APPLICATION OF THE KARHUNEN-LOEVE TRANSFORM TO THE REPRESENTATION OF VECTORCARDIOGRAMS

by

WILLIAM CLARK KESSEL

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January, 1977

Signature of Author

Department of Electrical Engineering, January 1977

Certified by

Thesis Supervisor

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

One of the more powerful tools in the diagnosis of heart disease is the vector cardiogram (VCG). Except for a difference in placement of electrical potential detectors (leads) on the body, a VCG is exactly the same as an electrocardiogram (ECG), and is used in a similar fashion. In this project we have concerned ourselves solely with the processing of VCG's.

The VCG consists of three equivalent leads, each of which represents a different component of a total waveform. Normally, these are represented as shown in figure 1, with electrical potential (vertical axis) plotted against time (horizontal axis). The separation into three waveform complexes (P, QRS, and T waves) is also shown. Taken together, the three leads will hereafter be referred to as a single VCG waveform.

The purpose of analyzing a VCG is to perform diagnoses concerning cardiac activity of patients. For the purposes of our computer diagnosis research, we consider approximately 30 diagnostic catagories. Class 1 indicates a normal, or undiseased, heart, while the remaining classes indicate various heart diseases of varying degrees of severity. Patients are classified according to both the rhythm (variance in the periodic timing of cardiac events) and the actual shape of the waveforms.

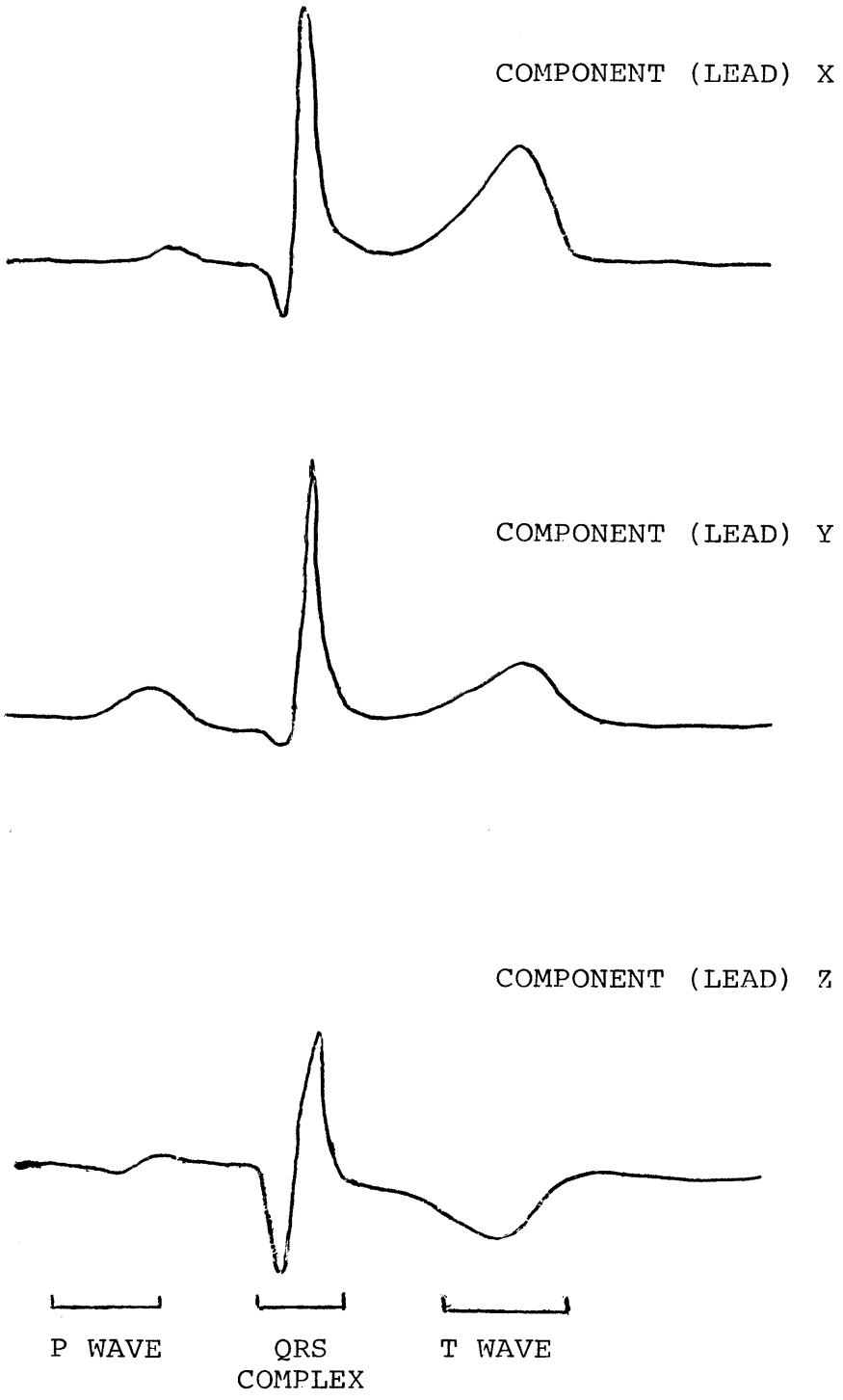


Figure 1

Sample Waveform

The purpose of this project, in simple terms, is to match the diagnosis of a cardiologist with a computer program. The classification error can then be defined as the proportion of VCG's that are diagnosed incorrectly by an algorithm (i.e. the cases in which the computer decision did not match that of the cardiologist).

The part of this overall project described in this thesis concerns analysis of the shape of the waveforms (morphology) as opposed to that of rhythm diagnosis. Therefore, we are considering those VCG's that are characterized by regular (i.e. periodic) patterns; hence the waveform of only one heartbeat is needed for each patient. In order to obtain a relatively noise-free, representative waveform, the average of several heartbeats is taken for each patient, the baseline fluctuations are removed, and the electrical potential is sampled at intervals of between two and four milliseconds. The number of samples taken per second is the sampling rate, and each time at which a sample is taken is referred to as a sample point.

This project has employed a more mathematical approach than most previous attempts at computer classification. In particular, while other people have tried to simulate the pattern recognition capabilities of a cardiologist (e.g. look for a specific slope of the S-T segment, or examine the relative sizes of the P and R waves), we have tried to use the statistics of a large set of patients' waveforms for which we knew the diagnosis(training set) and apply them to find diagnoses for

unknown waveforms (test set). In otherwords, we have applied supervised computer learning to the problem. It is our belief that this method offers the most promising avenue for computer diagnosis, as it is the one that takes maximal advantage of the capabilities of the computer (in computing statistical correlations, orthagonal expansions, etc.), rather than attempting to perform a task for which the computer is not as suited (emulating a human pattern recognizer).

In the waveform morphology analysis of VCG's the task can be separated into two problems: representation and classification. The classification problem involves considering each waveform of N samples as a single point in N-dimensional space, and applying clustering techniques to find the class to which it belongs. This thesis will not deal with this problem except to note that the difficulty of clustering increases rapidly as the number of dimensions increases. This fact suggests that there should be an initial process employed before classification, the purpose of which is to represent each waveform as accurately as possible with as few numbers as possible. A solution to this problem is the goal of this thesis.

CHAPTER 2

ERROR CRITERIA

Since our purpose is to represent waveforms with as small an error as possible, we must first decide what we mean by the representation error. Ideally, one would like to have a scalar measure, e, of representation error for each representation method. If, when comparing two methods, R_1 and R_2, e_1 was smaller than e_2, we would know that R_1 was better that R_2.

There are two approaches we could take to finding such a number. We could either start with an overall system viewpoint and find the classification error induced by a representation method, which would be the most correct way to go about it (since representation errors that don't affect classification are irrelevant), or we could examine the representation by itself, regarding all inaccuracies in the representation as contributing to our overall measure of representation error.

First, let us consider the overall method, that is using the classification error induced by the representation method. Unfortunantly, the classification error is a function of both the representation error and the choice of classification method. Since no specific classification, or clustering, method is assumed during the representation process, there can be no formal, correct method for assigning a numerical value to a representation error. Even if the classification method is

known, the relation between representation error and classification error is highly nonlinear, so that the only way to compute the effect of the representation error would be to complete the classification process for each possible representation scheme. This is clearly prohibitive in cost if any reasonable number of representation methods are to be tried.

Thus, we must turn to the second approach. From this microscopic viewpoint, the final error for a representation method would be computed from the individual errors at each sample of each patient's waveform. This means that the representation error must be a function of (in our case) over 100,000 numbers.

Finding such a function seems to be a quite difficult task, so the first thing to do is to break the problem down into several simpler ones. This can be accomplished by finding a representation error for each single waveform, and using these intermediate results to define a single, global error over all waveforms.

First, we treat the problem of computing a single error for a single waveform from the errors at each sample of the waveform (local errors). What we want is a number which gets larger as the representation becomes less accurate, where we would ideally want to define "accurate" in terms of the effect on classification. Since we cannot find an absolutely correct function (because of the unknown effect of a clustering algorithm) we must examine possibilities and decide which is most appro-

priate. Specifically, we will examine three functions:

$$f(x_1,x_2,\ldots,x_N)= \frac{1}{N}\sum_{i=1}^{N} x_i^2 \qquad (2.1)$$

x_i=error at $i^{\underline{th}}$ sample of waveform

N=number of samples

$$f(x_1,x_2,\ldots,x_N)= \frac{1}{N}\sum_{i=1}^{N} |x_i| \qquad (2.2)$$

and,

$$f(x_1,x_2,\ldots,x_N)= \max_{N}(|x_i|) \qquad (2.3)$$

The mean square function (2.1) is appealing because of its analytical tractability. One can directly compute expected values of representation errors for this criterion without performing a great deal of simulation. Besides this fact, there is little reason to use such an error criterion. In fact, such a criterion has a definite disadvantage: The contribution of the waveform reconstruction errors in areas of small magnitude (e.g. the P wave) will be relatively less than that of areas of large magnitude (QRS and T waves). This is caused by the extremely large weighting the mean square criterion gives to large magnitude errors. Hence, one gets proportionally greater accuracy in regions in which the signal is large.

Because of this fact we will turn to the sum of absolute values function (2.2). This function has the advantage of not

diminishing the importance of the P wave. It also better matches the intuitive idea that errors of equal amplitude are approximately equal in importance, as a general rule. The main drawback of this, and any function which involves absolute values is the algebraic difficulty in dealing with such functions. However, this may be a reasonable price to pay for the increased confidence that we have that this error measure more accurately measures how good a representation will be for classification purposes. This is especially true since the next next step in finding the final representation error (once an error is foundfor each of the individual waveforms) must also introduce operations which do not lend themselves to algebraic manipulation (see the following discussion).

Finally, we will examine the worst case function (2.3). This approach, though useful in that it gives a definite upper bound on the reconstruction errors, shares with the mean square function the disadvantage of concentrating on only part of the waveform. For example, a reconstruction error of a magnitude which is acceptable in the R wave region may be unacceptable in the P wave area. It is this consideration which has led us to discard the worst case approach for the purposes of this specific portion of our study.

We have, therefore, decided on the use of the average of the absolute values of the sample by sample reconstruction errors as the representation error for a single waveform. Now we must find a function which combines the errors for each waveform into

a single error for a representation algorithm.

In this situation a worst case approach makes more sense. Since our hope is to represent each waveform accurately, the waveforms in which we are most interested are those for which we do the worst job (remember that it is these waveforms that are more likely to be misclassified and this will comprise a major part of the overall classification error). Because all waveforms are quite similar we can be reasonably well assured that an acceptable error in one waveform would also be acceptable in any other. Thus, we will use some form of the worst case approach.

It must be remembered, however, that we cannot have complete confidence in the error which we have assigned to any specific waveform, because of the ad hoc approach. Therefore, instead of examining only the waveform with the worst error, we will will examine the ten waveforms with the worst errors.

Again, because we cannot have complete confidence in the error computations, and because we have reduced the problem to only ten waveforms, we will stop short of our original goal of finding one number to represent the error for a representation algorithm. Instead, we have found that it is feasible to examine the actual reconstructions of the ten worst waveforms for each method, and to make a judgement on each representation technique based on a visual interpretation of the data.

In summary, the method we will use in judging among several representation algorithms is as follows:

1. Reconstruct each waveform from its representation and find the error at each sample point.
2. Use the sum of the absolute values of these errors as the global error for each waveform.
3. Make a visual comparison between the waveforms with the ten worst errors for each representation algorithm.

CHAPTER 3

BASIC KARHUNEN-LOEVE TRANSFORM

As previously stated, the purpose of the preprocessing stage is to represent all waveforms as accurately as possible with as few numbers as possible. The Karhunen-Loeve transform is one of the simplest and most effective approaches to this problem. In fact, if a mean square error criterion is used (see chapter 2) it can be shown that this transform is the optimal linear solution. Thus, it certainly seems to be a reasonable starting point in the search for an adequate preprocessing method.

3.1) Theory of the Karhunen-Loeve Transform

Any single waveform can, of course, be represented by a set of points in two dimensional space. In our case, the dimensions are time vs. electrical potential. An accurate representation of one lead of a single waveform requires on the order of 100 samples (see chapter 4) if linear interpolation is used between sample points. One lead of each patient's waveform is the specified by:

$$x = (x_i, t_i) \quad i=1,\ldots,\text{no. of samples} \tag{3.1}$$

Another way to represent the same waveform (of n samples) is as a single point in n-dimensional space, where each component represents the electrical potential at a particular point

in time:

$$x = (x_1,\ldots,x_n); \quad n = \text{no. of samples} \tag{3.2}$$

If we now consider a large ensemble of VCG's we obtain a distribution of points (one for each waveform) in this n-dimensional space. Thus, for the $j^{\underline{th}}$ member of the ensemble:

$$x^j = (x_1^j,\ldots,x_n^j) \tag{3.3}$$

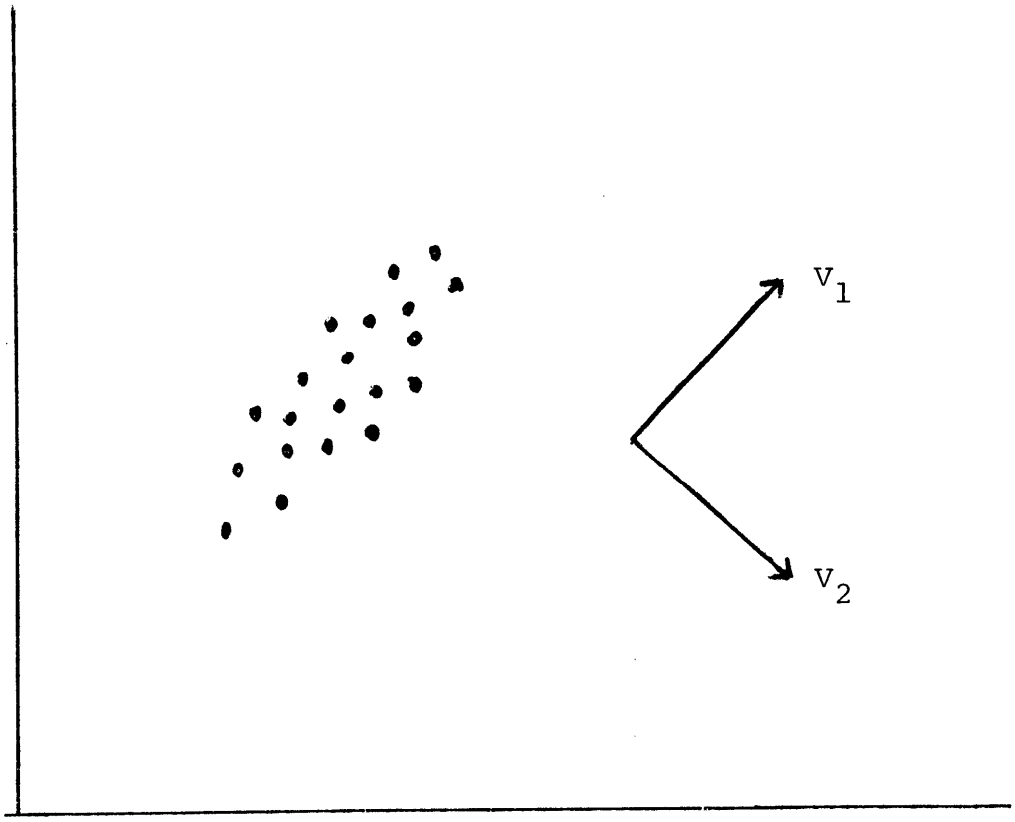
We now define the (k,h) member of the covariance matrix A of this data by:

$$a_{kh} = \frac{1}{P-1} \sum_{j=1}^{P} (x_k^j - m_k)(x_h^j - m_h) \tag{3.4}$$

P = no. of patients

$$m_i = \frac{1}{P} \sum_{j=1}^{P} x_i^j \tag{3.5}$$

The covariance matrix can be loosely said to define a hyper-ellipsoid where the length of each axis is proportional to the variance of the data along that axis. These axes are defined by the eigenvalues and eigenvectors of the matrix, where each eigenvector defines an axis, while the corresponding eigenvalue indicates the variance along that axis. Thus, the eigenvector corresponding to the largest eigenvalue of the covariance matrix of all patients' data defines the direction of the greatest variance of the data, as illustrated in figure 2. If the data is randomly distributed over n-dimensional space, then the n x n covariance matrix will be diagonal, with equal ele-



Two dimensional example of eigenvectors of covariance matrix over twenty data points.

V_1 = eigenvector corresponding to dominant eigenvalue of 2 x 2 covariance matrix

V_2 = eigenvector corresponding to minor eigenvalue

Figure 2

Eigenvectors

ments, thus having only one distinct eigenvalue with multiplicity n (indicating the equal variance in all directions of a hyper-spherical distribution), and all vectors as eigenvectors. If, however, there is any similarity between the waveforms, distinct eigenvalues and eigenvectors will exist, and the greater the similarity, the greater will be the disparity in magnitude between the largest and smallest eigenvalues.

As stated earlier, the eigenvectors define the principal axes of the hyper-ellipse enclosing the data. This statement contains an important fact: that the eigenvectors form an orthagonal set. It is this property of the eigenvectors of any real, symmetric matrix which makes the Karhunen Loeve transform possible. For, if we have a set of n orthagonal vectors in n-dimensional space, then these vectors completely span the space and can be used as a basis for representing any point in that space. The standard set of orthagonal vectors is:

$$U_1 = (1,0,\ldots,0)$$

$$U_2 = (0,1,\ldots,0)$$

$$\ldots$$

$$U_n = (0,0,\ldots,1)$$

Thus any point in n-dimensional space can be represented as:

$$x = x_1U_1+x_2U_2+\ldots+x_nU_n \tag{3.6}$$

which is normally written in shorthand form as:

$$x = (x_1,x_2,\ldots,x_n) \tag{3.7}$$

If we are given the eigenvectors $E_1 \ldots E_n$, x can also be represented as:

$$x = a_1E_1+a_2E_2+\ldots+a_nE_n \tag{3.8}$$

or, in shorthand form:

$$x = (a_1,a_2,\ldots,a_n) \tag{3.9}$$

So far, we have not reduced the number of coefficients needed to represent any point x. Instead of representing each waveform as n samples of electrical potential, we have shown how to represent the waveform by n coefficients as in (3.8).

But now let us return to the earlier discussion of an eigenvalue of a covariance matrix as representing the variance of all data along its corresponding eigenvector. What this means is that if all data is represented as:

$$x^j = (a_1^j,a_2^j,\ldots,a_n^j) \quad j=1,2,\ldots,P \tag{3.10}$$

where each a_i^j is a coefficient as in (3.8) for the $j^{\underline{th}}$ patient, then the $k^{\underline{th}}$ eigenvalue (associated with the $k^{\underline{th}}$ eigenvector) will be equal to the variance of a_k over all patients. If the data is highly correlated (as is true in our case since all VCG's are fairly similar in shape) then some of the eigenvalues will be very small, implying that the corresponding coefficients vary very little from patient to patient, and can be effectively ignored in relation to the information conveyed by the coefficients coresponding to the large eigenvalues.

This is the power of the Karhunen Loeve transform. It allows one to represent each of a set of vectors (in our case VCG waveforms) with a relatively small number of coefficients. The actual number of coefficients needed is determined by the

correlation of the data (i.e. how many small and large eigenvalues there are) and the accuracy desired.

3.2) Algorithm for Performing the Karhunen-Loeve Transform

The algorithm used in performing the Karhunen-Loeve transform is composed of four parts: creation of the covariance matrix, finding the dominant eigenvalues of this matrix, finding the coefficients in equation (3.8) (the pattern vector) of each waveform, and reconstructing each waveform using the K-L expansion.

If the covariance matrix were created directly equations (3.4) and (3.5), then to add new data would require a complete repetition of the process. If, however, the matrix is stored as the vector q:

$$q = \sum_{j=1}^{P} x_j \tag{3.11}$$

where x_i is a vector containing the waveform of the $i^{\underline{th}}$ patient, and the P x P matrix S:

$$S = \sum_{j=1}^{P} x_j x_j^T \tag{3.12}$$

then new data can be added without recomputing S and q for all old data. The covariance matrix A can then be computed when needed by the use of the equation:

$$A = (S - \frac{qq^T}{P})/(P-1) \qquad (3.13)$$

The eigenvectors of this matrix are computed using the program BASIS, implemented in FORTRAN by J. S. Halliday (2). Briefly, the symmetric matrix A is tridiagonalized through an orthagonal transformation. The eigenvalues of this tridiagonal matrix are computed and transformed into the eigenvalues of A through the inverse of the original transformation. See (2) for a listing of the program and details of the process.

Once m eigenvectors of the P x P matrix A are found, they are stored columnwise in the P x m matrix E. The pattern vector a_i for the $i^{\underline{th}}$ patient can then be computed with the equation:

$$a_i = E^T x_i \quad i = 1,\ldots,n \qquad (3.14)$$

Finally, the waveform must be reconstructed. Although this process is not actually a part of the Karhunen-Loeve transform, it is necessary in order to evaluate the accuracy of the representation. The reconstructed vector x_i' can be computed using the equation:

$$x_i' = Ea_i \qquad (3.15)$$

CHAPTER 4

VARIABLE SAMPLING RATES

One major trade-off in the representation process is that of computational simplicity versus accuracy. This trade-off is most obvious in the decision involving the sampling rate (i.e. the elapsed time between each value of electrical potential, or sample). If one uses a high sampling rate (a short time between samples) one obtains a very accurate initial representation of each waveform, but the cost of the Karhunen-Loeve transform becomes prohibitive. A low sampling rate makes it easy to perform the expansion, but introduces errors before one even begins the Karhunen-Loeve approximation.

Cardiologists have generally agreed that a rate of 250 samples per second will not cause the loss of any diagnostic information. Unfortunantly, sampling at this rate means that to perform the Karhunen-Loeve transform one would have to find eigenvalues of a 600 x 600 matrix, an infeasible task even on the largest modern computers. The alternative is to break the waveformup into its components and perform a separate expansion on each. This is feasible, but undesirable, since it complicates the algorithms and cannot take into account inter-component correlations.

A different approach to the problem would be to take advantage of the fact that some regions of each waveform obviously

do not require such a high sampling rate. For example, the S-T segment is always very smooth, and need not be represented by the 25 to 50 samples which the 250 samples per second rate allows. The obvious thing to do then, is to use a variable sampling rate. This does not mean that each waveform will be sampled differently. If this were the case the Karhunen-Loeve transform would not make sense, since a specific sample would refer todifferent parts of different waveforms. Instead, the sampling will be consistent between waveforms, but at a varying rate. The goal is to represent each waveform with 300 samples (equivalent to an average rate of 125 samples per second), a number which we have found is definitely numerically feasible for a Karhunen-Loeve transform.

An initial constraint in deciding where to sample a waveform is that we do not do our own sampling, but instead receive representations of waveforms sampled at a rate of 500 samples per second. Thus, any samples we use must come from this initial set of 1200 samples (3 leads of 0.8 seconds each). This would suggest an approach involving finding the samples which are least important, instead of those which are most important.

The first thing to do is to decide on the basis for choosing one sample over another. In other words, if we are given a collection of data:

$$(x_1^i, x_2^i, \ldots, x_n^i) \quad i = 1,2,\ldots,P$$

P = no. of patients

n = no. of samples (in this case, 500)

where x_j^k is the value of the $j^{\underline{th}}$ sample of the $k^{\underline{th}}$ patient, we would like an error criterion:

$$e_j \quad j = 1,2,\ldots,n$$

which would indicate the cost of ignoring the $j^{\underline{th}}$ sample on all the patients' waveforms.

The first step is to define an error for each sample for each individual patient:

$$a_j^i \quad i = 1,2,\ldots,P \quad j = 1,2,\ldots,n$$

The simplest way to calculate such an error is assuming linear interpolation to determine the value at the eliminated sample. Thus:

$$a_j^i = \left| x_j^i - (x_{j-1}^i + x_{j+1}^i)/2 \right| \tag{4.1}$$

Next, these individual errors must be combined into an error over all patients, the original goal. As in the situation in chapter 2, where the reconstruction errors for each patient were combined to yield a global error, the worst case criterion seems to be the best method. It is valid because we are comparing similar portions of each waveform. Therefore, we know that if a_j^i is greater than a_j^k, then the error is almost certainly more serious in patient i than in patient k. As a result, we can use the equation:

$$e_j = \max_i (a_j^i) \tag{4.2}$$

Now we must extend the criterion to cover the case of removal of more than one consecutive sample. What we want is a criterion (for a single patient initially) a_{kj}^i which represents

the error caused by removal of k consecutive samples, beginning with sample j, from the waveform of patient i. This cannot be a simple function of the a^i_j's defined in (4.1), because the interpolation endpoints are different (i.e. a^i_j assumes interpolation from x^i_{j-1} to x^i_{j+1}, while $a^i_{2,j}$ must interpolate from x^i_{j-1} to x^i_{j+2}). We must instead use a method as shown in figure 4.1, where:

$$a^i_{kj} = \max_{1\leq m\leq k} \left(\left| x^i_{j-1+m} - \left[x^i_{j-1} + \frac{m}{k+1}(x^i_{j+k} - x^i_{j-1}) \right] \right| \right) \qquad (4.3)$$

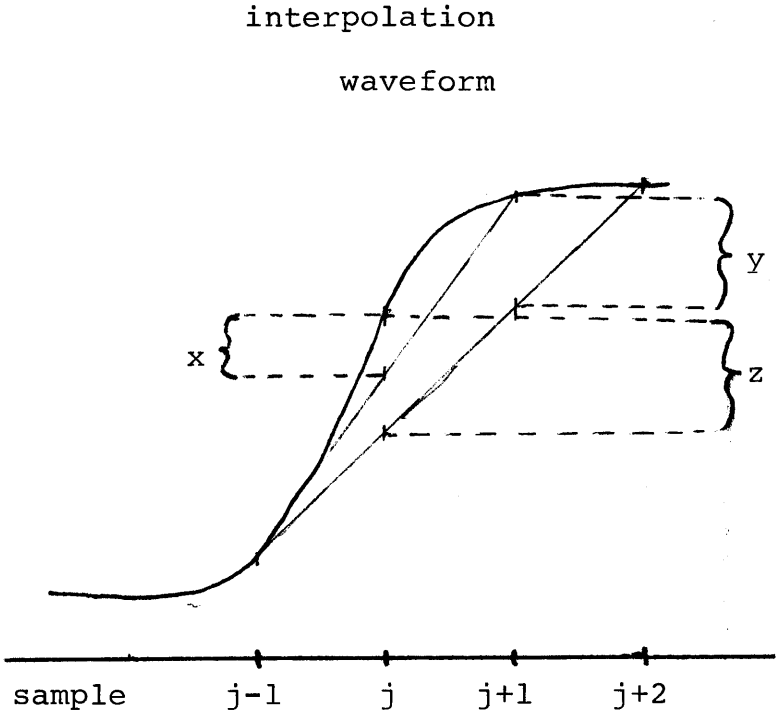
Notice that (4.3) simply chooses the worst interpolated approximation to any of the k removed samples. Notice also that if k is equal to one, this equation is the same as (4.1).

Again, we must now extend the criterion for a single patient to a criterion over all patients, and this can be done exactly as in (4.2):

$$e_{kj} = \max_i (a^i_{kj}) \qquad (4.4)$$

This then, will be the error criterion to be used in finding the samples most suited for removal.

What remains now is to present an algorithm for successively removing samples until a desired number is reached. First however, one point must be emphasized. That is, each e_{kj} represents a region of k samples to be removed. It is important to realize that this number is only valid if the sample immediately preceeding and the sample immediately following this region are not removed, since they are used for interpolation.



$$a^i_{1j} = x$$

$$a^i_{2j} = \max(y,z)$$

Figure 4.1

Interpolation Across Two Samples

This does not mean that if sample j is removed, sample j-1 can never also be removed. What it does mean is that the error caused by removing sample j-1 after removing sample j is the same as that caused by removing both simultaneously. In other words, after sample j is removed (corresponding to error e_{1j}) the error $e_{1(j-1)}$ is meaningless. The error caused by removal of sample j-1 is actually $e_{2(j-1)}$ (since sample j has already been removed). In general, if region A is removed, any error is invalid which would correspond to removal of the sample immediately preceeding or following region A without removing all of region A.

Finally, an algorithm can be presented, which will successively remove samples with the smallest error in the worst case:

1. Remove every second sample, to reduce the sampling rate to 250 samples per second.
2. Initialize a K x 200 matrix, E, to all zeroes (where K is an apriori choice of the most consecutive samples which may be removed). The elements of E (e_{kj}; k=1,K; j=1,200) will eventually contain the error criteria described in the preceeding discussion.
3. Compute a K x 200 matrix, A, for one patient, where:

$$a_{kj} = \max_{1 \leq m \leq k} \left(\left| x_{j-1+m} - \left[x_{j-1} + \frac{m}{k+1} (x_{j+k} - x_{j-1}) \right] \right| \right)$$

4. Update E where:

$$e_{kj} = \max(e_{kj}, a_{kj})$$

5. Repeat steps 3 and 4 for all patients.

When the first five steps are completed E contains the the error criteria for sample removal. What remains is to perform that removal:

6. Find the smallest element in E, and record the corresponding samples as removed (i.e. if e_{ij} is the smallest element, sample j and the i-1 samples following j are removed).

7. Delete element e_{ij} (by setting to a very large value so that it will not be found in step 6) and any elements which become invalid when the region corresponding to e_{ij} is removed. Specifically, the elements deleted are (in FORTRAN implied DO loop notation):

$$e_{ij}$$
$$e_{mn} \quad ((n=j+1,j+i),m=1,K)$$
$$e_{mn} \quad ((n=j-m,j-m-1+i),m=1,K)$$

8. Repeat steps 6 and 7 until the desired number of samples have been removed, or the errors are no longer neglegible.

RESULTS

The preceeding algorithm was implemented in FORTRAN as the programs WRSTRED (steps 2 through 5) and WRSTFIND (steps 6 through 8). As a value for K we used 8. The output of WRSTFIND

for the last 24 samples (columns) to be removed is shown in figure 4.2. The worst error was approximately 0.025 millivolts, or about 2.5% of the maximum electrical potential shown by most waveforms. It is questionable whether this is a neglegible value, and on the basis of results presented in the next chapter, it was decided that it would be better to use a constant sampling rate of 250 samples per second and expand each lead separately. However, I believe that this method is generally an effective preprocessing step for simplifying the numerical complexity of the Karhunen-Loeve transform.

```
 1 COLUMNS OUT OF   1 FOLLOWING   35 DELETED
  WORST PATIENT 8323     VALUE     0.15738960E-01
 1 COLUMNS OUT OF   1 FOLLOWING   44 DELETED
  WORST PATIENT 9572     VALUE     0.15996248E-01
 1 COLUMNS OUT OF   3 FOLLOWING  193 DELETED
  WORST PATIENT    0     VALUE     0.16055465E-01
 1 COLUMNS OUT OF   1 FOLLOWING   41 DELETED
  WORST PATIENT 9803     VALUE     0.16170364E-01
 1 COLUMNS OUT OF   1 FOLLOWING   46 DELETED
  WORST PATIENT 8331     VALUE     0.16213413E-01
 1 COLUMNS OUT OF   1 FOLLOWING  102 DELETED
  WORST PATIENT 9702     VALUE     0.16577777E-01
 1 COLUMNS OUT OF   2 FOLLOWING  129 DELETED
  WORST PATIENT 9034     VALUE     0.17475124E-01
 1 COLUMNS OUT OF   1 FOLLOWING  105 DELETED
  WORST PATIENT 9505     VALUE     0.17805677E-01
 1 COLUMNS OUT OF   2 FOLLOWING   22 DELETED
  WORST PATIENT 9424     VALUE     0.17931163E-01
 1 COLUMNS OUT OF   3 FOLLOWING  122 DELETED
  WORST PATIENT 9034     VALUE     0.19475937E-01
 1 COLUMNS OUT OF   4 FOLLOWING  193 DELETED
  WORST PATIENT    0     VALUE     0.21746404E-01
 1 COLUMNS OUT OF   2 FOLLOWING   33 DELETED
  WORST PATIENT 8969     VALUE     0.21931276E-01
 1 COLUMNS OUT OF   3 FOLLOWING  174 DELETED
  WORST PATIENT 9836     VALUE     0.22622164E-01
 1 COLUMNS OUT OF   3 FOLLOWING  132 DELETED
  WORST PATIENT 9034     VALUE     0.23239136E-01
 1 COLUMNS OUT OF   8 FOLLOWING    4 DELETED
  WORST PATIENT 9426     VALUE     0.24033587E-01
 1 COLUMNS OUT OF   5 FOLLOWING  187 DELETED
  WORST PATIENT 9769     VALUE     0.24035212E-01
 1 COLUMNS OUT OF   3 FOLLOWING  149 DELETED
  WORST PATIENT 9135     VALUE     0.24035931E-01
 1 COLUMNS OUT OF   1 FOLLOWING   97 DELETED
  WORST PATIENT 9630     VALUE     0.24131890E-01
 1 COLUMNS OUT OF   2 FOLLOWING   41 DELETED
  WORST PATIENT 9678     VALUE     0.24179161E-01
 1 COLUMNS OUT OF   3 FOLLOWING  155 DELETED
  WORST PATIENT 9440     VALUE     0.24337526E-01
 1 COLUMNS OUT OF   3 FOLLOWING   16 DELETED
  WORST PATIENT 9498     VALUE     0.24886966E-01
 1 COLUMNS OUT OF   2 FOLLOWING   38 DELETED
  WORST PATIENT 9851     VALUE     0.24953604E-01
 1 COLUMNS OUT OF   4 FOLLOWING  182 DELETED
  WORST PATIENT 9173     VALUE     0.25216613E-01
 1 COLUMNS OUT OF   2 FOLLOWING  104 DELETED
  WORST PATIENT 9505     VALUE     0.25258940E-01
TOTAL OF  100 COLUMNS DELETED
```

Figure 4.2

Removed Samples (26)

CHAPTER 5

KARHUNEN-LOEVE STUDIES

This chapter is a presentation of actual reconstruction results obtained. For a good part of this research effort the actual error criterion to be used had not yet been formalised. Because decisions had already been made regarding some of these methods before the error criterion presented in chapter 2 had been decided upon, and because of the expense of going back and getting missing reconstructions, complete data is not always available. Thus, the approach to be taken in presenting this data will be to show corresponding reconstructions of the same waveforms produced by different methods.

In examining the data, two major decisions must be made. First, the number of features necessary for a sufficiently accurate representation must be set. Second, it must be decided what kind of sampling is to be used.

5.1) Studies on Number of Features

As will demonstrated in the next section, the method of sampling is a less crucial decision than that of how many features to use. Therefore, we will first examine the effect of the number of features on the accuracy of representation.

The most complete results were obtained using the variable sampling rate described in the last chapter. As a result,

comparisons will be made of reconstructions using 30, 50, and 60 eigenvectors with the variable sampling rate.

Patient 8328

This patient has a reasonably close to normal waveform, and should not be terribly difficult to represent. A 20 feature representation (figure 5.1) is clearly inadequate, and the 30 feature reconstruction (figure 5.2) still misrepresents the Q and T waves. Even with 50 features (figure 5.3) the T wave is not represented correctly. It is only when 60 features are used (figure 5.4) that an adequate representation is achieved.

Patient 8334

Figure 5.5 again shows the inadequacy of the 30 feature representation. The errors in the P wave region are especially bad. In this case however, there is little difference between the use of 50 (figure 5.6) and 60 (figure 5.7) features.

After examining numerous results along these lines, it becomes apparent that it is not possible to find an exact number features necessary to insure proper classification. It is certain that 30 is not enough, and that 60 is enough. An exact determination can only be made during the clustering phase of the project. Therefore, all representations should be carried out to 60 features, keeping in mind, while clustering, that it may be possible to use fewer features.

5.2) Studies on Sampling

In deciding what method of sampling is to be used, we will make two comparisons: using first 30 features, and then 60 features. With each number of features, a comparison will be made between waveforms sampled with the variable method (see chapter 4) and waveforms sampled at 250 samples per second with a separate Karhunen-Loeve transform performed on each lead, with one third the features for each lead. For example, the separate lead transform with 30 features actually is composed of a transform with 10 features on each lead.

5.2.1) Sampling Studies - 30 Features

Patient 8323

This is a normal patient with a definite P wave which must be represented correctly. In this respect, the separate lead expansion (figure 5.8) does a better job than the variable sampling rate expansion (figure 5.9). However, the variable expansion performs a more accurate reproduction of the T wave, and so there is noobvious superiority of one method over the other.

Patient 8324

In this case, where there is little P wave present, the variable sampling (figure 5.11) does a slightly better job than the separate lead expansion (figure 5.10). The important difference is in the S-T segment, which is crucial for an accurate diagnosis (see Burch and Winsor (1)).

5.2.2) Sampling Studies - 60 Features

When examining waveforms which have more pronounced abnormalities, a 30 feature expansion is clearly not adequate, as shown in figure 5.12, a 30 feature variable sampling rate reconstruction. As a result, we will compare 60 feature reconstructions for these cases (20 features on each lead with the separate lead expansion).

Patient 8331

Again, with a pronounced P wave, the separate lead expansion (figure 5.13) seems to do slightly better than the variable sampling expansion (figure 5.14). The significant difference, however, does not occur in the P wave, but in the S wave region.

Patient 8370

This is a decidedly abnormal waveform, but there is still very little difference between the separate lead reconstruction (figure 5.15) and the variable sampling (figure 5.16). Again, the separate lead expansion has a slightly smaller cost (error as described in chapter 2).

As a result of these studies it is fairly obvious that there is only a minimal difference in accuracy between using a variable sampling rate expansion and a separate lead expansion, with twice the number of samples. The separate lead expansion is slightly more expensive to run due to the three expansions necessary, but uses slightly simpler programs.

One major advantage that the separate lead expansion does have however, is that while the algorithm for deciding which samples to use for the variable sampling rate does provide an upper bound for the interpolation error, that bound applies only to the specific data base which was used as a training set. This disadvantage applies to some extent to any implementation of the Karhunen-Loeve transform (since the features, or eigenvectors, are computed on the basis of that set), but the variable sampling algorithm provides an extra possibility in the choosing of samples. Therefore, we have decided to use the constant sampling rate of 250 samples per second, with separate expansions for each lead.

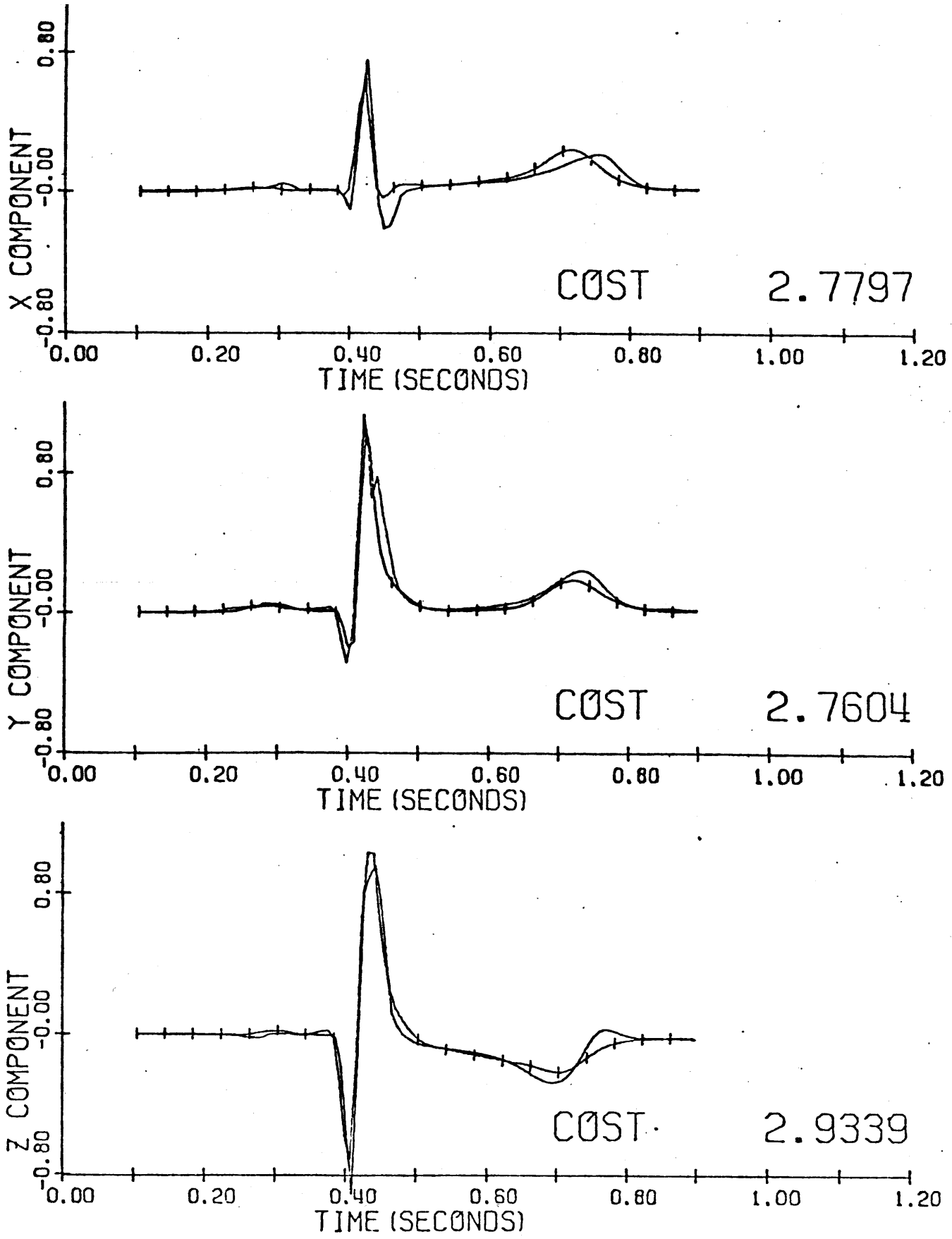


Figure 5.1

Patient 8328 20 Features

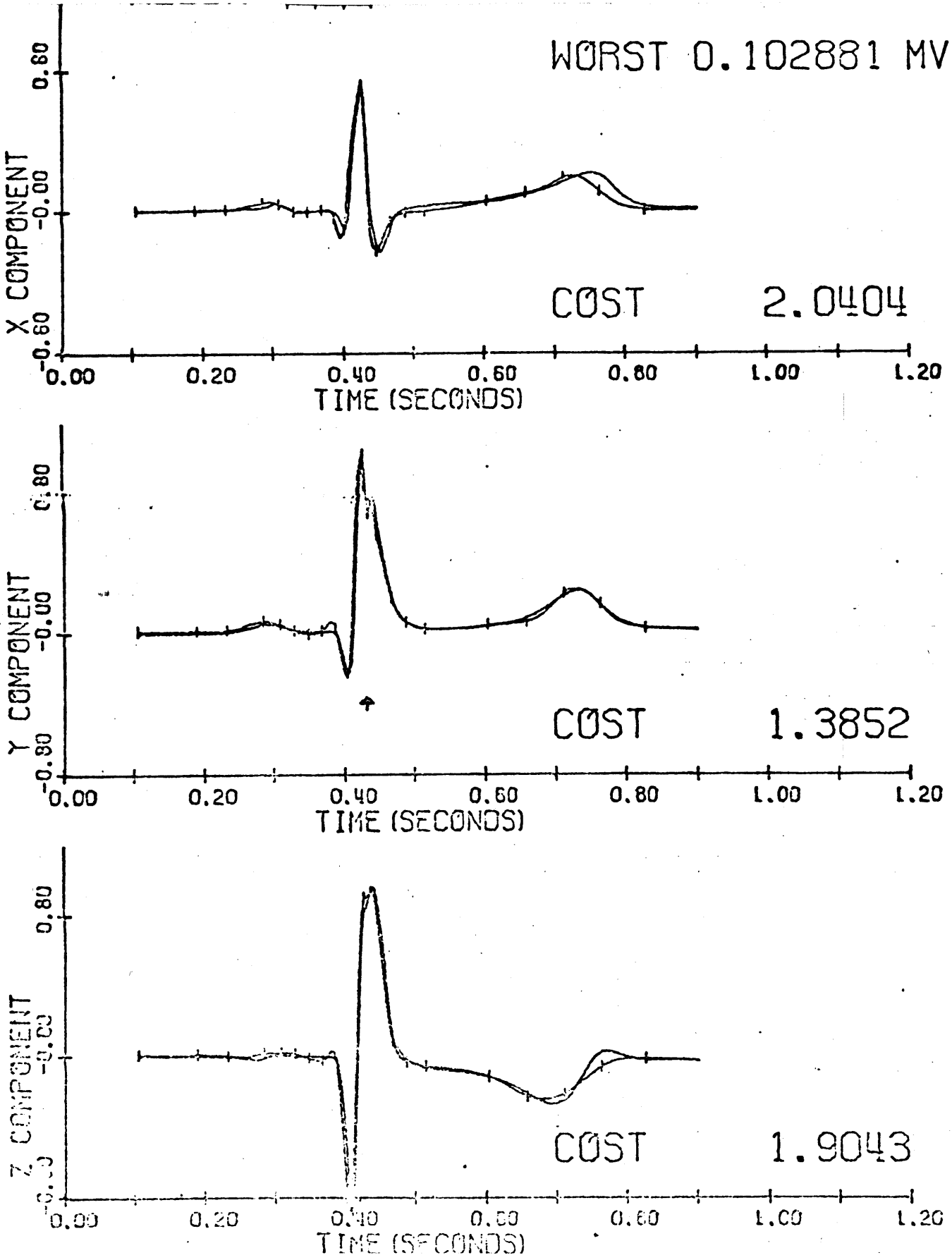


Figure 5.2

Patient 8328 30 Features

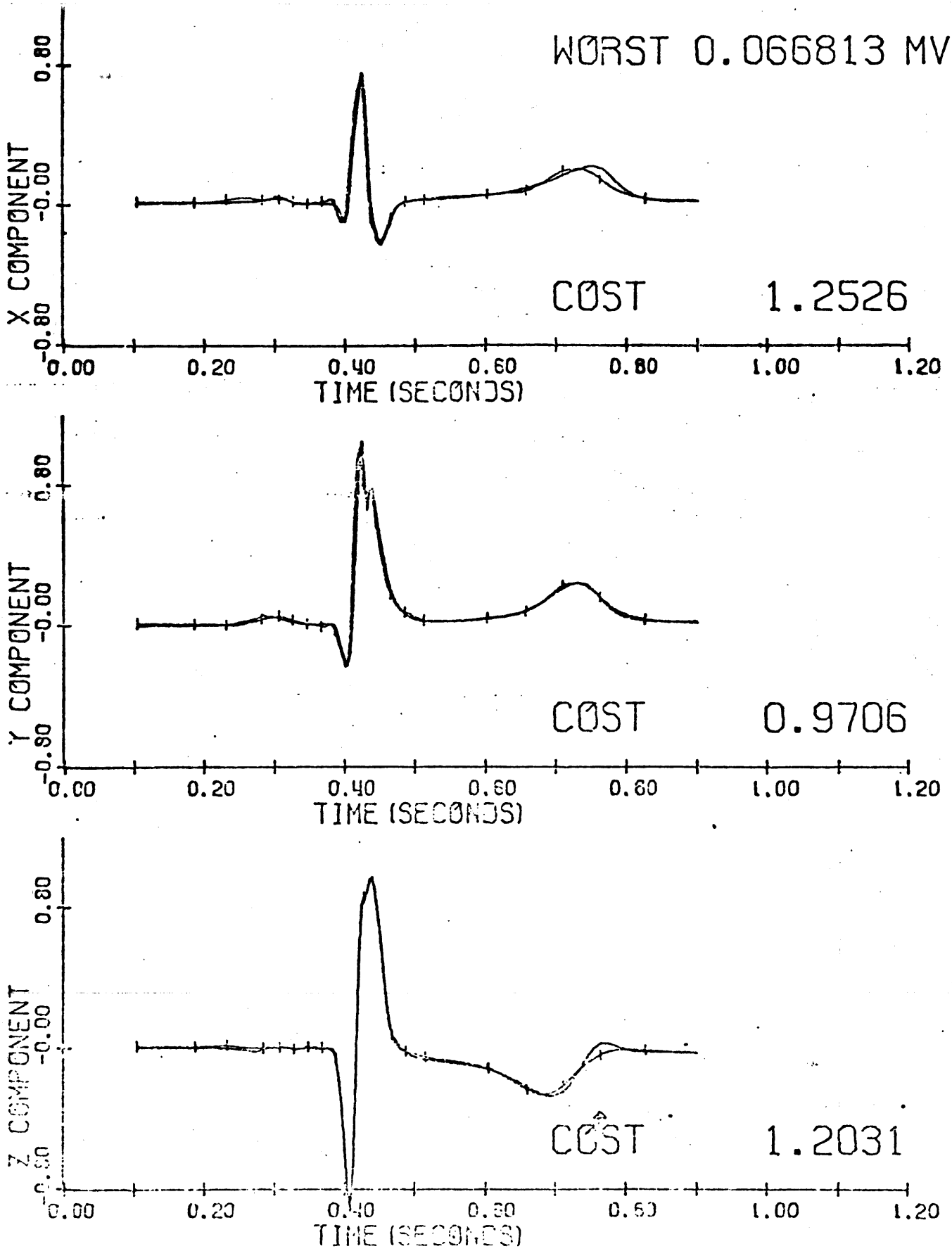


Figure 5.3

Patient 8328 50 Features

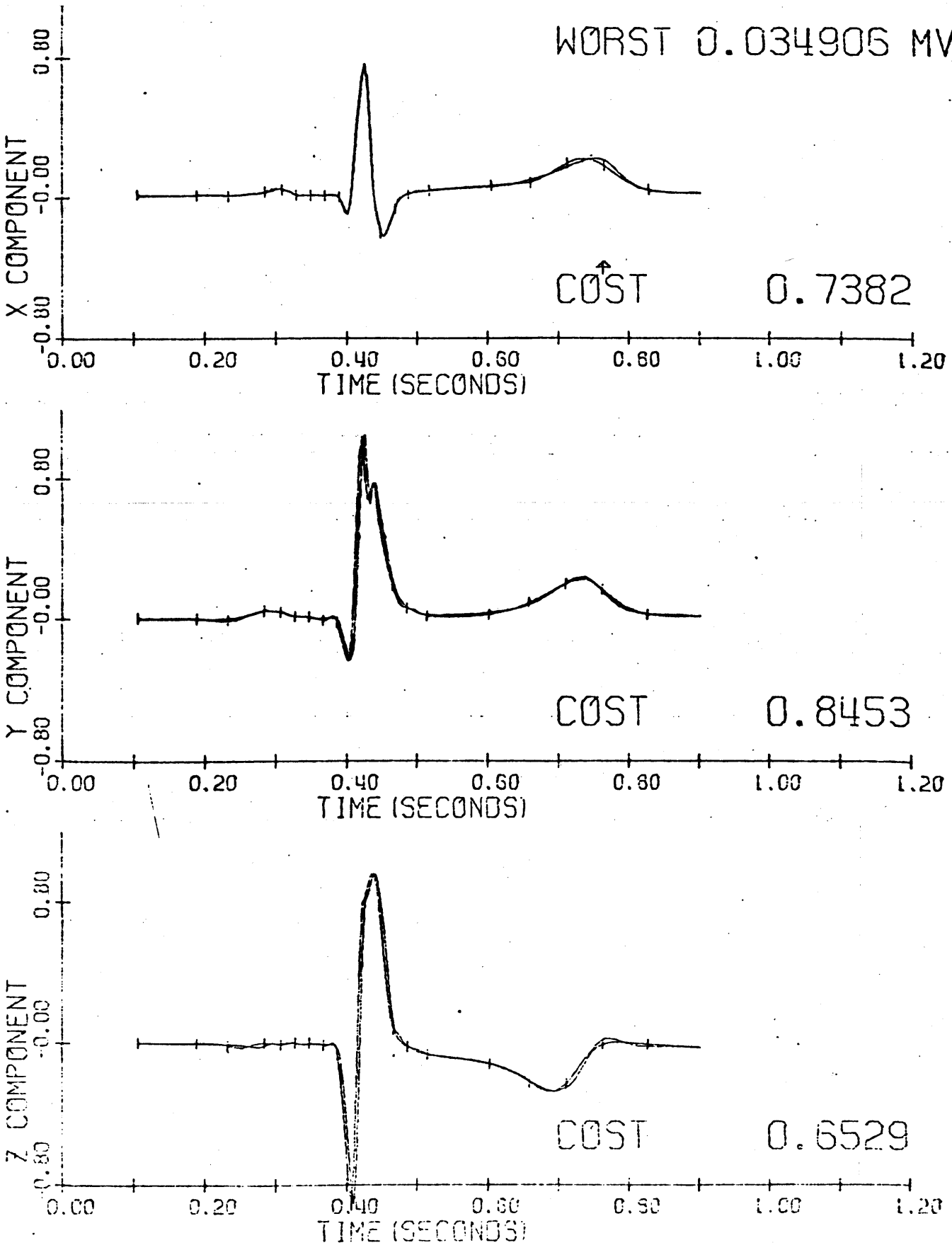


Figure 5.4

Patient 8328 60 Features

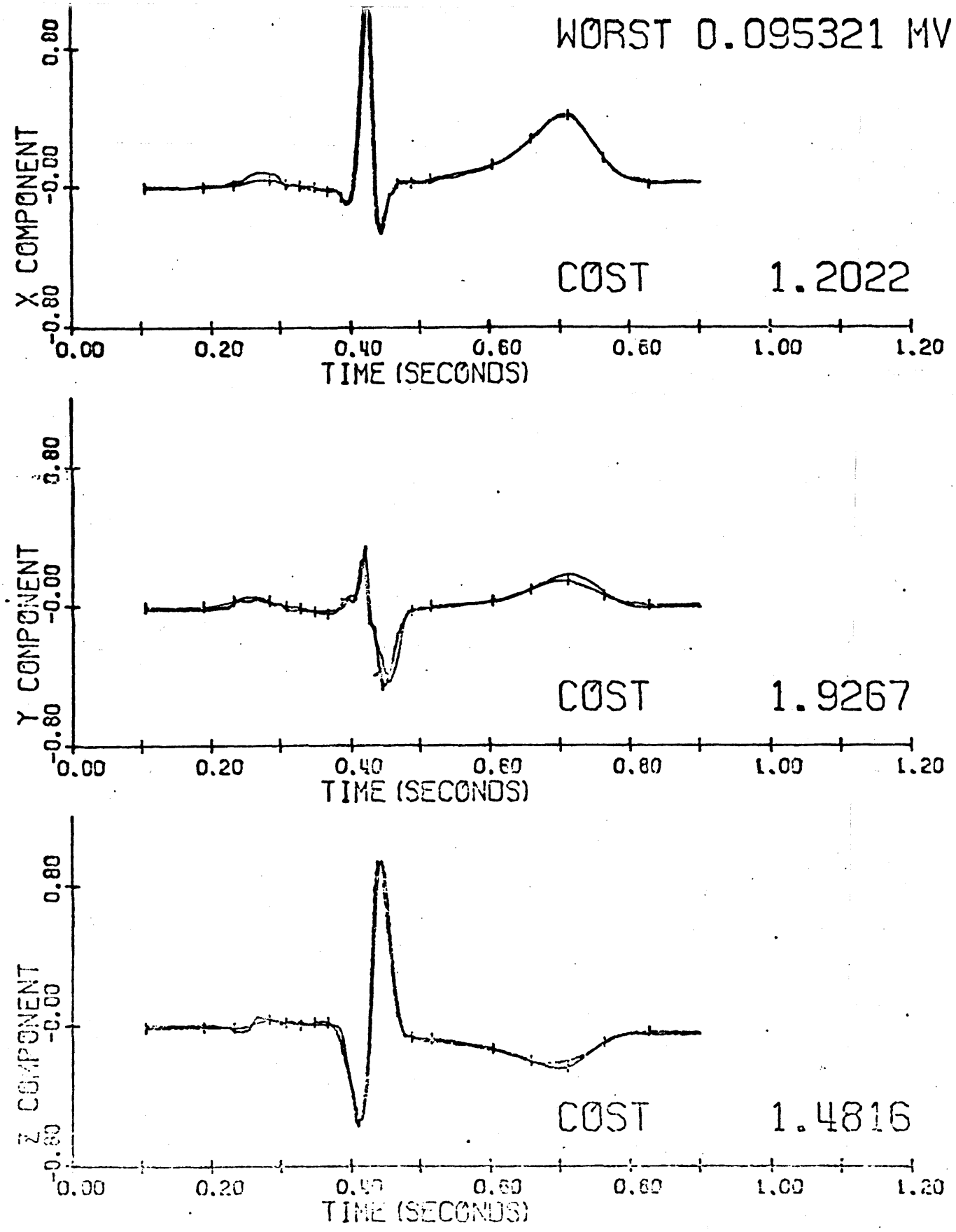


Figure 5.5

Patient 8334 30 Features

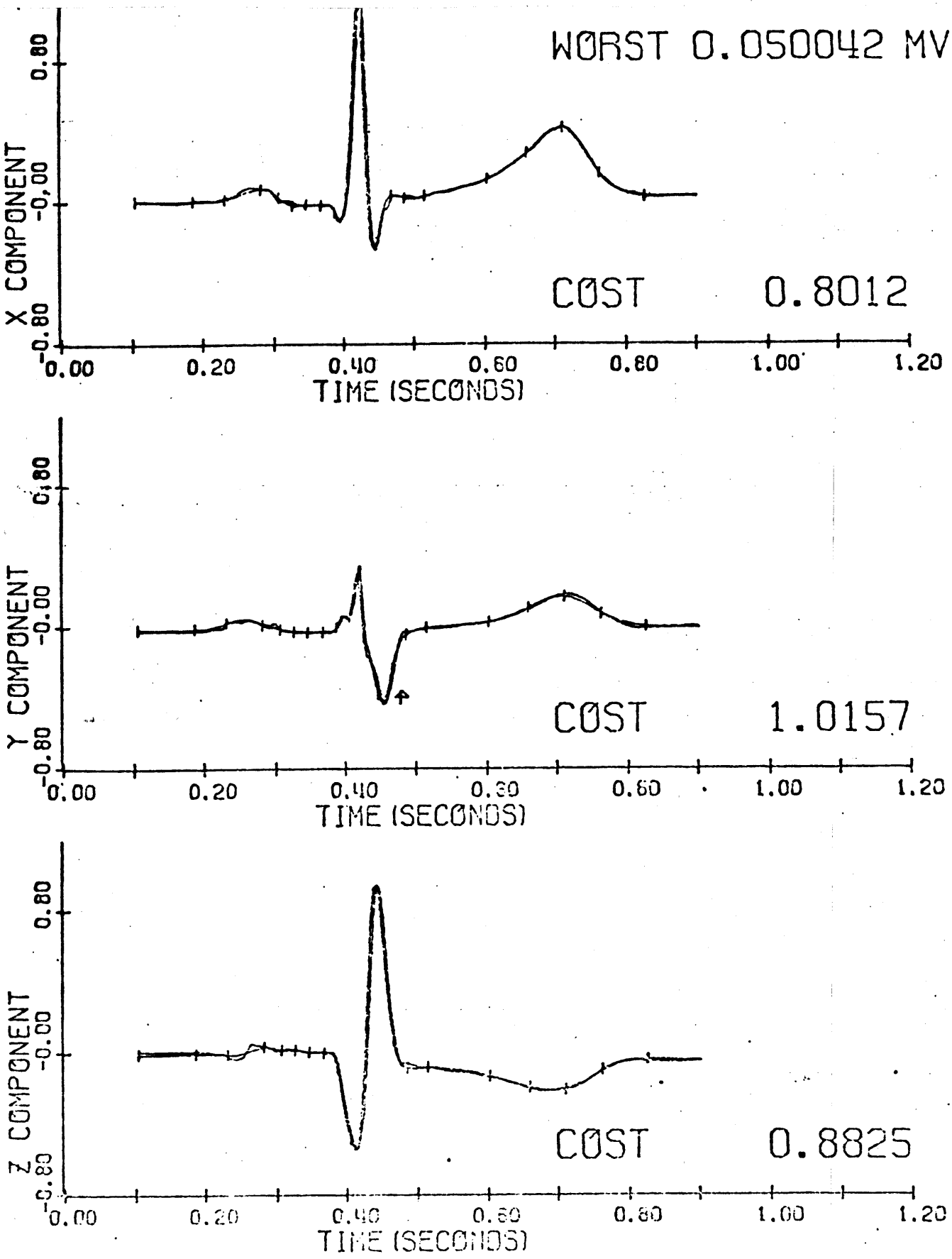


Figure 5.6

Patient 8334 50 Features

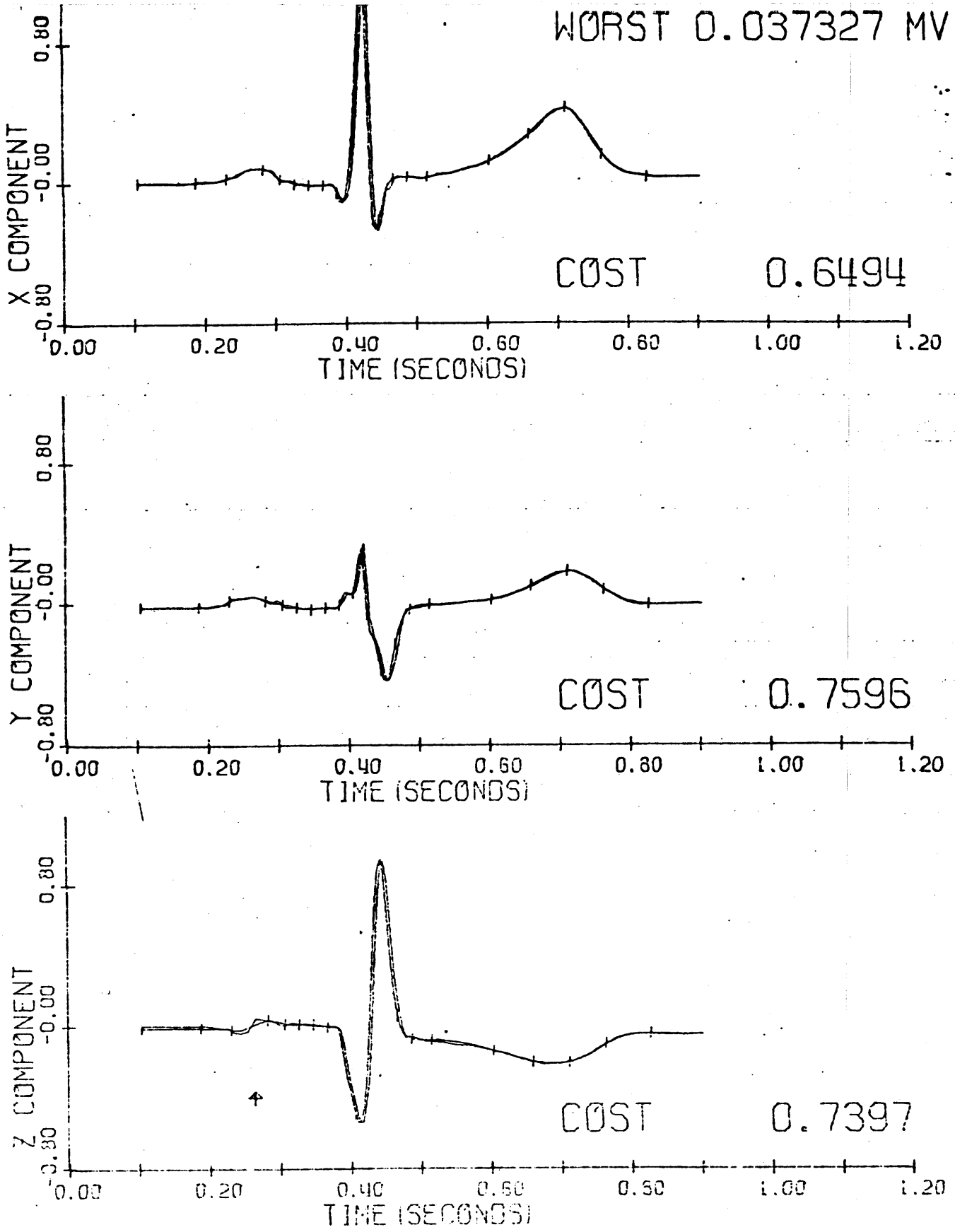


Figure 5.7

Patient 8334 60 Features

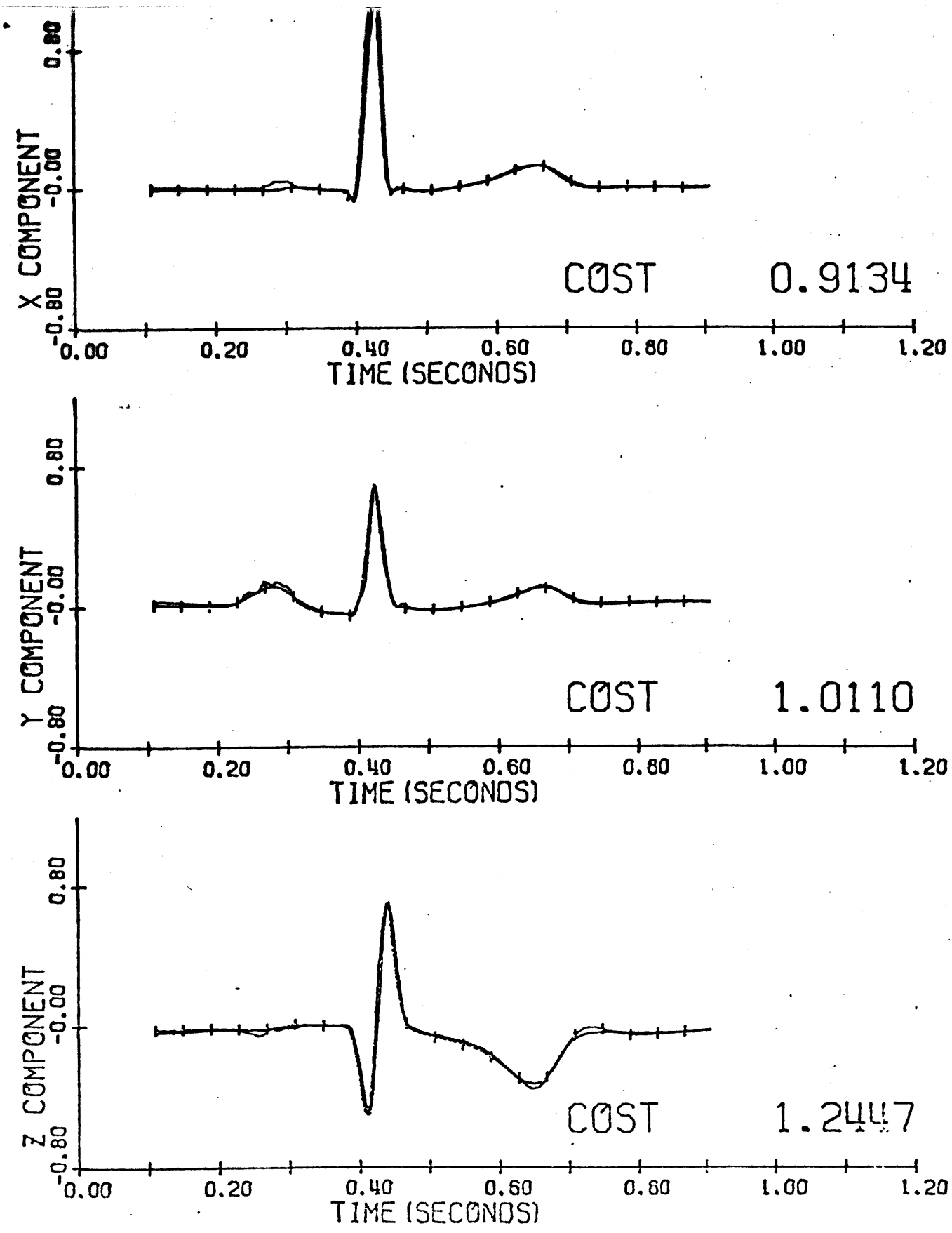


Figure 5.8

Patient 8323 10 Features Each Lead

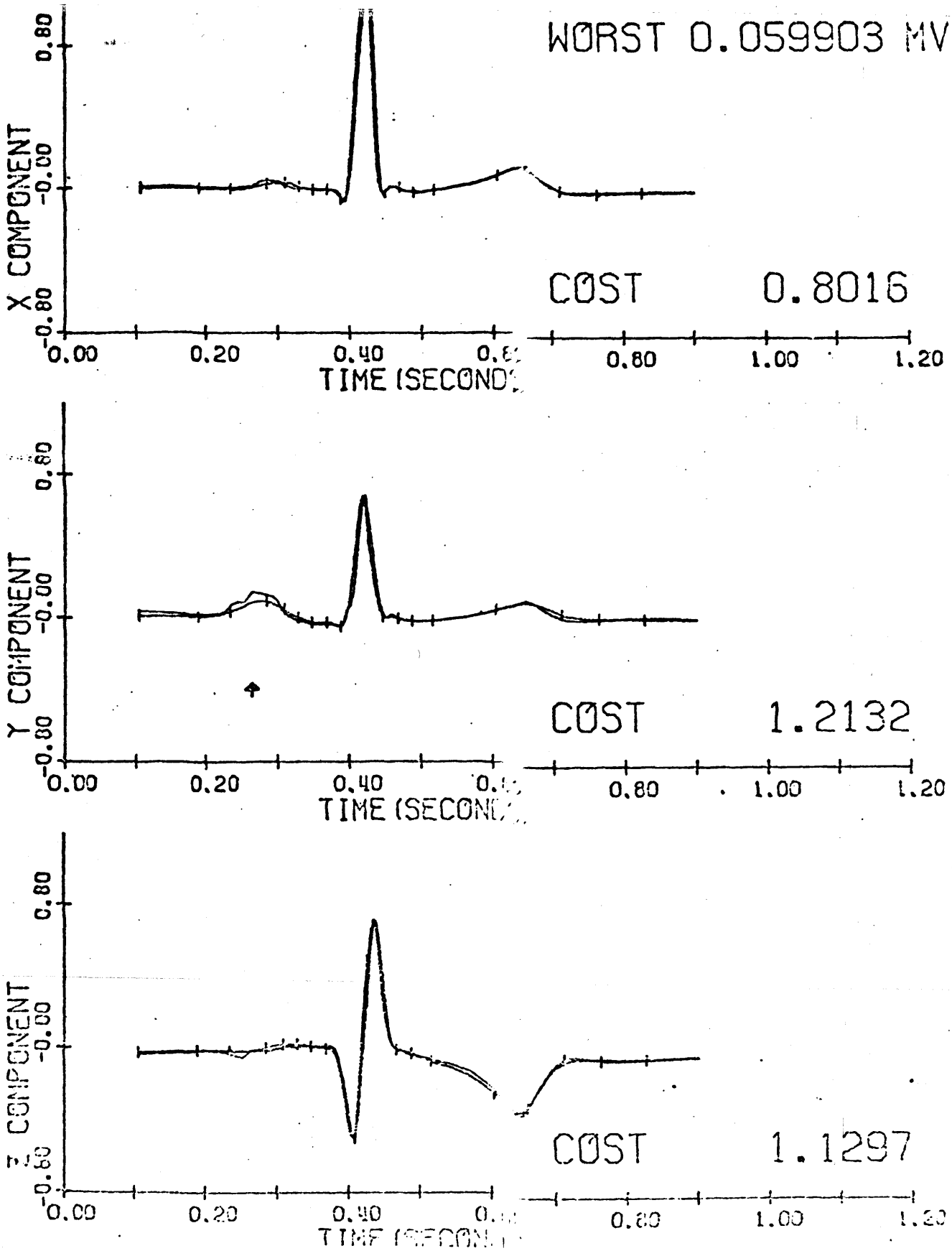


Figure 5.9

Patient 8323 30 Features

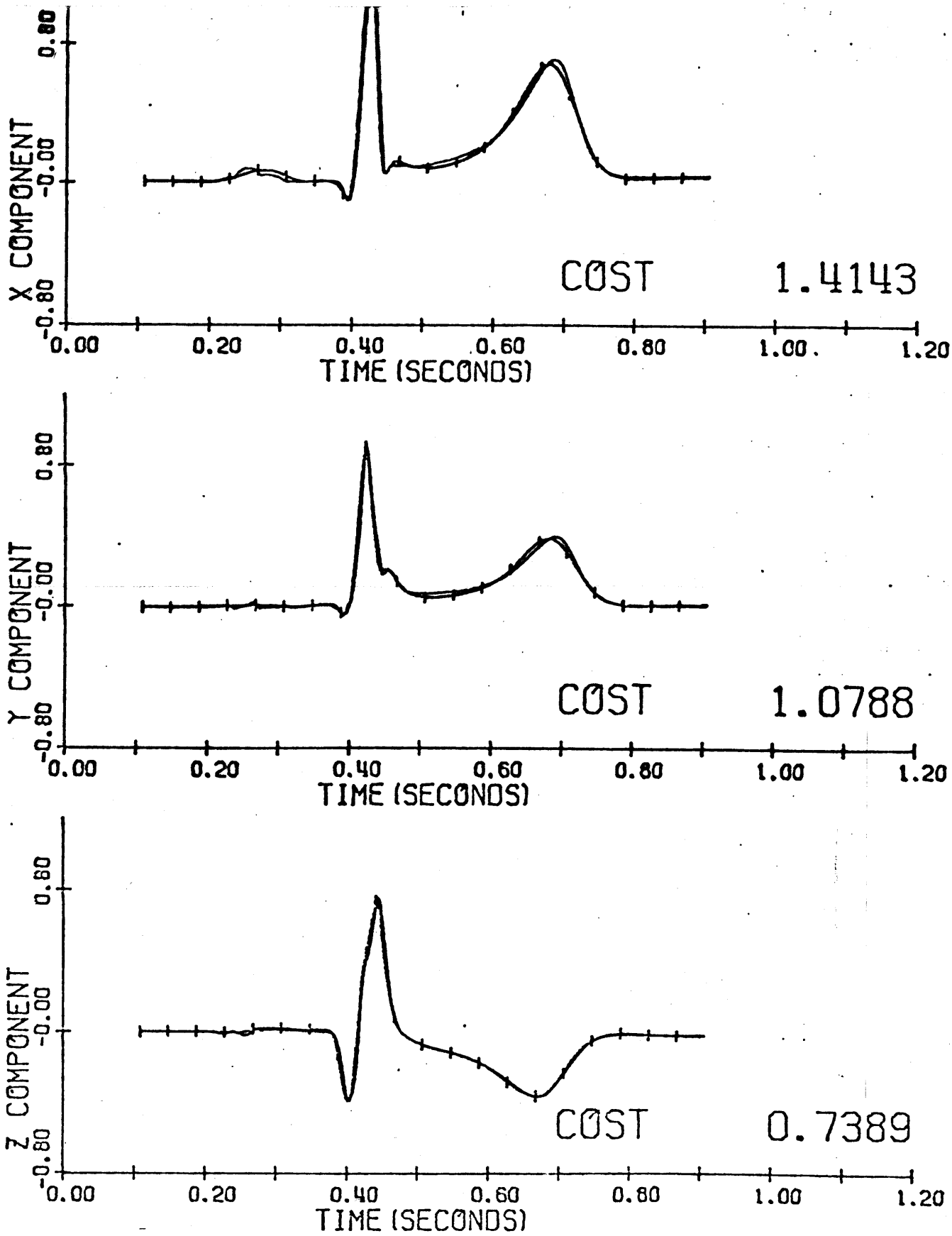


Figure 5.10

Patient 8324 10 Features Each Lead

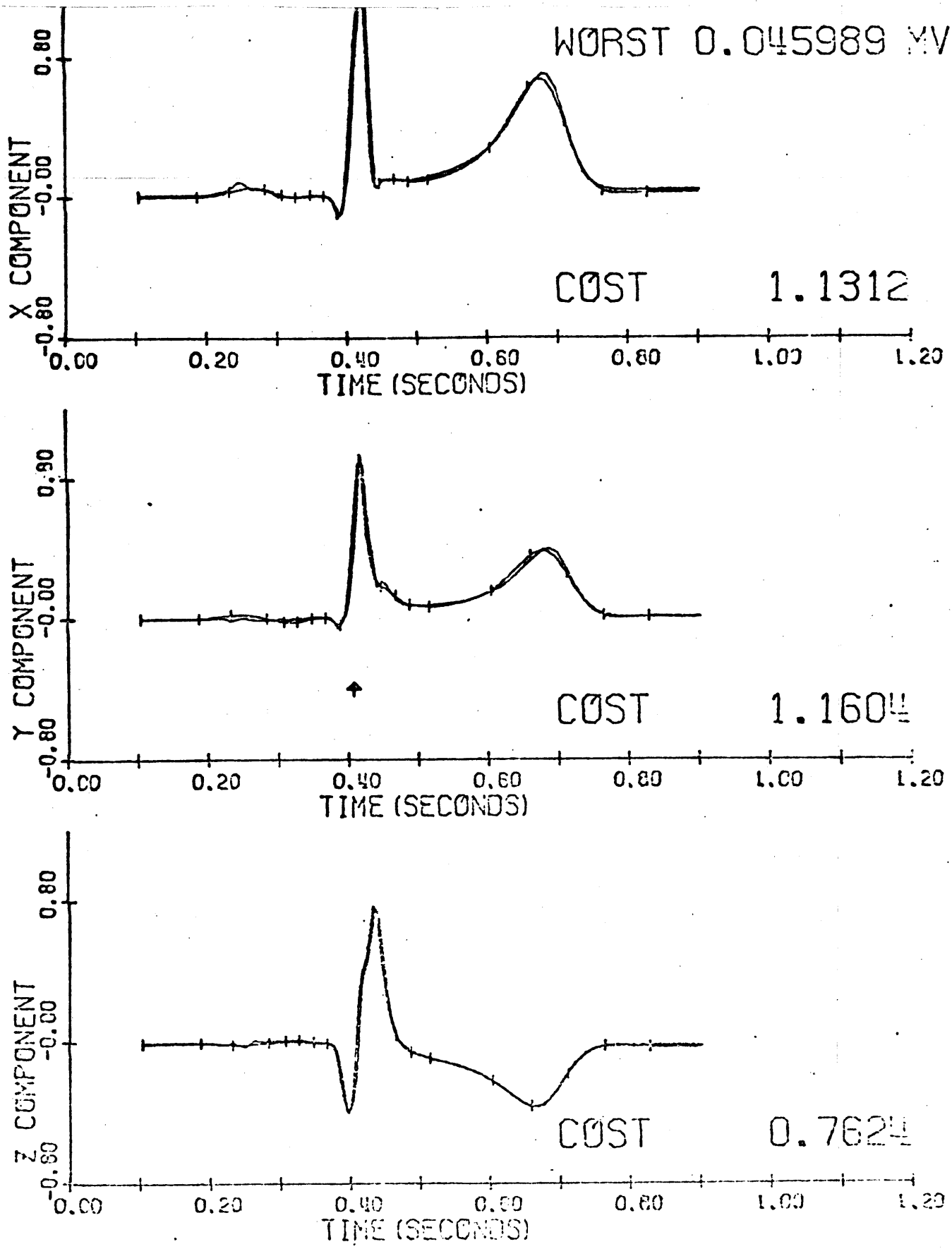


Figure 5.11

Patient 8324 30 Features

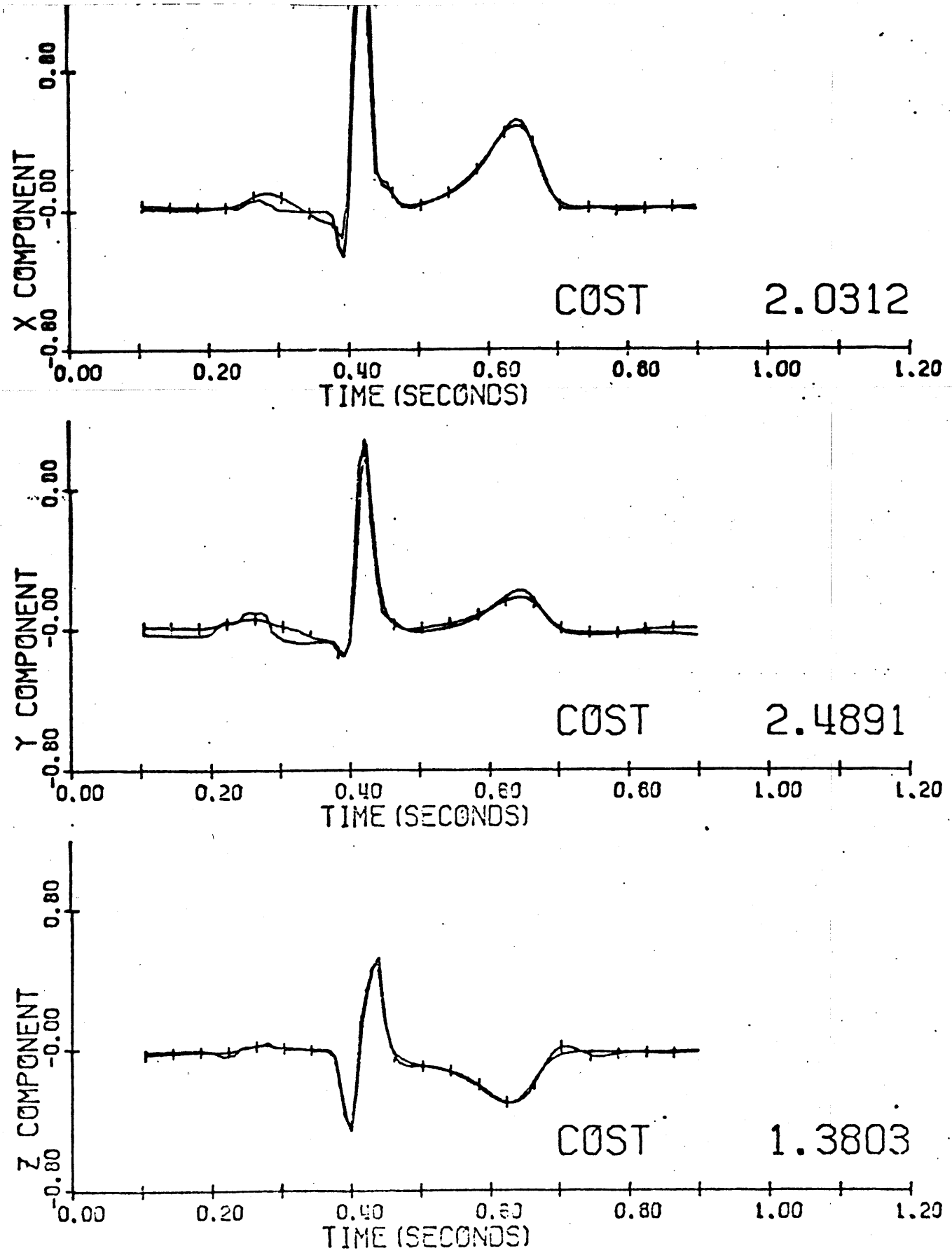


Figure 5.12

Patient 8331 30 Features

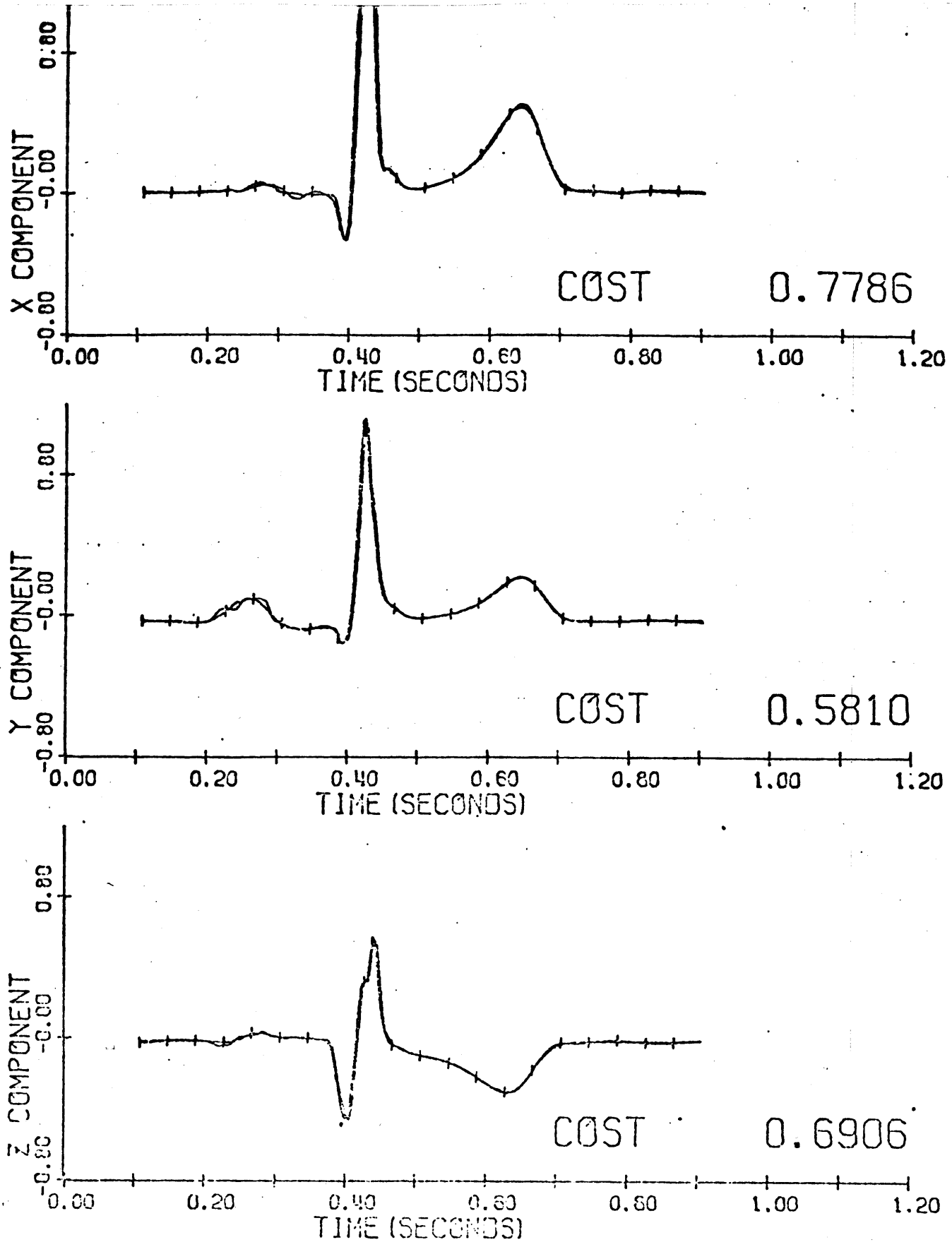


Figure 5.13

Patient 8331 20 Features Each Lead

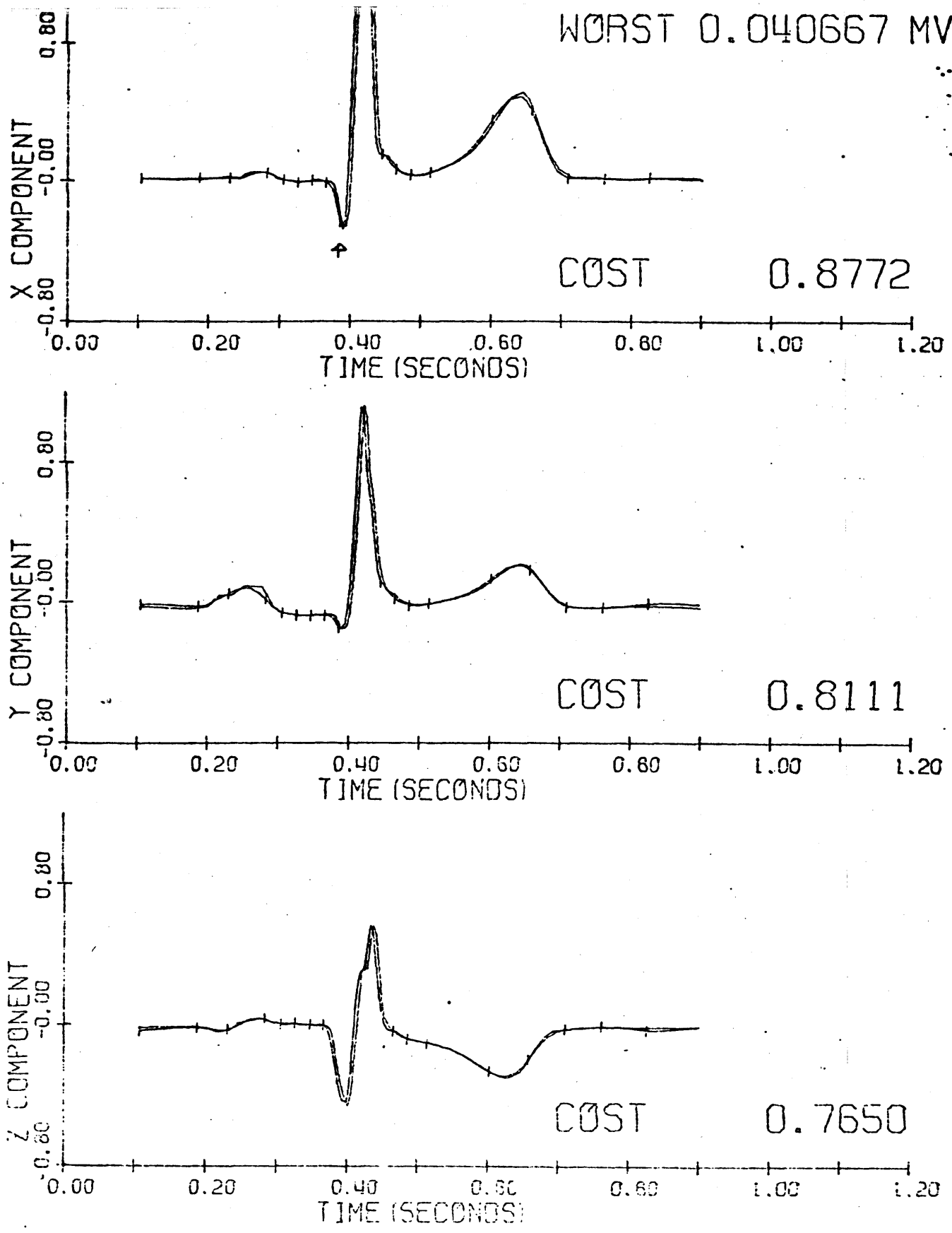


Figure 5.14

Patient 8331 60 Features

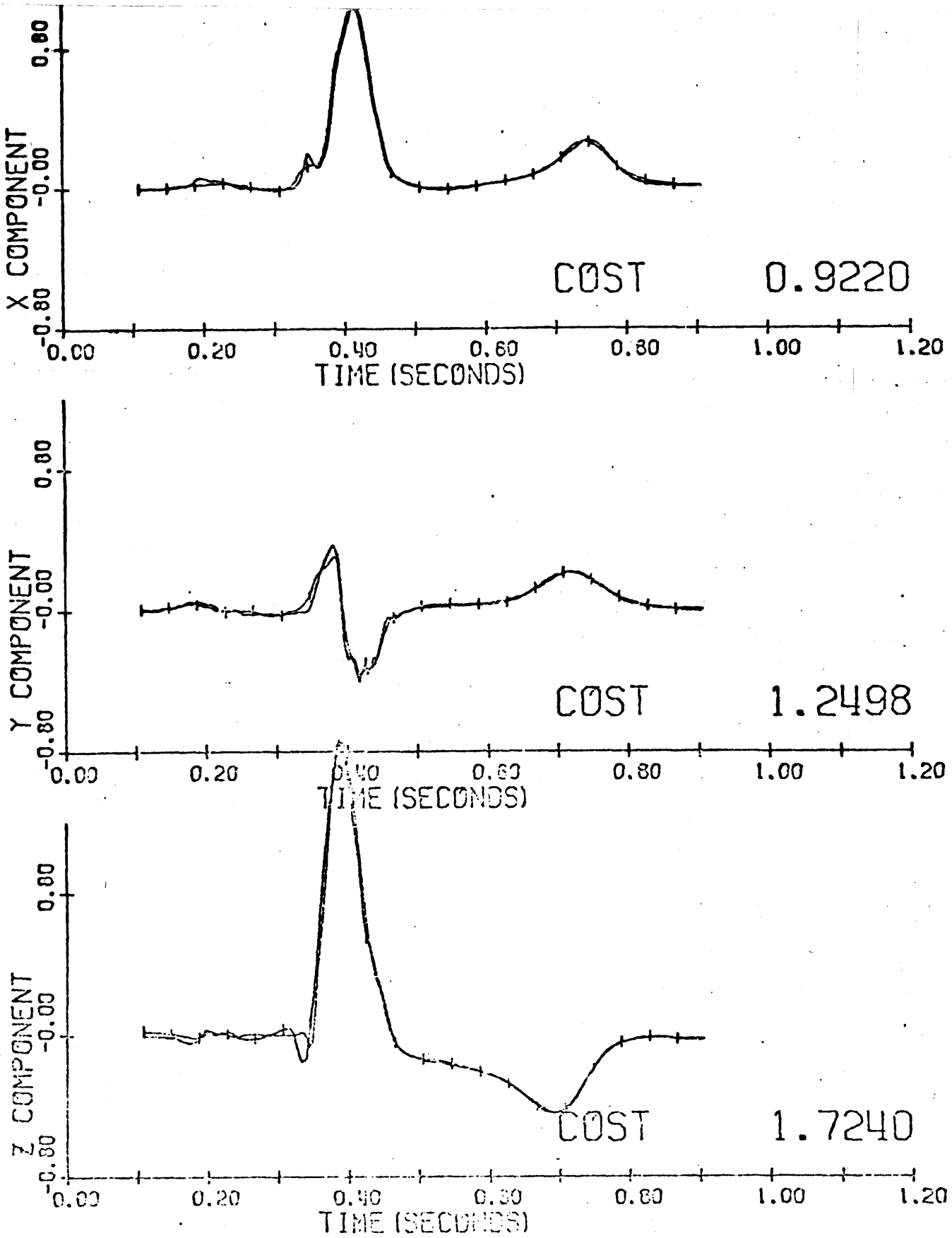


Figure 5.15

Patient 8370 20 Features Each Lead

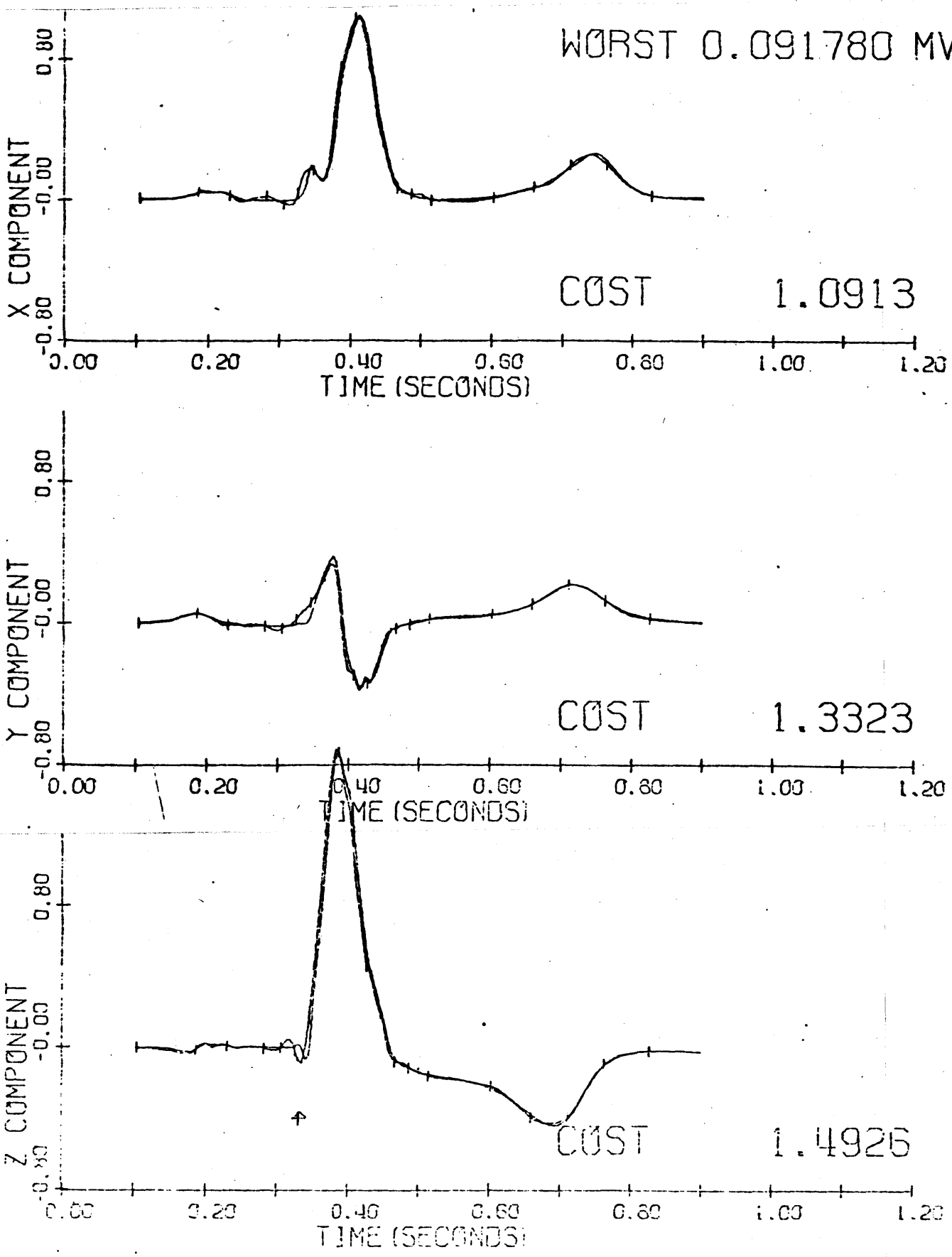


Figure 5.16

Patient 8370 60 Features

CHAPTER 6

CONCLUSION

In the final analysis, the decision that must be made is whether or not the Karhunen-Loeve transform is an effective enough tool in the representation and reduction of VCG data to form the basis for a classification scheme. A definitive answer to this question can be found, of course, only by successfully incorporating the process into an effective diagnostic scheme, which implies the need for a good clustering, or classification algorithm.

Let us assume then, that a good clustering algorithm can be developed. If that is the case, is the Karhunen-Loeve transform a viable pre-clustering process? It is my belief that the studies outlined in this thesis indicate that it is. Since all existing clustering algorithms are extremely dependent for their success on the dimensionality of the data, it is extremely important that such dimensionality be reduced as much as possible, and in this respect the Karhunen-Loeve transform is quite impressive, in that we have been successful in reducing the number of dimensions from 600 (200 samples on each of three leads) to 60.

In addition, the Karhunen-Loeve transform has proved itself useful in another respect: it is a relatively simple, mathematically rigorous process which is very easy to modify. Indeed

one of the major drawbacks of an ad-hoc pattern recognition scheme is that once developed, it becomes a huge, extremely complicated and interrelated system in which all but the simplest modifications are all but impossible to perform. The approach of modular, easy to understand and modify, programs which is possible because of the rigorous mathematical basis of the Karhunen-Loeve expansion has been invaluable in the work which went into this thesis.

Unfortunantly, in looking at this work as a whole, it must be decided that attempts at modifying the transform have not yielded any real, significant improvement. Since the Karhunen-Loeve transform is the optimal linear process in preserving the overall information content of a data base, the only successful improvements can be in the area of preserving that information which is useful for classification, as opposed to that which is not. Any such modification must either be of an ad-hoc variety, which lessens the advantageousness of the expansion, or must be non-linear, which is extremely difficult to deal with.

As a result, it is my belief that the main thrust of our work in the future must be in the area of clustering. It is in this area that hte success or failure of this project will be determined.

REFERENCES

(1) Burch, G.E., and Winsor, T., A Primer of Electrocardiography, Philadelphia, PA, Lea & Febiger, 1966

(2) Halliday, J.S., "The Characterization of Vectorcardiograms for Pattern Recognition", MS Thesis, MIT E.E. Dept., C.S. Draper Lab. Rept. T-583, June, 1973

APPENDIX

PROGRAM LISTINGS

1) Listing of WRSTRED

(see following pages)

```
                C  WRITTEN BY BILL KESSEL 3/12/76                                  00000001
                C   FINDS WORST CASE MATRIX                                       00000002
                C     DEPTH 8- TO CHANGE MUST ALTER LINES 70,170,190,200,240,290   00000003
                C   INPUT: TAPE OF ALL PATIENTS----NEWROT                         00000004
                C   OUTPUT: WORST CASE MATRIX                                     00000005
                C  THIS IS A CHANGE TO EXTEND TABLE TO 12 DEEP                     00000006
                C     WITHOUT REDOING REST OF TABLE                                00000007
0001                  REAL*4 A(3) ,Y(300,3),WRST(200,12)/2400*0./                  00000010
0002                  INTEGER PWRST(200,12)/2400*0/                                00000020
0003                  DO 100 I=1,199                                               00000021
0004                  READ(10,50) (WRST(I,J),J=1,4)                                00000022
0005                100 READ(10,60) (WRST(I,J),J=5,8)                              00000023
0006                  DO 110 I=1,199                                               00000024
0007                  READ(10,90) (PWRST(I,J),J=1,4)                               00000025
0008                110 READ(10,90) (PWRST(I,J),J=5,8)                             00000026
0009                 70 READ(9,END=20) IP,RT,A,((Y(I,J),J=1,3),I=1,300)            00000030
                C  SET FINAL ENDPOINT= 0, SO MIGHT ELIMINATE LAST POINT            00000035
0010                  DO 10 J=1,3                                                  00000040
0011                 10 Y(201,J)=0.                                                00000050
                C  THIS LOOP IS FOR ELIMINATION STARTING AT EACH POINT             00000055
0012                  DO 30 I=1,199                                                00000060
                C  THIS LOOP IS OVER MAXIMUM LENGTH OF ELIMINATION(DEPTH)          00000065
0013                  DO 30 J=9,12                                                 00000070
                C  THIS LOOP FOR EACH LEAD(WORST LEAD FOUND IN EACH CASE)          00000075
0014                  DO 30 K=1,3                                                  00000080
                C  FIND HGT DIFFERENCE BETWEEN BEGIN AND END                       00000085
0015                  DIF=(Y(I+J+1,K)-Y(I,K))/(J+1)                                00000090
0016                  DO 30 JJ=1,J                                                 00000100
0017                  BADNES=ABS(Y(I,K)+DIF*JJ-Y(I+JJ,K))                          00000110
                C  IS IT WORST YET AT THAT POINT?                                  00000115
0018                  IF(BADNES .LE. WRST(I,J))  GO TO 30                          00000120
                C  YES- REPLACE OLD VALUE                                          00000125
0019                  WRST(I,J)=BADNES                                             00000130
0020                  PWRST(I,J)=IP                                                00000140
0021                 30 CONTINUE                                                   00000150
0022                  GO TO 70                                                     00000160
                C  NOW MAKE SURE ELIMINATIONS DO NOT GO PAST END OF TIME           00000165
0023                 20 DO 40 J=1,11                                               00000170
0024                  DO 40 K=1,J                                                  00000180
0025                  ILAB=188+J                                                   00000190
0026                  JLAB=(12-J)+K                                                00000200
0027                 40 WRST(ILAB,JLAB)=99.                                        00000210
0028                   DO 50 I=1,199                                               00000220
                C                                                                  00000222
                C                                                                  00000224
                C  FINALLY OUTPUT MATRIX                                           00000226
0029                 50 WRITE(10,60) (WRST(I,J),J=9,12)                            00000240
0030                 60 FORMAT(4E18.8)                                             00000250
                C  AND OUTPUT CORRESPONDING PATIENTS                               00000255
0031                  DO 80 I=1,199                                                00000260
```

```
0032          80 WRITE(10,90) (PARST(I,J),J=9,12)                    00000275
0033          90 FORMAT(4I5)                                         00000280
0034             CALL EXIT                                           00000290
0035             END                                                 00000300
```

2) Listing of WRSTFIND

(see following pages)

```
                C    WRITTEN BY BILL KESSEL 3/13/76                                    00000001
                C    FIND BEST COLUMNS TO ELIMINATE SEQUENTIALLY BY                    00000002
                C     USING WORST CASE MATRIX CREATED BY WRSTRED                       00000003
                C    LOWEST VALUE FOR WORST CASE IS, OF COURSE, ELIMATED FIRST         00000004
                C   INPUT:  WORST CASE MATRIX FROM WRSTRED                            00000005
                C   OUTPUT: LIST OF 200 POSSIBLE TIME POINTS, 1 IF DELETED,0 IF NOT   00000006
                C    PLOT OF DATA SAMPLE TIMES TO BE USED                              00000007
                      REAL*4 WRST(204,12)/2448*99./,Y(102)/100*1.,0.,1./               00000010
0001                  REAL*4 X(102)/101*0.,0.2/,BUFFER(304)                            00000020
0002                  INTEGER PWRST(200,12)/2400*0/,DEL(200)/200*0/,DELTOT             00000030
0003                  NB=304                                                           00000040
0004                                                                                   00000045
                C    DELTOT IS RUNNING TOTAL POINTS DELETED AT ANY TIME                00000050
                      DELTOT=0                                                         00000055
0005            C   READ WORST CASE MATRIX AND CORRESPONDING PATIENT NUMBERS          00000060
                      DO 10 I=1,199                                                    00000070
0006                  READ(9,30) (WRST(I,J),J=1,4)                                     00000075
0007                10 READ(9,30) (WRST(I,J),J=5,8)                                    00000080
0008                  DO 20 I=1,199                                                    00000090
0009                  READ(9,40) (PWRST(I,J),J=1,4)                                    00000095
0010                20 READ(9,40) (PWRST(I,J),J=5,8)                                   00000100
0011                30 FORMAT(4E18.8)                                                  00000102
0012                  DO 400 I=1,199                                                   00000104
0013                400 READ(9,30) (WRST(I,J),J=9,12)                                  00000106
0014                  DO 410 I=1,199                                                   00000108
0015                410 READ(9,40) (PWRST(I,J),J=9,12)                                 00000110
0016                 40 FORMAT(4I5)                                                    00000112
0017            C                                                                      00000114
                C                                                                      00000116
                C                                                                      00000118
                C   FIND LOWEST ENTRY IN MATRIX                                      00000120
                  110 WLOW=99.                                                         00000130
0018                  DO 50 I=1,199                                                    00000140
0019                  DO 50 J=1,12                                                     00000150
0020                  IF(WLOW .LE. WRST(I,J)) GO TO 50                                 00000160
0021                  ILOW=I                                                           00000170
0022                  JLOW=J                                                           00000180
0023                  WLOW=WRST(I,J)                                                   00000190
0024                 50 CONTINUE                                                       00000192
0025            C                                                                      00000194
                C                                                                      00000196
                C   FIND OUT HOW MANY SAMPLES ARE ACTUALLY DELETED                    00000198
                C     (SOME MAY BE GONE ALREADY)                                       00000200
                      NDEL=0                                                           00000210
0026                  DO 60 K=1,JLOW                                                   00000220
0027                  IF (DEL(ILOW+K) .EQ. 0) NDEL=NDEL+1                              00000230
0028                 60 DEL(ILOW+K)=1                                                  00000235
0029            C      ERROR IF NONE DELETED                                           00000240
                      IF(NDEL .EQ. 0) GO TO 70                                         00000245
0030            C   AND ADD TO RUNNING TOTAL                                         00000250
                      DELTOT =DELTOT+NDEL
0031
```

```
                C  OUTPUT MOST RECENT ELIMINATION                                  00000255
                                                                                   00000260
0032                   WRITE(6,80) NDEL,JLOW,ILOW                                  00000270
0033                80 FORMAT(1X,I2,' COLUMNS OUT OF ',I2,' FOLLOWING',I4,         00000280
                      1   ' DELETED')                                              00000290
0034                   WRITE(6,90) PWRST(ILOW,JLOW),WLOW                           00000300
0035                90 FORMAT('    WORST PATIENT',I5,'     VALUE',E18.8)          00000302
                C                                                                  00000304
                C                                                                  00000306
                C  NOW UPDATE MATRIX TO SHOW RECENT ELIMINATION                    00000308
                C   FIRST OVERWRITE ENTRY JUST ELIMINATED                         00000310
0036                   WRST(ILOW,JLOW)=99.                                         00000330
0037                   DO 100 K=1,12                                               00000340
0038                   DO 100 J=1,JLOW                                             00000345
                C  OVERWRITEENTRIES STARTING ON ELIMINATED POINT                   00000350
0039                   WRST(ILOW+J,K)=99.                                          00000355
                C    THEN ENTRIES ENDING ON EIMINATED POINT                        00000360
0040                   INDX=(ILOW-(K+1))+J                                         00000370
0041                100 WRST(INDX,K)=99.                                           00000375
                C  CHECK TO SEE IF ENOUGH POINTS HAVE BEEN ELIMINATED              00000380
0042                   IF(DELTOT .LT. 100) GO TO 110                               00000385
                C  AND OUTPUT LIST OF ALL POINTS                                   00000390
0043                   WRITE(10) DEL                                               00000400
0044                   WRITE(6,200) DELTOT                                         00000410
0045                200 FORMAT(' TOTAL OF ',I4,' COLUMNS DELETED')                 00000412
                C                                                                  00000414
                C                                                                  00000416
                C                                                                  00000418
                C  NOW PLOT SAVED POINTS                                           00000420
0046                   CALL PLOTS(BUFFER,NB,1)                                     00000430
0047                   CALL PLOT(2.,2.,-3)                                         00000440
0048                   CALL AXIS(0.,0.,4HTIME,-4,6.,0.,0.,.2,10.)                  00000450
0049                   CALL SYMBOL(2.,8.,0.2,12HCOLUMNS USED,0.,12)                00000460
0050                   INDX=1                                                      00000465
0051                   DO 120 I=1,200                                              00000470
0052                   IF (DEL(I) .EQ. 1) GO TO 120                                00000480
0053                   TIME=.1+.004*I                                              00000490
0054                   X(INDX)=TIME                                                00000500
0055                   INDX=INDX+1                                                 00000510
0056                120 CONTINUE                                                   00000520
0057                   CALL LINE(X,Y,100,1,-1,13)                                  00000530
0058                   CALL WHERE(A,B,FACT)                                        00000540
0059                   CALL PLOT(A,B,999)                                          00000550
0060                   CALL EXIT                                                   00000560
0061                 70 WRITE(6,210) DELTOT,DEL                                    00000570
0062                210 FORMAT('  PROGRAM LOGIC ERROR'/I4,5(/1016))                00000580
0063                   CALL EXIT                                                   00000590
0064                   END
```