF,RATE,(CASE(I),I=1,3)
999 FORMAT(1H14,F20.10,3A2)
DO 150 J=1,3
DO 400 I=1,LEN
SVEC(I)=X(I,J)
400 CONTINUE
DO 200 I=1,LEN
200 SH(I,J)=SH(I,J)+SVEC(I)
DO 190 I=1,LEN
INDEX=(I*(I-1))/2
DO 180 K=1,I
IX=INDEX+K
180 SS(IX,J)=SS(IX,J)+SVEC(I)*SVEC(K)
IF(IDF.GE.IUP)GO TO 100
GO TO 80
100 DO 360 J=1,3
DO 360 I=1,200
IF(IADD.LE.0)GO TO 113
360 SH(I,J)=SH(I,J)/IADD
WRITE(10)RATL,RATH
DO 39 LEAD=1,3
39 WRITE(10)SH(J,LEAD),J=1,200
RATL=RATL*60.
RATH=RATH*60.
WRITE(6,45)RATL,RATH
45 FORMAT(1E18.13,'SET OF 20 % L VECTORS PER LEAD...'
1 'FOR VCG(S) WITH HR BETWEEN ',F6.1,' AND ',F6.1)
DO 40 LEAD=1,3
ENERGY=0.
K=0
DO 30 I=1,200
DO 30 J=1,I
K=K+1
SS(K,LEAD)=(SS(K,LEAD)-SM(I,LEAD)*SM(J,LEAD)*INDEX)/(IADD-1.)
30 IF(I.EQ.J)ENERGY=ENERGY+SS(K,LEAD)
CALL BASIS(200,20,SS(1,LEAD),RT,VEC,&101)
ICHECK=ICHECK+1
WRITE(6,341)ICHECK,ALFABT(5)
RSUM=0.
WRITE(6,50)LEAD,ENERGY
50 FORMAT(5H51LEAD,12,/(3X,3HENERGY=,E13.6//9X,5HVALUE,7X,
1 11HRUNNING SUM,3X,14HFRAC OF ENERGY)
DO 65 I=1,20
II=II+I
RSUM=RSUM+RT(II)
FPAC=RSUM/ENERGY
WRITE(6,71)II,RT(II),RSUM,FPAC
71 FORMAT(13,3(2X,E13.6))
WRITE(10)RT(II),VEC(J,II),J=1,200
55 CONTINUE
DO 206 J=1,20,2
DO 206 K=1,2
DO 206 L=1,200
DO 598 K=1,20,2
DO 601 L=1,200
VEC1(L)=VEC(L,K)
601 VEC2(L)=VEC(L,K+1)
CALL BGNSTEP(0,-1,KLVEC,3,0)
CALL BGNTRL(KLD(LEAD),1,14,2.5,0.,0,0.)
CALL NEWPEN(1)
CALL BGNSSCL(1,212.,1012.,0.,1.,0,0,3.1496,.7874,1.5748,.7874,
10.,0.,0.,0.,14)
TX(1)=212.
TX(2)=1012.
TY(1)=6.
CALL NEWPEN(2)
DO 590 J=1,2
TY(1)=TY(1)-2.
TY(2)=TY(1)
CALL BGPLOT(1,2,1,TX,TY,1,0.,0,0,0.)
590 CALL BOVPLOT(1)
CALL NEWPEN(3)
CALL BGNTRL(LDV(21-K),1,14,2.5,4.,0,0.)
CALL BGPLOT(1,200,1,2,VEC1,1,4.0,0,0,0.)
CALL BOVPLOT(1)
CALL BGNTRL(LDV(20-K),1,14,2.5,2.0,0,0.)
CALL BGPLOT(1,200,1,2,VEC2,1,4.0,0,0,0.)
CALL BOVPLOT(1)
598 CONTINUE
DO 605 I=1,200
605 VEC3(I)=SM(I,LEAD)+4.
TY(1)=4.
TY(2)=4.
CALL NEWPEN(1)
CALL BGNSTEP(0,-1,KLVEC,3,0)
CALL BGNTRL(KLD(LEAD),1,14,2.5,0.,0,0.)
CALL BGNSSCL(1,212.,1012.,0.,1.,0,0,3.1496,.7874,1.5748,.7874,
10.,0.,0.,0.,14)
CALL NEWPEN(2)
CALL BGNTRL(LAV,2,.14,2.5,4.,0,0.)
CALL BGPLOT(1,2,1,TX,TY,1,0.,0,0,0.)
CALL BOVPLOT(1)
CALL BGPLOT(1,200,1,2,VEC3,1,4.,0,0,0.)
CALL BOVPLOT(1)
40 CONTINUE
CALL ENDPLOT(1)
ENDFILE 1C
GO TO 112
101 WRITE(6,110)
110 FORMAT(14H  CANT DO IT)
112 CONTINUE
GO TO 113
113 WRITE(6,115)
115 FORMAT(17H  SORRY NO DATA)
CALL EXIT
STOP
END

SUBROUTINE BASIS(N, LL, R, ROOT, VEC, *)
REAL*8 ROOT(LL), LB, UB, OLB, OLB
REAL*8 DIAG(200), VEC(200, LL), SUB(200), SUDDIA(200), BETA(200)
REAL R(I)
INTEGER STM, CNT(60)
EXTERNAL STM

CALCULATES AND PUNCHES THE KARHUNEN BASIS. WRITES EIGENVALUE FRACTIONS
EIGENVALUES IN ROOT(I) I=1,LL INCREASING IN MAGNITUDE.
EIGENVECTORS IN VEC(J,I) I=1,H J=1,N CORRESPONDING TO THE EIGENVALUES.
DIMENSIONS ALLOW FOR ORDER UP TO 300, AND 20 EIGENVALUES.
R IS STORED AS A LINEAR ARRAY. MATRIX(I,J) STORED AS R(I*(I-1)/2+J)

CALL STRIL(N, R, DIAG, SUBDIA)
DO 5 I=1,N
  SUB(I) = SUBDIA(I)
  6  BETA(I) = SUBDIA(I)**2

COMPUTE ENERGY AND BOUNDS FOR EIGENVALUES

ENERGY = 0.
DO 5 I=1,N
  5  ENERGY = ENERGY + DIAG(I)

FIND LOWER BOUND

LB = ENERGY*1.0D-8
OLB = 0.0D0
UB = ENERGY*1.5
OUB = ENERGY*1.5
YF = 0.5
I = 0
44 : = H-STM(1, N, DIAG, SUBDIA, BETA, LB, !)
  I = I+1
  IF (I<500) 52, 53, 53
53 WRITE(6, 54)
  54 FORMAT(' Routines did not find a lower bound after 500 its.'
  RETURN 1
52 IF (H-LL) 41, 42, 43
41 OUB = LB
DECREASE LB
  LE = (1.-YF) * LB + YF * CLB
  GO TO 44
43 OLB = LB
INCREASE LB
  LB = YF * OUB + (1.-YF) * LB
  GO TO 44
42 CONTINUE

COMPUTE THE NUMBER OF EIGENVALUES BETWEEN LB AND UB

!
IF (M-60) 7,7,8
8 WRITE(6,9) LB,UB,M
9 FORMAT(' REQUIRED NUMBER OF EIGENVALUES IS MORE THAN 60'/'
1 ' LB=',E15.6/' UB=',E15.6' M=',I3)
   CALL EXIT
C
C COMPUTE THE M MOST SIGNIFICANT EIGENVALUES AND VECTORS OF R
C
7 CALL TRIST(N,LIB,UB,DIA,BETA,H,ROOT,VEC,CNT,&12)
  GO TO 11
12 WRITE(6,13) M
13 FORMAT(' CALC UNSUCCESSFUL. M=',I2//' ITS. PER VECTOR')
   WRITE(6,14) (CNT(I),I=1,M)
14 FORMAT(12)
   CALL EXIT
11 CONTINUE
C
C TRANSFORM BACK TO ORIGINAL CO-ORDINATE SYSTEM
C
M IS THE NUMBER OF ROOTS
C
ROOT CONTAINS THE EIGENVALUES
C
VEC CONTAINS THE EIGENVECTORS
C
CNT IS THE NUMBER OF ITERATIONS PER VECTOR
C
   CALL SBK(N,R,SUB,M,VEC)
C
20 CONTINUE
RETURN
END
SUBROUTINE TRIST(N,LIB,UB,C,BETA,H,ROOT,VEC,CNT,*)
C
INTEGER I,P,Q,R,M1,M2,N,M,CNT(N),FAIL,STM,J,K,S,ITS,GRP,NUM,
1 INT(200),MDUM,PDUM,PDUM,PDUM
REAL*8 C(N),B(N),VEC(200,N),BETA(N),LB,UB,ROOT(N),MA,
1 X(200),WU(200),Z(200),D(200),E(200),F(200),Y(200),
1 EPS1,EPS2,EPS3,EPS4,X1,X0,XU,U,V,8
EXTERNAL STM

C C CONTAINS DIAGONAL TERMS, B THE SUB-DIAGONAL TERMS, BETA THE SQUARES
C OF THE SUB-DIAGONAL TERMS OF THE SYMMETRIC TRIDIAG MATRIX, ORDER N
C
U NEED REQUIRE THE EIGENVALUES LESS THAN UB AND GREATER THAN LB.
C EIGENVALUES IN ROOT( ), EIGENVECTORS IN VEC( ), STORED COLUMNWISE.
C PROCEDURE FAILS IF M ON ENTRY IS LESS THAN THE NUMBER OF
C EIGENVALUES IN THE RANGE. A CALL WITH M+1 DETERMINES THIS NUMBER.
C ON EXIT M GIVES THE NUMBER OF EIGENVALUES ACTUALLY FOUND.
C NUMBER OF ITERATIONS PER VECTOR IS STORED IN CNT( ). PROCEDURE
C FAILS IF 5 ITERATIONS IS NOT ENOUGH. THEN CNT=6.
C NA IS THE RELATIVE MACHINE PRECISION. EPS1 IS THE ERROR TO BE
C TOLERATED IN THE SMALLEST EIGENVALUE.
C PROCEDURE REPLACES NEGLIGIBLE ELEMENTS OF B BY ZERO.
C
C DIMENSIONS ALLOW FOR MATRICES UP TO ORDER 300.
C
C THIS SUBROUTINE USES FUNCTION SUBPROGRAM STM
C
!M=2.22D-16
200 EPS1=(DAABS(LB)+DAABS(UB))**M
C
C LOOK FOR SMALL SUB-DIAGONAL ENTRIES
C
201 DO 202 I=2,N
   IF (DABS(B(I)) - MA*DABS(C(I-1))) < 203,203,202
   B(I) = 0.
   BETA(I) = 0.
202 CONTINUE
C
C CALCULATE THE NUMBER OF EIGENVALUES REQUIRED. FAIL IF MORE THAN M (IN)
C
204 R=M
   M = STM(1,N,C,B,BETA,UB,N) - STM(1,N,C,B,BETA,LB,N)
   IF (M-R) < 205,205,206
206 RETURN
C
C LOOK FOR INDEPENDANT SUB-MATRICES
C
205 Q=0
   R=1
250 P=Q+1
   NDUM=N-1
   IF (NDUM-P) < 209,209,208
208 DO 207 Q=P,NDUM
   IF (DABS(B(Q+1))) < 207,300,207
207 CONTINUE
C
209 Q=N
C
C LOOK FOR ISOLATED ROOT ON DIAGONAL
C
300 IF (P-Q) < 409,310,409
C
C ISOLATED ROOT
C
310 IF (LB-C(P)) < 311,311,312
311 IF (C(P)-UB) < 313,313,312
313 DO 314 I=1,N
314 VEC(I,R)=0.
   ROOT(R)=C(P)
   VEC(P,R)=1.
   CNT(R)=0
   R=R+1
312 GO TO 1200
C
C NUMBER OF ROOTS IN MATRIX P TO Q
C
409 M1=STM(P,Q,C,B,BETA,UB,N)+1
   M2=STM(P,Q,C,B,BETA,UB,N)
   IF (M1-M2) < 415,415,1200
415 CONTINUE
C
C FIND ROOTS BY BISECTION
X(0)=UB
   DO 416 I=M1,M2
   X(I)=UB
416 \ WU(I) = LB

LOOP FOR KTH EIGENVALUE

\[ MDUM = M2 - M1 + 1 \]
\[ DO 400 KDUM = 1, MDUM \]
\[ K = N2 - KDUM + 1 \]
\[ XU = LB \]
\[ IDM2 = K - H1 + 1 \]
\[ DO 420 IDM1 = 1, IDM2 \]
\[ I = K - IDM1 + 1 \]
\[ IF (XU - WU(I)) \]
\[ 419, 420, 420 \]

419 \ XU = WU(I) \]
GO TO 402

420 CONTINUE

402 IF (X0 - X(K)) \]
421, 421, 422

422 \ X0 = X(K) \]

421 IF (X0 - XU - 2.0*MA*(DABS(XU) + DABS(X0)) - EPS1) \]
423, 423, 424

424 \ X1 = (XU + X0) * 0.5 \]
\[ S = STM(P, Q, C, B, BETA, XL, N) \]
\[ IF (S - X) \]
425, 425, 426

425 \ IF (S - H1) \]
427, 427, 428

427 \ XU = X1 \]
\[ XU (M1) = X1 \]
GO TO 421

422 \ XU = X1 \]
\[ XU (S+1) = X1 \]
\[ IF (X(S) - X1) \]
430, 430, 430

430 \ X(S) = X1 \]
429 GO TO 421

426 \ X0 = X1 \]
GO TO 421

423 \ X(K) = (X0 + XU) * 0.5 \]
400 CONTINUE

FIND VECTORS BY INVERSE ITERATION

\[ \text{NORM} = DABS(C(P)) \]
\[ \text{PDUM} = P + 1 \]
\[ DO 500 I = PDUM, 0 \]

500 \ \[ \text{NORM} = \text{NORM} + DABS(C(I)) + DABS(B(I)) \]

EPS2 IS THE CRITERION FOR GROUPING
EPS3 REPLACES ZERO PIVOTS & EQUAL ROOTS ARE MODIFIED BY EPS3
EPS4 IS TAKEN VERY SMALL TO AVOID OVERFLOW

EPS2 = NORM * 0.001
EPS3 = MA * NORM
EPS4 = EPS3 * (Q - P + 1)
GRP = 0
S = P
DO 1150 K = H1, N2
IJS = 1
ROOT(P) = X(K)
X1 = X(K)
LOOK FOR CLOSE OR COINCIDENT ROOTS

IF(K=M1) 501,502,501
501 IF(X1-X0-EPS2) 503,504,504
503 GRP=GRP+1
GO TO 505
504 GRP=0
505 IF(X1-X0) 506,506,502
506 X1=X0+EPS3

502 U=EPS4/SQRT(Q-P+1.)
DO 507 I=P,Q
507 Z(I)=0

ELIMINATION WITH INTERCHANGES

U=C(P)-X1
V=B(P+1)
DO 708 I=PDUM,Q
BI=B(I)
IF(DABS(BI)-DABS(U)) 709,710,710
710 INT(I)=1
Y(I)=U/BI
XU=Y(I)
D(I-1)=BI
E(I-1)=C(I)-X1
IF(I=Q) 711,712,711
711 F(I-1)=S(I+1)
GO TO 713
712 F(I-1)=0.0
713 U=V-XU*E(I-1)
V=-XU*F(I-1)
GO TO 708
709 INT(I)=0
Y(I)=BI/U
XU=Y(I)
D(I-1)=U
E(I-1)=V
F(I-1)=0.0
U=C(I)-X1-XU*V
IF(I=Q) 714,709,714
714 V=BI+1
706 CONTINUE
IF(U) 715,716,715
715 D(Q)=U
GO TO 717
716 D(Q)=EPS3
717 E(Q)=0.0
F(Q)=0.0

BACK-SUBSTITUTION

850 IDM1=Q-P+1
DO 300 IDM2=1,IDM1
I = Q - IDM^2 + 1
Z(I) = (Z(I) - U*E(I) - V*F(I)) / D(I)
V = 0
U = Z(I)
800 CONTINUE

C ORTHOGONALISE WITH RESPECT TO PREVIOUS MEMBERS OF GROUP

L1 = R - GRP
L2 = R - L1
IF (L2 - L1) 904, 950, 950
950 DO 900 J = L1, L2
XU = 0.0
DO 901 I = P, Q
901 XU = XU + Z(I) * VEC(I, J)
DO 902 I = P, Q
902 Z(I) = Z(I) - XU * VEC(I, J)
900 CONTINUE
904 NORM = 0.0
DO 903 I = P, Q
903 NORM = NORM + DABS(Z(I))

C FORWARD SUBSTITUTION

IF (NORM - 1.0) 1001, 1002, 1002
1001 IF (ITS - 5) 1003, 1004, 1003
1004 NORM = 6
RETURN MERR
1003 IF (NORM) 1005, 1006, 1005
1006 Z(S) = EPS4
IF (S - Q) 1007, 1008, 1007
1007 S = S + 1
GO TO 1010
1008 S = P
GO TO 1010
1005 XU = EPS4 / NORM
DO 1011 I = P, Q
Z(I) = Z(I) * XU
1011 CONTINUE
1010 DO 1012 I = P DUM, Q
IF (INT(I) - 1) 1015, 1016, 1015
1016 U = Z(I - 1)
Z(I - 1) = Z(I)
Z(I) = U - Y(I) * Z(I)
GO TO 1012
1015 Z(I) = Z(I) - Y(I) * Z(I - 1)
1012 CONTINUE
ITS = ITS + 1
GO TO 1050

C NORMALISE SO THAT SUM OF SQUARES IS 1, & EXPAND TO FULL ORDER

1002 U = 0.0
DO 1101 I = P, Q
1101 U = U + Z(I) ** 2
XU=1.0/DSQRT(U)
DO 1102 I=P,Q
1102 VEC(I,R)=Z(I)*XU
PD2=P-1
IF(PD2=1) 1111,1110,1110
1110 DO 1103 IDM1=1,PD2
I=P-IDM1
1103 VEC(I,R)=0.0
1111 QDUM=Q+1
IF(I=QDUM) 1112,1113,1113
1113 DO 1104 I=QDUM,N
1104 VEC(I,R)=0.0
1112 CNT(R)=0
R=R+1
X0=X1
1150 CONTINUE
1200 IF(Q.LT.N) GO TO 250
RETURN
END

INTEGER FUNCTION STM(P,Q,D,E,F,LAMBDAN,N)
INTEGER P,Q,CNT
REAL*8 D(N),E(N),F(N),LAMBDAN,MA,X

C STM GIVES THE NUMBER OF EIGENVALUES LESS THAN LAMBDAN IN THE SUB-MATR:
C OF THE SYMM TRIDIAG MATRIX DEFINED BY D,E,F.
C D IS THE DIAGONAL, E IS THE SUBDIAGONAL.
C F CONTAINS THE SQUARES OF THE TERMS IN D.
C
MA=2.22E-16
CNT=0
X=1.0
DO 105 I=P,Q
X=D(I)-LAMBDAN-DABS(E(I)/MA)
IF(X) 101,102,101
101 X=D(I)-LAMBDAN-P(I)/X
102 X=D(I)-LAMBDAN-E(I)/MA
103 IF(X) 104,105,105
104 CNT=CNT+1
105 CONTINUE
STM=CNT
RETURN
END

SUBROUTINE SBK(N,A,E,M,Z)
INTEGER N,M,I,J,X,L
REAL*8 E(N),Z(200,H),S
REAL A(1)

C TRANSFORMS THE EIGENVECTORS IN Z BACK TO THE ORIGINAL COORD SYSTEM.
C USING THE TRANSFORMATION INFO (A) PRODUCED BY STRI1. E CONTAINS THE
C SUBDIAGONAL OF THE TRIDIAGONAL MATRIX PRODUCED BY STRI1
C
DO 2 I=2,11
XT=I*(I-1)/2
IF(Z(I)) 4,2,4
2 L=I-1
\[ H = E(I) \times A(K+L) \]

\[ \text{DO } 3 \ J = 1, M \]
\[ S = 0.0D0 \]
\[ \text{DO } 5 \ K = 1, L \]
\[ S = S + A(K+K) \times Z(K,J) \]
\[ S = S / H \]
\[ \text{DO } 6 \ K = 1, L \]
\[ Z(K,J) = Z(K,J) + S \times A(K+K) \]
\[ 3 \text{ CONTINUE} \]
\[ 2 \text{ CONTINUE} \]

C

RETURN
END

SUBROUTINE STRI1 (NORDER, A, DIAG, SUBDIA)

C

DOUBLE PRECISION DIAG, SUBDIA, RELTOL, F, G, H
INTEGER NDIM, NORDER, I, II, NP1, IM1, K, JP1
DIMENSION A(1), DIAG(NORDER), SUBDIA(NORDER)
DATA RELTOL/2.44E-63/

C TRANSFORMS A SYMMETRIC MATRIX A TO TRIDIAGONAL FORM.
C TRANSFORMATION INFO IS STORED IN A ON OUTPUT.
C A(I,J) IS STORED AS A(I*(I-1)/2+J).
C
C SAVE DIAGONAL ELEMENTS.
C 10 DO 1 = 1, NORDER
   10 DIAG(I) = A(I*(I+1)/2)

C PERFORM REDUCTIONS IN REVERSE ORDER OF SUBSCRIPTS.
   NP1 = NORDER + 1
   DO 120 II = 1, NORDER
      I = NP1 - II
      KI = I*(I-1)/2
      WRITE(6,9999) II, NORDER
      9999 FORMAT(217)
   120 H = 0.0D0
   IM1 = I - 1
   IF(IM1.LE.0) GO TO 30
   DO 20 K = 1, IM1
      20 H = H + A(K+K)*2
   C IF H IS TOO SMALL, UNDERFLows MAY JEOPARDIZE ORTHOGONALITY.
C SO SKIP TRANSFORMATION.
   30 IF(H.GT.RELTOL) GO TO 40
   SUBDIA(I) = 0.0D0
   GO TO 110

C 40 F = A(K+IM1)
   G = DSIGN(DSQR(H), F)
   SUBDIA(I) = G
   H = H - F*G
   A(K+IM1) = F - G

C F = 0.0D0
DO 30 J=1,IM1
  KJ=J*(J-1)/2
  C
  FORM ELEMENT OF A*U
  G=0.0D0
  DO 50 K=1,J
  50  G=G+A(KJ+K)*A(KI+K)
  C
  IF (IM1.LT.JP1) GO TO 70
  DO 60 K=JP1,IM1
  60  KK=K*(K-1)/2
     G=G+A(KK+J)*A(KI+K)
  C
  FORM ELEMENT OF P
  70  G=G/H
  SUBDIA(J)=G
     F=F+G*A(KI+J)
  80  CONTINUE
  C
  H=F/(2.*H)
  C
  FORM REDUCED A
  DO 100 J=1,IM1
  F=A(KI+J)
  G=SUBDIA(J)-H*F
  SUBDIA(J)=G
  C
  KJ=J*(J-1)/2
  DO 20 K=1,J
     A(KJ+K)=A(KJ+K)-F*SUBDIA(K)-G*A(KI+K)
  20  CONTINUE
  C
  100 CONTINUE
  C
  110 H=DIAG(I)
  ID=I*(I+1)/2
  DIAG(I)=A(ID)
  A(ID)=H
  120 CONTINUE
  C
  RETURN
END
/*/
C PROGRAM TO GENERATE 20 RL VECTORS AND COEFFICIENTS
C ISW CONTAINS RANGES OF SERIAL NUMBERS TO BE PROCESSED
C
REAL VEC1(50), VEC2(50), VEC3(50), X(200,3)
REAL*8 RT(30), VEC(50,50)
REAL SM(50,3), SS(1275,3), SVEC(50)
INTEGER*2 CASE(3)
DIMENSION IGW(500), KLVEC(4), LDV(20), TY(2), TX(2), LAV(2)
DIMENSION ALFABT(26)
DIMENSION KLD(3)

INTEGER IF(1000), ILC(1000)
DATA KLD/'L# 1, 'L# 2, 'L# 3'/
DATA ALFABT/'M-A', 'M-B', 'M-C', 'M-D', 'M-E', 'M-F', 'M-G', 'M-H', 'M-I',
1 'M-J', 'M-K', 'M-L', 'M-M', 'M-N', 'M-O', 'M-P', 'M-Q', 'M-R', 'M-S',
2 'M-T', 'M-U', 'M-V', 'M-W', 'M-X', 'M-Y', 'M-Z'/
DATA KLVEC/'K-L', 'VECT', 'ORS '/
DATA LAV/'AVER', 'VECT'/
DATA LDV/' 1', ' 2', ' 3', ' 4', ' 5', ' 6', ' 7',
1 ' 8', ' 9', '10', '11', '12', '13', '14', '15',
2 '16', '17', '18', '19', '20'/
ICHCK=0
341 FORMAT('0 CHECK', '2X, I4', ' AT CHECKPOINT', '2X, A4')
LEN=50
LMT=(LEN*(LEN+1))/2
IK=0
DO 7 I=1, 500
7 ISW(I)=-1
DO 12 I=1, 500, 2
READ(5,11)(ISW(I+J-1), J=1,2)
IF(ISW(I).LT.0)GO TO 13
IK=IX+1
K=I
11 FORMAT(2I6)
12 CONTINUE
13 IUP=ISW(K+1)
C READ IN LIMITS ON HEART RATE IN */MIN
READ(5,10) RATL, RATH
10 FORMAT(2F6.1)
RATL=RATL/60.
RATH=RATH/60.
C INITIALIZE RUNNING TOTAL
14 DO 60 J=1,3
DO 70 I=1, LEN
60 DD(I,J)=0.
70 SK(I,J)=0.
DO 60 I=1, LMT
60 IADD=0
000 READ P WAVE LOCATIONS
80 READ(10,END=100)ID,IREC,IFLG,LOC,COR,XCOR
20 READ(9,END=100)IDP,RATE,(CASE(I),I=1,3),K
    READ(9) ((X(I,J),I=1,200),J=1,3)
    IF(IDF.NE.IDP)GO TO 20
    DO 61 J=1,3
    DO 61 I=1,200
    IF((X(I,J)) .GT. 10.) GO TO 80
    CONTINUE
    IX2=IX*2
78 DO 10 K=1,IK2,2
    IF((RATE.LT.RATL).OR.(RATE.GT.RATH))GO TO 80
    IF(IFLG.NE.1)GO TO 80
    IF(ISW(K).LE.0)GO TO 80
    IF((IDF.GE.ISW(K)).AND.((IDF.LE.ISW(K+1))))GO TO 72
10 CONTINUE
    GO TO 80
72 IADD=IADD+1
    IDP=IDP+1
    ILC(IDP)=ID
    WRITE(6,999)IDP,RATE,(CASE(I),I=1,3)
999 FORMAT(1H,i4,F20.10,3A2)
    DO 130 J=1,3
    DO 400 I=1,LEN
    LOC=I+ILC(IDP)-31
    IF(LOC.LT.1)GO TO 941
    SVEC(I)=K(LOC,J)
    GO TO 400
941 SVEC(I)=0.
400 CONTINUE
    DO 200 I=1,LEN
    SM(I,J)=SM(I,J)+SVEC(I)
    DO 190 I=1,LEN
        INDX=(I*(I-1))/2
    DO 190 K=1,I
        IX=INDX+K
190 SS(IX,J)=SS(IX,J)+SVEC(I)*SVEC(K)
    IF(IDF.GE.1UP)GO TO 100
    GO TO 80
100 DO 360 J=1,3
    DO 360 I=1,30
    IF(IADD.LE.0)GO TO 113
360 SM(I,J)=SM(I,J)/IADD
500 ECHO FWAVE LOCATIONS

    DO 730 I=1,IDP
    WRITE(6,740)IP(I),ILC(I)
    IF(I .EQ. IDP)GO TO 730
    IF(IP+1) .GT. IF(I) GO TO 730
730 WRITE(6,750)
    CALL EXIT
740 FORMAT(215)
750 FORMAT(1SH ERROR-OUT OF ORDER)
WRITE(4) RATL, RATH
DO 39 LEAD=1,3
39 WRITE(4) (SM(J, LEAD), J=1,50)
RATL=RATL*60.
RATH=RATH*60.
WRITE(6,45) RATL, RATH
45 FORMAT(1H, 'SET OF 20 K L VECTORS PER LEAD.. P WAVES..',
1 'FOR VCG(6) WITH HR BETWEEN ',F6.1,' AND ',F6.1)
DO 40 LEAD=1,3
ENERGY=0.
K=0
DO 30 I=1,50
DO 30 J=1,1
K=K+1
SM(K, LEAD)=(SM(K, LEAD)-SM(I, LEAD)*SM(J, LEAD)*IADD)/(IADD-1.)
30 IF(I.EQ.J) ENERGY=ENERGY+SS(K, LEAD)
CALL BASIS (50,20, SS(1, LEAD), RT, VEC, &101)
ICHECK=ICHECK+1
WRITE(6,341) ICHECK, ALFABT(8)
RSUM=0.
WRITE(6,50) LEAD, ENERGY
50 FORMAT(5H1LEAD, 12, /3X, SENERGY=.B13.6//3X, SVALUE, 7X,
1 11RUNNING SUM, 3X, 14LFRAC OF ENERGY)
DO 65 I=1,20
II=II-I
RSUM=RSUM+RT (II)
FRAC=RSUM/ENERGY
WRITE(6,71) II, RT (II), RSUM, FRAC
71 FORMAT(1J, 2X, RT (II), VEC (J, II), J=1,50)
CONTINUE
DO 206 J=1,20,2
DO 206 K=1,2
DO 205 L=1,50
205 VEC (L, J+K-1)=VEC (L, J+K-1)-2.0*K+6.
DO 598 K=1,20,2
DO 601 L=1,50
VEC1 (L)=VEC (L, K)
501 VEC2 (L)=VEC (L, K+1)
CALL BGNSTP (0, -1, KLVEC, 4, 0)
CALL BGNTTL (L, LEAD), 1., 14, 2.5, 0., 0., 0.
CALL NEWSEN(1)
CALL BGMINC (1, 000., 0200., 0., 1., 0., 0., 3.1495, 7.674, 1.5740, .7874,
10., 0., 0., 0., 10.)
TX (1)=0.
TX (2)=200.
TY (1)=6.
CALL NEWSEN (2)
DO 590 J=1,2
TX (1)=TY (1)-2.
TX (2)=TX (1)
CALL DPLT1 (1, 2, 1, TX, TY, 1., 0., 0., 0., 0.)
590 CALL DOWL (1)
CALL BGNPEN(3)
CALL BGNPTT(LLD(V21-K),1,14,2.5,4.,0,0.)
CALL BPL0T(1,050,1,2,VEC1,1,4.0,0,0,0.)
CALL B0VPLT(1)
CALL BGNPTT(LLD(V20-K),1,14,2.5,2.0,0,0.)
CALL BPL0T(1,050,1,2,VEC2,1,4.0,0,0,0.)
CALL B0VPLT(1)
500 CONTINUE
DO 505 I=1,50
505 VEC3(I)=SH(I,LEAD)+4.
TY(1)=4.
TY(2)=4.
CALL BGNPEN(1)
CALL BGNPTH(K-L,VEC,4,0)
CALL BGNPTT(LLD(LEAD),1,14,2.5,0.,0,0.)
CALL B0C1SCL(1,0001,0203,0310,0313,1326,.7074,1.5748,.7874,
10.,0.,0.,0.,14)
CALL BGNPEN(2)
CALL BGNPTT(LLD,2,14,2.5,4.,0,0.)
CALL BPL0T(1,2,1,tx,ty,1,0.,0,0,0.)
CALL B0VPLT(1)
CALL BPL0T(1,050,1,2,VEC3,1,4.,0,0,0.)
CALL B0VPLT(1)
40 CONTINUE
CALL BNDPLT(1)
ENDTWLC 4
GO TO 112
101 WRIT5(6,110)
110 FORMAT(14H CANT DO IT)
112 CONTINUE
GO TO 110
113 WRIT5(6,115)
115 FORMAT(17H SORRY NO DATA)
118 CALL EXIT
STOP
END

SUBROUTINE BASIS(N,LL,R,ROOT,VEC,)

END

GO.FT0SFO01 DD DISP=OLD,
GO.FT0SFO02 DD DISP=OLD,
GO.FT0SFO03 DD DISP=OLD,
GO.FT0SFO04 DD DISP=OLD,
GO.FT0SFO05 DD DISP=OLD,
GO.FT0SFO06 DD DISP=OLD,
GO.FT0SFO07 DD DISP=OLD,
GO.FT0SFO08 DD DISP=OLD,
GO.FT0SFO09 DD DISP=OLD,
GO.FT0SFO10 DD DISP=OLD,
GO.FT0SFO11 DD DISP=OLD,
GO.FT0SFO12 DD DISP=OLD,
GO.FT0SFO13 DD DISP=OLD,
GO.FT0SFO14 DD DISP=OLD,
GO.FT0SFO15 DD DISP=OLD,
GO.FT0SFO16 DD DISP=OLD,
GO.FT0SFO17 DD DISP=OLD,
GO.FT0SFO18 DD DISP=OLD,
GO.FT0SFO19 DD DISP=OLD,
GO.FT0SFO20 DD DISP=OLD,
C PROGRAM TO COMPUTE PATTERN VECTOR BASED ON SEGMENTED KL
C AND TO RECONSTRUCT A TEST WAVEFORM FROM ITS EXPANSION

C REAL*8 PVEC(200, 20, 3)
C REAL*8 RT(20)
C REAL*8 PVEC(50, 20, 3)
C REAL PHN(200, 3), BUF(900, 50)
C REAL PHN(50, 3)
C REAL WAVPH(200, 3)
C INTEGER LP(900), LZ(900), LOC(900)
C INTEGER IT(75), IX(75), IX(75)
C INTEGER*2 CASE(3)
C COMMON FVEC, PVEC, PHN, PHN, WAVPH, IT, IX, ITU, LENPV, RATE
C COMMUN BUF, N, IX
C CALL PLOTS(0, 0, 3)
C CALL PLOT(2, -12, -3)
C CALL PLOT(0, 5, -3)
C CALL FACTOR(75)
C DO 16 A = 1, 900
  151  IZ(I) = 0
C
C READ P WAVE MEANS AND KL VECTORS
C READ(4) RATL, RATH
C
C READ P WAVE BASIS VECTORS
C DO 50 I = 1, 3
  50  READ(4) (PHN(J, I), J = 1, 50)
C DO 20 I = 1, 3
  20  J = 1, 20
C READ(4) (RT(J), (PVEC(X, J, I), K = 1, 50)
C READ(4, END = 2) (RT(J), J = 1, 20)
C
C READ FULL WAVE MEANS AND KL VECTORS
C 2 READ(10) RATL, RATH
C DO 30 I = 1, 3
  30  READ(10) (PHN(J, I), J = 1, 200)
C READ FULL WAVE BASIS VECTORS
C DO 10 I = 1, 3
  10  J = 1, 20
C READ(10) (RT(J), (PVEC(X, J, I),X = 1, 200)
C CONTINUE
C
C READ WAVE LOCATIONS (FLOC BIDS UP CONTAINING NUMBER READ IN)
C FLOC = 1
C IF Patient Number 0, END OF WAVE LOC
C 30  READ(11, END = 70) LP(HLOC), IREC, IFLAG, LOC(HLOC), COR, NCOR
C IF(LP(HLOC).EQ. 0) GO TO 70
C 60  FORMAT(314)
C HLOC = HLOC + 1
C GO TO 30
C 70  HLOC = HLOC - 1
C
DO 171 I=1,900
READ(5,50)K
IF(K.EQ.0)GO TO 172
I2(K)=1
171 CONTINUE
172 CONTINUE
C INITIALIZE POINTER TO CURRENT LOCATION (BEGINNING OF LIST)
   LOCPR=1
C READ ITAU, LOCATION OF SEGMENT BREAK
READ (5,30) ITAU
WRITE(5,220) ITAU
220 FORMAT(24HSEGMENT BREAK AT SAMPLE,I4)
   FORMAT(15)
READ FORMAT OF PATTERN VECTOR
READ(5,30) LENPV
   FORMAT(5)
READ LENGTH OF PATTERN VECTOR
READ(5,20) LENPV
   FORMAT(3)
READ COMPOSITION OF EACH MEMBER OF PATVEC
   FORMAT(5)
IPV(I)= 0100 RATE
   FORMAT(5)
   0200 P-R INTERVAL
   FORMAT(5)
   1YXX XXVECTOR, Y LEAD, PRWAVE
   FORMAT(5)
   2YXX XX VECTOR, Y LEAD, FULL WAVE (QRST)
   FORMAT(5)
   DO 100 I=1,LENPV,5
100 =I+4
   FORMAT(5)
   100 READ(5,110)((IT(J),IY(J),IX(J)),J=1,I4)
110 FORMAT(5(211,12,1X))
C NOW INPUT STAGE IS OVER, BEGIN PROCESSING OF PATIENTS
C
READ PATIENT TO BE PROCESSED. IF IP NOT EQUAL TO 0, PROCESS
   FORMAT(5)
   PATIENT WITH THAT ID. IF P=0, PROCESS FIRST IP PATIENTS.
   FORMAT(5)
   IF IP LESS THAN 0, QUIT
   FORMAT(5)
   I=0
200 READ(5,90) IP,NP
   FORMAT(5)
   IF(IP .LT. 0) GO TO 140
   FORMAT(5)
   IF(IP .EQ. 0) GO TO 120
   FORMAT(5)
PROCESS PATIENT IP
   FORMAT(5)
   FIXD PRWAVE LOCATION
   FORMAT(5)
   150 IF(LP(LOCPR) .EQ. IP) GO TO 130
   FORMAT(5)
   LOCPR=LOCPR+1
   FORMAT(5)
   IF(LOCPR .LE. NLOC) GO TO 150
   FORMAT(5)
   C COMES HERE IF LOCATION OR WAVEFORM CANT BE FOUND
   FORMAT(5)
   170 FORMAT(25H MISSING DATA FOR PATIENT ,I5)
   FORMAT(5)
   140 CONTINUE
   FORMAT(5)
   CALL WHERE(X,Y,FACT)
   CALL PLOT(X,Y,295)
   FORMAT(5)
   DO 285 K=1,5
   FORMAT(5)
WRITE(6,190)K
WRITE(6,190) (BUF(K,I),I=1,20)
WRITE(6,190) (BUF(K,I),I=21,40)
WRITE(6,190) (BUF(K,I),I=41,60)

FORMAT( IH,14)
FORMAT( IH ,2OF6.3)
WRITE(12)K, (BUF(K,I),I=1,60)

CONTINUE
CALL EXIT

C LOCATION FOUND, LOCK FOR WAVEFORM
READ(9,END=160)JP, RATE, (CASE(I), I=1, 3), K
READ(9) (WAVEFM(J, I), J=1, 200), (I=1, 3)
IF (JP .NE. IP) GO TO 130

C WAVEFORM FOUND, PROCESS IT
CALL SEG(R(JP, LOC(LOCPTR)))

C SEE IF ANY MORE PATIENTS TO DO
GO TO 200

C IP=0, SO PROCESS NEXT NP PATIENTS

DO 210 IPAT=1, NP
READ(9,END=220)JP, RATE, (CASE(I), I=1, 3), K
READ(9) (WAVEFM(J, I), J=1, 200), (I=1, 3)

C CHECK TO SEE IF WAVEFORM AND LOC ARE THE SAME PATIENT
IF (JP .EQ. LP(LOCPTR)) GO TO 250

C IF JP IS LESS THAN LP READ NEXT WAVEFORM
IF (JP .LT. LP(LOCPTR)) GO TO 260

C IF JP IS GREATER THAN LP GO TO NEXT WAVEFORM LOC
LOCPTR=LOCPTR+1
IF (LOCPTR .LE. NLOC) GO TO 270

C COMES HERE IF NO MATCH POSSIBLE
WRITE(6,280)

FOR 120 IPAT=1, NP

END SUBROUTINE SEG(JP, LOC)

REAL* E PVEC(200, 20, 3), PvEC(50, 20, 3)
COMMON PVEC, PvEC, PXN, PKN, WAVEFM, IT, IX, ITA, LENPV, RATE
COMMON BUF, H, IZ
REAL PAVEC(75), PXH(200, 3), PNN(50, 3), WAVEFM(200, 3)
REAL PIXM(200, 3), COST(3), BUF(505, 15), IZ(50)
INTEGER IX(75), IX(75), IX(75)

C FIRST FORM HYBRID MENU FOR EACH LEAD
DO 16 I=1, 3
C INITIALIZE PORTION WHICH MIGHT NOT BE OVERWRITTEN
DO 20 J=1,ITAU
20 HNN(J,I)=0.
C FIND DISTANCE (TO LEFT) P MEAN MUST BE SHIFTED SO THAT MEAN
C P WAVE LOCATION (31ST SAMPLE) IS LINED UP WITH LOCATION FOR
C THIS PATIENT
ISHFT=31-LOC
C DON'T MOVE P WAVE MEAN
C DO NOT WRITE BEFORE FIRST SAMPLE
IBGN=1
IF(ISHFT.GT.0)IBGN=1+ISHFT
DO 30 J=IBGN,50
30 HNN(J-ISHFT,I)=PMN(J,I)
C MOVE QRS MEAN INTO HYBRID
DO 40 J=ITAU,200
40 HNN(J,I)=PMN(J,I)
10 CONTINUE
C WRITE(6,50) JP
50 FORMAT(1H0,2H1PATIENT ,15,15H PATTERN VECTOR)
60 FORMAT(15)
C MEAN IS READY, FIND PATTERN VECTOR ELEMENT BY ELEMENT,
C COMPUTE RECONSTRUCTION AS YOU GO

C CHECK TYPE
M=M+1
DO 70 I=1,LEMPV
70 IF(IT(I).EQ.0.AND.IY(I).EQ.1)GO TO 30
IF(IT(I).EQ.0)GO TO 70
IF(IT(I).EQ.1)GO TO 100
C
C FOLLOWING CODE IS FOR QRS SEGMENT
LEAD=IY(I)
NUM=IX(I)
C COMPUTE MEMBER OF PATTERN VECTOR
PATVEC(I)=0.
DO 110 J=ITAU,200
110 PATVEC(I)=PATVEC(I)+(WAVEF(J,LEAD)-PMN(J,LEAD))*FVEC(J,NUM,LEAD)
C PRINT OUTPUT
WRITE(6,120)I,LEAD,NUM,PATVEC(I)
120 FORMAT(16,1H LEAD) I,NUM,PATVEC(I)
C COMPUTE RECONSTRUCTION AND ADD TO HYBRID MEAN
DO 130 J=ITAU,202
130 HNN(J,LEAD)=HNN(J,LEAD)+PATVEC(I)*FVEC(J,NUM,LEAD)
C RETURN FOR NEXT ELEMENT OF PATVEC
GO TO 70
C
C FOLLOWING CODE P WAVE SEGMENT
LEAD=IY(I)
NUM=IX(I)
C COMPO TE PATVEC MEMBER
PATVEC(I)=0.
IEND=IATA-1
C MAKE SURE SHIFT DOES NOT CAUSE OVERWRITE OF QRST SEGMENT
IF(ISHFT .LT. 0) IEND=ISHFT+IEND
IF(IEND .GT. 50) IEND=50
DO 150 J=ISHFT,IEND
150 PATVEC(I)=PATVEC(I)+(WAVEFM(J-ISHFT,LEAD)-PMN(J,LEAD))
* PVEC(J,NUN,LEAD)
C PRINT OUTPUT
C WRITE(6,160) I,LEAD,NUN,PATVEC(I)
160 FORMAT(14,11H PHASE LEAD,12,7H VECTOR,13,7H VALUE,F15.7)
BUF(K,I)=PATVEC(I)
C COMPUTE RECONSTRUCTION AND ADD TO HYBRID MEAN
DO 170 J=ISHFT,IEND
170 HNN(J-ISHFT,LEAD)=HNN(J-ISHFT,LEAD)+PATVEC(I)*PVEC(J,NUN,LEAD)
AND GO TO NEXT MEMBER
. GO TO 70
C COMES HERE FOR RATE AS ELEMENT OF PATVEC
C GO PATVEC(I)=RATE
C OUTPUT
C WRITE(6,150) I,PATVEC(I)
150 FORMAT(14,11H HEART RATE,12,7H VALUE,F15.7)
BUF(K,I)=PATVEC(I)
C DOES NOT AFFECT RECONSTRUCTION, SO GO TO NEXT ELEMENT
GO TO 70
C COMES HERE FOR P-R INTERVAL AS MEMBER
C PATVEC(I)=(83-LOC)*.004
C WRITE(6,200) I,PATVEC(I)
200 FORMAT(14,13H P-R INTERVAL,10X,7H VALUE,F15.7)
BUF(K,I)=PATVEC(I)
C THIS IS END OF LOOP
C CONTINUE
C FIND RECONSTRUCTION ERROR
CALL FCO S(T,WAVEF M,NI3,COST,TCOST,200)
END R ET U R N
C 111 SUBROUTINE FCO S(K,REC,COST,TCOST,M)
REAL X(1,3),REC(1,3),COST(3)
TCOST=0.
TO 10 J=1,M
C=ABS(X(J,1)-REC(J,1))
COST(I)=COST(I)+C
10 TCOST=TCOST+C
WRITE(6,30) TCOST
30 FORMAT(1H,14H PATIENT COST ,F15.7)
RETURN
END
SUBROUTINE PLOT(X,REC,ITIT,IP,COST,TCOST,K)
COMMON FVEC,PVEC,TM,TM,PM,PAFEM,IT,IX,ITAU,LENPV,RATE
REAL X(N,3),REC(N,3),COST(3),T(202),SCL(202),X(3),Y(3)
INTEGER LAX(3)
DATA LAX/'LD X','LD Y','LD Z'/
DO 30 I=1,N
30 T(I)=I-1.
T(I+2)=N/6.
CALL PLOT(0.,9.,-3)
CALL SYMBOL(2.,0.,2.,13HSEGMEN T RECON,0.,13)
CALL NUMBER(2.,-3.,2.,FLOAT(ITIT),0.,-1)
CALL SYMBOL(2.7,-3.2,4,IT,0.,4)
CALL SYMBOL(1.6,-3.2,7,IT,0.,7)
CALL NUMBER(2.6,-3.2,5,IT,0.,-1)
CALL SYMBOL(4.6,-3.2,4,TCOST,0.,4)
SAVE=X(N-1,3)
SAV2=X(N,3)
CALL SCALE(X,2.5,N*3-2,1)
SCL(I+1)=X(I-1,3)
SCL(I+2)=X(N,3)
X(N,3)=SAVE
X(I,3)=SAV2
DO 45 I=1,3
RM(I)=0.
DO 45 J=1,6
DIF=ABS(X(J,I)-REC(J,I))
IF (DIF.LT.RM(I)) GO TO 45
RM(I)=DIF
V(I)=X(J,I)
45 CONTINUE
WRITE(6,100)(RM(I),I=1,3)
100 FORMAT(1H,'ABSOL. MAX. ERROR IS ',3F10.5)
WRITE(6,200)(V(I),I=1,3)
200 FORMAT(1H,'AT ORIGINAL VALUE OF ',3F10.5)
DO 10 I=1,3
CALL PLOT(0.,-3.,-3)
DO 20 J=1,11
SCL(J)=X(J,I)
20 CONTINUE
CALL HEYPEM(1)
CALL AXIS(0.,0.,4,TIME,-4,6.,0.,0.,8/5.)
CALL AXIS(0.,0.,LAX(1),4.2.5,50.,SCL(+1),SCL(+2))
CALL SYM2SCL(4.,2.15.,4TCOST,0.,6)
CALL NUMBER(4.5,1.5,COST(I),3.,4)
CALL LINE(T,SCL,0.,1,0,13)
DO 40 J=1,3
40 SCL(J)=REC(J,I)
10 CALL LINE(T,SCL,0.,1,3,13)
CALL PLOT(0,0,-3)
RETURN
END

/*
   //GO.FT09F001 DD DISP=OLD,
   // DCD=(LRECL=808,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
   // DSN=HB.H3H01.2332042.XYZ.GRS
   //GO.FT10F001 DD DISP=OLD,
   // DCD=(LRECL=808,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
   // DSN=HB.H3H01.2332042.KLM.GRS
   //GO.FT04F001 DD DISP=OLD,
   // DCD=(LRECL=208,BLKSIZE=2048,RECFM=VBS,BUFNO=1),
   // DSN=HB.H3H01.2332042.PKL.GRS
   //GO.FT11F001 DD DISP=OLD,
   // DSN=HB.H3H01.2332042.PAT.GRS
   //GO.FT12F001 DD DISP=(NEW,CATLG,DELETE),
   // VOLSER=0000062,UNIT=3330,
   // SPACE=TRK,(20,1)),DSN=HB.H3H01.2332042.REC.GRS
   //GO.SYSIN DD *
*/

0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0050
0010

1101,1102,1103,1104,1105
2101,2102,2103,2104,2105
2101,2107,2108,2109,2110
2111,2112,2113,2114,2115
1201,1202,1203,1204,1205
2201,2202,2203,2204,2205
2206,2207,2208,2209,2210
2211,2212,2213,2214,2215
1301,1302,1303,1304,1305
2301,2302,2303,2304,2305
2306,2307,2308,2309,2310
2311,2312,2313,2314,2315
001 5
0 0 5 0
-1 0

*/
C THIS PROGRAM SORTS AN ARRAY OF PATTERN VECTORS
C ACCORDING TO DIAGNOSTIC CLASSES

DIMENSION Buf(10, 50, 50), IV(10), ID(10, 50)
DIMENSION IS(10), REJ(50), Z(50)

IE=0
DATA IS/1, 4, 7, 8, 12, 13, 15, 23, 25, 29/
DO 10 I=1, 10
DO 12 J=1, 60
DO 12 K=1, 50
Buf(I, J, K)=0.
10 L(J)=0.
12 CONTINUE
DO 29 I=1, 10
29 IV(I)=0
L=1
IR=1

15 READ(5, 20) IC
20 FORMAT(212)
"="IC
IF(IS(I).EQ.0) GO TO 30
DO 22 I=1, 10
IF(IS(I).EQ.0) GO TO 122
22 CONTINUE
IF(IS(I).EQ.29) GO TO 23
CONTINUE
255 READ(9, END=30) IT, (REJ(I), I=1, 60)
GO TO 23
23 IF(IS(I).GE.10) GO TO 256
IR=IR+1
K=IS(IT)
74 DO 120 I=1, 10
74 IF(IS(IT).EQ.0) GO TO 122
120 CONTINUE
122 IC=I
IV(I)=IV(I)+1
IF=IV(I)
READ(5, END=30) IA, (REJ(K), K=1, 50)
ID(IA, IF)=IA
DO 32 K=1, 10
 TF(K, IA, IF)=REJ(K+5)
SUP(IA, IF, K+10)=REJ(K+25)
SU(K, IF, K+20)=REJ(K+45)
32 CONTINUE
25 GO TO 15
55 DO 50 J=1, 10
IF(J+200) GO TO 40
DNUM=DNUM+1
DO 55 J=1, DNUM
WRITE(10) ID(I, J), (SUP(I, J, K), K=1, 50)
WRITE(6,101)IS(I),ID(I,J)
35  CONTINUE
   IF(I.EQ.10)GO TO 444
   WRITE(10)IB,(Z(K),K=1,60)
444 WRITE(6,101)ID
   CONTINUE
   ENDFILE 10
101 FORMAT(1H,2I4)
   END

/*
 //LKED.SYSMOD DD DSN=SYS1.TESTLIB(HE33SOR),DISP=CHR
 //GC.PT0F001 DD DISP=OLD,
 // DSNAME=HD.H3H01.2332042.REC.ALL
 //GO.PT10F001 DD DISP=(OLD),
 // DCB=(LRZCL=156,BLCIDE=2040,RECNO=2000,DSVNO=1),
 // DSNAME=HD.H3H01.2332042.SOR.ALL,
 // UNIT=3330,VOL=800062,SPACE=(TRK,(10,2))
 //GO.SYSl DD *
22
55
99
12
13
99
// HB333PPV JOB (3H01,5020), 'HB30420910R RZED'
// JOBPAR: L=5, E=0, RESTART
// /* PPV EXEC PLOTCLG, REGION.GO=124K, TIME=6
// /* PORT.SYSIN DD *

C THIS PROGRAMS PLOTS AN ASSEMBLY OF PATTERN VECTORS
C ACCORDING TO DIAGNOSTIC CLASSES

DIMENSION BUF(10,60,12), IV(10), ID(10,50)
INTEGER*4 TIT1(4), TIT2(4)
INTEGER*2 FOUR, ONE, SEVH, SIGHT, CEE, DEE, EF, GEE, ETH, EYE
INTEGER*2 SYM

DIMENSION IS(10), X(60), Y(60), REJ(60), Z(60)
DATA TIT1/'FEAT','URE','SPAC','E 12'/
DATA TIT2/'FEAT','URE','SPAC','E 23'/
DATA FOUR/'4' /
DATA ONE/'1' /
DATA SEVH/'7' /
DATA SIGHT/'8' /
DATA CEE/'C' /
DATA DEE/'D' /
DATA EF/'E' /
DATA GEE/'G' /
DATA ETH/'H' /
DATA EYE/'I' /
IT=-1
ID=0
DATA IS/1,4,7,8,12,15,21,23,25,29/
DO 10 I=1,10
DO 12 J=1,60
DO 12 K=1,12
BUF(I,J,K)=0.
10 Z(J)=0.
12 CONTINUE
DO 20 I=1,10
20 IV(I)=0
L=1
IT=0
DATA IS/1,4,7,8,12,15,21,23,25,29/
READ (5,20) IC
FORMAT (2I2)
:=IC
IF (I.EQ.0)GO TO 50
DO 22 I=1,10
IF (IS(I).EQ.0) GO TO 122
22 CONTINUE
IF (I.EQ.20)GO TO 23
CONTINUE
255 READ (5,255) IT, (ADJ(I), I=1,50)
GO TO 15
23 IF (I.EQ.10)GO TO 265
IF (I.EQ.1)
:=IS(IP)
DO 120 IP=1,10
IF (I.EQ.IS(IP)) GO TO 122
120 CONTINUE
GO TO 99
220 IF(JC.NE.9)GO TO 940
SYN=ETH
GO TO 99
940 IF(JC.NE.10)GO TO 999
SYN=EYE
99 CALL EPLT(1,IV(JC),0,X,Y,1,0.,1,SYN,0.07)
CALL BOVPLT(1)
100 CONTINUE
CALL ENDPLT(0)
501 CONTINUE
CALL DNDPLT(1)
999 STOP
END

/ * 
//LKD.SYSMOD DD DSN=SYS1.TESTLIB(DB233PPV),DISP=SHR
//GO,FT600001 DD DISP=OLD,
//DSNAME=HB.H3001.2333042.REC.ALL
//GO,SYSIN DD *
22
26
99
99
12
13
22
99
99
15
97
99
21
25
29
99
99
13
REAL X(300,30), PI(465,30), X(30,30), P(465,30), PHI(30,30), Z(30)
REAL T(30), L, PHI(465,30), PHI(465,30), PHI(30,30), DU(62), ZIG(30)
REAL BKK(100), ALP(10), W1(30), WK2(30), DETOLD(30)
REAL E(30,30)
REAL CSMH(30)
REAL DS(30)
INTEGER IPS(30), JP(300)
DIMENSION ICI(10), IC2(10), IT(10), DI(10)
REAL ITP(100), N1(30), N2(30)
INTEGER IHE(30)
COMMON / LIN / L
COMMON / COST / TCOST, TCOST, TCCOST, TCOST
* , AER, CSR, TSR
COMMON / DISM / DISM
COMMON / SORI / INS(300,12), CMS(300,12), ISET(300), INX(300),
* , ICLS(300), IPRS(30)
COMMON / CH / PASS, CHAK, USUN, IFOLD, IFNEW
* , CJ, OF, OH
COMMON / FUSI / FUSI, FSC
COMMON / UP / ALPHA(10), M, L, LUPDS
EQUIVALENCE (SU(1), IPB)
DATA ALP/3., 2., 2., 2., 3., 3., 3., 2., 3., 3., 3., 3./
DATA / 0*/0./
DATA INH/30*/0/
DATA INT/0/
DATA INH/900*/0./
DATA SIG/0.2587, 0.0226, 0.1360, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
* , 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226, 0.0226,
DATA IPS1/30*/0./
DO 1111 I=1,10
1111 READ ALPA(1)=ALP(1)

READ AND ECHO INPUT OPTIONS
READ(5,10) MDIN, IFPS, NBS
READ(5,10) IFIN, IFSC
IF(IFPS .EQ. 1) GO TO 112
IF(10) IFSC=1
112 CONTINUE
10 FORMAT(3I3)
READ(5,20) L
20 FORMAT(2F7.3)
WRITE(6,300) MDIN, NBS, L
300 FORMAT(17H3 OF DIMENSIONS, L, /11H3 CM. PAST, IP/4H & L, =, F13.3)
IF(IFPS .EQ. 0) GO TO 310
WRITE(3,320) (ALPA(I), I=1,10)
320 FORMAT(18HMIFFUSSING IN EFFECT/ ON ALPA. =, 10F7.2)
WRITE(6,111) IFZM,IFSC
111 FORMAT(12H F U Z Z Y M E A N = ,I1,/12H F U Z Z Y C O V = ,I1)
GO TO 330
310 WRITE(6,340)
340 FORMAT(11H U N D O F U Z Z I N G)
330 CONTINUE
C SET INITIAL FILE DIRECTION
IFOLD=0
IFLEN=10
IPTR=1
INST=1
IF(IFPS .EQ. 0) GO TO 73
GO TO 71
73 CONTINUE
DO 72 I=1,10
72 ALPHA(I)=1.
71 CONTINUE
IPASS=0
IPSPTR=0
READ(5,10) IMPTSN,IOUTSN
READ(5,10) IUPDSN,ISIGS
READ(5,33) EIGS
33 FORMAT(F10.6)
READ(5,10) IMHS
READ(5,10) IDISS
WRITE(6,1) IMPTSN
WRITE(6,42) EIGS
WRITE(6,61) IMHS
WRITE(6,81) IDISS
F1 FORMAT(9H D I S T. S N : ,AN, 1H = ,I2, 1H N (O=M I N A L,
* /2X,13H 1 = L - E X P (-D/2)/2X,10H2 = E X P (D/2))
51 FORMAT(6H IMHS, 1H = ,I2, 16H (O=V G, 1=U P D S))
43 FORMAT(10H INITIAL "F I G H T = ,F10.6)
2 FORMAT(10H B I G "S W I T C H = ,I2, 16H (O=.001, 1=Z I G))
C SET VALUE FOR INITIAL DIAGONAL
IF(ISIGS .EQ. 1) GO TO 3
DO 3 I=1,30
3 EIG(I)=1.
C CONTINUE
IPTR=2
IP(1)=0
ICLC(1)=0
IPSPTR=1
INST=1
73 IF(IFPS .EQ. 0) GO TO 730
READ FROM CARDS
READ(5,350) IPE,(EI(I),I=2,11)
IF(IPD .EQ. 9999) GO TO 40
GO TO 50
360 FORMAT(I4,10F6.2)
C READ FROM TAPE
370 READ(11,END=40) IPB,(DU(I),I=2,61)
50 IP(IPS PTR)=IPB
ICL (IPS PTR)=IPS PTR
IPS PTR=IPS PTR+1
IP(IPB .EQ. 0) GO TO 50
C ADD TO MEAN
DO 30 I=1,NDIM
HN(IPSPTR,I)=(HN(IPSPTR,I)*IH(IPSPTR)+SU(I+1))/(IH(IPSPTR)
+1.)
60 X(IPPTR-1,I)=SU(I+1)
IH(IPSPTR)=IH(IPSPTR)+1
IMT=IMT+1
IPSTM(IPSPTR)=1
INSH=0
GO TO 70
30 IPS PTR=IPS PTR+1
INSH=1
GO TO 70
60 IF(INSH .EQ. 0) GO TO 62
C READ MEANS FROM CARDS
C DO 64 I=1,IPSPTR
64 READ(5,65) UK(I,J),J=1,NDIM
65 FORMAT(F10.4)
GO TO 62
C SET NUMBER OF PATIENTS AND CLASSES
62 IP=IPPTR-1
NC=IPS PTR
IPOT=(NDIM*(NDIM+1))/2
IHC=0
DO 100 IC=1,NC
IF(IPSPTR(IC) .EQ. 0) GO TO 100
C WRITE MEANS
C WRITE(6,1143) IC,(UK(IC,I),I=1,NDIM)
1143 FORMAT(16H SEED FOR CLASS ,IF(/5,10F11.5))
IHC=IHC+1
C INITIALIZE COVARIANCE
DO 450 I=1,NDIM
PJ(I,IC)=0.
450 PJU(I,IC)=0.
DETOLD(IC)=1.
DC 110 I=1,NDIM
DETOLD(IC)=DETOLD(IC)*EIG(I)*SIGS
IHN=(IP(INDX,IC)+1)/2
PJU(INDX,IC)=EIG(I)*SIGS
110 PJU(INDX,IC)=1./(EIG(I)*SIGS)
C INITIALIZE COSTS

130 NPASS=NPASS+1
WRITE(6,350) NPASS
ICOST=0
ITCOST=0
TER=0.
CER=0.
COMM=0.
TCOST=0.

350 FORMAT(SH1PASS,I6)
C GET COVARIANCE COMPUTED DURING PREVIOUS PASS
DO 120 IC=1,IC
IF(IPSW(IC) .EQ. 0) GO TO 120
IF(IPSW(IC))=999
DO 140 I=1,MTOT
P(I,IC)=PN(I,IC)
140 PHI(I,IC)=PHI(I,IC)
DO 150 I=1,NDIM
X(IC,I)=IN(IC,I)
SHX(IC)=0.
SSHX(IC)=0.
IF(IUPSSN .NE. 2) GO TO 120
C REDUCE COVARIANCE SO DETERMINANT IS CONSTANT
C
IDTSW=0
IF(NPASS .EQ. 1) IDTSW=1
203 CONTINUE
INDX=0
DO 121 J=1,NDIM
DO 121 I=1,NDIM
INDX=INDX+1
ISD=(I*(I-1))/2+J
IF(J .GT. 1) ISD=(J*(J-1))/2+1
121 DUMMY(INDX)=P(ISD,IC)
CALL MINV(DUMMY,NDIM,DT,M1,M2)
IF(IDTSW .NE. 1) GO TO 201
DETOLD(IC)=DT
WRITE(6,202) IC,DT
202 FORMAT(1SH ORIGINAL DET CLASS,I3,1H=,F15.6)
GO TO 120
201 CONTINUE
WRITE(6,204) IC,DT
204 FORMAT(1SH NEW DET CLASS,I3,1H=,F15.6)
DDT=(ABS(DDETCLD(IC)/DT))**(1./NDIM)
DO 122 I=1,MTOT
P(I,IC)=P(I,IC)*DDT
PI(I,IC)=PI(I,IC)/DDT
PHI(I,IC)=PHI(I,IC)/DDT
122 PHI(I,IC)=PHI(I,IC)/DDT
IDTSW=1
GO TO 203

120 CONTINUE
RETURN
REWIND 10

C INITIALIZE ERROR MEASURES
    DO 91 IC=1,NC
    DO 91 ICC=1,NC
91  E(IC,ICC)=0.
    OJ=0.
    OF=0.
    ON=0.
    UNAX=0.
    USUN=0.

C REVERSE DATA FLOW DIRECTION FOR IS

C
    IS=IFOLD
    IFOLD=IFNEW
    IFNEW=IS
    ICL=1
    ICSN=0

C GO THROUGH ALL PATHS TO COMPUTE NEW IS

    DO 150 IP=1,IP
      IF(IP(IP) .NE. 0) GO TO 170

C NEW CLASS, PRINT COSTS FOR LAST CLASS
      IF(ICSN .EQ. 0) GO TO 6
      IF(ICL .EQ. 1) GO TO 6
      OTP=ICCOST
      CTP=OTP/INCREMENT(ICL-1)
      CCOST=CCOST/INCREMENT(ICL-1)
      WRITE(6,31) CER
      31 FORMAT(24H0 CLASS ADDITIVE ERROR =,1X,F6.4)
      WRITE(6,5) CTP,CCOST
      5 FORMAT(19H AVG CLASS ERROR =,F7.5/21H AVG WEIGHTED ERROR =,
           * F12.5)

C REinitialize
    ICSN=0.
    CER=0.
    CCOST=0.
    WRITE(6,100) ICL
    100 FORMAT(7H0 ,CLASS ,13)
       ICL=ICL+1
    ICSN=0.
    GO TO 160

170  AL=0.
    ICSN=1
    BE=0.

C COMPUTE DISTANCE TO EACH CLUSTER
    TO 190 IC=1,NC
      IF(D(IP)'(IC) .LT. 0.) GO TO 190
      CALL DIST(IP,PI(1,IC),X,J,NDIM,IC,IC)
    ICSN=IC
    IF(D(IC) .LT. 0.) GO TO 410
    IF(D(IC) .LT. 10.**(-10)) GO TO 200
IF(IPS .LE. 0) GO TO 190
IF(ALPHA(IC) .GE. 2.9) GO TO 180
AI=1./(2.*D(IC)+AI)
GO TO 150

190 BE=1./(3.*D(IC))**(.5)+BE
190 CONTINUE
IF(IPZS .HE. 0) GO TC 210
C IF NO FUZZING, SET LARGEST D TO 1, ALL OTHERS TO 0
DLG=1000000.
DO 220 IC=1,HC
IF(IPSW(IC) .LE. 0) GO TO 220
IF(DLG .LT. D(IC)) GO TO 220
DLG=D(IC)
ICS=IC
220 CONTINUE
GO TO 200
C COMPUTE MEMBERSHIP FUNCTIONS

210 DO 230 IC=1,HC
IF(IPSW(IC) .LE. 0) GO TO 230
H=(BE**2+4.*AI)**(.5)
IF(ALPHA(IC) .C-T.2.9)GO TO 225
M(IC)=(1./AI+(BE/(2.*(AI**2)))*(BE-H))/(2.*D(IC))
C - 0  TC 230
225 W(IC)=((-BE)/(2.*AI)+H/(2.*AI))/(3.*D(IC))
230 CONTINUE
DO 231 IC=1,HC
IF(IPSW(IC) .EQ. 0) GO TC 231
IF(W(IC) .30. 0.) GO TO 231
C UPDATE NEW VERSIONS OF COV, INVERSE COV
CALL UPDT(ICS,IP,PX,PPN,PN1,IC,PN1,IC,PME,SH,SH,SH)
231 CONTINUE
250 CONTINUE
C WRITE PATIENT AND 10 NEAREST CLUSTERS
CALL OUTPT(IP,IP,IPSW,NCR,ICL,IC,IPSW,IPSW,ICL,ICL)
ICI=ICL(IP)
DO 92 IC=1,HC
IF(IPSW(IC) .LE. 0) GO TO 92
Z(IC,ICI)=W(IC,ICI)+W(IC)
92 CONTINUE
GO TO 160
C THESE LINES FOR EFFECTIVELY 0 D----AVOID OVER OR UNDERFLOW
260 DO 240 IC=1,HC
W(IC)=0.
W(IC)=1.
CALL UPDT(ICS,IP,PX,PPN,PN1,ICS,ICS,PN1,ICS,PN1,ICS)
GO TO 250
240 CONTINUE
156 CONTINUE
C ALL PATIENTS ADDED IN
OTP=ICOST
OTP=OTP/IMEM(ICL-1)
CCOST=CCOST/IMEM(ICL-1)
WRITE(6,31) CER
WRITE(6,5) OTP,CCOST
WRITE(6,32) TER
32 FORMAT(24H0 TOTAL ADDITIVE ERROR =,1X,F9.4)
OTP=TCOST
OTP=OTP/INT
TCOST=TCOST/INT
WRITE(6,7) OTP,TCOST
7 FORMAT(18H0 AVG TOTAL COST =,F7.5/
* 27H AVG WEIGHTED TOTAL COST =,F12.5)
C MAKE SURE INVERSE COVS ARE ACTUALLY INVERSE OF COVS
DO 400 IC=*1,NC
IF(PSW(IC) .EQ. 0) GO TO 400
CALL PSM(PN(1,IC),PNICl,IC),TMP,NDIM)
ICS-IC
D O 510 1=1,NDIM
INDX=d*(1+1) > / 2
IF(ABS(Cl.-TMP(INDX))  .GT. .01) GO TO 520
510 TMP(INDX)=0.
DO 530 I=1,NTOT
INDX-I
IF(ABS(TMP(I))  .GT. .0001) GO TO 520
530 CONTINUE
400 CONTINUE
560 IF(NPASS .GE. 2) WRITE(6 ,550) WMAX,WSUM
550 FORMAT(12H0MAX CHNG = ,E15.7,12H SUM CHNG = ,E15.7)
C WRITE CLUSTER TIGHTNESS MEASURES
OH—OH/NDIM
OF=OF/NDIM
WRITE(6,570) OJ,OF,OH
570 FORMAT(3H J»,24X,E14.7/26H PARTITION COEFFICIENT(F) = ,1X,
* E14.7/22H AVG CLASS ENTROPY(H)= ,5X,E14.7//12H CLASS MEANS)
C WRITE MEANS
DO 580 IC=1,NC
IF(IPSWIC) .EQ. 0) GO TO 580
WRITE(6,590) IC,(MN(IC,I),1=1,NDIM)
590 FORMAT(13,3(10(IX,E10.4)/))
580 CONTINUE
WRITE(6,101)
101 FORMAT(12H0COVARIANCES )
C WRITE COVARIANCES
DO 102 IC=1,NC
IF(IPSW(IC) .EQ. 0) GO TO 102
DO 103 I=1,NDIM
IBGN=(I*(I-1))/2+1
IEN=IBGN+I-1
IF(I .EQ. 1) WRITE(6,104) IC,PN(1,IC)
103 CONTINUE
WRITE(6,104)
104 FORMAT(1H0,12,IX,F9.4)
103 IF(I .NE. 1) WRITE(6,105) (PN(J,IC),J=IBGN,1EN)
105 FORMAT (4X,10(F9.4,1X))
102 CONTINUE
C WRITE E (ERROR MEASURE)
C WRITE(6,82)
82 FORMAT(27H5E(C1,C2)=AVG OF WEIGHTS OF,
* /34H MEMBERS OF CLASS C1 TO CLUSTER C2 )
   DO 99 IC=1,NC
   IF(IPSW(IC) .EQ. 0) GO TO 99
   ICI=IMEM(IC)
   DO 93 JC=1,NC
   IF(IPSW(JC) .EQ. 0) GO TO 93
   E(IC,JC)=E(IC,JC)/ICI
   93 CONTINUE
   99 CONTINUE
   WRITE(6,94)
94 FORMAT(1H ,6(7H C1, C2,4X,1HE, 6X))
   IR=1
   DO 95 I=1,NC
   IF(IPSW(I) .EQ. 0) GO TO 95
   DO 96 J=1,NC
   IF(IPSW(J) .EQ. 0) GO TO 96
   BT(IR)=E(I,J)
   ICI(IR)=I
   IC2(IR)=J
   IR=IR+1
   IF(IR .NE. 7) GO TO 96
   WRITE(6,97) (ICI(III),IC2(III),BT(III),III=1,6)
   IR=1
   96 CONTINUE
   95 CONTINUE
   IF(IR .EQ. 1) GO TO 98
   IR=IR-1
   WRITE(6,97) (ICI(III),IC2(III),BT(III),III=1,IR)
97 FORMAT(1H ,6(I3,IIX,I3,IIX,F8.6,2X))
98 CONTINUE
C COMPUTE AND WRITE DIV AND BATT DIST S
   WRITE(6,11)
11 FORMAT(1H50//4(29H C1 C2 BATT DIV ))
   IR=1
   NC1=NC-1
   DO 12 I=1,NC1
   IF(IPSW(I) .EQ. 0) GO TO 12
   I=I+1
   DO 14 J=1,NDIM
   M1(J)=MN(I,J)
   14 CONTINUE
   DO 15 K=1,NDIM
   IF(IPSW(J) .EQ. 0) GO TO 13
   DO 15 K=1,NDIM
   M2(K)=MN(J,K)
   CALL BATT(PN(1,I),PN(1,J),PN1(I),PN1(J),M1,J,M2,NDIM,
* DIV, BAT, TMP
I1(I0) = I
I2(I0) = J
BT(I0) = BAT
DV(I0) = DIV
IR = IR + 1
IF (IR .EQ. 5) CALL WRIC(1, IC1, IC2, BT, DV, IR)
13 CONTINUE
12 CONTINUE
IF (IR .NE. 1) CALL WRIC(1, IC1, IC2, BT, DV, IR)

WRITE SECOND (RESORTED) LIST OF PATIENTS
DO 21 IC = 1, NC
IF (IPSW(1) .EQ. 0) GO TO 21
IF (IPRS(I0) .EQ. 999) GO TO 21
WRITE(6, 22) IC

22 FORMAT (9HCLUSTER ,I2)
K = IFRS(1)
WRITE(6, 23) JP(K), ICLS(K), (IWS(K, I), OWS(K, I), I = 1, LIM)
K = INX(K)
IF (K .NE. 999) GO TO 24
21 CONTINUE
GO TO NEXT PASS
IF (NPASS .LT. NPS) GO TO 130
CALL EXIT

410 WRITE(6, 420) D(I0), IC
420 FORMAT (18HNEGATIVE DISTANCE,1X,F10.4,6H IC = ,I4)
CALL EXIT

520 WRITE(6, 540) IC, IND, TMP(IND)
540 FORMAT (28H*WARNING* CLASS, MEM, VAL, 3X, 215, E15.7)
GO TO 560
30 WRITE(6, 270)
270 FORMAT (13H PARITY ERROR)
CALL EXIT
END
SUBROUTINE PSM(A, B, C, NDIM)
DIMENSION A(I1), B(I1), C(I1)
K = 0
DO 10 I = 1, NDIM
DO 10 J = 1, I
K = K + 1
C(K) = 0.
DO 10 N = 1, NDIM
IAC = (I*(I-1))/2 + N
IF (N .GT. I) IAC = (N*(N-1))/2 + I
IBC = (J*(J-1))/2 + N
IF (N .GT. J) IBC = (N*(N-1))/2 + J
10 C(K) = C(K) + A(IAC) * B(IBC)
RETURN
END
SUBROUTINE UPDFP(IC, IP, X, M, P, PI, NDIM, SNW, N, SWH)

C
THIS SUBROUTINE UPDATES THE COV, INV COV, MEAN OF ALL CLASSES
C FOR DATA FOR ONE PATIENT
C
REAL SNH(30)
REAL X(300,NDIM)
REAL P(1),PI(1),N(30,NDIM),L,SNH(30)
COMMON /PUTS2/ IFZC,IFSC
COMMON /UP/ D(30),ALPHA(10),L,"C",IUPDSH
COMMON /DISHS/ IDISSY
REAL T(30),TP(465),TT(30,30)
REAL G(30)
MS=(NDIM*(NDIM+1))/2
IHRD=0
C FIND LARGEST /
MLG=0.
DO 290 I=1,NC
IF((I(J) .LE. MLG) GO TO 290
I=!J
MLG=I(J)
290 CONTINUE
C COMPUTE NEW SUM OF MS TERMS
IF(IS .EQ. IC) IHRD=1
IF(IHRD .EQ. 1) SNH(IC)=SNH(IC)+1.
WA=M(IC)**ALPHA(IC)
SNHIC=SNH(IC)+WA
SNHIC=SNH(IC)+L
ISAVE=IDISS(*)
IDISS=0
C FIND DISTANCE FOR PAT TO CLASS
CALL DIS(IP,PI,X,":NDIM,DS,IC)
ISAVE=ISAVE
C UPDATE .MS.
DO 10 I=1,NDIM
G(I)=X(IP/I+1)*IC)
IF(IFEM .EQ. 1) GO TO 200
IF(IHRD .EQ. 1) I!J=(IC,I)=(IC,I)*(SNH(IC)-1.)*X(IP/I)/SNH(IC)
GO TO 10
200 G(I,J)=(IC,I)*SNH(IC)+X(IP,I)*WA)/SNH
10 CONTINUE
IF(IUPDSH .EQ. 0) GO TO 30
IF(IFEC .EQ. 6 AND. IHRD .EQ. 0) GO TO 30
SAV=SNHC
IF(IFEC .EQ. 1) GO TO 210
SNH(I)=SNH(IC)-1.
SNHIC=SNH(IC)+L
SNHIC=SNH(IC)
IP=SNH(IC)
*=-1.
210 CONTINUE
C UPDATE COV
C
DO 20 I=1,NDIM
K=(I*(I-1))/2
DO 20 J=1,I
K=K+1
20 TP(K)=G(I)*G(J)
A=SHOPL/SMNPL
B=SMW(IC)*WA/(SMNW*SMNPL)
DO 40 I=1,NS
40 P(I)=A*P(I)+B*TP(I)
IF(IUPDSW.EQ.3) GO TO 90

UPDATE INVERSE COV

DO 60 I=1,NDIM
INDX=(I*(I-1))/2
DO 60 J=1,NDIM
JNDX=(J*(J-1))/2
TP3(I,J)=0.
DO 60 N=1,NDIM
NNDX=(N*(N-1))/2
IA=INDX+N
IF(N.GT.I) IA=NNDX+I
IB=JNDX+N
IF(N.GT.J) IB=NNDX+J
60 TP3(I,J)=TP3(I,J)+PI(IA)*TP(IB)
DO 70 I=1,NDIM
DO 70 J=1,I
INDX=(I*(I-1))/2+J
TP(INDX)=0.
DO 70 N=1,NDIM
JNDX=(J*(J-1))/2+N
IF(N.GT.J) JNDX=(N*(N-1))/2+J
70 TP(INDX)=TP(INDX)+TP3(I,N)*PI(JNDX)
DENOM=A+B*DS
DO 50 I=1,NS
50 PI(I)=(PI(I)-B*TP(I)/DENOM)/A
80 CONTINUE
SMW(IC)=SAV
RETURN
C COMES HERE IF CVOS ARE TO REMAIN DIAGONAL

90 INDX=0
DO 100 I=1,NDIM
DO 100 J=1,I
INDX=INDX+1
IF(I.EQ.J) GO TO 100
P(INDX)=0.
100 CONTINUE
DO 110 I=1,NS
110 PI(I)=0.
DO 120 I=1,NDIM
INDX=(I*(I+1))/2
120 PI(INDX)=1./P(INDX)
SMW(IC)=SAV
RETURN
END
SUBROUTINE DIS(IP, PI, X, M, N DIM, DIST, IC)
COMMON /DISMES/ IDISSW
REAL PI(1), M(30, N DIM), X(300, N DIM)
DIST = 0.
DO 10 J = 1, N DIM
   INDEX = J - 1
   IF (J .EQ. 1) GO TO 10
   DO 20 K = 1, JM1
      INDEX = (J * JM1) / 2 + K
      DIST = DIST + (X(IP, J) - M(IC, J)) * (X(IP, K) - M(IC, K)) * PI(INDEX) * 2.
      10 CONTINUE
IF (IDISSW .EQ. 0) RETURN
DIST = EXP(DIST / 2.)
IF (IDISSW .EQ. 2) RETURN
DIST = -1. / DIST
RETURN
END

SUBROUTINE OUTPT(IP, W, JP, IPSW, NCS, ICL, IOUTSW)
COMMON /CG/ NPASS, WMAX, WSUM, IFOLD, IPNEW
* , OJ, OF, OG
COMMON /COSTS/ ITCOST, TCOST, ICCOST, CCOST
* , AER, CER, TER
COMMON /SORT/ IWS(300, 12), OWS(300, 12), ILS(300), INX(300),
* ICLS(300), IFRS(30)
COMMON /LIM/ LIM
COMMON /UP/ D30, ALPHA(10), L, NC, IUPDSW
REAL L
REAL W(30), OW(12)
INTEGER OC12, IPSW(30), ICSW(30), JP(30)
AER = 0.
INTG = 1000
LIM = 11
IF (NCS .LT. 12) LIM = NCS
IF (NPASS .EQ. 1) GO TO 50
C READ OLD MEMBERSHIP FUNCTIONS AND COMPUTE CHANGES..
READ(IFOLD) (WO(I), I = 1, NC)
DO 60 I = 1, NC
   IF (IPSW(I) .EQ. 0) GO TO 60
   WCN = ABS(WO(I) - W(I))
   WSUH = WSUM + WCN
   IF (WMAX .LT. WCN) WMAX = WCN
60 CONTINUE
C WRITE NEW MEMBERSHIP FUNCTION
WRITE(IFNEW) (W(I), I = 1, NC)
C COMPUTE CLUSTER TIGHTNESS MEASURES
DO 70 I = 1, NC
   IF (IPSW(I) .EQ. 0) GO TO 70
   OF = OF + W(I) ** 2
   OJ = OJ + W(I) ** ALPHA * D(I)
70 CONTINUE
IF(W(I) .EQ. 0.) GO TO 70
   OH=OH+W(I)*ALOG(W(I))
70 CONTINUE
   DO 10 I=1,30
   10 ICSW(I)=IPSW(I)

C FIND CLASS WITH LARGEST MEMBERSHIP FUNCTION
   DO 20 J=1,LIM
      OW(J)=0.
   DO 30 I=1,NC
      IF(ICSW(I) .EQ. 0.) GO TO 30
      IF(W(I) .LT. OW(J)) GO TO 30
      IS=I
      OW(J)=W(I)
      OC(J)=I
30 CONTINUE
   ICSW(IS)=0
20 CONTINUE

C PUT PATIENT IN ORDERED LIST FOR THAT CLASS
   DO 1 I=1,LIM
      IWS(IP,I)=OC(I)
      OWS(IP,I)=OW(I)
      KK=OC(I)
      K=IFRS(KK)
      IF(K .NE. 999) GO TO 2
      ILST(IP)=0
      IFRS(KK)=IP
      INX(IP)=999
      GO TO 3
2 IF(OWS(K,1) .LT. OWS(IP,1)) GO TO 4
   KK=INX(K)
   IF(KK .EQ. 999) GO TO 5
   K=KK
   GO TO 2
5 INX(K)=IP
   ILST(IP)=K
   INX(IP)=999
   GO TO 3
4 INX(IP)=K
   ILST(IP)=ILST(K)
   IF(ILST(K) .EQ. 0) GO TO 6
   KT=ILST(K)
   ILST(K)=IP
   INX(KT)=IP
   GO TO 3
6 IFRS(KK)=IP
   ILST(K)=IP
3 IF(OC(1) .EQ. ICL-1) GO TO 80

C COMPUTE ERROR FUNCTIONS
   AER=OW(1)-W(ICL-1)
CER=CER+AER
TER=TER+AER
ICTCST=ICTCST+I
ICTCST=ICTCST+I
IF(W(I=WCL-1)) .EQ. 0., GO TO 80
ICTCST=ICTCST+ON(1)/W(I=WCL-1)
ICTCST=ICTCST+ON(1)/W(I=WCL-1)

C WRITE LINE OF OUTPUT
80 IF(IOUSW .EQ. 1) WRITE(6,40) JP(IP),(OC(I),CN(I),I=1,LIM)
   * ,INTG,AER
40 FORMAT(1H,I4,2X,12(I2,5X,F6.4))
RETURN
END
SUBROUTINE WRW(I1,I2,BT,DV,IR)
   DIMENSION ICL(4),IC2(4),BT(4),DV(4)
   N=IR-1
   IR=1
   WRITE(6,10) (IC1(I),IC2(I),BT(I),DV(I),I=1,N)
10 FORMAT(2H,4(I2,1X,12E10.4,1X,12E10.4,2X))
RETURN
END
SUBROUTINE DTR(TM,TMI,TMP,NDM,DET)
   REAL TM(1),TMI(1),TMP(NDM,NDM),D1(10),D2(20)
   NDM=NDM+1
   DO 10 I=2,NDM
      DO 10 J=2,I
         INDX=(I-1)/2+J
         TMP(I-1,J-1)=TM(INDX)
         TMP(J-1,I-1)=TM(INDX)
10      TMP(J-1,I-1) = TM(INDX)
   CALL MINV(TMP,NDM,NDM,NM,D1,D2)
   DET=DET/TI(1)
RETURN
END
SUBROUTINE BATT(A,B,AM,BM,NDM,NDM,DIV,BAT,TMP)
   REAL A(I),B(I),AM(I),BM(I),TMP(NDM,NDM),D1(10),D2(10)
   REAL AM(I),BM(I)
   NDM=NDM-1
   CALL DTR(A,AM,TMP,NDM,NDM)
   CALL DTR(B,BM,TMP,NDM,NDM)
   DIV=0.
   INDX=0.
   DO 10 I=1,NDIM
      DO 10 J=1,I
         INDX=INDX+1
         FAC=2.
         IF(I .EQ. J) FAC=1.
      DIV=DIV+(A(INDX)*BM(INDX)+B(INDX)*AM(INDX))*FAC
10    DIV=DIV+(AM(J)-BM(J))*(AI(INDX)+BI(INDX))*(AM(I)-BM(I))*FAC
      DIV=DIV/2.-NDIM
   INDX=0.
   DO 10 I=1,NDIM
      DO 10 J=1,I
         INDX=INDX+1
         TMP(I,J)=A(INDX)+B(INDX)
20 TMP(J,I) = TMP(I,J)
    CALL HIY(TMP,NDIM,TDT,D1,D2)
    DAT = 0.
    DO 30 I = 1,NDIM
    DO 30 J = 1,NDIM
 30   DAT = DAT + (AH(J) - B(N(J))) * TMP(J,I) * (AH(I) - B(N(I)))
    DAT = DAT/2.
    DAT = DAT + ALOG(TDT*2.**(-NDIM))/SORT(ADT*BDT)
    DAT = DAT/2.
    RETURN
END

/*
//LK2D.SYSLMOD DD DSM=SYS1.TSTLIB(HB233FO2),DISP=CHR
//GO.FT10F001 DD DISP=OLD,
// DSRNAME=HD.H3H01.233042.SOR.ALL
//GO.FT0F001 DD DISP=(NEW,PASS),
// DBC=(LRECL=120,BLKSIZE=2048,RECFM=VBS,DUFN=1),
// UNIT=TAPE,VOL=SER=SCRCH,
// DSRNAME=HD.H3H01.2333042.FOZ.CDD
//GO.FT10F001 DD DISP=(NEW,PASS),
// DBC=(LRECL=120,BLKSIZE=2048,RECFM=VBS,DUFN=1),
// UNIT=TAPE,VOL=SER=SCRCH,
// DSRNAME=HD.H3H01.2333042.FOZ.EVEN
//GO.SYSL DD *
010001003
 01001
 011.000
 000001
 011000
+00.001000
000
+00
*/
BIBLIOGRAPHY


