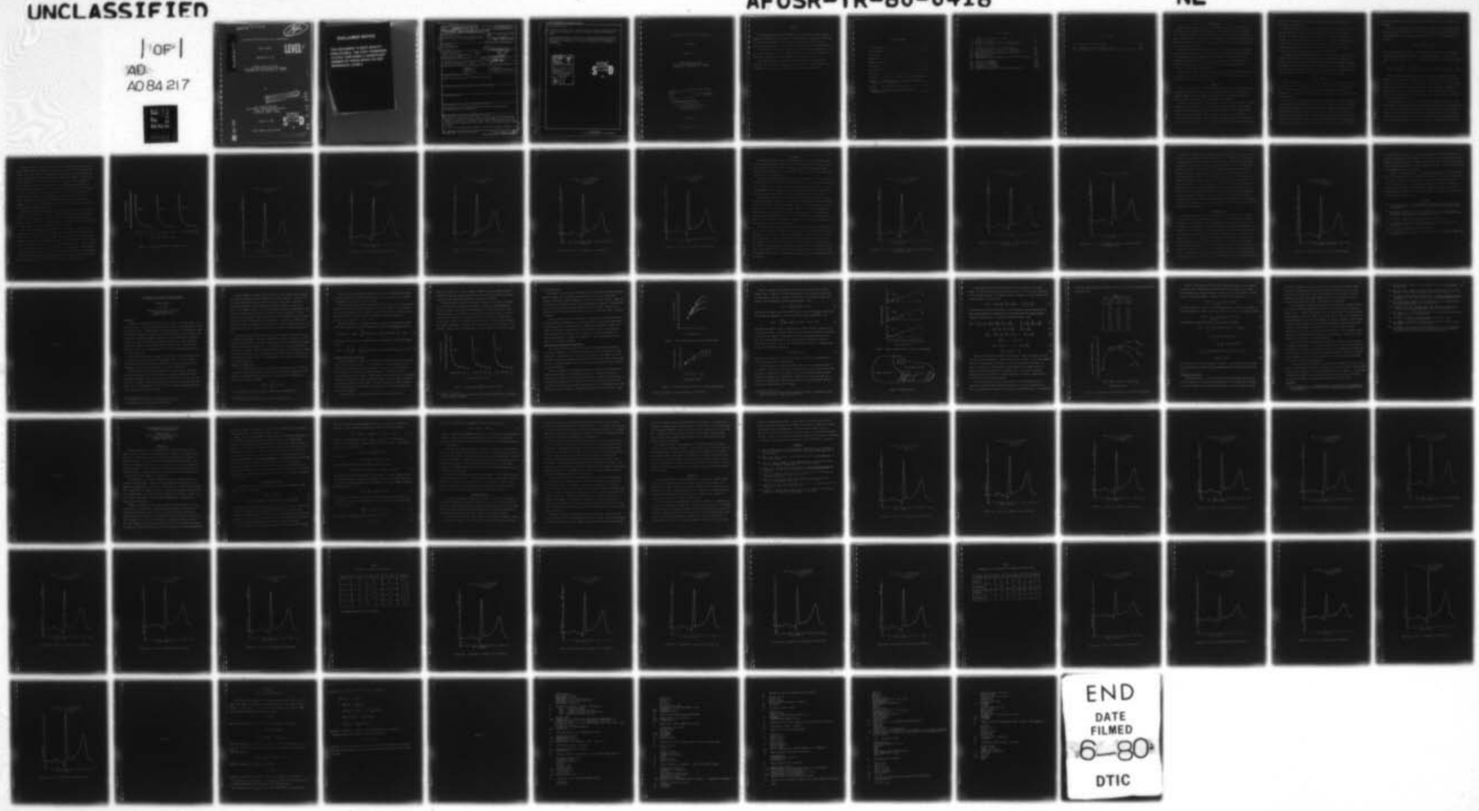
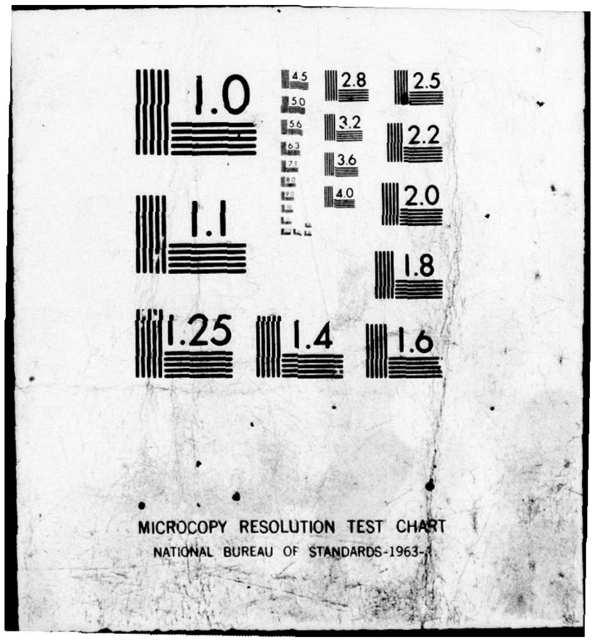
AD-A084 217 TEXAS A AND I UNIV KINGSVILLE DEPT OF ELECTRICAL ENG--ETC F/G 9/4
AN ADAPTIVE PREDICTOR FOR DATE COMPRESSION.(U)
MAR 80 M HANKAMER AFOSR-79-0074
UNCLASSIFIED AFOSR-TR-80-0418 NL

1 OF 1
AD
A084 217



END
DATE
FILMED
6-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 80-0418

ADA084217

AN ADAPTIVE PREDICTOR FOR DATA COMPRESSION

LEVEL II

FINAL REPORT

SUBMITTED TO THE

UNITED STATES AIR FORCE
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
BUILDING 410 BOLLING AFB, DC 20332

BY

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

DR. MICHAEL HANKAMER
DEPARTMENT OF ELECTRICAL ENGINEERING
TEXAS A&I UNIVERSITY
KINGSVILLE, TEXAS 78363

DTIC
ELECTE
MAY 19 1980
S D
D

MARCH 31, 1980

GRANT NUMBER AFOSR-79-0074

DDC FILE COPY

80 5 14 073

Approved for public release;
distribution unlimited.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-80-0418
2. GOVT ACCESSION NO.: AD-A084 247
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): AN ADAPTIVE PREDICTOR FOR DATE COMPRESSION
5. TYPE OF REPORT & PERIOD COVERED: Final
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Michael Hankamer
8. CONTRACT OR GRANT NUMBER(s): AFOSR-79-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Texas A&I University
Dept. of Electrical Engineering
Kingsville, TX 78363
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304 A6
11. CONTROLLING OFFICE NAME AND ADDRESS:
Air Force Office of Scientific Research/NM
Bolling AFB, Washington, DC 20332
12. REPORT DATE: Mar[redacted], 1980
13. NUMBER OF PAGES: 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report):
Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
ECG, Digital ECG, Data Compression, Digital Encoding, Entropy, Linear Minimum-Mean-Square Error Estimation
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The problem of predicting the n^{th} sample of a periodically stationary random sequence (a digitized ECG) using a set of L prior samples is considered. The entropy of the source is calculated, using a Markov source model, to find that entropy decreases rapidly with source order. Only a very short predictor should be needed.

The linear, least-mean-square estimator is derived and computer simulated. It

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 411754

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract cont.

is shown to be short (L=1), relatively robust, moderately accurate (usually within 10%), and adaptive in that the estimator improves from period to period.

Data compression ratios of about 4:1 can reasonably be expecte from direct application of the predictor; however, by judicious deletion and later regeneration of samples, it is felt that an additional 4:1 compression is achievable.

Accession For	
NTIS GRA&I	X
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	23

DTIC
ELECTE
MAY 19 1980
S D
D

AN ADAPTIVE PREDICTOR FOR DATA COMPRESSION

Final Report

submitted to the

United States Air Force
Air Force Office of Scientific Research
Building 410 Bolling AFB, D.C. 20332

by

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

Dr. Michael Hankamer
Department of Electrical Engineering
Texas A&I University
Kingsville, Texas 78363

March 31, 1980

Grant Number AFOSR-79-0074

ABSTRACT

The problem of predicting the n^{th} sample of a periodically stationary random sequence (a digitized ECG) using a set of L prior samples is considered. The entropy of the source is calculated, using a Markov source model, to find that entropy decreases rapidly with source order. Only a very short predictor should be needed.

The linear, least-mean-square estimator is derived and computer simulated. It is shown to be short (L=1), relatively robust, moderately accurate (usually within 10%), and adaptive in that the estimator improves from period to period.

Data compression ratios of about 4:1 can reasonably be expected from direct application of the predictor; however, by judicious deletion and later regeneration of samples, it is felt that an additional 4:1 compression is achievable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Direct digital transmission of electrocardiagram data is increasingly important to the USAF. The School of Aerospace Medicine, Brooks AFB, Texas has over 800,000 ECG's stored in its Central Electrocardiographic Library and the number is growing at well over 100 per day. In an effort to ease the workload on Air Force physicians and make the Library more accessible for medical research, the ECG data is being converted to a machine accessible format.

American Heart Association standards [1] call for a sampling rate of 500 samples/second and sample quantization of 9 bits/sample. The data rate and machine storage capacity implied by these requirements is not acceptable; thus the search for an efficient method of data compression is on. This research proposal was based on a linear least mean-square error predictor derived by the author during the 1978 USAF/ASEE Summer Faculty Research Program [2].

The objective of this research was threefold: to develop a software simulation of the algorithm; to study its performance; and to develop a data base sufficient to estimate performance of a practical system.

NARRATIVE

Prior to the grant receipt a "mini-program" (referred to in the original proposal) had been written and tested on fake data. While the results based on the fake data were not conclusive, they did indicate that the prediction algorithm would tend to follow the data. On that basis, the grant proposal was submitted.

During the spring semester, 1979, the author suggested a graduate student research problem: to compute the entropy of a digitized ECG, assuming a Markov source model. Although this was not part of the original grant proposal, it later proved to be one of the more interesting aspects. The entropy of a data source is well-known to lower-bound the average number of bits required to transmit a sample; thus a computation of the entropy should give some insight into how well the predictor algorithm can do its work. The results of the entropy computation program (ENTROPY) will be discussed in the next section.

After the grant was received in the Spring of 1979, there proved to be a great deal of difficulty in obtaining the digitized VCG's from the School of Aerospace Medicine. The author, having never been exposed to the vagaries of magnetic tape transfer, was totally unprepared for the difficulties. Personnel changes within the branch that was to supply the VCG's also contributed to the problem, which was eventually solved when the author went to Brooks AFB and brought back listings of several VCG's for later entry into the Texas A&I computer system.

The entropy computation program (ENTROPY) was finally running on actual data in August 1979; the first version of the prediction algorithm (ECG1) was running in October; and an improved version of the predictor (ECG2) was established in December. The delay associated with getting data from the School of Aerospace Medicine and the difficulties associated with performing a tape-to-tape transfer precluded trying the algorithm on abnormal VCG's, but the remaining objectives were satisfied.

SUMMARY OF RESULTS

The details of the research carried through the programs ENTROPY and ECG are to be found in the papers "The Entropy of a Digitized Electrocardiagram" (Appendix A) and "A Least Mean-Square Prediction Algorithm for Digital Electrocardiagraphy" (Appendix B). In this summary, we paraphase the results which are described in detail in the Appendices.

First we consider the meaning and calculations of entropy. Shannon's noiseless coding theorem [3] roughly states that for any data source there exists a code whose average message length is lower-bounded by the source entropy; the Huffman coding procedure [4] explicitly generates that code. Thus it follows that the entropy measures the average number of bits required to transmit a sample from an ECG. But the entropy offers more than just a lower bound for direct encoding of the samples; entropy can be defined for sources with memory, in which case the change in entropy with memory length may prove useful in defining the proper length of a prediction algorithm.

Let S be a discrete-time source with each message quantized to one of 2^N levels (the source contains 2^N messages). S emits a message sequence with each message s_i drawn independently from S with probability p_i. Then the (zero-order) entropy is defined as

$$H_0(S) \triangleq - \sum_{i=1}^{2^N} p_i \log(p_i)$$

The messages may not be independent. Suppose that the probability of the message s_i depends on the preceeding M messages; e.g., $p(s_i)=p(s_i|s_{j1},s_{j2},\ldots,s_{jM})$. Then the source is called a Markov source of order (memory) M, and the source entropy is given by

$$H_M(S) \triangleq - \sum_{j1=1}^{2^N} \sum_{j2=1}^{2^N} \cdots \sum_{jM=1}^{2^N} \sum_{i=1}^{2^N} p(s_{i1}s_{j1},s_{j2},\ldots,s_{jM}) \log p(s_i|s_{j1},s_{j2},\ldots,s_{jM})$$

Most prediction algorithms attempt to predict the next sample from the prior samples, as for instance, in differential PCM; the current sample is treated as the estimate of the next sample. This is equivalent to treating the source as an order 1 Markov source. Similarly, linear extrapolation can be considered as equivalent to treating the sou-ce as an order 2 Markov source. Data compression occurs if the prediction is good and the difference are small relative to the samples, as only the differences need be transmitted.

The entropy thus measures the data compression capability in two ways. First, the entropy of the order M Markov source bounds the performance of M-length predictor: e.g., if $H_2(S)$ = 2.5 bits/sample, then it follows that, on the average, the best a length 2 predictor can do is to get within 2.5 bits/sample of the original message. Indirectly, the entropy measures the compressibility of the data by indicating the loss associated with lowering the sample rate or more coarsely quantizing the data.

Results are fully described in Appendix A; the fundamental results being three. First, a short prediction algorithm is clearly most appropriate. As can be seen from Figure 1, the source entropy drops dramatically when modeled as a first-order Markov source. Successive reduction in the entropy for higher order models is evident, but not as dramatic as the reduction in going from a zero-order to first-order Markov model. Second, there is some loss (e.g., decrease in entropy) as the sample rate decreases and the quantization is made coarser, but as is pointed out in the Appendix, it is not clear how significant this loss actually is. It is clear, however, that quantization and sample rate are interdependent. Third, the first order entropy $H_1(S)$ -- approximately 1 bit/sample -- probably represents the limit in "easy" compression; as it can be shown (Appendix A) that $H_1(S)$ is a measure of the beat-to-beat variation of the electrocardiagram.

The adaptive predictor program (ECG) was written in two versions (ECG1 and ECG2) ECG1 was less complex, but more unstable. It worked satisfactorily for a predictor length L of 1 but would tend to come unglued for $L \geq 2$. ECG2 resolved the problem of ECG1, but at the cost of added complexity required to compute the correlation functions exactly. Only ECG2 (listing in Appendix D) will be discussed.

The predictor algorithm behaved quite well: independently of sample rate and quantization level, the prediction was within 10% of the true value most of the time. Figure 2((a)-(e)) shows predictor performance versus the actual digitized VCG at 500 samples/second and 11 bit quantization per sample. The maximum error on the first beat occurs at the peak of the R-wave and is a bit over 16% off. The average error was only 19.9 or a little over 4 bits/sample as compared to the 11 bit/sample original message. In particular, the improvement from first to fifth beat must be noted; as the predictor algorithm continuously updated itself with improved correlation information from the prior samples of the VCG being estimated, its prediction clearly improves.

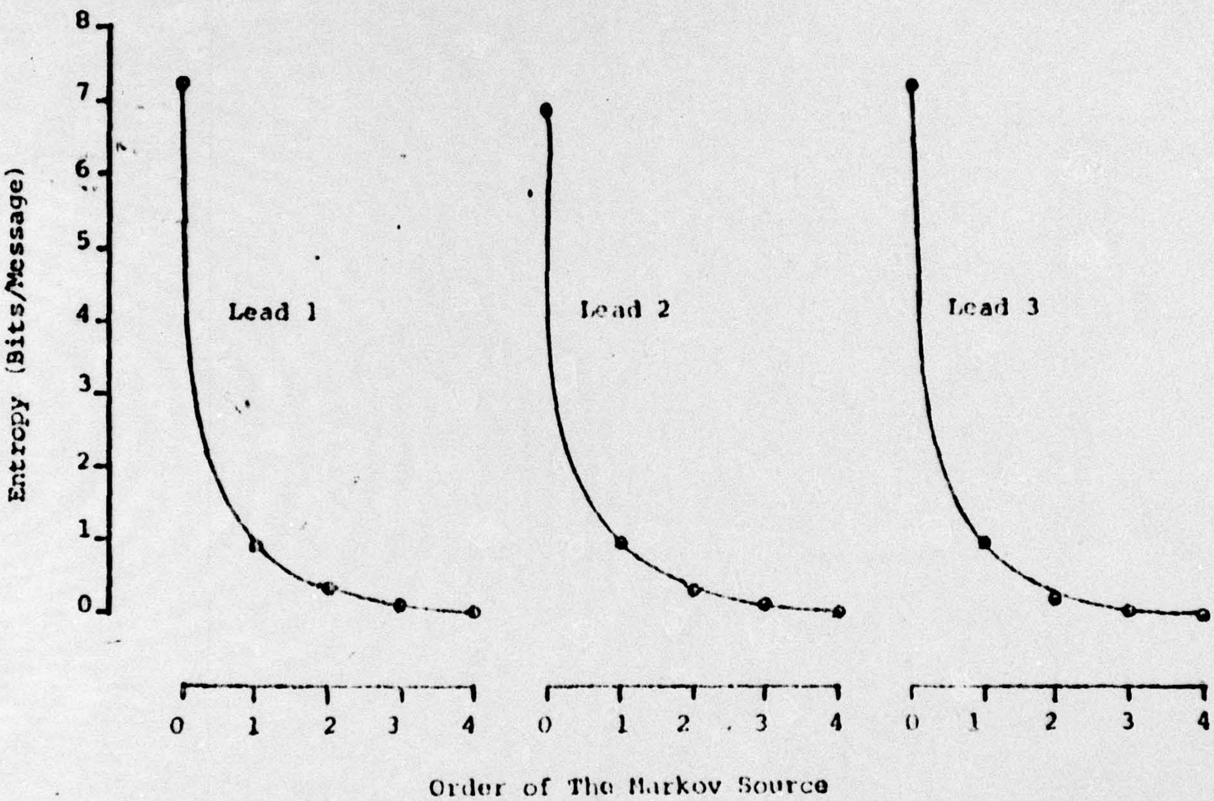


Figure 1: Entropy of VCG Modeled as a Markov Source

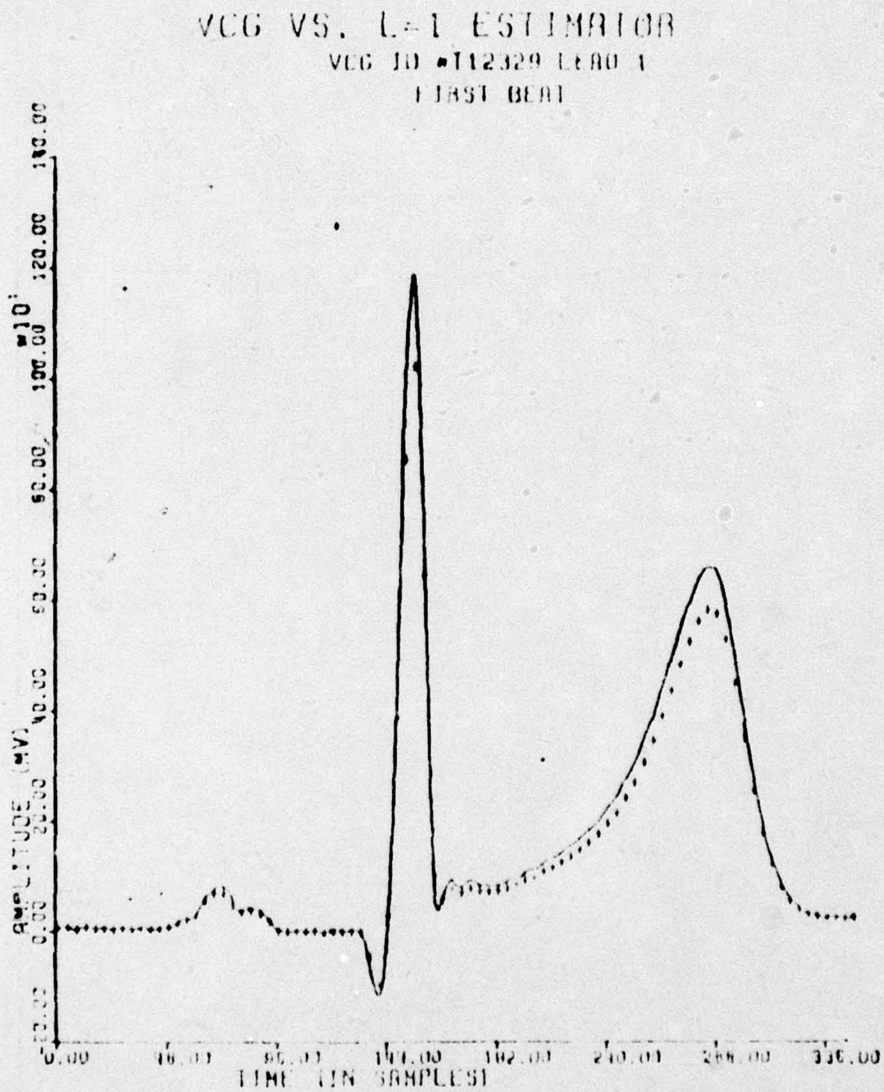


Figure 2(a): VCG vs. L=1 Estimator (Q=11, R=500 s/s)

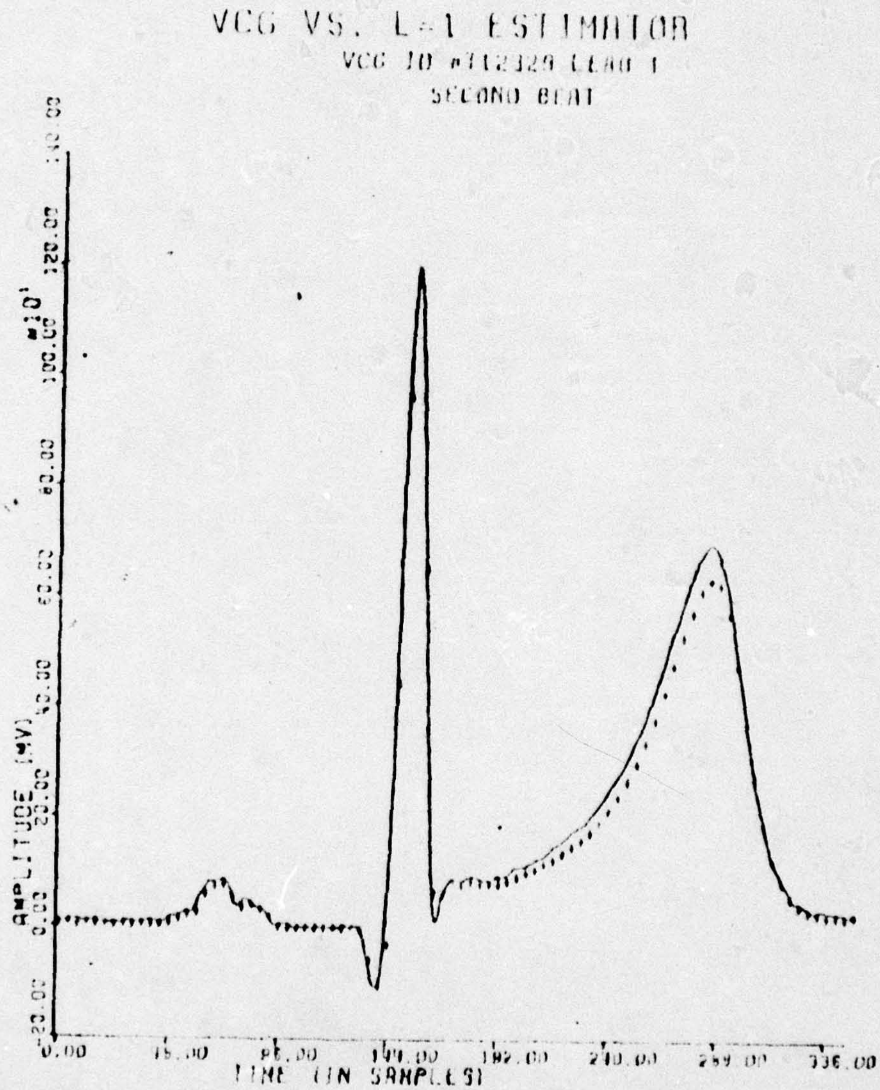


Figure 2(b): VCG vs. L=1 Estimator (Q=11, R=500 s/s)

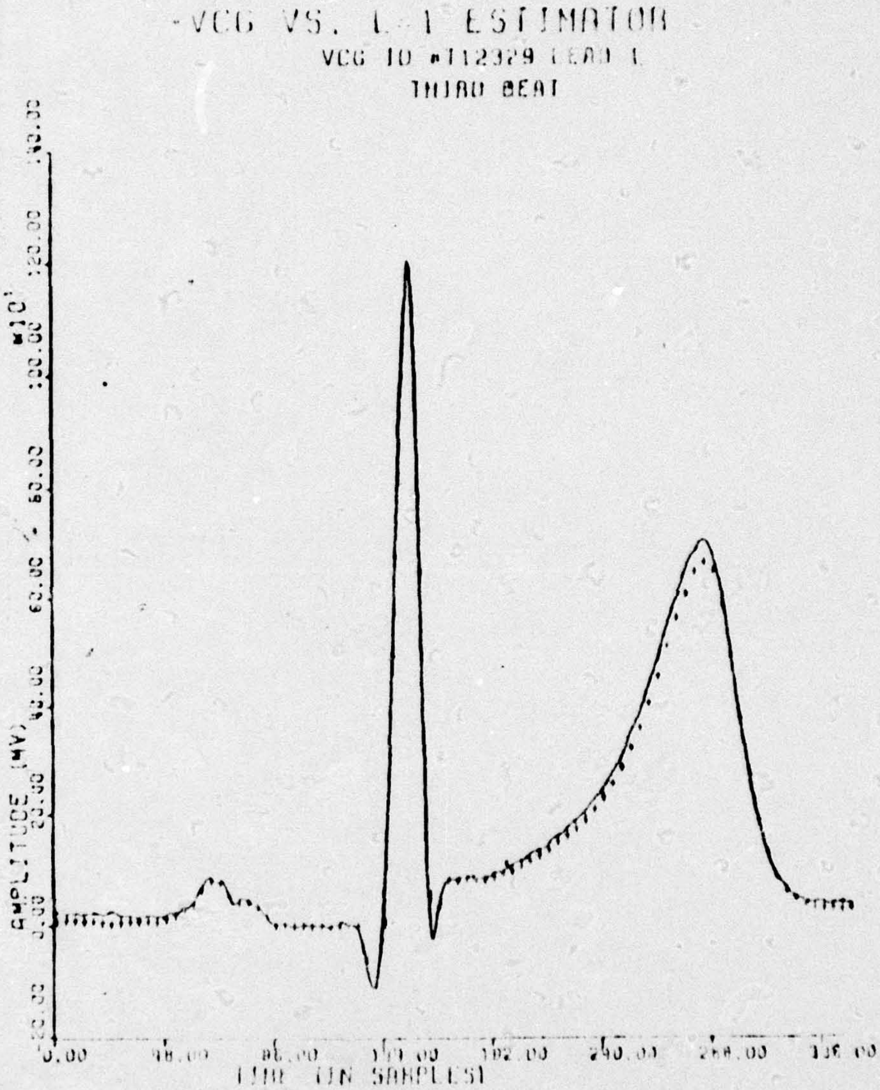


Figure 2(c): VCG vs. L=1 Estimator (Q=11, R=500 s/s)

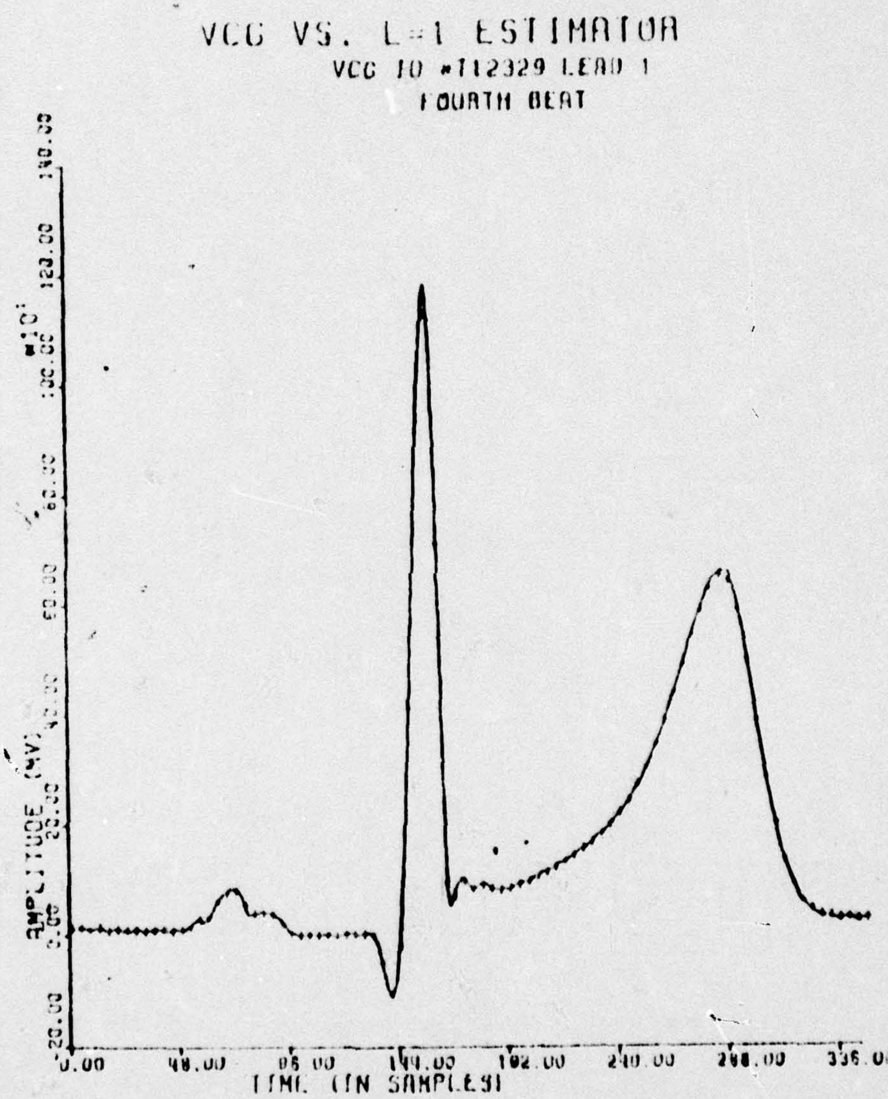


Figure 2(d): VCG vs. L=1 Estimator (Q=11, R=500 s/s)

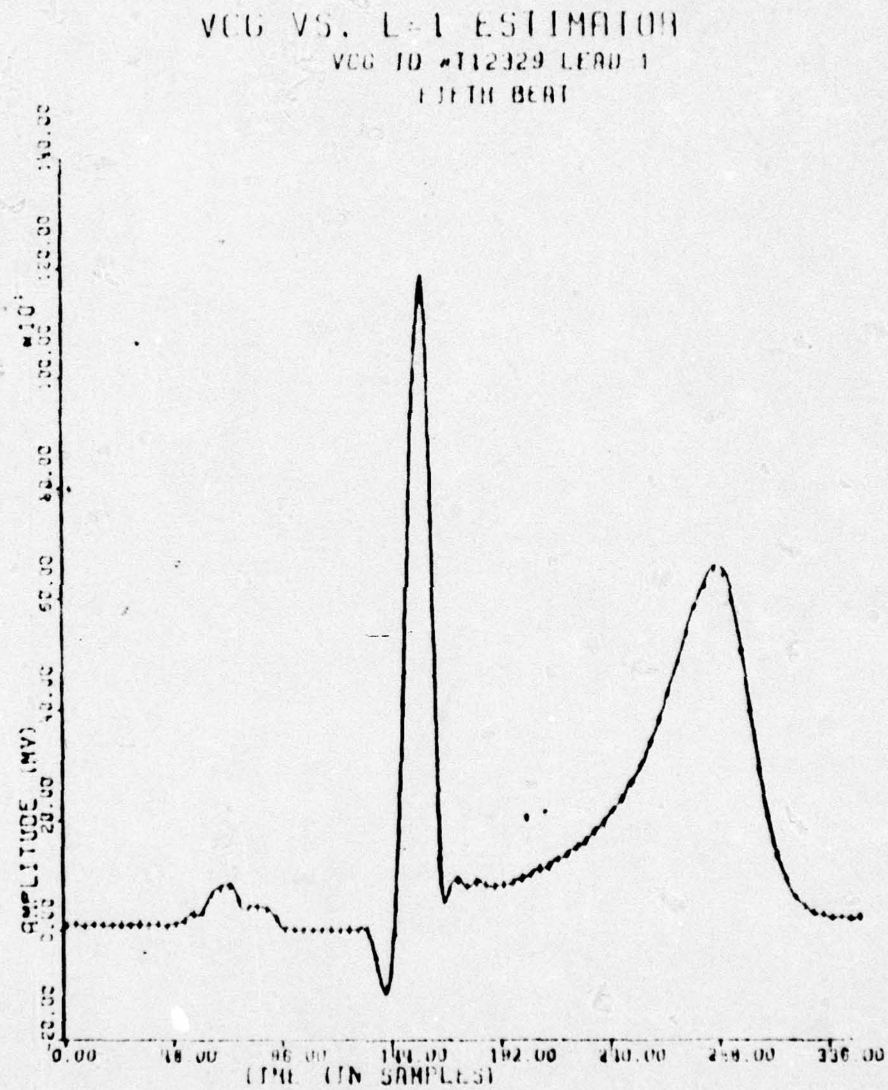


Figure 2(e): VCG vs. L=1 Estimator (Q=11, R=500 s/s)

DISCUSSION

The adaptive least-mean-square algorithm proposed in the 1978 USAF/ASEE Summer Faculty Program and studied as part of this grant request should prove useful on any discrete signal sequence that can be modeled as a periodically stationary random sequence. The algorithm has one free parameter -- the length L -- which is best chosen on the basis of source statistics.

The notion of entropy and the Markov source model have been shown to be useful in analyzing the source. The L^{th}-order entropy has been found to be useful in determining the best length for the predictor algorithm. The entropy was shown to be less useful in determining an optimum sample rate or quantization level, primarily because entropy is not a function of the samples themselves, but of their probabilities instead.

With particular emphasis on the electrocardiagram problem data analysis suggests that the L=1 predictor is the best. It is the least complex, and the longer predictors (L=2 and L=3) appear to have slightly higher average and mean-squared errors. Moreover $H_L(S) < 1.0$ for $L > 1$, which implies a more complicated ... but not impossibly so ... Huffman coding procedure would be required to take advantage of the reduced entropy. (This presumes some improvement in the L=2 and L=3 predictors.) Since $H_1(S)$ can be shown to be a measure of the statistical "irregularity" of the electrocardiagram, it is doubtful that such an improvement is possible.

The optimum sample rate and quantization level are 250 samples/second and 8 bits/sample. This is, however, a judgment call based on the fact that the first evidence of distortion is seen in the 250 sample/second simulation. Consider Figure 3, in which the L=1 predictor is used at (a) 500 samples/second, (b) 250 samples/second, and (c) 125 samples/second. The "glitches" that are clearly evident at the beginning of the QRS complex in the 125 samples/second plot first appear in the 250 samples/second data. There is no evidence of such an appearance at 500 samples/second. The choice of 8 bit quantization results from the interdependence of sample rate and quantization described in Appendix A.

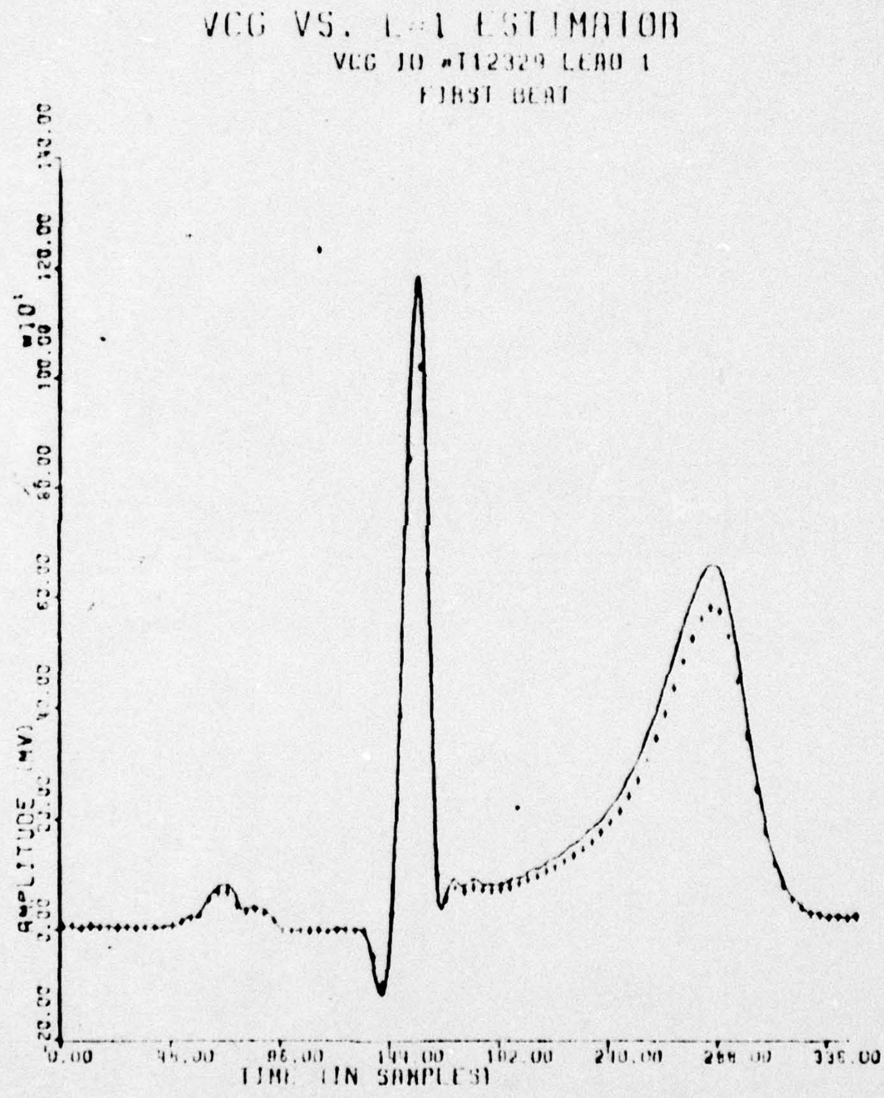


Figure 3(a): VCG vs. L=1 Estimator: Effects of Varying Data Rate (500 samples/second)

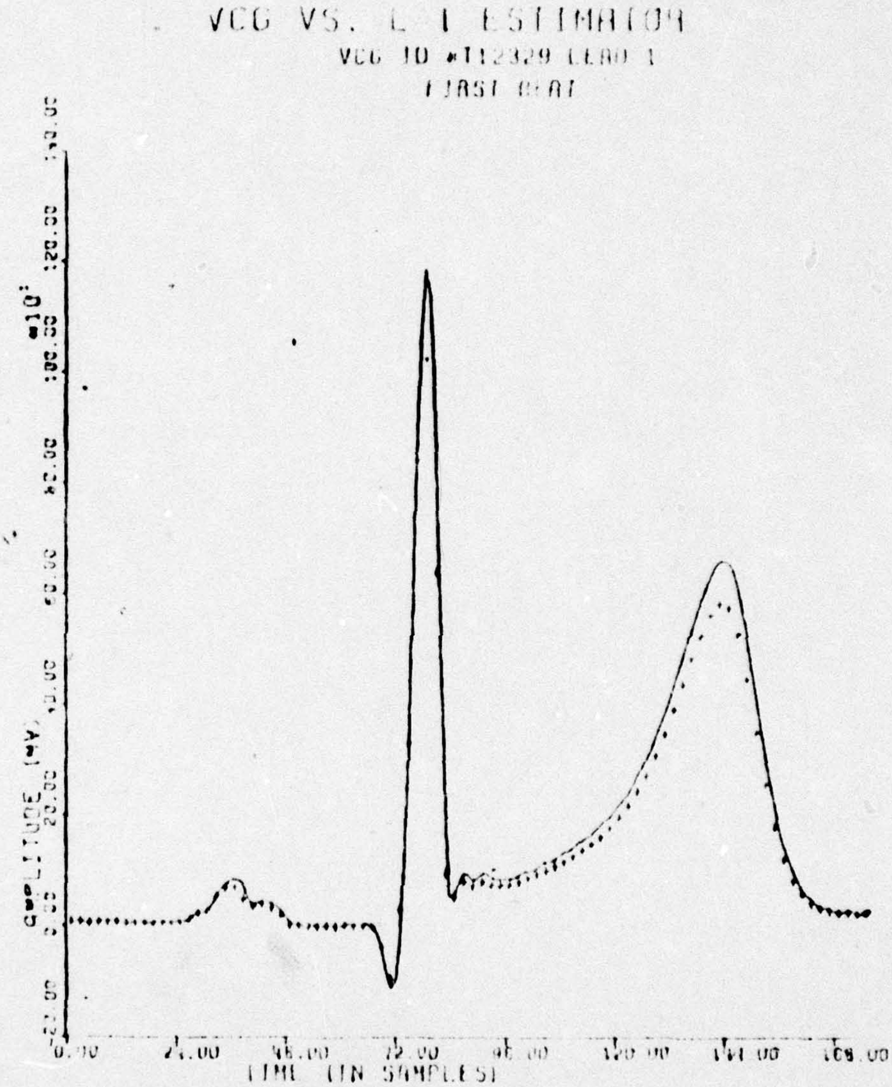


Figure 3(b): VCG vs. L=1 Estimator: Effects of Varying Data Rate (250 samples/second)

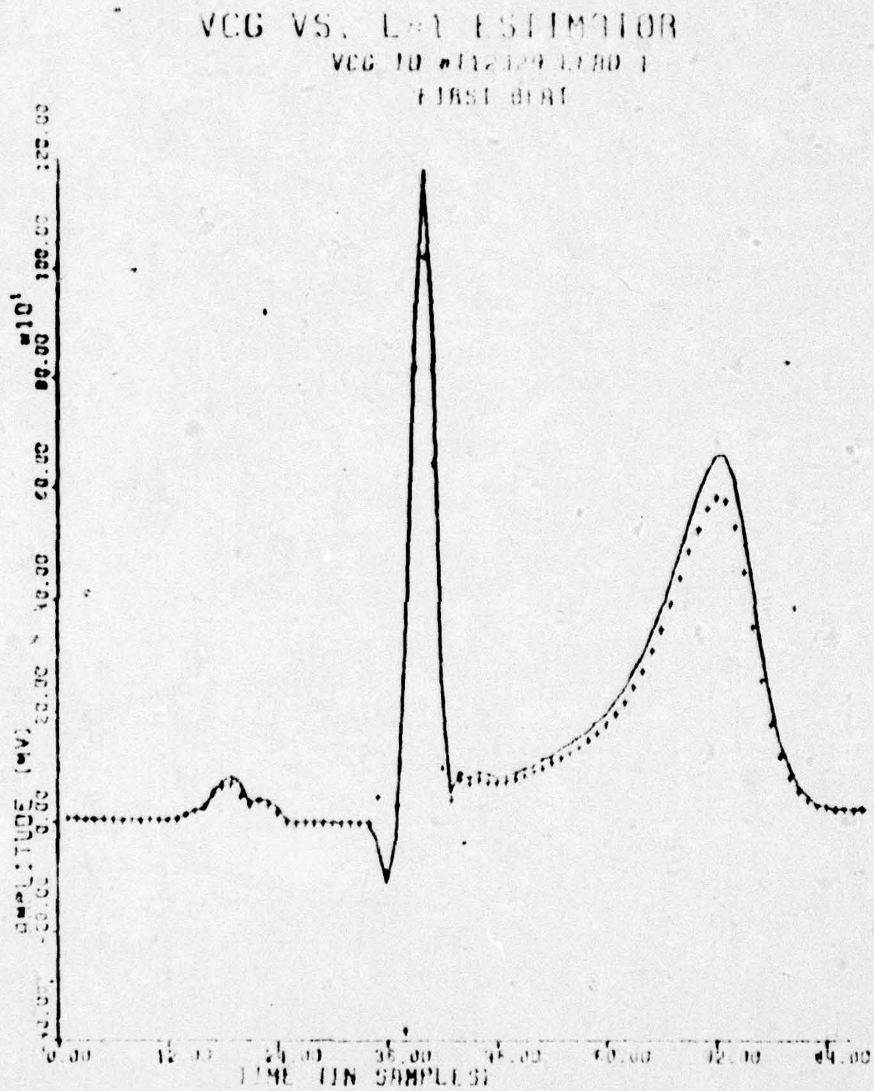


Figure 3(c): VCG vs. L=1 Estimator: Effects of Varying Data Rate (125 samples/second)

Using the L=1 estimator, the maximum possible data compression can be no more than N:1; e.g., 9:1 assuming the original electrocardiagram is sampled to 9 bit accuracy. Practically, however, the data suggests a maximum of 4:1 is realistically achievable. To compress the data further would require deleting samples. This is both possible, and entirely feasible. As part of a class in digital signal processing the author assigned a computer problem to devise an interpolation algorithm to generate the 500 samples/second data from 125 samples/second data; e.g., every 4^{th} point was used to regenerate the deleted samples. Using a standard procedure [5], the students used 4-point and 6-point least squares quadratics and 6- and 8-point quadratic functions to regenerate the data. The 4-point quadratic result is given in Figure 4, where it is often difficult to determine that there are in fact two plots. (One must comment here that this is another argument for a lower sampling rate.) A more elegant method of interpolation is also possible: the adaptive predictor algorithm can be easily extended to full beat interpolation as a single large scale matrix operation. The details of the extension are in Appendix C.

RECOMMENDATIONS

It is now clear that the prediction algorithm works, and works well. Further research in this area should progress on two fronts. First, using a L=1 predictor, sampling rate of 250 samples/second and quantization of 8 bits/sample concentrate on the design of the compressor itself. As described in the original report [2], the differences can be transmitted in either of two ways: using a Huffman encoder, or by a more recent approach called tree encoding [6]. Ordinary predictive DPCM makes a LMS prediction of the next sample based on some statistical knowledge of the source and transmits the difference--irregardless of its size. A tree encoder uses the same predictor with an addition; it can look at the difference and, if necessary, modify its prediction. Since each set of prior samples can have multiple predictions extending from it, the prediction sequence has the branchlike structure of a tree and hence the name tree encoding. Huffman coding offers the possibility of exact reproduction of the

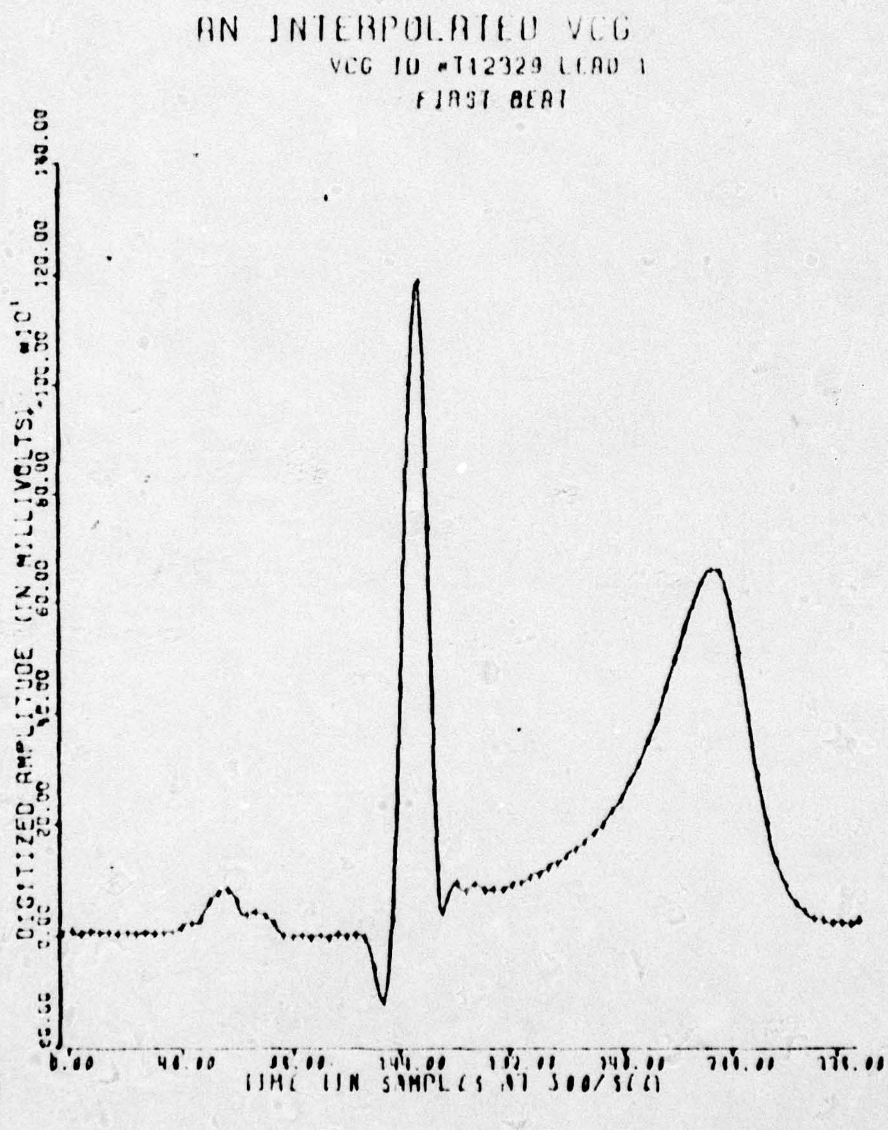


Figure 4: A Simple Interpolator to Recover Deleted Samples

transmitted sequence, but at a cost (in bits transmitted per sample) dependent on the probability structure of the differences. Tree encoding, on the other hand, offers the ultimate in compression (one bit transmitted per sample), but at the cost of only being able to reproduce the sequence to within a distortion measure. Which of the two methods is more appropriate is an open question.

Second, it is now appropriate to consider deleting samples as a means of further reducing the total number of bits required for transmission and storage. Judging from the quality of the interpolated waveform of Figure 4 it is apparent that there is room for significant gain. One possibility worth examining is the least-mean-square predictor in its non-adaptive full beat form. However, the success of the simple 4-point quadratic interpolator certainly suggests that it and other relatively simple algorithms should not be neglected.

REFERENCES

[1] H.V. Pipberger, et.al., "Recommendations for standardization of instruments in electrocardiagraphy and vectorcardiagraphy", IEEE Trans. Biomed. Engr., BME-14 (Jan 67) 60-68.

[2] M. Hankamer, "Adaptive/predictive data compression for electrocardiagrams". Participant's Final Report to the 1978 USAF/ASEE Summer Faculty Research Program under Contract No. F44620-75-C-0031 (Aug. 18, 1978).

[3] N. Abramson, Information theory and coding (McGraw-Hill) 1963.

[4] D.A. Huffman, "A method for the construction of minimum redundancy codes", Proc. IRE, Vol. 40, No. 10 (Sep 1952) 1098-1101.

[5] R.W. Hamming, Digital Filters (Prentice-Hall) 1977.

[6] J.B. Anderson and J.B. Bodie, "Tree Encoding of Speech", IEEE Trans. Inform. Thy. IT-21 No. 4 (Jul 75) 379-387.

Appendix A

THE ENTROPY OF A DIGITIZED ELECTROCARDIAGRAM

Michael Hankamer

F. Q. Khatib

Department of Electrical Engineering
Texas A&I University
Kingsville, Texas 78363

Introduction

Digitization of electrocardiagrams (ECG's) has become increasingly popular, for a variety of reasons. Digital transmission has a much greater noise immunity for a fixed signal-to-noise ratio. The decreasing cost of microprocessors and other digital logic has provided the ability to do significant signal processing and control cheaply; thus the ECG can be economically sampled, digitized, and pre-processed into an efficient transmission format. Mass storage is becoming economical: a received ECG may be electronically stored in lieu of being restored to analog form. Finally, digital processing of ECG's is an accomplished fact: there are now practical algorithms for routine diagnostic use.

American Heart Association standards [1] for digitizing electrocardiagrams call for an effective bit rate of 4500 bps(1) per lead of data. A "dial-up" digital telephone modem typically operates at 2400 bps, so it follows that real-time data transmission is not feasible without some form of data compression(2).

Algorithms fall into two general catagories: time and frequency compression. Both have been well-covered in the literature; for example, representative time compression algorithms can be found in Dower and Stewart [2], Cox, et.al. [3], and Weaver [4]. Frequency compression algorithms can be found in Young and Huggens [5], Ahmed, et.al. [6], and Womble, et.al. [7]. In both catagories, maximum compression ratios of about 10:1 have been reported.

(1) 500 samples per second at 9 bit quantization per sample.

(2) Compressed data also requires much less storage.

Recently Shannon's noiseless coding theorem [8] and the Huffman coding procedure [9] have been discovered by those interested in ECG data compression. Roughly, the noiseless coding theorem declares that for any source there exists a code whose average message length is lower-bounded by a quantity called the entropy of that source. The Huffman procedure is an explicit construction method of generating a code most nearly meeting the lower bound. For example, suppose a digitized ECG has an entropy of 3.8 bits/sample. Then there exists a code for transmitting that ECG having average word length lower-bounded by 3.8 bits/sample. Assuming 500 a sample/second rate and a Huffman code meeting the lower bound, the average transmission rate for the coded ECG is 1900 bps, a compression ratio of 2.37:1 from the standard rate of 4500 bps.

The notion of entropy offers more than just a lower bound for directly encoding the digitized ECG messages. Entropy can be defined for sources with memory, in which case the change in entropy with memory length may be useful in determining the optimum length of prediction algorithms used for data compression. Entropy may change with quantization (the number of possible messages), from which it may be possible to define an optimal quantization level. Entropy may vary with sample rate, in which case an optimum sample rate may be found. These possibilities are examined in more detail in the following sections.

<u>Entropy of a Markov Source</u>

Let S be a discrete-time source with each message quantized to one of 2^N levels (the source contains 2^N messages). The source S emits a message sequence with each message s_i drawn independently from the set of all messages with probability p_i. Then the entropy $H_0(S)$ is defined

$$H_0(S) \triangleq - \sum_{i=1}^{2^N} p_i \log (p_i) \tag{1}$$

If the logarithm is base-2, the entropy is expressed in bits/message(3).

(3) In this paper the logarithms will always be expressed base 2.

The source S is a zero-memory Markov source if it can be completely described by the source messages s_i and their probabilities p_i; i.e., the occurrence of a message is independent of occurrence of a prior message.

The zero memory source is quite restrictive for some applications. A more general model for S is one in which the occurrence of a symbol s_i depends on a finite number (M) of preceding messages. Such a source is called a Markov source of order M and is specified by giving the source messages S and their conditional probabilities $p(s_i|s_{ji},s_{j2},\ldots,s_{jm})$ for $i,j = 1,2,\ldots, 2^N$.

The ordered sequence of the M prior samples is known as the state of the source. The M^{th} order Markov source has 2^{NM} states; each state has state entropy defined by

$$H(S|s_{j1},s_{j2},\ldots,s_{jM}) \triangleq - \sum_{i=1}^{2^N} p(s_i|s_{j1},s_{j2},\ldots,s_{jM}) \log(p(s_i|s_{j1},s_{j2},\ldots,s_{jM})) \qquad (2)$$

The average of (2) over all the possible state is the entropy of the M^{th} order Markov source.

$$H_M(S) \triangleq - \sum_{j1=1}^{2^N} \sum_{j2=1}^{2^N} \cdots \sum_{jM=1}^{2^N} \sum_{i=1}^{2^N} p(s_i,s_{j1},s_{j2},\ldots s_{jM}) \log(p(s_i|s_{j1},s_{j2},\ldots,s_{jM})) \qquad (3)$$

Entropy and Prediction Algorithms

Most time compression schemes use a prediction algorithm to predict the next data sample from some prior knowledge -- only the difference from the predicted value is transmitted. Compression occurs if the predictor is good and the differences are small compared to the samples. The simplest example is differential PCM: the difference between adjacent samples, rather than the samples themselves, is transmitted. Differential PCM treats the source as Markov of order 1; the next sample is assumed to not differ much from the current sample. An order 2 approximation might be that of linear extrapolation; the next sample is estimated to be the linear extrapolate of the two prior samples.

Suppose entropy is a non-increasing function of source order; that is,

$H_M(S) \leq H_K(S)$ for $M \geq K$. Since the order of a Markov source corresponds roughly[1] to predicto length, the change is source entropy with source order should give some indication of the expected effectiveness of a prediction algorithm.

A FORTRAN program has been run on an IBM 360 computer to calculate the entropy (up to fourth order) of any given data sequence. Digitized vectorcardiagram data has been supplied by the School of Aerospace Medicine, Brooks AFB, Texas for use in computing entropies. The results of one such test are given in Figure 1. The data from which the entropies were computed was taken at 500 samples per second with a message quantization of 11 bits per sample. The reduction in entropy with increasing source order is dramatic. From order zero to order one, a reduction of about 7:1 is achieved. Further reduction of typically 3-4 to 1 is possible for each unit increase

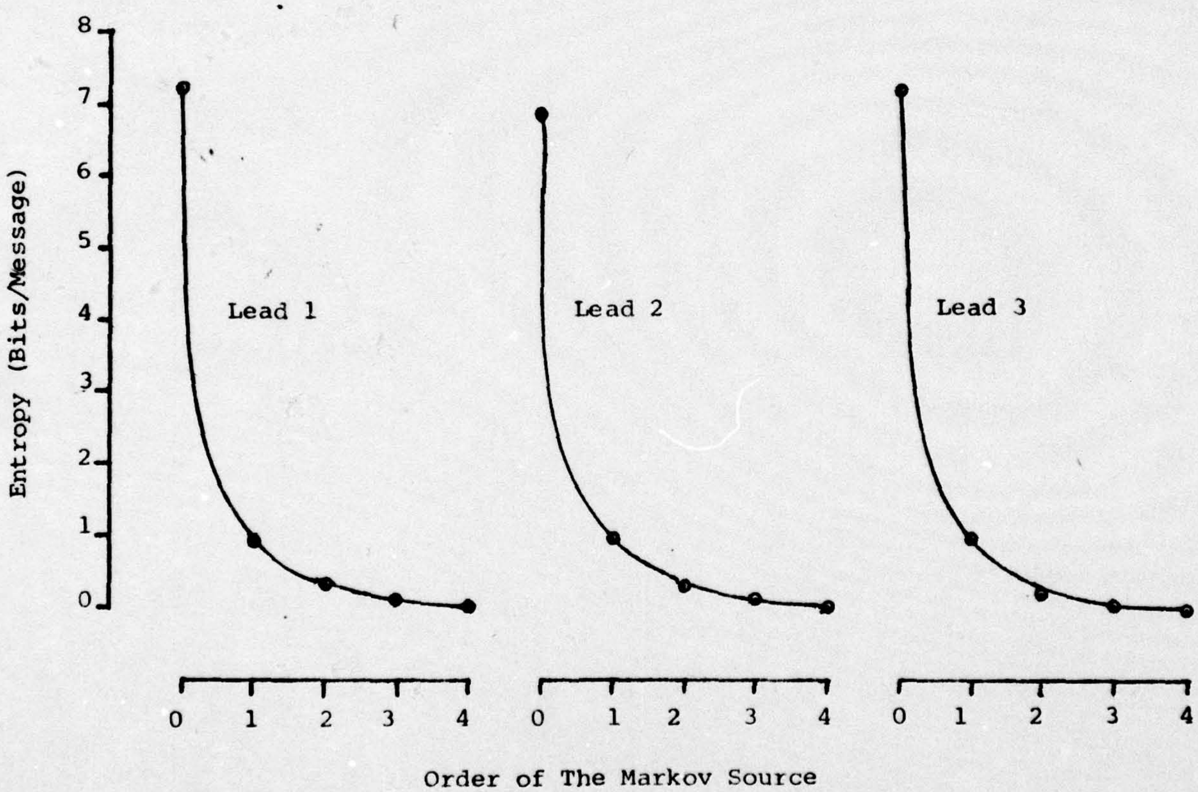


Figure 1: Entropy of VCG Modeled as a Markov Source

---

(1) Roughly, since the entropy is defined for the message probabilities; any predictor uses the messages themselves.

of the source order.

From Figure 1 it is clear that the digitized electrocardiagram is highly correlated sample-to-sample; thus significant data compression is probably achievable with a prediction algorithm of limited complexity. Some success has been reported. Weaver [4] has achieved a 4:1 compression using a clever second-order interpolator. Hankamer [10] proposed an adaptive variable-length estimation algorithm in 1978; research now in progress with short length versions indicates 3-4 to one compression ratios are easily achievable.

From the entropy versus source order data, it appears that short algorithms, such as those of Weaver and Hankamer, offer the most potential for significant compression. The ratio $H_{i-1}(S)/H_i(S)$, which we presume to measure the compression capability of a prediction algorithm, is greatest for $i=1$ for each lead. For $i>1$, the savings are smaller; moreover, the reduction of the entropy below 1 bit/message implies that multiple messages must be combined for transmission. This is possible, but at the added cost of increased complexity.

The Relationship of Entropy to Data Quality

Data quality is clearly affected by both sample rate and sample quantization. It would be convenient if the source entropy were also directly affected by rate and quantization. Unfortunately it is not to be, for entropy is not defined in terms of the number of messages or message precision, but in terms of the message probabilities instead. The message probabilities are only indirectly affected by changes in sample rate and quantization.

The data from which the entropies in Figure 1 were taken has been "massaged" to reflect varying sample rates and quantization levels. The results are shown in Figure 2 and 3 for lead 1 of the test vectorcardiagram. Consider first the entropy as a function of sample rate -- Figure 2. The increase in entropy clearly slows as the sample rate increases; but at what sample rate the law of diminishing return takes effect is not clear. Similarly, consider the quantization curve of Figure 3. Some

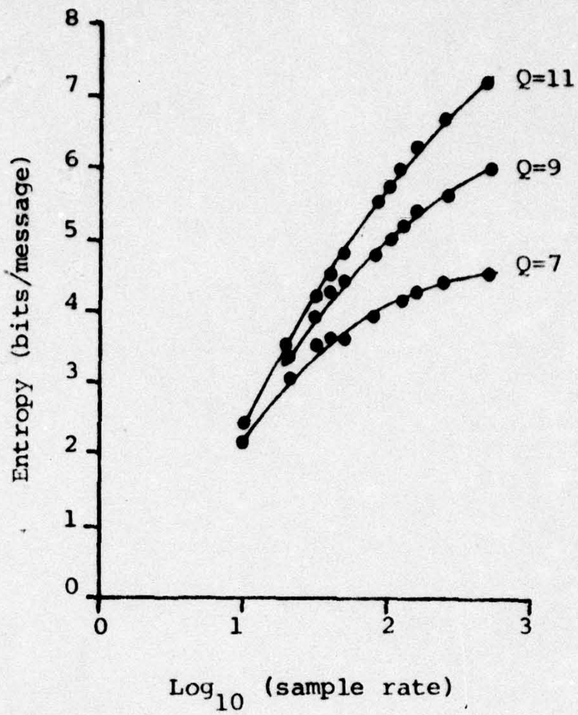


Figure 2: Zero Order Entropy as a Function of Sample Rate

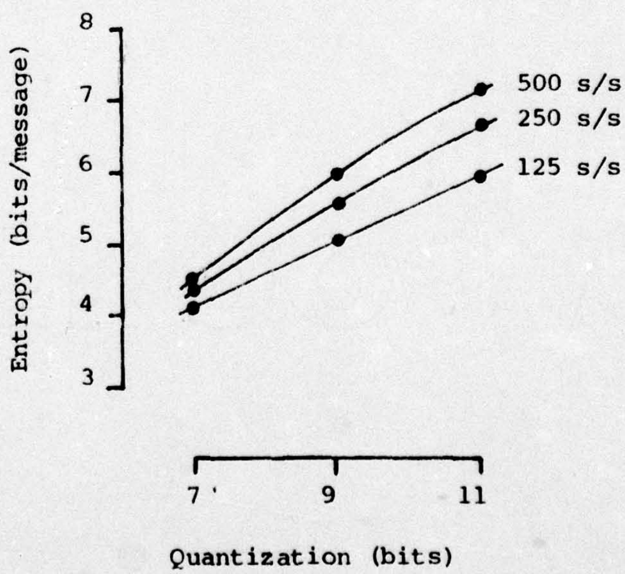


Figure 3: Zero Order Entropy as a Function of Sample Quantization

flattening is apparent, but any significance is not obvious.

There is no simple interpretation of the flattening of the entropy curves of Figures 2 and 3. However, we can give some insight into the effects of sample rate and quantization on entropy. Consider first the change of entropy with sample rate. Suppose S has M independent, equally probable messages. Then

$$H_0(S) = -\sum_{i=1}^{M} \frac{1}{M} \log \left(\frac{1}{M}\right) = \log M \quad (4)$$

Suppose that the sample rate is increased K times (S has now KM messages) and that the new messages are independent of the original set and equally probable[(4)]. Then

$$H_0(S) = -\sum_{i=1}^{KM} \frac{1}{KM} \log \left(\frac{1}{KM}\right) = \log (KM) = \log M + \log K \quad (5)$$

The maximum increase in entropy corresponding to a K-fold increase in sample rate is log K bits per sample. Conversely, suppose that in increasing the sample rate K times, each of the K new samples is identical to the old sample immediately preceding it. Then the relative probabilities of the messages remain unchanged, and hence the entropy does not change. We see, then, that the entropy change due to sampling rate, $\Delta H_0(K)$, is bounded above and below by

$$0 \leq \Delta H_0(K) \leq \log K \quad (6)$$

The bounds and entropies for each of the 3 leads of the test vectorcardiagram are given in Figure 4 for 11 bit quantization.

The data points clearly split the middle between the bounds suggesting that, on the average, new states are created by increased sampling a little over half the time: about what one would expect "at random" (e.g., if the increased sampling rate were measuring an additive noise fluctuation). Conversely, it is also true that for a resting electrocardiagram, the electrical activity is essentially dormant about half the time and would probably not be changing.

---

(4) Note the assumption that the number of possible messages is presumed to be much larger than the actual number of messages in S.

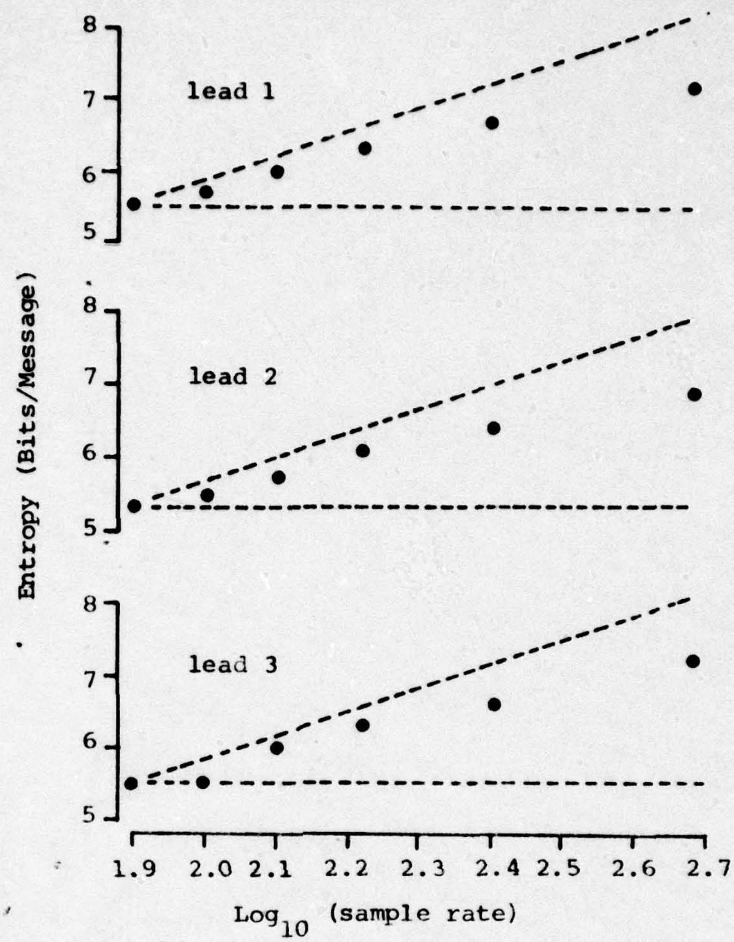


Figure 4: Upper and Lower Bounds on the Zero Order Entropy

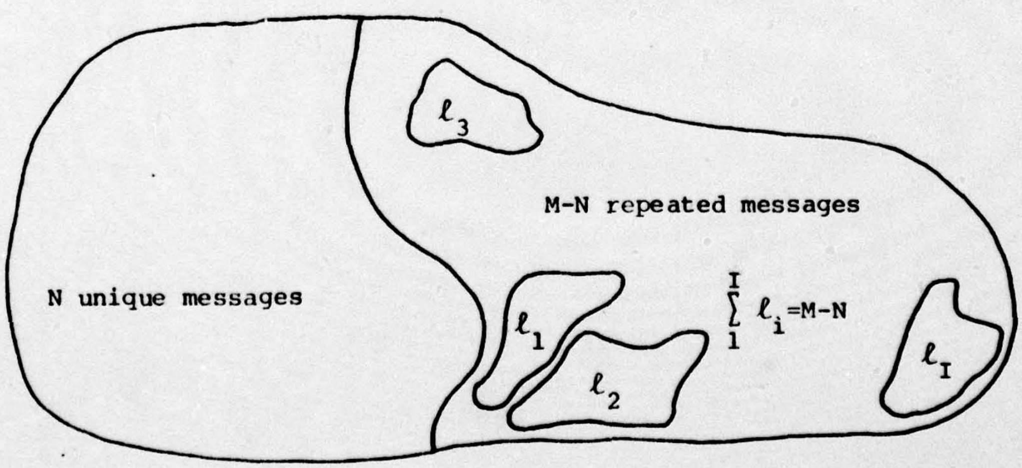


Figure 5: Message Structure

On the change in entropy with quantization, we again consider a very simple model. Let S be a source with zero-order entropy $H_0(S)$. S has a total of M messages, N (<M) of which are unique. The remaining (M-N) are repeated. Figure 5 indicates the source structure. For this source

$$H_0(S) = -\frac{N}{M}\log\left(\frac{1}{M}\right) - \frac{\ell_1}{M}\log\left(\frac{\ell_1}{M}\right) - \dots - \frac{\ell_I}{M}\log\left(\frac{\ell_I}{M}\right) \tag{7}$$

Now suppose that each message belonging to S is requantized using one additional bit. The N unique messages are unaffected, but on the average the N-M repeated messages subdivide into two messages. Suppose they subdivide equally. Then

$$H_0^1(S) = -\frac{N}{M}\log\left(\frac{1}{M}\right) - \frac{\ell_1}{2M}\log\left(\frac{\ell_1}{2M}\right) - \frac{\ell_1}{2M}\log\left(\frac{\ell_1}{2M}\right) - \dots - \frac{\ell_I}{2M}\log\frac{\ell_I}{2M} - \frac{\ell_I}{2M}\log\left(\frac{\ell_I}{2M}\right) \tag{8}$$

$$= -\frac{N}{M}\log\left(\frac{1}{M}\right) - \frac{\ell_1}{M}\log\left(\frac{\ell_1}{2M}\right) - \dots \frac{\ell_I}{M}\log\left(\frac{\ell_I}{2M}\right) \tag{9}$$

$$H_0^1(S) = -\frac{N}{M}\log\left(\frac{1}{M}\right) - \frac{\ell_1}{M}\log\left(\frac{\ell_1}{M}\right) - \dots - \frac{\ell_I}{M}\log\left(\frac{\ell_I}{M}\right) \tag{10}$$

$$+ \frac{\ell_1}{M}\log 2 + \dots + \frac{\ell_I}{M}\log 2$$

$$= H_0(S) + \frac{1}{M}(\ell_1 + \dots + \ell_I) = H_0(S) + \frac{M-N}{M} \tag{11}$$

$$H_0^1(S) = H_0(S) + 1 - \frac{N}{M} \tag{12}$$

Equation (12) provides an estimate of the maximum increase in entropy for a unit increase in source quantization. The results are given in Table 1. $H_0^1(S)$ was computed from (12) after examining the VCG data to find N for each quantization level. ΔH appears to maximize at a quantization of 8 bits/sample, which suggests that for Q samll, the increasing quantization is effective, and for Q large the increasing quantization may be ineffective -- actually measuring noise effects rather than any changes in the electrocardiagram itself.

A particularly striking aspect of this study is the interdependence of the sampling rate and the sample quantization using the first order entropy $H_1(S)$ as a tool. Each sampling rate appears to have an optimum quantization level. (See Figure 6.)

While the results presented in Figure 6 are for lead 1, they are equally sharp for the other two leads.

TABLE 1
Actual vs. Computed Entropies

Q	$H_0(S)$	$H_0^1(S)$	$\Delta H_0(S)$
3	0.81	-	-
4	1.77	1.81	0.04
5	2.75	2.77	0.02
6	3.59	3.74	0.15
7	4.47	4.57	0.10
8	5.22	5.42	0.20
9	5.97	6.10	0.13
10	6.61	6.74	0.13
11	7.16	7.28	0.12

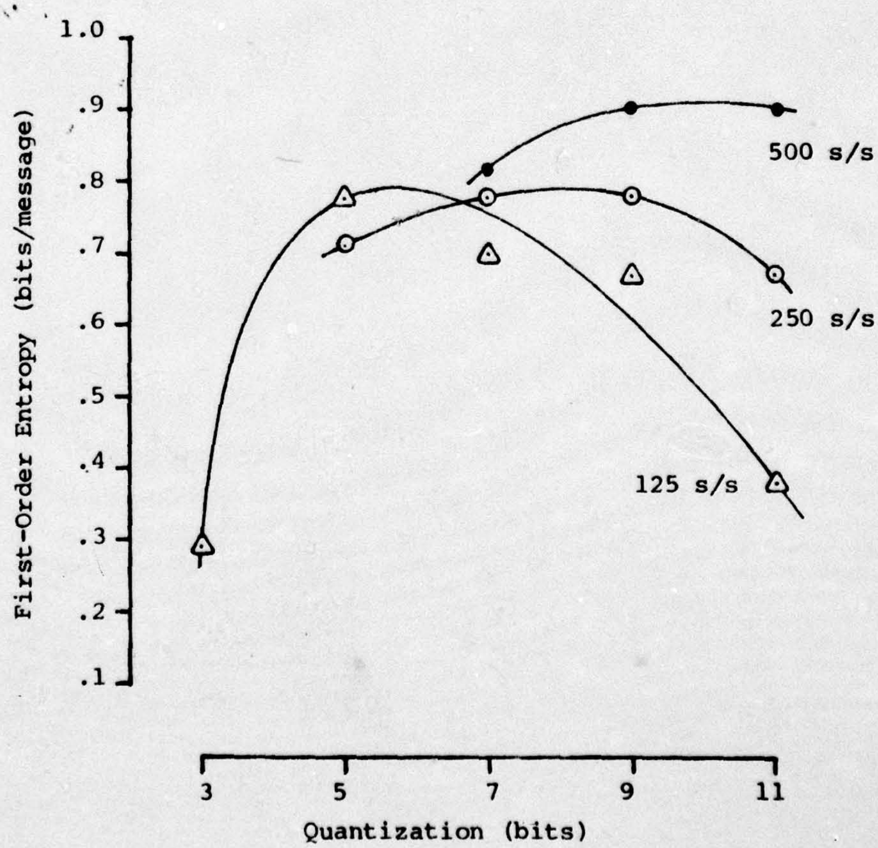


Figure 6: Quantization and Rate Dependence of the First-Order Entropy

The first order entropy also has another interesting property: it is a measure of the statistical "regularity" of the electrocardiagram. Suppose S is a Markov source transmitting a message sequence of N words with each word quantized to k bits. There are 2^k possible messages. Consider the first order entropy of S:

$$H_1(S) \triangleq - \sum_{s \in S} \sum_{m \in S} p(s,m) \log p(s|m) \tag{13}$$

The double summation is over all of the states belonging to S ($s \in S$) and the number of messages belonging to S ($m \in S$). For a first order source, the two are identical; hence

$$H_1(S) = - \sum_{s_i \in S} \sum_{s_j \in S} p(s_i,s_j) \log p\,(s_i|s_j) \tag{14}$$

The conditional probability $p(s_i|s_j) = p(s_i,s_j)/p(s_j)$ from which

$$H_1(S) = - \sum_{s_i} \sum_{s_j} p(s_i,s_j)\,[\log p(s_i,s_j) - \log p(s_j)] \tag{15}$$

$$= - \sum_{s_i} \sum_{s_j} p(s_i,s_j) \log p(s_i,s_j) + \sum_{s_i} \sum_{s_j} p(s_i,s_j) \log p(s_j) \tag{16}$$

$$= - \sum_{s_i} \sum_{s_j} p(s_i,s_j) \log p(s_i,s_j) + \sum_{s_j} p(s_j) \log p(s_j) \tag{17}$$

$$= H_0(S,S) - H_0(S) \tag{18}$$

The term $H_0(S,S)$ is the joint entropy of two beats. It follows that the first order entropy measures the average uncertainty between different heartbeats from the same source.

Conclusions and Caveats

The dramatic decrease of source entropy with increasing memory length clearly shows the potential of relatively simple predictors in data compression algorithms. For all three leads of the vectorcardiagram studied, one bit per message should be sufficient to completely define the next sample given only the single prior sample. Of course, the entropy function states only that some relation exists; it does not give the actual relationship. Prediction algorithms are usually linear and time-invariant; the actual relationship symbolized by H(S) need be neither. Thus the entropy function practically gives a lower bound on the achievable.

The relationship of entropy to data quality is not yet clear. Certainly there is some relation, as perhaps symbolized by the law of diminishing returns, and seen in Figures 2 and 3. In both cases (sampling rate and quantization), it is clear that the entropy improves as rate and quantization increase, but as judged from the simplistic models presented here, it is not clear whether the change in entropy is a true quality increase or simply a reflection of the increased randomness generated by having more possibilities for messages. It follows, then, that choice of sample rate and quantization are best left to the user, with one limitation. The first-order entropy emphasizes the interdependency of sample rate and quantization. They must be chosen together to best optimize the overall performance.

Finally, the first-order entropy clearly expresses a limits on the practicality of compression algorithms. In an earlier section the first order entropy was shown to be a measure of the statistical "regularity" of the electrocardiagram: one might think of $H_1(S)$ as what is "left over" after the information common to all ECG's is removed. Thus a fundamental result of this research is that 1 bit/sample probably represents the limit in "easy" time data compression.

It must be pointed out that there is one caveat to be applied to the data in this paper: it was derived from one beat of one patient's electrocardiagram. Certainly these results are not sufficient to uncritically apply to an entire population. Yet the data has been tested against other beats from the same patient, and against other ECG's. The numbers do change, but the general characteristics remain the same.

References

[1] H.V. Pipberger, et.al., "Recommendations for Standardization of Instruments in Electrocardiagraphy and Vectorcardiagraphy", IEEE Trans. Bio-Med. Engr., BME-14 (Jan 67) 60-68.

[2] G.E. Dower and D. Stewart, "An ECG Compression Code", J. Electrocardial. VI-2 (1973) 175-176.

[3] J.R. Cox, et.al., "AZTEC, a Preprocessing Program for Real Time ECG Rhythm Analysis", IEEE Trans. Bio-Med. Engr., BME-15 (apr 68) 128-129.

[4] C.S. Weaver, "Digital ECG Data Compression", in Digital Encoding of Electrocardiagrams, H.K. Wolf, Ed., Springer-Verlag (Berlin, New York), 1979.

[5] T.Y. Young and W.H. Huggins, "On the Representation of Electrocardiagrams", IEEE Trans. Bio-Med. Engr., BME-10 (Jul 63) 86-95.

[6] N. Ahmed, et.al., "Electrocardiagraphic Data Compression via Orthogonal Transforms", IEEE Trans. Bio-Med. Engr., BME-22 (Nov 75) 484-492.

[7] M.E. Womble, et.al., "Data Compression for Storing/Transmitting ECG's/VCG's", Proc. IEEE, Vol. 65, No. 5 (1977) 702-706.

[8] N. Abramson, Information Theory and Coding, (McGraw-Hill), 1963.

[9] D.A. Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proc. IRE, Vol. 40, No. 10 (Sep 1952) 1098-1101.

[10] M. Hankamer, "Adaptive/Predictive Data Compression for Electrocardiagrams", Participants Final Report to the 1978 USAF/ASEE Summer Faculty Research Program under contract No. F44620-75-C-0031 (Aug 18, 1978).

Appendix B

A LEAST MEAN-SQUARE PREDICTION ALGORITHM FOR DIGITAL ELECTROCARDIOGRAPHY

Michael Hankamer
Department of Electrical Engineering
Texas A&I University
Kingsville, Texas 78363

INTRODUCTION

Digital transmission of electrocardiagrams (ECG's) has become increasingly popular. Digital transmission has much greater noise immunity for a fixed signal-to-noise ratio. The decreasing cost of microprocessors and other digital logic has provided the ability to do significant signal processing and control cheaply. Thus the ECG can be sampled, digitized, and pre-processed into an efficient transmission format economically. Finally, the cost of mass storage is becoming economical: the received ECG may be electronically stored in discrete form in lieu of being restored to analog form for later analysis, processing, etc.

Preprocessing into an efficient transmission format is a current problem in digital electrocardiagraphy. American Heart Association standards [1], call for 500 samples/second per lead at a precision of 9 bits/sample: for a 3-lead vectorcardiagram (VCG) a data rate of 13.5 Kbps is called for. Since an unconditioned (dial-up) voice-grade telephone modem has a typical data rate capability of 2400 bps, it follows that for real-time transmission, the VCG/ECG must be preprocessed: compressed into a fewer number of bits/second.

Compression algorithms fall into two categories: time and frequency. Both are well covered in the literature. Representative time compression algorithms can be found in Dower and Stewart [2], Cox, et.al., [3], and Weaver [4]; representative frequency compression algorithms are described in Young and Huggins [5], Ahmed, et.al. [6], and Womble, et.al. [7]. In both categories, compression ratios of about 10:1 have been reported. The frequency representation has received somewhat more emphasis in light of its traditional attachment to pattern recognition while time representations have received somewhat less emphasis in light of their attachment to the transmission problem--appropriate algorithms have not been economical.

Womble, et.al. used the optimum least mean-square (LMS) frequency representation in a compression algorithm. This paper considers one part of the complementary solution: a least mean-square time representation of the digitized electrocardiagram. Womble used a large ensemble of patient electrocardiagrams to find the eigenvectors of the Karhuuen-Loeve expansion. Then, using the eigenvectors, an individual electrocardiagram was decomposed, and the 20 or so largest eigenvalues transmitted to the receiver, whereupon the least-mean-square estimate of the transmitted electrocardiagram was assembled. The complementary solution to be discussed here uses the same statistical ensemble to produce an estimate of the digitized electrocardiagram; the estimate is then subtracted out and only the differences transmitted. At the receiver, the estimate is regenerated and the differences added back in to produce the original electrocardiagram.

THE PREDICTION ALGORITHM

The sampled electrocardiagram is modeled as a periodically stationary random sequence; that is, one for which

$$E\{s(n)\} = E\{s(n+kN)\} \tag{1}$$

$$R_{ss}(n,m) \triangleq E\{s(n)s(m)\} = E\{s(n+kN)s(m+\ell N)\} \triangleq R_{ss}(n+kN,m+\ell N) \tag{2}$$

for some positive integer N and any integers k,ℓ,m, and n. It must be noted that the actual ECG data sequence is not periodically stationary; however, with proper "massaging" (baseline removal, gain control, blocking and centering about a fiducial point, etc.) it can be made so. Womble, et.al., used these techniques in preparing ECG data for frequency compression; in this paper such steps are assumed.

Given the periodically stationery random sequence $\underline{s}$ we wish to predict the nth member $s(n)$ given the L preceeding members $s(n-1), s(n-2), \ldots, s(n-L)$. We restrict ourselves to linear, minimum mean-square-error (LMMSE) predictors of the form

$$\hat{s}(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots + a_L s(n-L) = A^T \underline{s} \tag{3}$$

where $\hat{s}(n)$ is the prediction, and $A^T = \{a_1, a_2, \ldots, a_L\}$ and $\underline{s}^T = \{s(n-1), s(n-2), \ldots, s(n-L)\}$ are Lx1 row vectors. The mean-square prediction error, $e^2(n) = [\hat{s}(n) - s(n)]^2$, can be written in matrix form as

$$e^2(n) = [A^T \underline{s} - s(n)][A^T \underline{s} - s(n)]^T$$

which becomes

$$e^2(n) = A^T \underline{ss}^T A - 2A^T \underline{s}s(n) + s^2(n) \tag{4}$$

Taking the expected value of the mean-squared error (MSE) gives

$$\varepsilon^2(n) \triangleq E\{e^2(n)\} = A^T E\{\underline{ss}^T\} A - 2A^T E\{\underline{s}s(n)\} + E\{s^2(n)\} \tag{5}$$

The matrix $\underline{ss}^T$ is LxL; its ij*th* element is $s(n-i)s(n-j)$. Taking the expectation over all elements yields the LxL symmetric correlation matrix Λ_s. The column vector $\underline{s}s(n)$ has as the *ith* element $s(n-i)s(n)$; taking the expectation over all elements yields the correlation vector Γ. Thus

$$\varepsilon^2(n) = A^T \Lambda_s A - 2A^T \Gamma + E\{s^2(n)\} \tag{6}$$

The elements of the column vector A have not yet been chosen--we will use them to minimize $\varepsilon^2(n)$. To effect the minimization, set the derivative of $\varepsilon^2(n)$ with respect to A equal to zero.

$$\frac{\partial \varepsilon^2(n)}{\partial A} = \Lambda_s A + (A^T \Lambda_s)^T - 2\Gamma = \underline{0} \tag{7}$$

Noting that $\Lambda_s = \Lambda_s^T$ by symmetry, we can solve for A to get

$$A_{opt} = \Lambda_s^{-1} \Gamma \tag{8}$$

from which it follows that the minimum mean-squared error is given by

$$\varepsilon^2(n) = E\{s^2(n)\} - \Gamma^T \Lambda_s^{-1} \Gamma \qquad (9)$$

for each n. Note that the minimum mean-squared error $\varepsilon^2(n)$ depends on n: the predictor is adaptive. This results from the random sequence $\underline{s}$ being at most periodically stationary. Wide-sense stationarity would be required to make the minimum mean-square error independent of n.

Suppose M=N: one full period is used in forming the estimate of the next sample. Since the random sequence is periodically stationary, it can be shown that the matrix Λ_s is circulatory; $\Lambda_s(n+1)$ differs from $\Lambda_s(n)$ by just a row and column shift. For this case the column vector Γ is identically the last column of Λ_s and it follows that the optimal predictor A_{opt} $(=\Lambda_s^{-1}\Gamma)$ is exactly the vector $\{0,0,\ldots,1\}^T$. The optimum prediction is the sample value from one period earlier; the optimal LMMSE predictor of a heartbeat is the prior beat.

This answer is intuitive, and not particularly helpful, since by implication the first beat must be sent in full. In this paper we utilize a short predictor (L=1,2,3) compared to the electrocardiagram period (N=351). The predictor is adaptive from sample-to-sample (both Λ and Γ depend on n) and period-to-period (the correlations comprising Λ and Γ are continuously updated as new samples are received).

SIMULATION RESULTS

The prediction algorithm, as defined by equations (3) and (8), was simulated on an IBM 360/65 computer using predictor lengths L of 1, 2, and 3. The algorithm was fully adaptive, in that the correlation functions comprising the matrices Λ and Γ were updated each sample. Digitized vectorcardiagram data was supplied by the School of Aerospace Medicine at Brooks AFB, Texas, for use in testing the algorithm.

Results were quite pleasing. Before considering the Figures and Tables in detail, we can summarize as follows. The prediction algorithm generally behaves quite well.

The predicted value is nearly always within 10% of the true value. The predictor seems to be relatively insensitive to parameter changes. It is adaptive. The original correlations on which the prediction is based are generated on the basis of an "average" heartbeat; as the original correlations are updated with "personal" information on the heartbeat being predicted, the prediction clearly improves.

Figure 1 ((a)-(e)) gives the predictor performance versus the original data for lead 1 of a sample vectorcardiagram. The data was taken at 500 samples/second and 11 bits/sample. The original data is shown as a solid line; the predicted data (every second point) is given by the (+) signs. The error at the peak of the R-wave on beat 1 is 16%; by beat 5 that error has decreased to less than 4%. The L=2 and L=3 estimators are shown for the same lead of the vectorcardiagram in Figures 2 and 3 respectively. Only the first and last beats of the 5 beat sequence are given. It is to be noted that increasing the length of the estimator does not appear to significantly improve the quality of the estimates, for in both cases the first beat peak error is about 17%, decreasing to about 5% on the fifth beat.

Table 1 is a quantization of the results shown in Figure 1, giving the maximum error, average error, standard deviation of the error, and entropy of the error for the five beats of the vectorcardiagram. The vectorcardiagram range is from about -150 to +1200, for a total range of 1350. Thus a maximum error of 150 represents about 11% of full scale. The entropy is more fully discussed in a companion paper [8], but roughly can be said to measure the minimum number of bits required to transmit a sample, on the average. Starting with 11 bits/sample, the predictor represents a compression gain of about 2:1 for the first beat, increasing to about 3:1 by the fifth beat. Although the tabular data for the L=2 and L=3 predictors are not given, they are typically the same.

The adaptive nature of the predictor is clearly desirable. Figure 4 shows the predicted versus actual vectorcardiagram that was used in Figure 1: the only difference being that the predictor was never updated as the new samples entered. The first beat ((a) in both Figures 1 and 4) shows that neither is on the mark -- if anything, the nonadaptive estimator might be a little closer. But by beat five ((e) in both Figures), the non-adaptive estimator is still as far away from the actual beat as it was in beat one. The adaptive predictor, on the other hand, is very close to the true value. Table 2 compares the predictor errors for beats one and 5; the superiority of the adaptive predictor is evident.

The adaptive predictor is also insensitive to parameter changes: as evidence, consider the vectorcardiagram of Figure 5. The VCG #T12329 was effectively reduced to 125 samples/second by using every fourth sample and 7 bit quantization by dividing each sample by $16(2^4)$. Nonetheless, as evidenced by the Figure, the predicted value is still close to the actual value, and converging as the number of beats increases. There is some evidence of a loss of performance during the Q wave and the S-T interval; but this loss is most likely due to the low sampling rate rather than any inadequacy of the algorithm.

CONCLUSIONS

This algorithm for data prediction should prove useful for any discrete signal sequence that can be modeled as a periodically stationary random sequence. The algorithm error seems to be relatively robust: independent of both quantization and sample rate -- at least within reasonable limits. Sampling rate and quantization are interdependent [8], and because of aliasing, it is doubtful if the predictor is capable of operating correctly below the Nyquist rate.

The L=1 predictor algorithm appears to be the most practical for implementation. The L=2 and L=3 predictors, although good, consistently had average error and standard deviations close to or slightly worse than those associated with the less complex L=1 predictor. Other research (e.g., [8]) also implies that unit length predictors offer the best "gain" per-unit complexity. Presuming a unit-length predictor, this research tends to indicate that a practical maximum compression of 3-4 to 1 is the ultimate achievable by this method. Further compression would require less than perfect reproduction, perhaps by not sending all the samples, or possibly by sending only approximations to the differences. The latter appears chancy, since the differences are used to reconstruct the succeeding samples. The former method offers some hope, since it is not difficult to extend the prediction algorithm to a full-period interpolation. That extension is in progress and will be reported at a later date.

REFERENCES

[1] H.V. Pipberger, et.al., "Recommendations for standardization of instruments in electrocardiagraphy and vectorcardiagraphy", IEEE Trans. Biomed. Engr. BME-14 (Jan 67) 60-68.

[2] Dower, G.E., and D. Stewart, "An ECG compression code", J. Electrocardial., VI, No. 2 (1973) 175-176.

[3] Cox, J.R., et.al., "AZTEC, a preprocessing program for real-time ECG rhythm analysis", IEEE Trans. Biomed. Engr., BME-15 (Apr 68) 128-129.

[4] Weaver, C.S., "Digital ECG data compression", in Digital Encoding of Electrocardiagrams. H.K. Wolf, Ed., Springer-Verlag (Berlin, New York), 1979.

[5] Young, T.Y., and W.H. Huggins, "On the representation of electrocardiagrams", IEEE Trans. Biomed. Engr., BME-10 (Jul 63) 86-95.

[6] Ahmed, N. et.al., "Electrocardiagraphic data compression via orthogonal transforms", IEEE Trans. Biomed, Engr., BME-22 (Nov 75) 484-492.

[7] Womble, M.E., et.al., "Data compression for storing and transmitting ECG/VCG's", Proc. IEEE, Vol. 65, No. 5 (1977) 702-706.

[8] Hankamer, M. and F.Q. Khatib, "The entropy of a digitized electrocardiagram", submitted to the IEEE Trans. Biomed. Engr. (Jan 10, 1980).

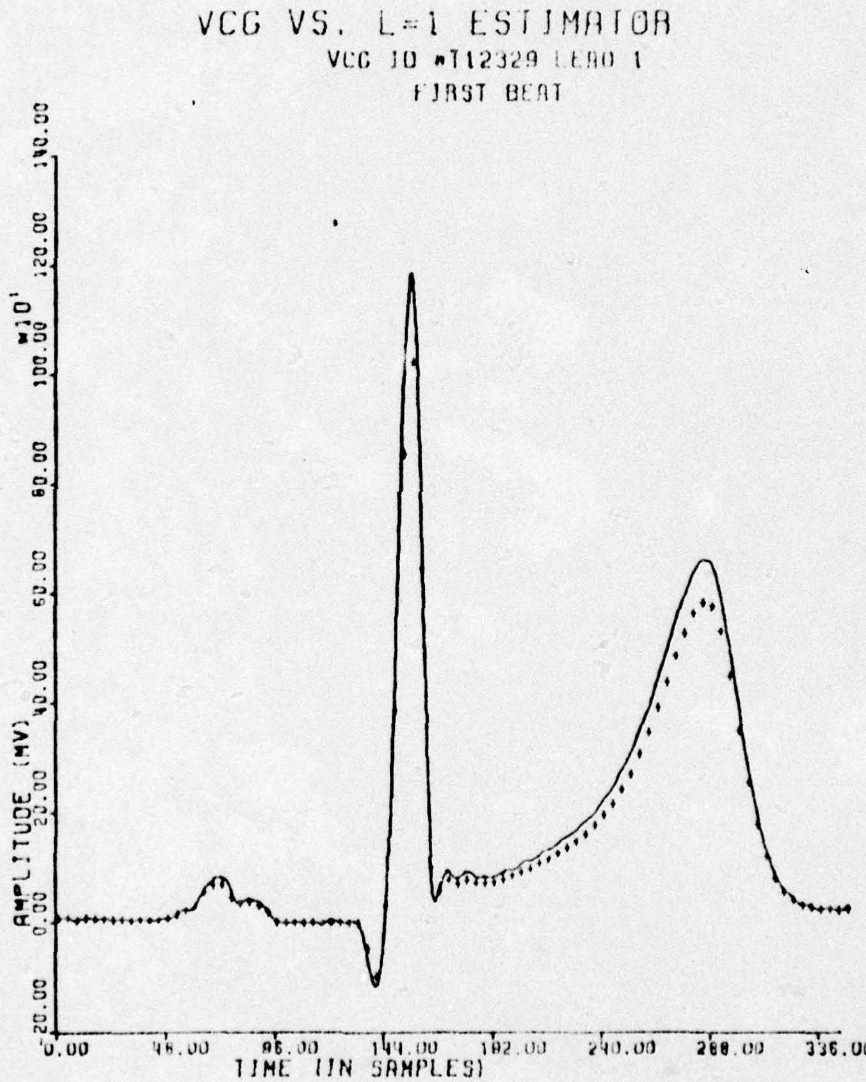


Figure 1(a): A VCG vs. L=1 Estimator (Q=11, R=500 s/s)

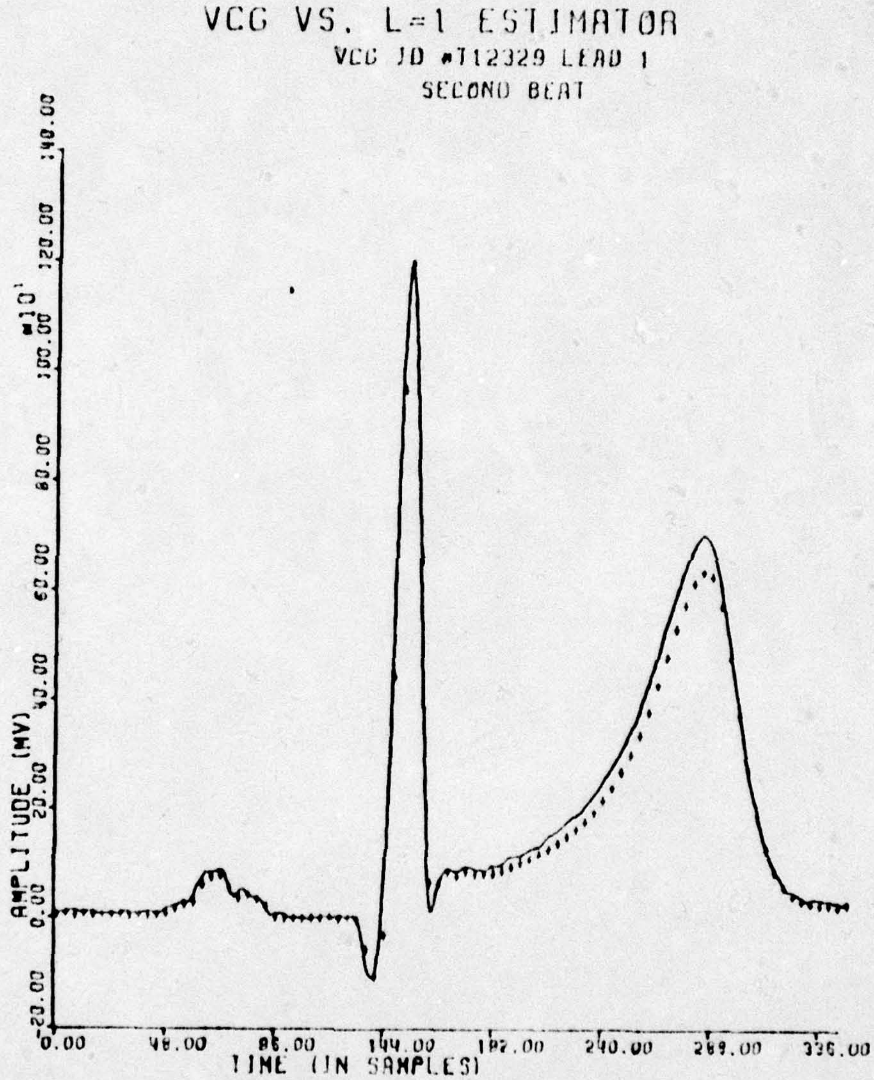


Figure 1(b): A VCG vs. L=1 Estimator (Q=11, R=500 s/s)

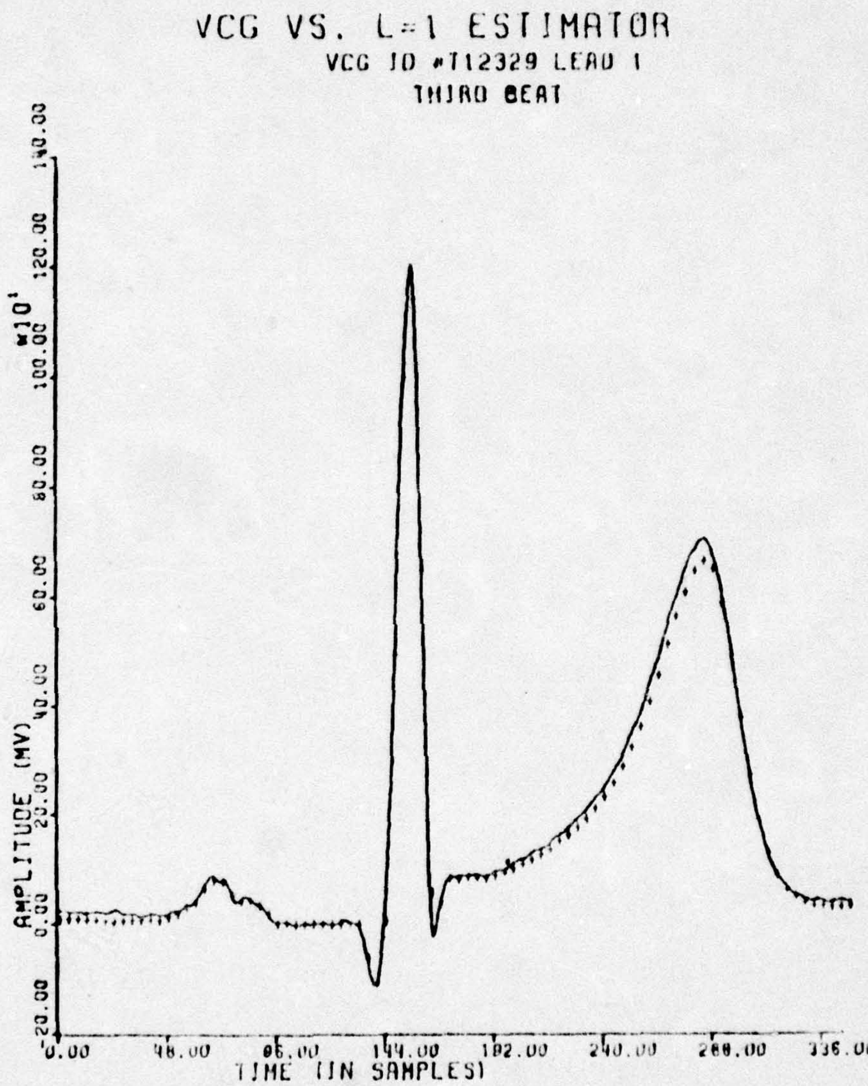


Figure 1(c): A VCG vs. L=1 Estimator (Q=11, R=500 s/s)

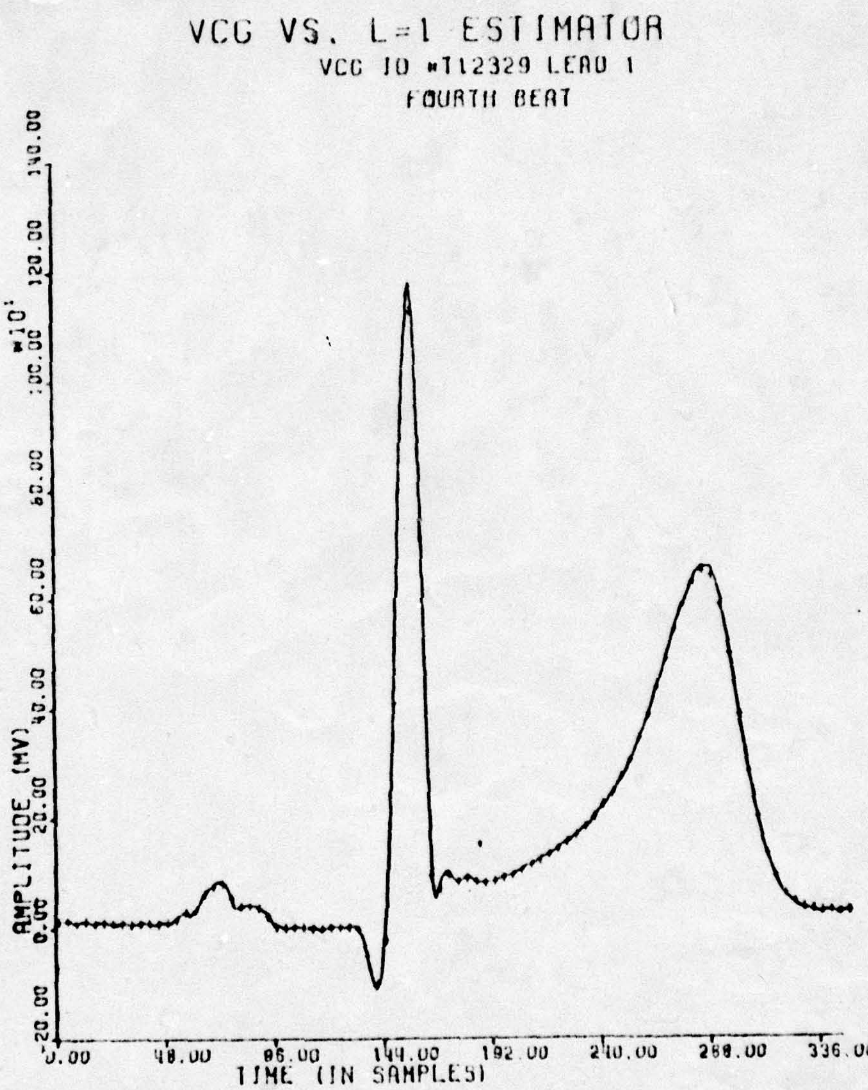


Figure 1(d): A VCG vs. L=1 Estimator (Q=11, R=500 s/s)

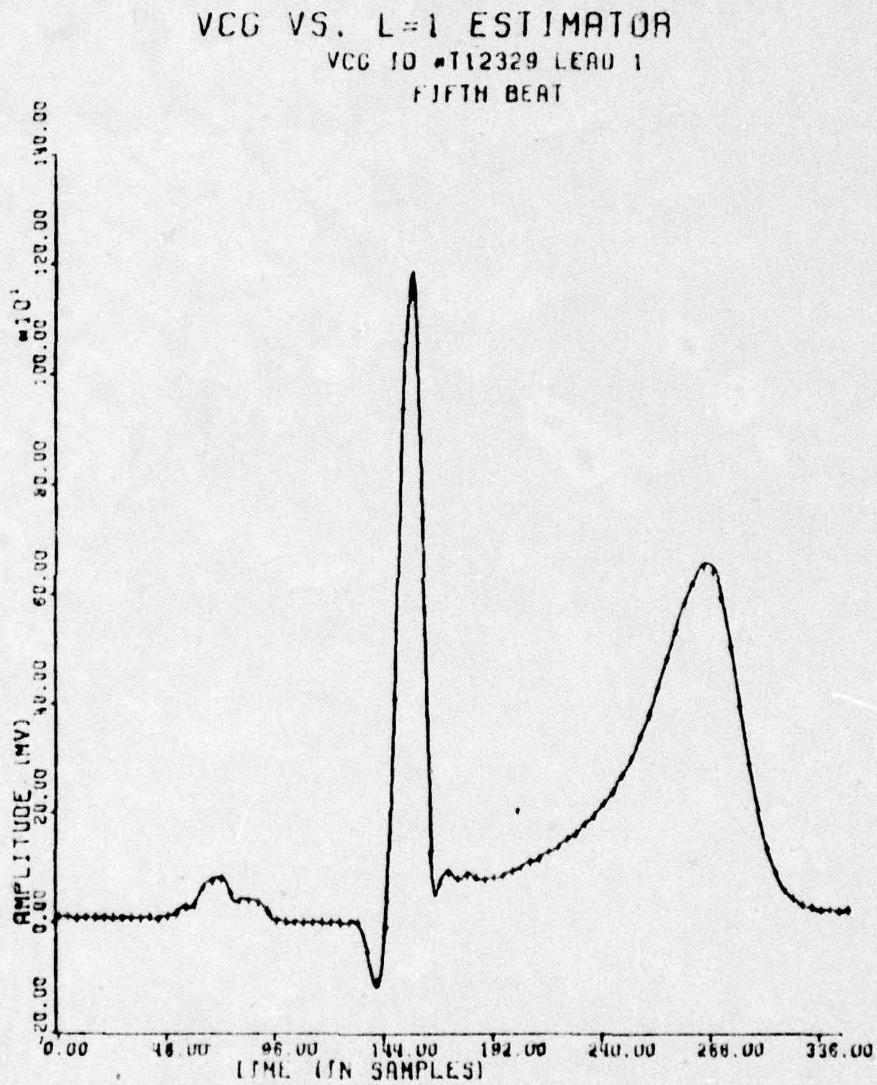


Figure 1(e): A VCG vs. L=1 Estimator (Q=11, R=500 s/s)

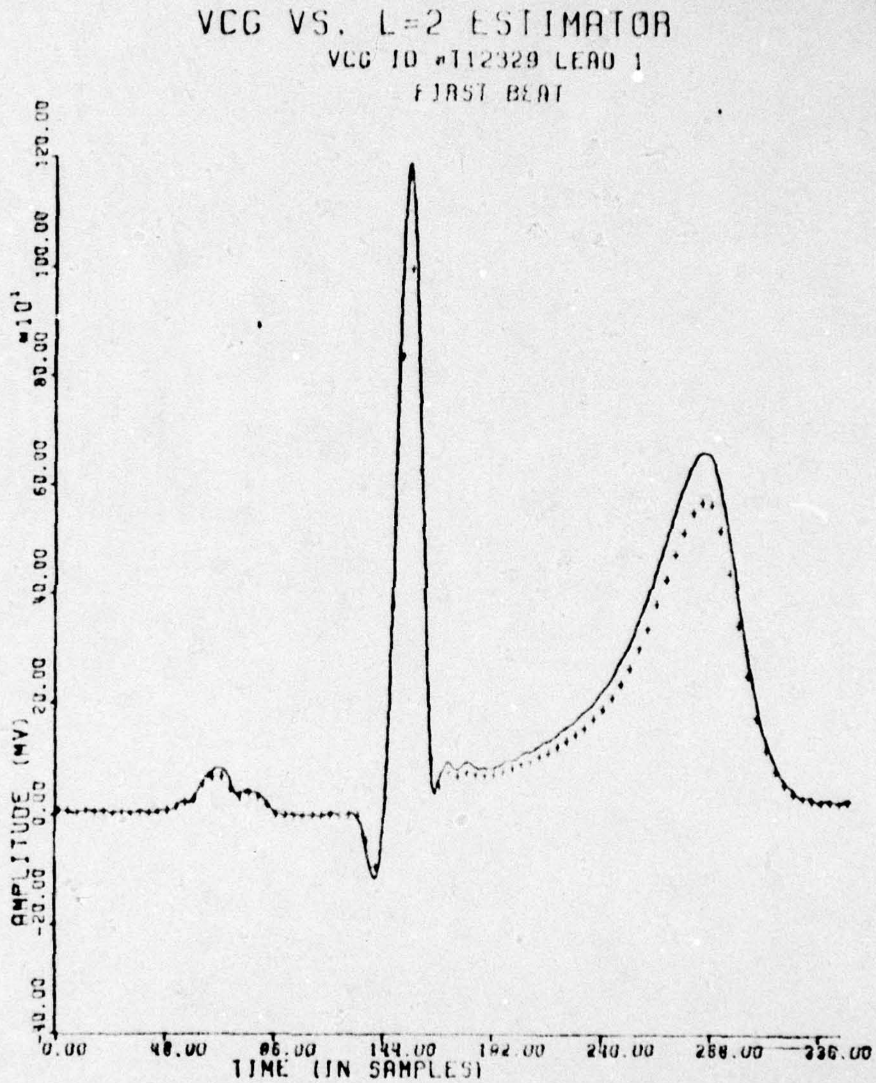


Figure 2(a): A VCG vs. L=2 Estimator (Q=11, R=500 s/s)

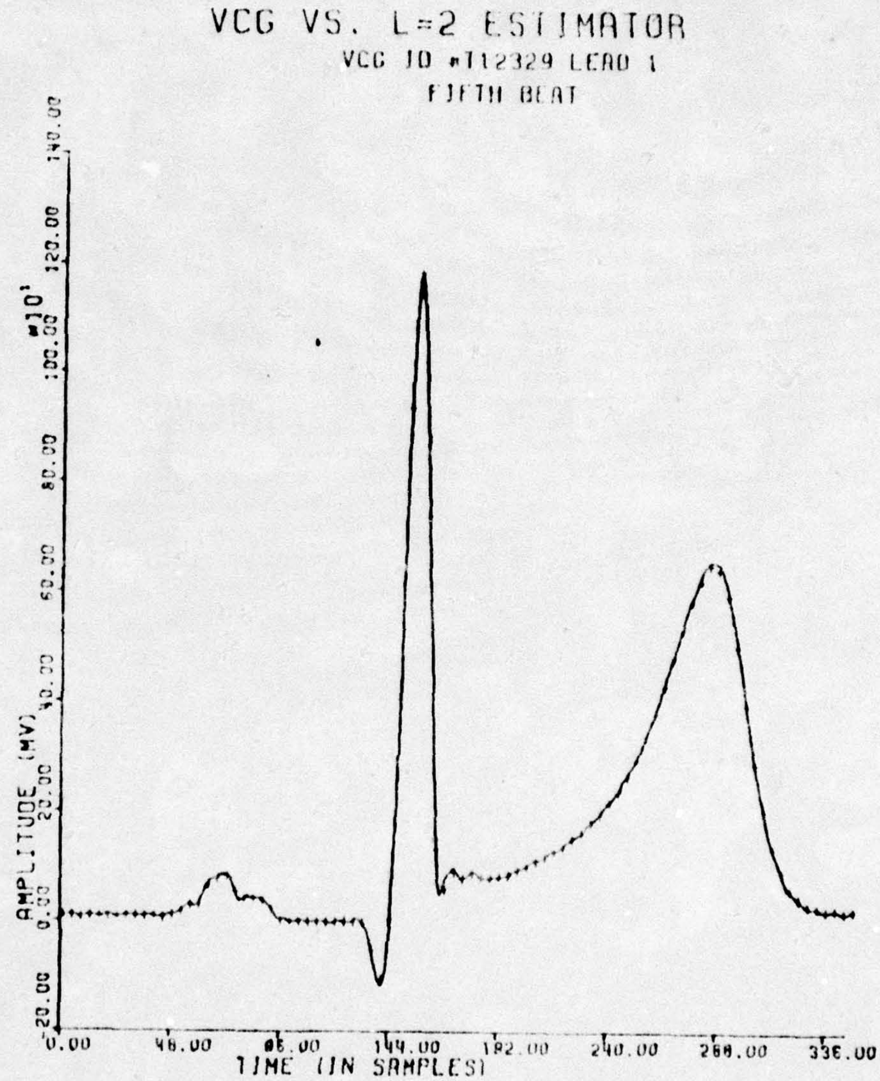


Figure 2(b): A VCG vs. L=2 Estimator (Q=11, R=500 s/s)

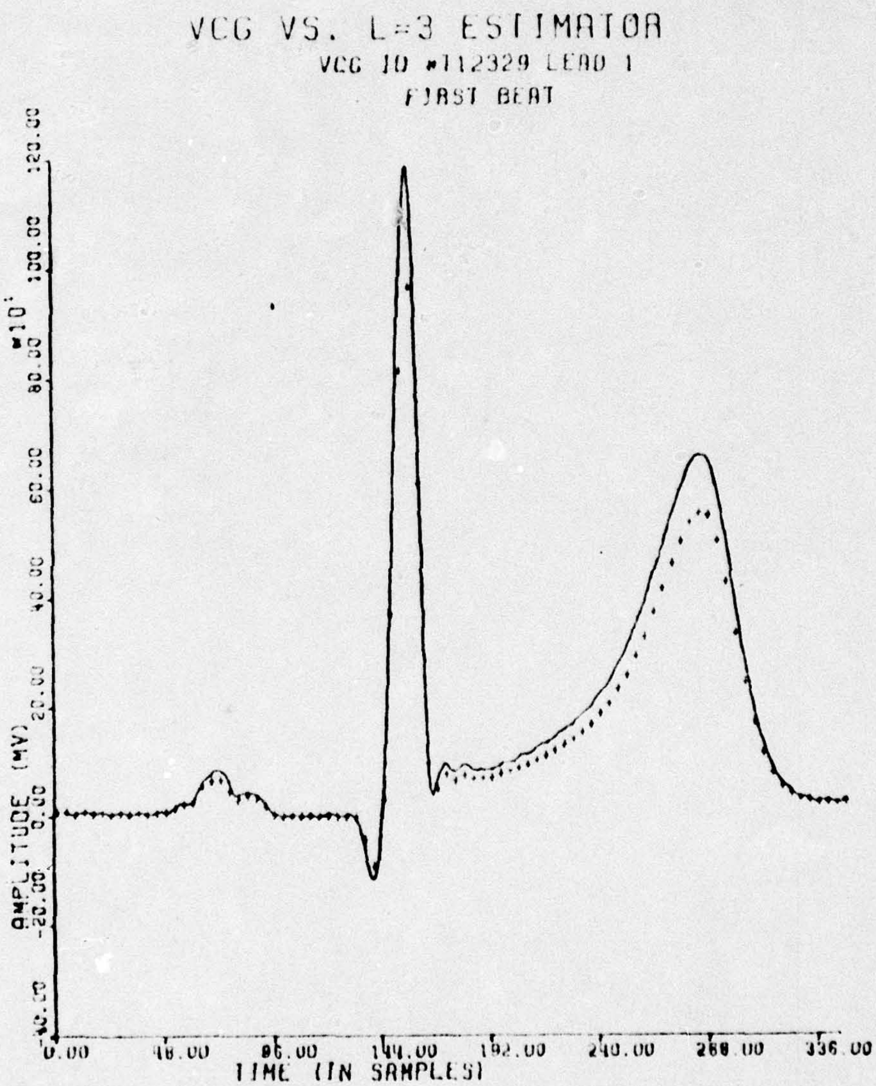


Figure 3(a): A VCG vs. L=3 Estimator (Q=11, R=500 s/s)

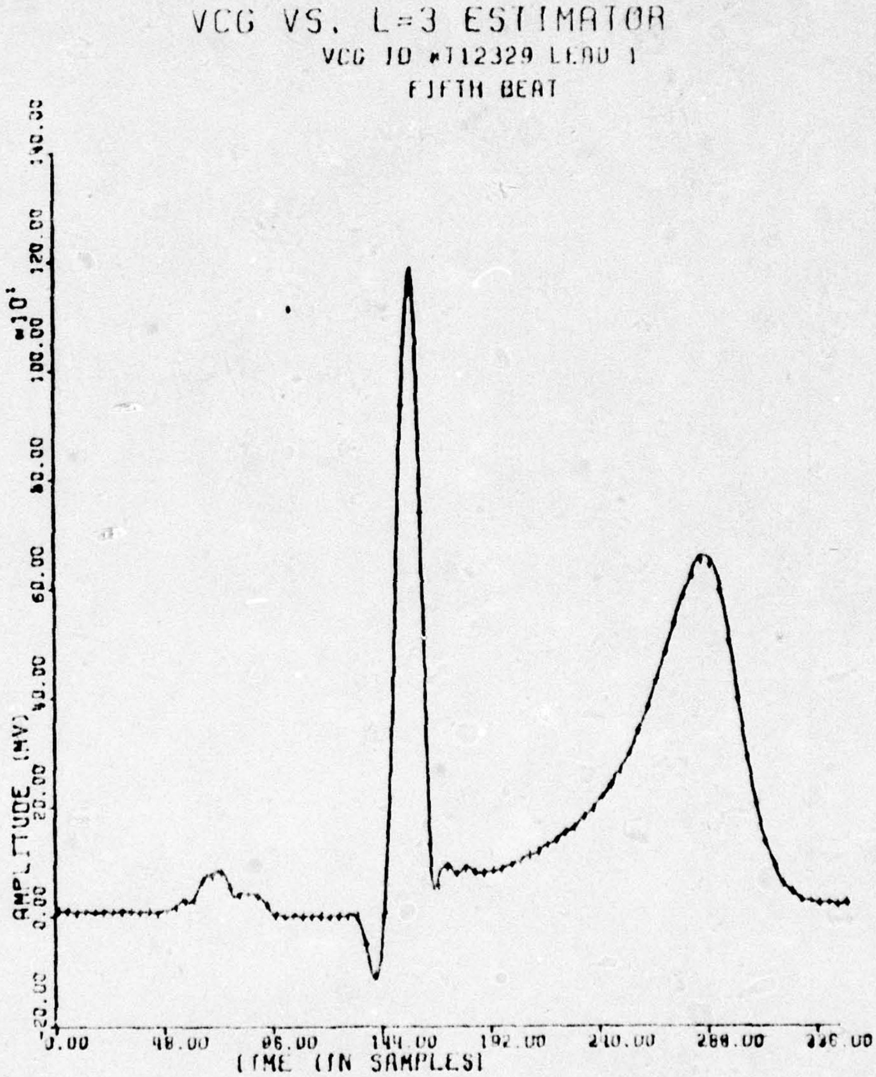


Figure 3(b): A VCG vs. L=3 Estimator (Q=11, R=500 s/s)

TABLE 1

Predictor Errors for VCG #T12329 Lead 1

Beat No.	Max. Error	Avg. Error	Std. Dev.	Entropy
1	165	19.9	31.2	5.5
2	147	17.3	28.3	5.6
3	72	10.8	14.3	5.2
4	129	4.1	17.4	5.9
5	76	2.9	12.0	3.8

500 samples/second at 11 bit quantization

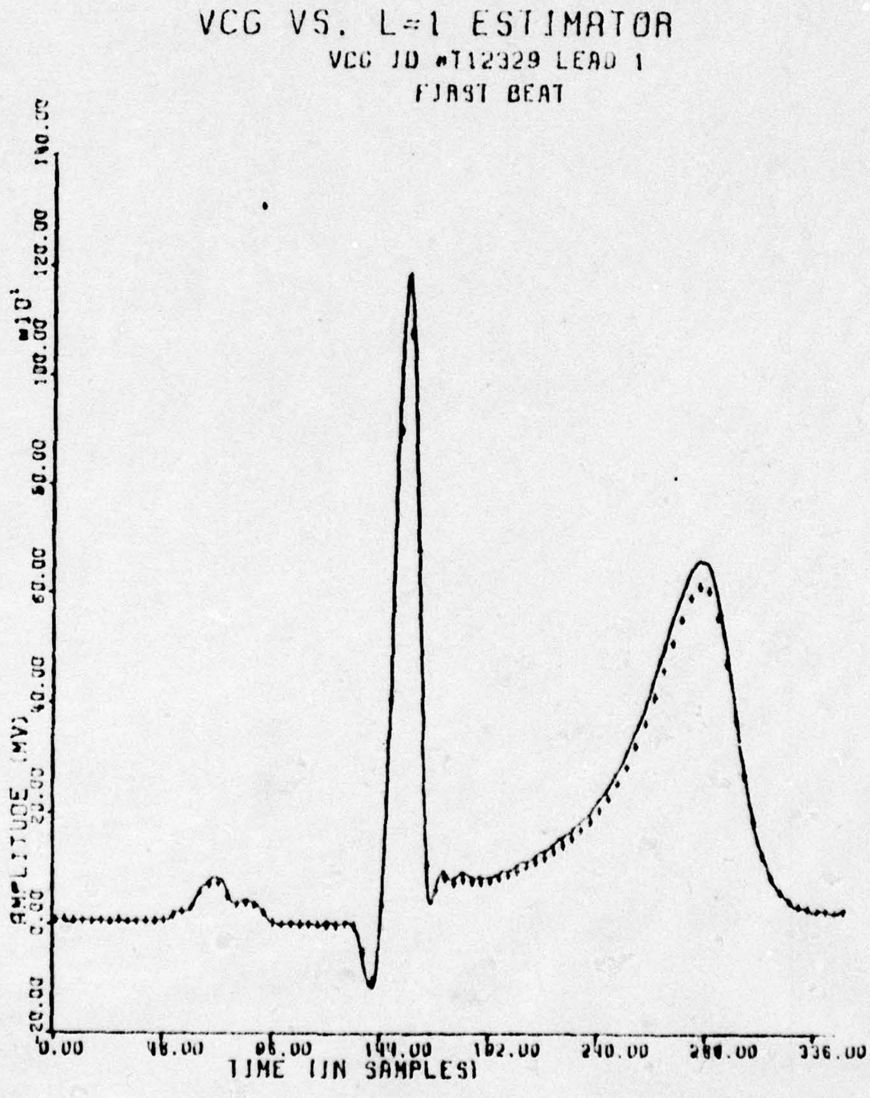


Figure 4(a): Non-Adaptive L=1 Estimator (Q=11, R=500 s/s)

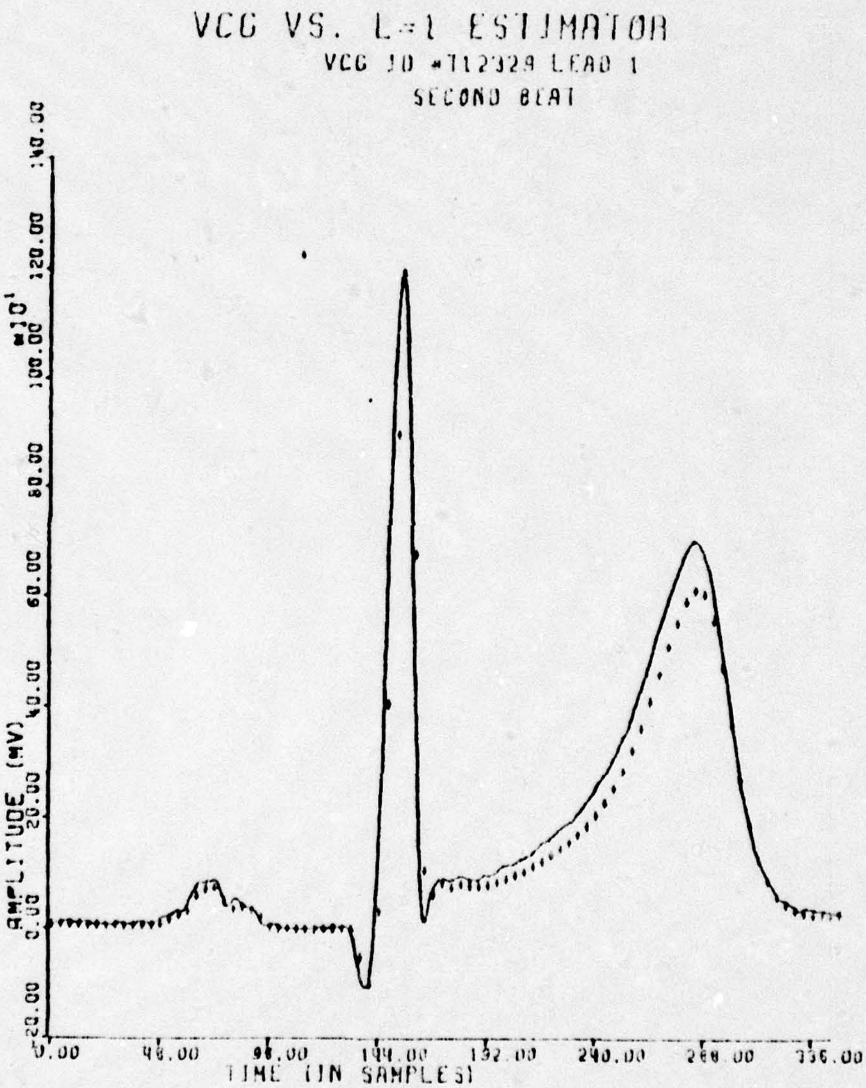


Figure 4(b): Non-Adaptive L=1 Estimator (Q=11, R=500 s/s)

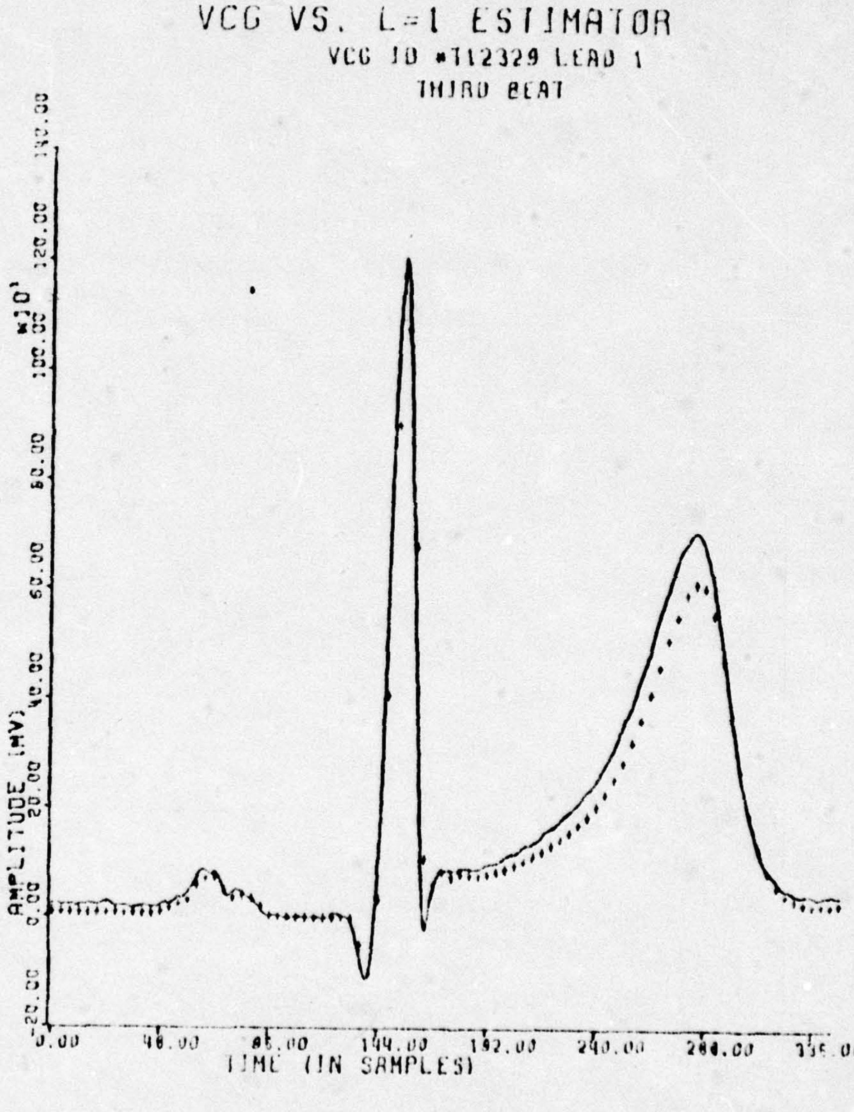


Figure 4(c): Non-Adaptive L=1 Estimator (Q=11, R=500 s/s)

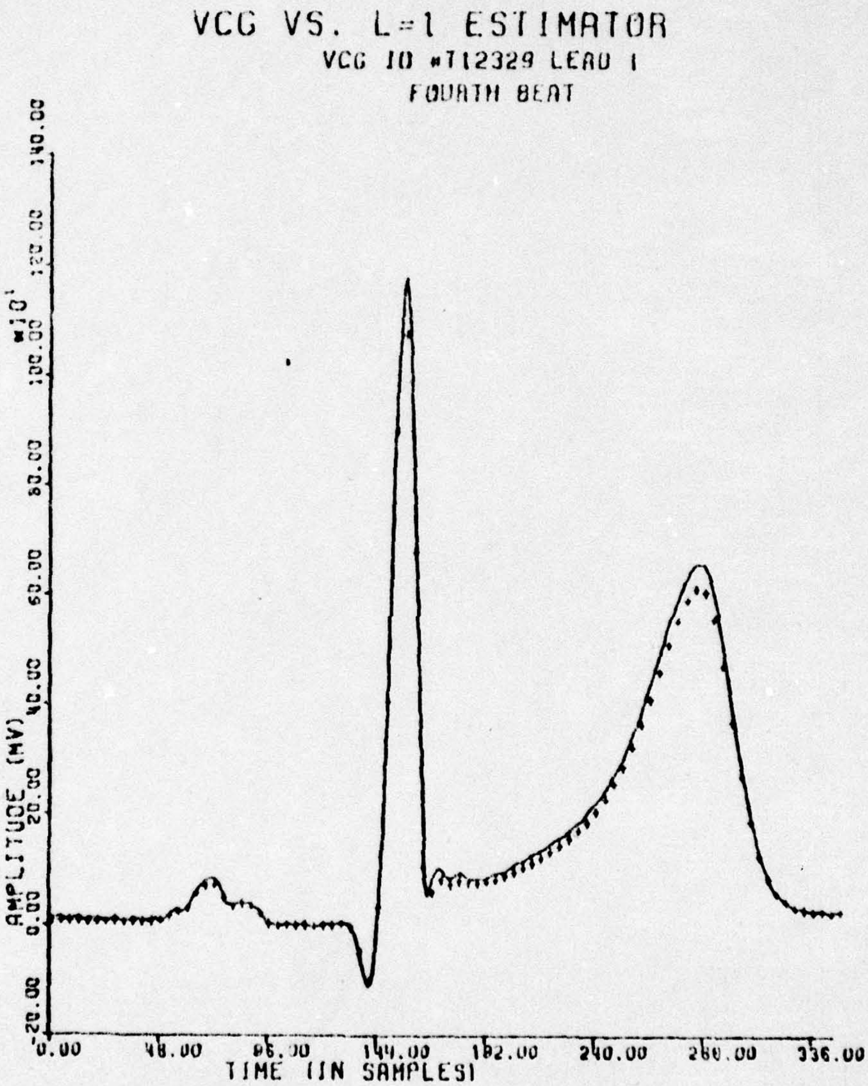


Figure 4(d): Non-Adaptive L=1 Estimator (Q=11, R=500 s/s)

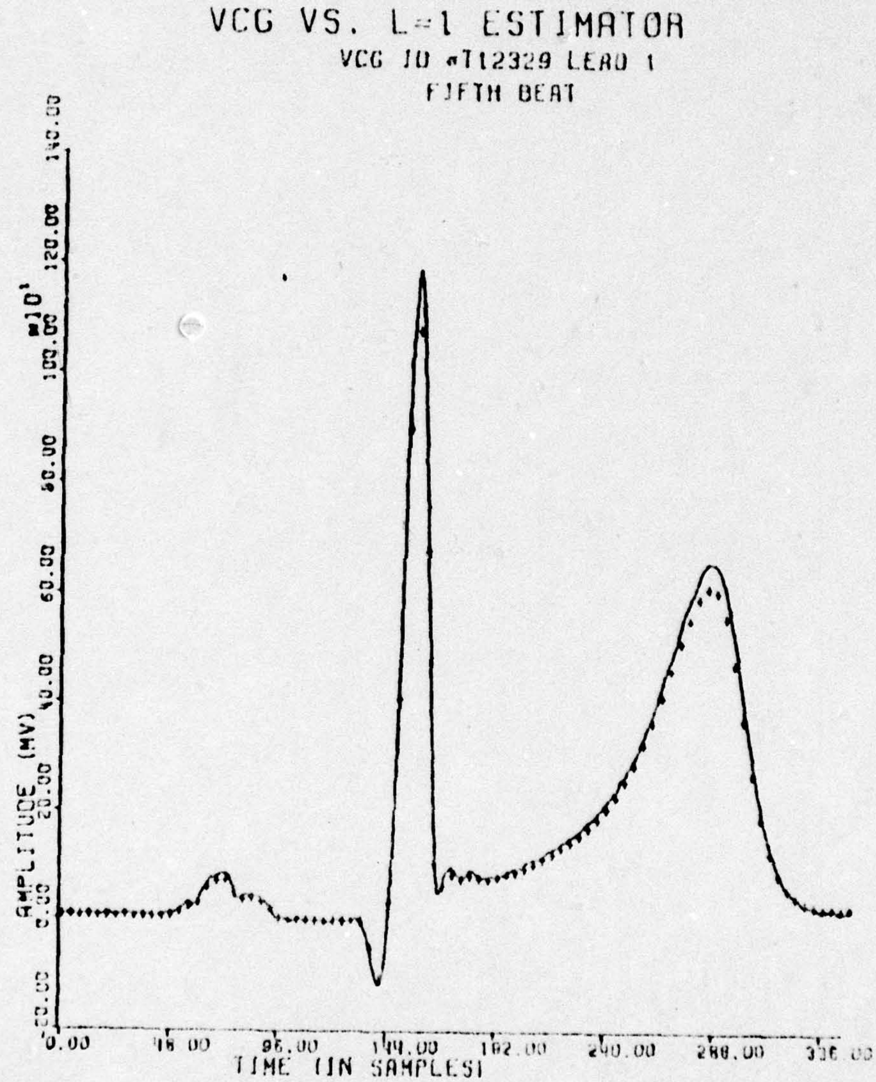


Figure 4(e): Non-Adaptive L=1 Estimator (Q=11, R=500 s/s)

TABLE 2

A Comparison of the Non-Adaptive and Adaptive Predictor Errors

Predictor	Beat No.	Max. Error	Avg. Error	Std. Dev.	Entropy
Adaptive	1	165	19.9	31.2	5.5
Nonadaptive	1	110	12.3	20.6	5.2
Adaptive	2	76	2.9	12.0	3.8
Nonadaptive	2	140	11.9	23.5	5.3

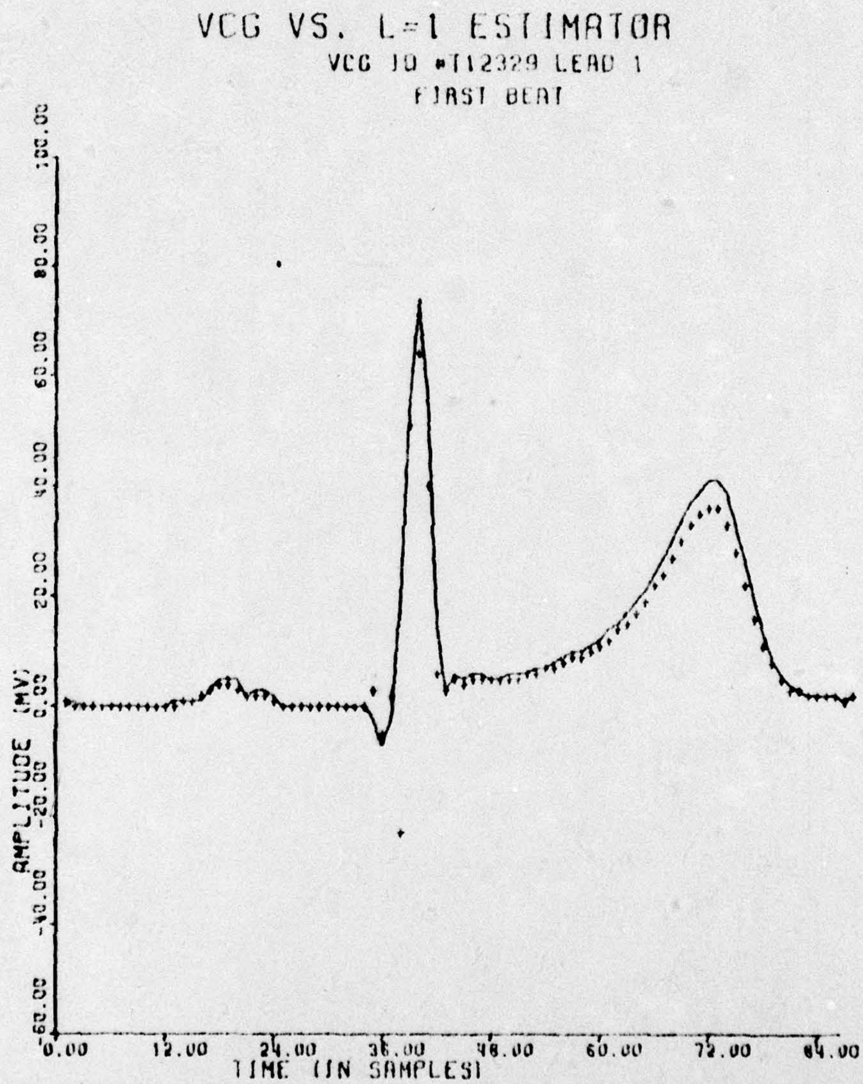


Figure 5(a): VCG vs. L=1 Estimator (Q=7 and R=125 s/s)

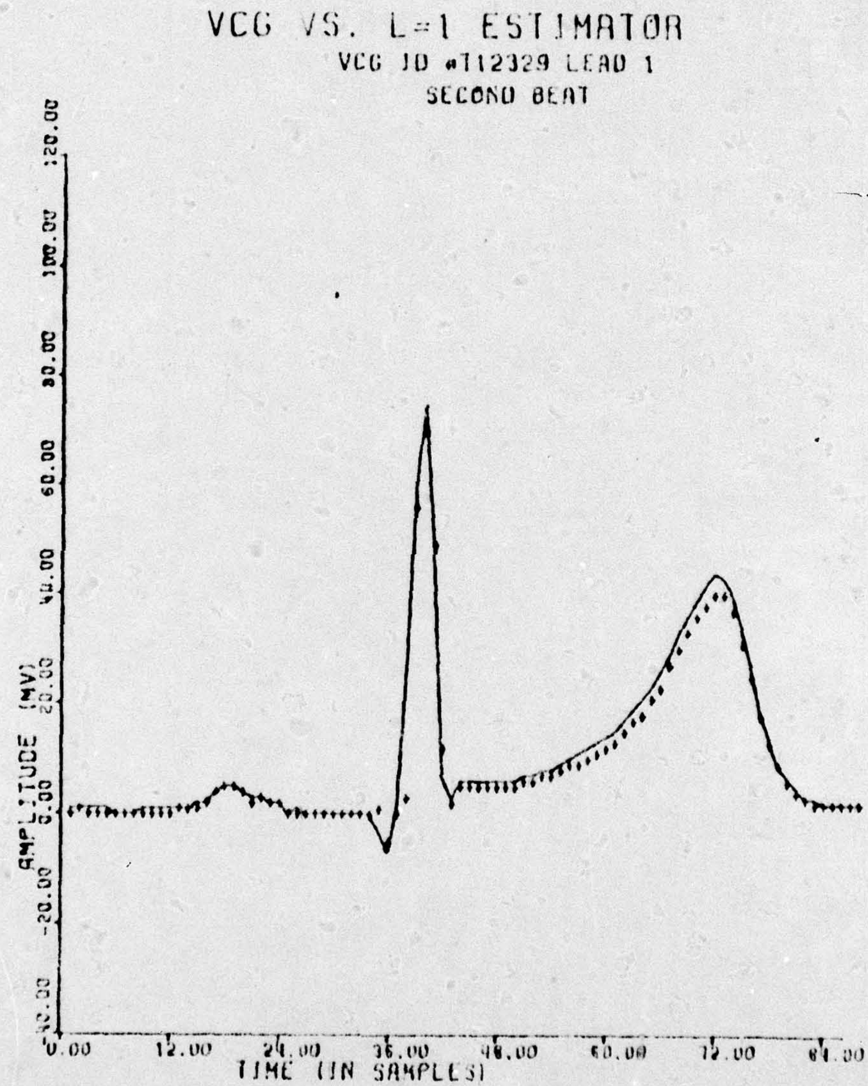


Figure 5(b): VCG vs. L=1 Estimator (Q=7 and R=125 s/s)

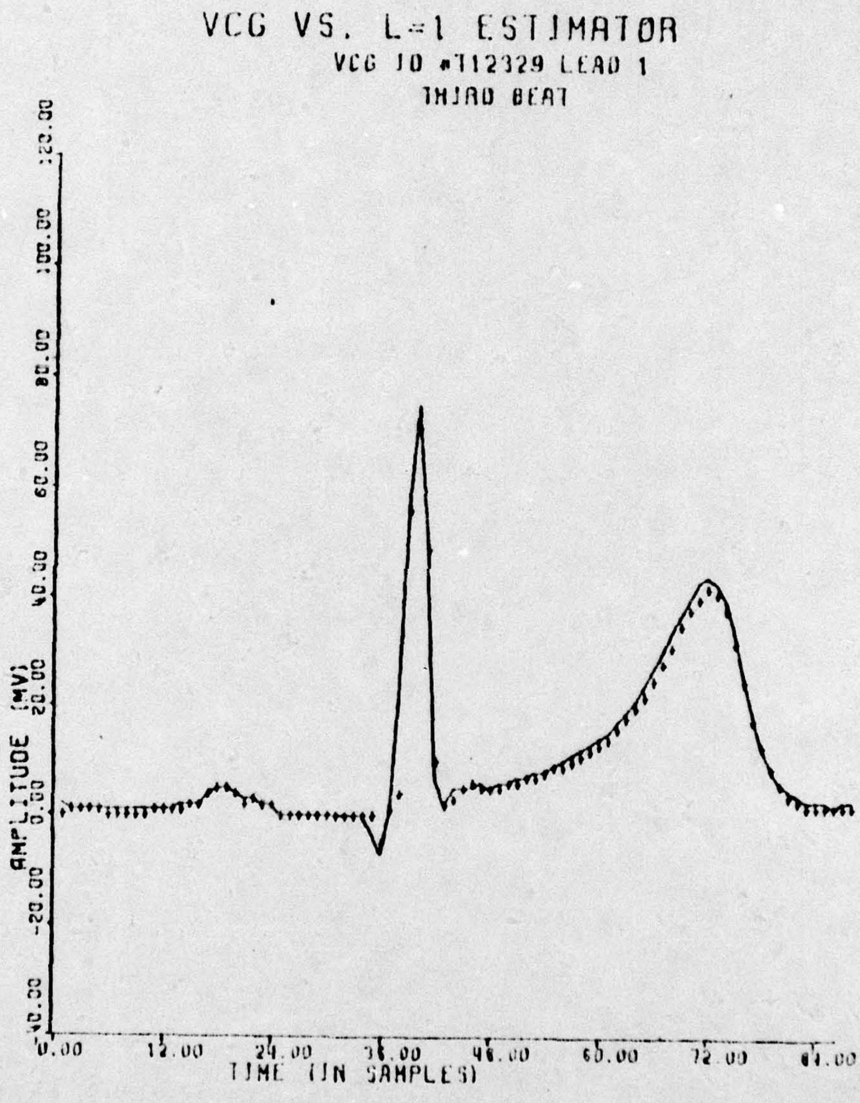


Figure 5(c): VCG vs. L=1 Estimator (Q=7 and R=125 s/s)

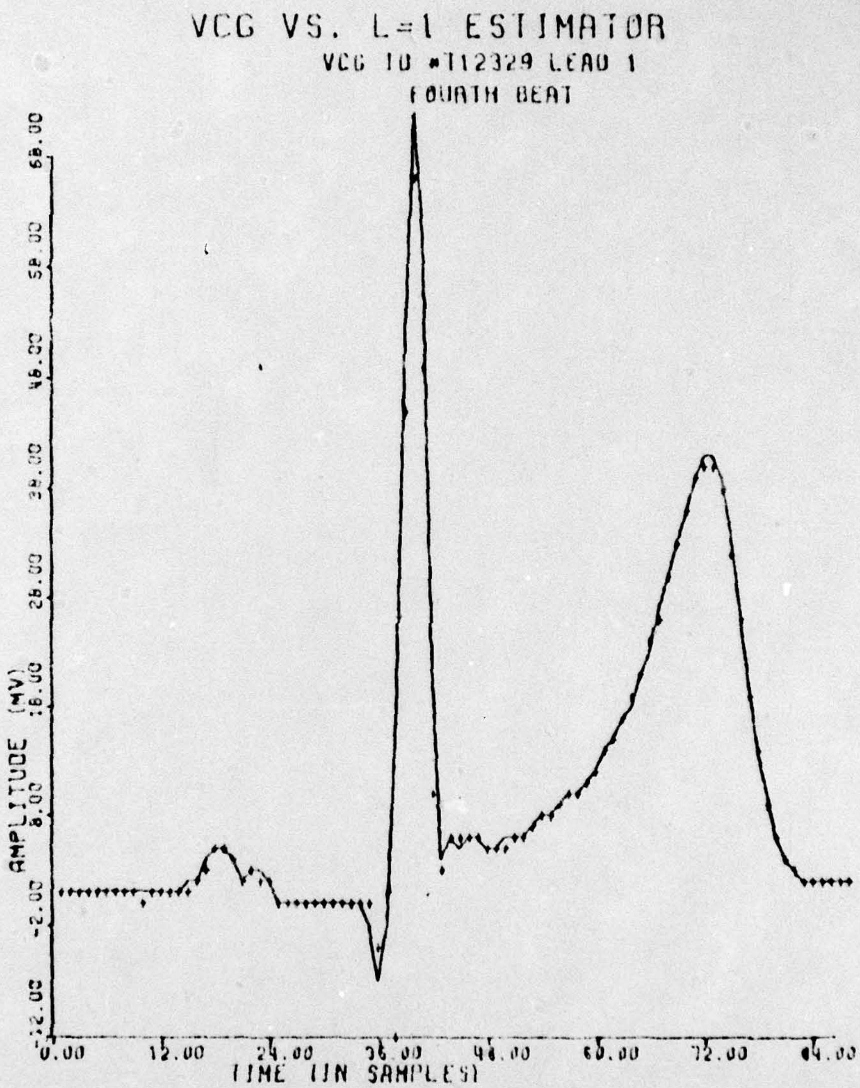


Figure 5(d): VCG vs. L=1 Estimator (Q=7 and R=125 s/s)

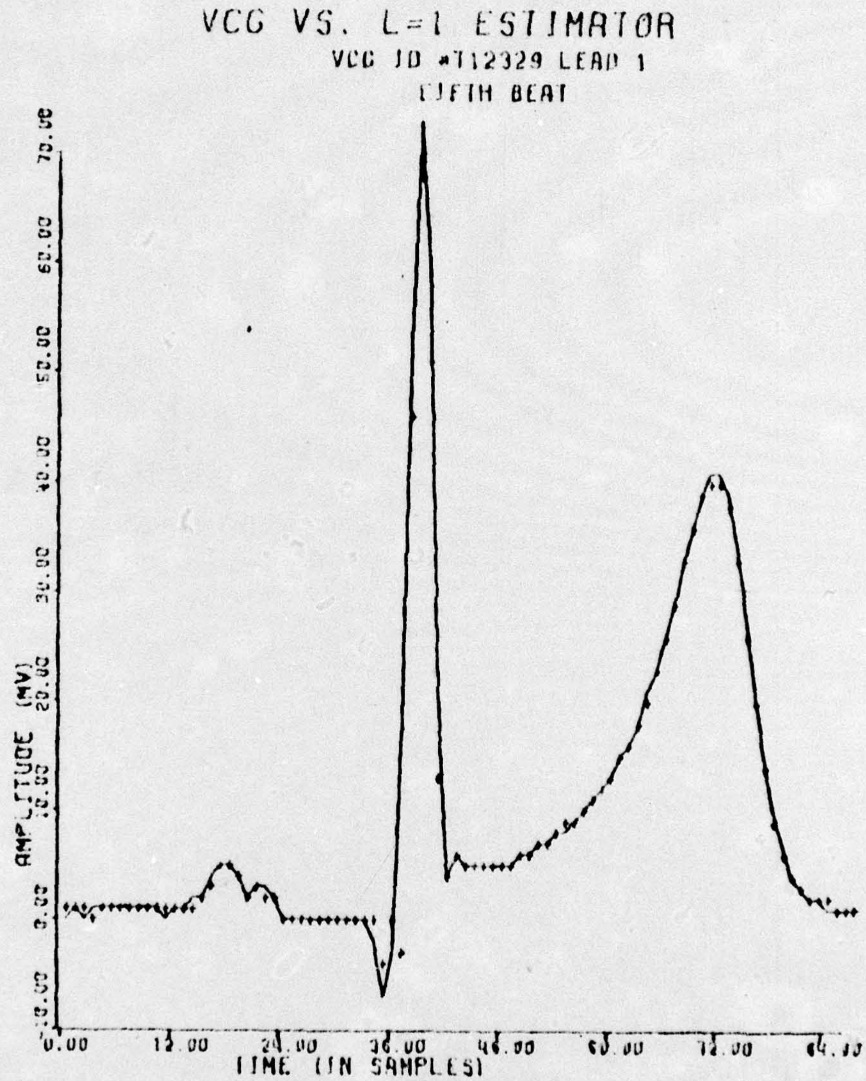


Figure 5(e): VCG vs. L=1 Estimator (Q=7 and R=125 s/s)

Appendix C

APPENDIX C

The Full Period Interpolator

Consider an N-element sample, of which only every kth equally spaced element is known. The remaining N-K elements must be interpolated from the K known elements. Let $\underline{S}_T^T = \{S_1, S_{k+1}, S_{2k+1}, \ldots, S_N\}$ be the vector of known samples, where $k = \frac{N-1}{K-1}$. Let S_ℓ be the sample being interpolated, and suppose

$$\hat{S}_\ell = A_\ell^T \underline{S}_T$$

Then the error $\epsilon_\ell \triangleq (S_\ell - \hat{S}_\ell) = (S_\ell - A^T \underline{S}_T)$. Squaring the error gives

$$\epsilon_\ell^2 = (S_\ell - A_\ell^T \underline{S}_T)(S_\ell - A_\ell^T \underline{S}_T)^T$$

$$\epsilon_\ell^2 = S_\ell^2 - 2A_\ell^T \underline{S}_T S_\ell - A_\ell^T \underline{S}_T \underline{S}_T^T A_\ell$$

This is the mean-squared error for the estimate $\hat{S}_\ell$. The total mean-squared error is $\sum_{\text{all } \ell} \epsilon_\ell^2$. Minimizing the total mean-squared-error is equivalent to minimizing each term individually, so

$$\frac{\partial \epsilon_\ell^2}{\partial A_\ell} = -2(\underline{S}_T^T S_\ell) + 2A_\ell^T \underline{S}_T \cdot \underline{S}_T^T = 0$$

Taking the expectation and solving for A_ℓ yields

$$A_\ell = \Lambda^{-1} \Gamma_\ell$$

where Λ is the symmetric correlation matrix of the transmitted samples (that is, $\lambda_{11} = E\{S_1,S_1\}$; $\lambda_{12} = E\{S_1,S_{k+1}\}$; $\lambda_{22} = E\{S_{k+1},S_{k+1}\}$) and Γ_ℓ is the correlation vector of the ℓth interpolate with the members of the transmitted samples ($\gamma_{11} = E\{S_\ell,S_1\}$; $\gamma_{12} = E\{S_\ell,S_{k+1}\}$). Since this representation is valid for each interpolate $\hat{S}_\ell$, $1 \le \ell \le N-k$, we see that the vector of interpolates

$$\underline{\hat{S}} \triangleq [\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_{N-k}]^T$$

$$= \{ [A_1^T, A_2^T, \ldots, A_{N-k}^T]^T \underline{S}_T \}$$

$$= \{ [(\Lambda^{-1}\Gamma_1)^T, (\Lambda^{-1}\Gamma_2)^T, \ldots, (\Lambda^{-1}\Gamma_{N-k})^T]^T \underline{S}_T \}$$

$$= \{ [\Gamma_1^T \Lambda^{-1}, \Gamma_2^T \Lambda^{-1}, \ldots, \Gamma_{N-k}^T \Lambda^{-1}]^T \underline{S}_T \}$$

$$= \{ [\Gamma_1^T, \Gamma_2^T, \ldots, \Gamma_{N-k}^T]^T \Lambda^{-1} \underline{S}_T \}$$

Now define $\Gamma = [\Gamma_1^T, \Gamma_2^T, \ldots, \Gamma_{N-k}^T]^T$ to be the $(N-k) \times (k)$ matrix of interpolate correlations and we have the desired full beat interpolator.

$$\underline{\hat{S}} = \Gamma \Lambda^{-1} \underline{S}_T$$

Note that Γ and Λ are invariant; they do not change as new beats enter the interpolator. The interpolation scheme is thus a large matrix product; it does not involve any matrix inversion.

Appendix D

```
      INTEGER ID(3)
      REAL LAMDA(4,4),DET
      DIMENSION L(4),M(4)
      DIMENSION H(352,6),GAMA(4),ALFA(4)
      DIMENSION S(351),X(351),DIFF(351)
      DATA NDEX1/1/,NDEX2/1/
C
C     READ IN THE PREDICTOR CONTROL INFORMATION
C        LEN -- PREDICTOR LENGTH
C        NRATE -- SAMPLE RATE (MAX IS 500)
C        MAXI -- NUMBER OF BEATS TO BE PROCESSED
C        IQ -- QUANTIZATION LEVEL IN BITS
C
      WRITE(6,890)
890   FORMAT(1H0,'GIVE THE PREDICTOR CONTROL INFORMATION'/5X,
     2'PREDICTOR LENGTH (I5)'/5X,'SAMPLE RATE (I5)'/5X,
     3'NUMBER OF BEATS (I5)'/5X,'QUANTIZATION LEVEL (I5)'/5X,'X',4X,
     4'X',4X,'X',4X,'X')
      READ(6,910) LEN,NRATE,MAXI,IQ
910   FORMAT(4I5)
C
C     READ AND ECHO THE VCG IDENTIFICATION DATA
C
      READ(5,900) ID,LEAD
900   FORMAT(3A2,4X,I1)
      WRITE(6,901) ID,LEAD
901   FORMAT(1H1,'VCG ID NUMBER ',3A2,' LEAD ',I1)
C
C     READ IN THE AVERAGE HEARTBEAT
C
      READ(5,920) (H(I,1),I=1,351)
920   FORMAT(7F10.0)
C
C     RELOCATE SAMPLES APPROPRIATE TO THE DESIRED SAMPLING RATE
C
      NA=INT(500./NRATE+0.5)
      N=INT(351./NA+0.5)
      IF(NA.EQ.1) GO TO 12
      NX=N+1
      DO 329 I=2,NX
      ND=NA*(I-1)+1
329   H(I,1)=H(ND,1)
12    IDIV=2**(11-IQ)
      DO 330 I=1,N
330   H(I,1)=H(I,1)/IDIV
C
C     INITIALIZE THE CORRELATION COMPUTATIONS
C
      LX=LEN+1
      DO 8 I=1,N
```

```
      II=N-I+1
      DO 8 J=1,LX
      JJ=J+1
      JK=II-J+1
      IF(JK.LE.0) JK=JK+N
      H(II,JJ)=H(II,1)*H(JK,1)*EXP(-0.1*J)
8     CONTINUE
990   FORMAT(1X,6F10.0)
C
C     READ IN THE SAMPLE VALUES FOR A BEAT
C
13    READ(5,1010) (S(I),I=1,351)
1010  FORMAT(12F6.0)
C
C     SELECT ONLY THOSE SAMPLES TO BE USED
C
      IF(NA.EQ.1) GO TO 334
      DO 333 I=2,N
      ND=NA*(I-1)
333   S(I)=S(ND)
334   CONTINUE
      DO 332 I=1,N
332   S(I)=S(I)/IDIV
C
C     COMPUTE THE CORRELATIONS--MATRIX LAMDA AND VECTOR GAMA
C
17    DO 10 I=1,LEN
      II=N+1-I
      K=2
      DO 10 J=I,LEN
      LAMDA(I,J)=H(II,K)
      LAMDA(J,I)=LAMDA(I,J)
10    K=K+1
      DO 20 J=1,LEN
20    GAMA(J)=H(1,J+2)
C
C     START TO FORM THE ESTIMATE.  INVERT THE MATRIX LAMDA.
C
      IF(LEN.GT.1) GO TO 22
      LAMDA(1,1)=1./LAMDA(1,1)
      GO TO 35
22    CALL INVERT(LAMDA,LEN,DET)
      IF(DET.NE.0.) GO TO 30
      WRITE(6,2000) NDEX2
2000  FORMAT(1H ,'DETERMINANT ALMOST SINGULAR -- EXTRAPOLATE ESTIMATE',I
     24)
25    X(NDEX2)=2*H(N,1)-H(N-1,1)
      GO TO 50
30    CONTINUE
C
```

```
C     COMPUTE THE OPTIMAL PREDICTOR COEFFICIENTS
C
35    DO 40 I=1,LEN
      ALFA(I)=0.
      DO 40 J=1,LEN
      ALFA(I)=ALFA(I)+LAMDA(I,J)*GAMA(J)
40    CONTINUE
C
C     ESTIMATE THE NEXT SAMPLE
C
      X(NDEX2)=0.
      DO 50 I=1,LEN
      X(NDEX2)=X(NDEX2)+ALFA(I)*H(N+1-I,1)
50    CONTINUE
      X(NDEX2)=AINT(X(NDEX2)+0.5)
C
C     COMPUTE THE DIFFERENCE BETWEEN THE DATA AND THE ESTIMATE.
C
55    DIFF(NDEX2)=S(NDEX2)-X(NDEX2)
C
C     UPDATE THE CORRELATION FUNCTIONS
C
      LY=LX+1
      DO 60 J=2,LY
60    H(352,J)=H(1,J)
      DO 65 I=1,N
      DO 65 J=1,LY
65    H(I,J)=H(I+1,J)
      H(N,1)=S(NDEX2)
      DO 70 J=2,LY
70    H(N,J)=(NDEX1*H(352,J)+H(N,1)*H(N+2-J,1))/(NDEX1+1)
C
C     HAVE WE REACHED THE END OF THE DATA?
C
      NDEX2=NDEX2+1
      IF(NDEX2.LE.N) GO TO 17
      NDEX2=NDEX2-N
C
C     PRINT THE DATA FOR THIS BEAT
C
      WRITE(6,2010) NDEX1
2010  FORMAT(1H0,5X,'DATA FOR BEAT NUMBER',I2/6X,'SAMPLE',
     25X,'ESTIMATE',5X,'DIFFERENCE')
      WRITE(6,2020) (S(I),X(I),DIFF(I),I=1,N)
2020  FORMAT(6X,F6.0,6X,F6.0,8X,F6.0)
      WRITE(7,2020) (S(I),X(I),DIFF(I),I=1,N)
C
C     COMPUTE THE DIFFERENCE AVERAGE, MEAN-SQUARE ERROR, AND ENTROPY.
C
      SUM=0.
```

```
      SUM2=0.
      ENTRPY=0.
      NEX=N-1
      DO 120 I=1,NEX
      IF(DIFF(I).EQ.999999.) GO TO 120
      SUM=SUM+DIFF(I)
      SUM2=SUM2+DIFF(I)*DIFF(I)
      COUNT=1.
      IEX=I+1
      DO 110 J=IEX,N
      IF(DIFF(I).NE.DIFF(J)) GO TO 110
      SUM=SUM+DIFF(J)
      SUM2=SUM2+DIFF(J)*DIFF(J)
      DIFF(J)=999999.
      COUNT=COUNT+1.
110   CONTINUE
      ENTRPY=ENTRPY-1.4427*(COUNT/N)*ALOG(COUNT/N)
120   CONTINUE
      SUM=SUM/N
      SUM2=SUM2/N
      WRITE(6,2030) SUM,SUM2,ENTRPY
2030  FORMAT(1H0,'THE AVERAGE OF THE DIFFERENCES IS ',F6.1/1H ,'THE MEAN
     2-SQUARE ERROR IS ',F10.2/1H ,'THE ENTROPY OF THE DIFFERENCES IS ',
     3F5.1,' BITS')
C
C     STOP?  OR ANOTHER BEAT?
C
      NDEX1=NDEX1+1
      IF(NDEX1.LE.MAXI) GO TO 13
      ENDFILE 7
      STOP
      END
      SUBROUTINE INVERT(LAMDA,LEN,DET)
      REAL LAMDA(4,4),A(4,8)
      DET=1.
C
C     INITIALIZE THE A MATRIX
C
      DO 5 I=1,4
      DO 5 J=1,8
5     A(I,J)=0.
      DO 20 I=1,LEN
      DO 10 J=1,LEN
10    A(I,J)=LAMDA(I,J)
20    A(I,4+I)=1.
C
C     PERFORM THE INVERSION BY ELEMENTARY ROW REDUCTIONS
C     ON THE MATRIX A
C
      DO 45 I=1,LEN
```

```
      IF(A(I,I).EQ.0.) GO TO 70
      DO 30 J=I,LEN
      IF(A(J,I).EQ.0.) GO TO 30
      TEMP=A(J,I)
      DO 30 K=I,8
      A(J,K)=A(J,K)/TEMP
30    CONTINUE
      ID=I+1
      IF(ID.GT.LEN) GO TO 40
      DO 40 J=ID,LEN
      TEMP=A(J,I)
      DO 40 K=I,8
      IF(TEMP.EQ.1.) A(J,K)=A(J,K)-A(I,K)
40    CONTINUE
45    CONTINUE
C
C     LAMDA IS IN UPPER TRIANGULAR FORM--COMPLETE THE REDUCTION
C
      LL=LEN-1
      DO 55 I=1,LL
      IJ=I+1
      DO 50 J=IJ,LEN
      TEMP=A(I,J)
      DO 50 K=J,8
50    A(I,K)=A(I,K)-TEMP*A(J,K)
55    CONTINUE
C
C     RETURN THE INVERSE MATRIX TO LAMDA
C
      DO 60 I=1,LEN
      DO 60 J=1,LEN
      LAMDA(J,I)=A(J,4+I)
60    LAMDA(I,J)=LAMDA(J,I)
      GO TO 80
70    DET=0.
80    RETURN
      END
```

ILM