FINAL REPORT

to

Jet Propulsion Laboratory

Subcontract No. 957030

JPL Technical Officer: Dr. Steve Townes

for

Low Cost Voice Compression for Mobile Digital radios

Principal investigator

Jim K. Omura

Electrical Engineering Department

University of California

Los Angeles, California 90024

June 1985

This report was prepared for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

PREFACE

The final report for this grant consists of the following three parts:

Part I : Executive Summary

Part II : Report on Computer Simulations

Part III: Audio Tape of Simulations

This document includes Part I and Part II along with a summary description of the contents of the audio tape. Part II provides a detail description of the specific algorithms and parameters employed including parameter quantization levels. Also included are the Language C computer programs of the simulations used on UCLA's MASSCOMP computer

PART I

EXECUTIVE SUMMARY

Objective

The goal of this contract was to develop a new technique for low cost robust voice compression at 4800 bits per second. Our approach was based on using a cascade of digital biquad adaptive filters with simplified multipulse excitation followed by simple bit sequence compression.

Initial Results

Digital biquad adaptive filters are relatively easy to implement and compare well with the more commonly used LPC filters. This was shown by Martin and Sun [1,2] of UCLA. Work in this contract applied these biquad adaptive filter results to voice compression at 4800 bits per second. The generation of multiple excitation was based on combining the well known (M,L) tree search algorithms [3] followed by short block compression algorithms.

The work on this contract started with the basic block diagram shown in Figure 1. Here speech sampled 8,000 times a second with 12 bit quantization is denoted by $\underline{s}$. Eight adaptive biquad filter coefficients $\underline{k}$ corresponding to a cascade of four biquad filters were computed (using the Martin and Sun algorithms) and sent to the receiver once every 160 samples. The same coefficients were used in the speech synthesis model in the (M,L) tree search agorithm.

The (M,L) tree search algorithm assumes that binary symbols enter the cascade of four biquad filters at a rate of 8,000 bits per second. After each bit enters the filter an estimate of the speech sample exits. The inputs and outputs of these filters are represented by a binary tree illustrated in Figure 2. Starting at some initial filter condition, all possible binary inputs

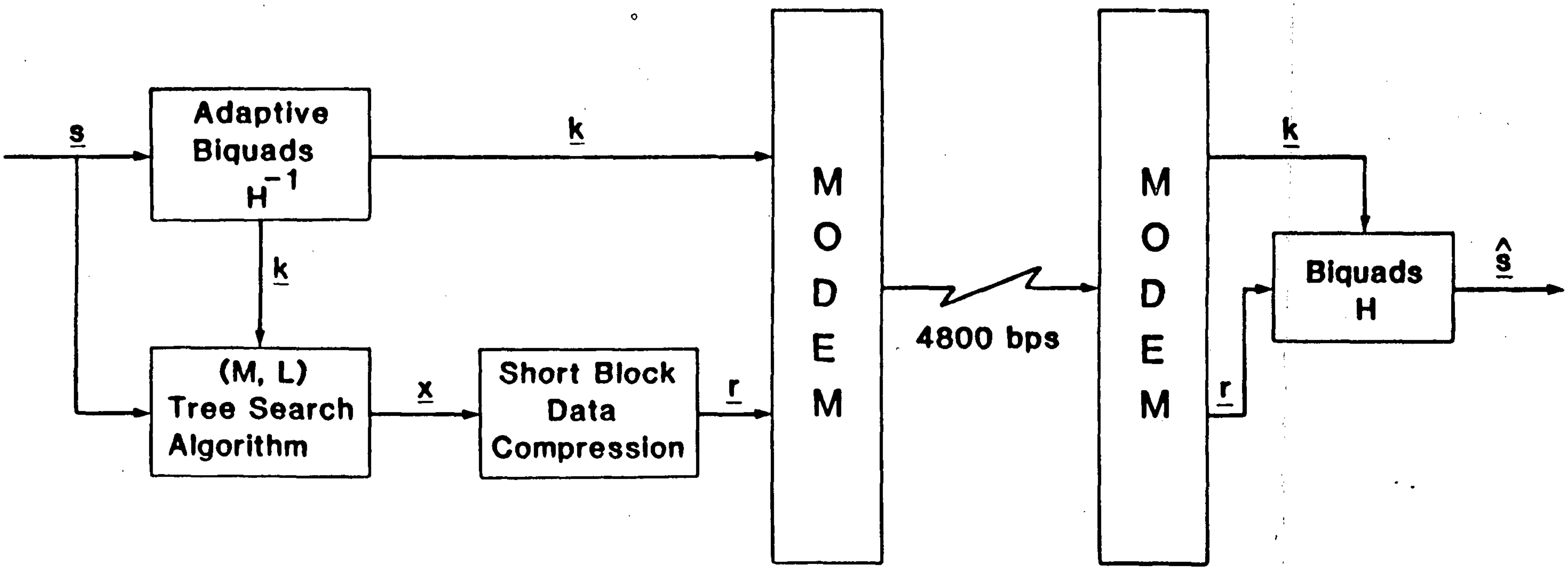
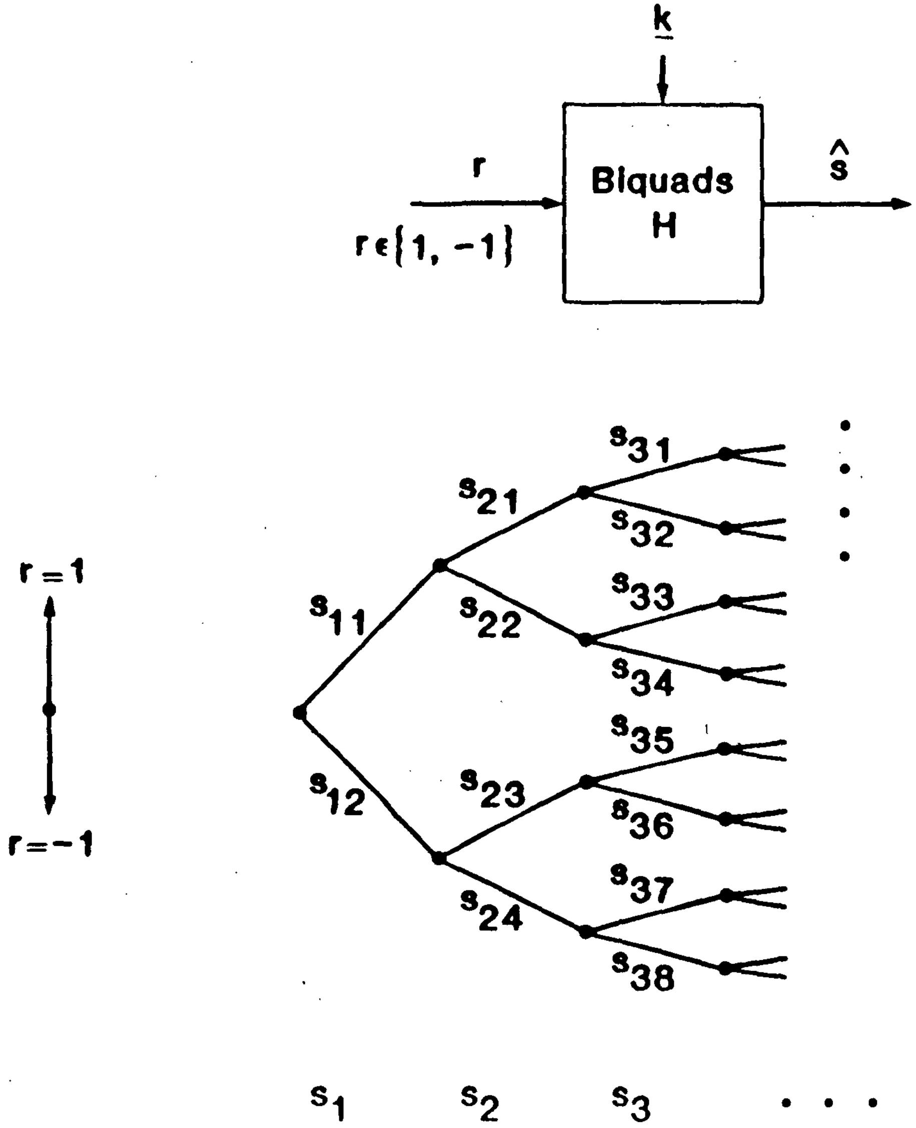

s
Adaptive Biquads H^{-1}
k
k
(M, L) Tree Search Algorithm
x
Short Block Data Compression
r
MODEM
4800 bps
MODEM
k
r
Biquads H
$\hat{s}$

Figure 1. Original Block Diagram

Figure 2. Model of Speech Process



to the cascade of filters and their corresponding outputs are represented by the tree. Potential estimated speech samples are labeled on each branch of this tree. Shown below are the actual speech samples denoted by s_1 . s_2....

The goal of the tree search algorithm is to find the input binary sequence to the cascade of four biquad filters so that the corresponding outputs "match" the actual speech as close as possible. This work initially examined the following criteria:

- mean square error $(s-s)^2$
- magnitude. $|s-s|$.
- third power magnitude. $|s-s|^3$
- fourth power. $|s-s|^4$

Subjective listening to compressed speech for each of these criteria showed that the fourth power was slightly better than the fifth power. third power. and the mean squared error. Differences between these criteria were small. Finding the "best" binary sequence amounts to searching all possible paths in the representation tree and comparing each path output sequence with the actual speech sequence using some criterion as given above. Since the number of paths grows exponentially with the number of tree branches (depth of the tree) a more practical tree search approach is required. Also. because there was only small differences in the above criteria. we selected the man square error criterion for the remainder of this work.

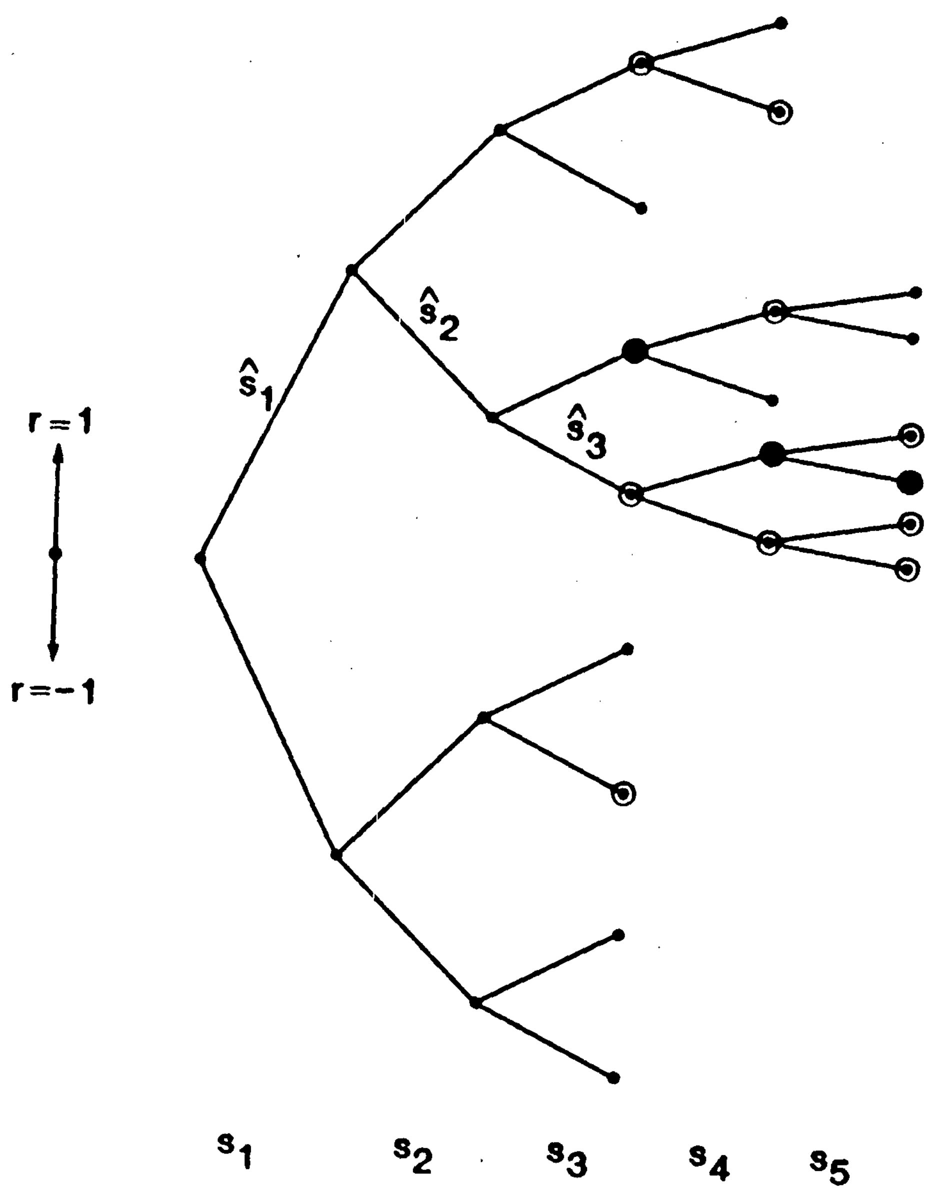
The (M.L) tree search algorithm is a suboptimum tree search algorithm that keeps track of only M survivor paths of L branchs in length at any given time. It also requires that all survivor paths originate from the same node L branches from the end. At a rate of 8.000 times a second in the (M.L) tree search algorithm, each of at most M surviving paths are extended by one branch forming temporarily up to 2M paths. The single best path for L+1 branches is computed and its initial leftmost branch path is chosen. Among the M-1 next best paths only those following this leftmost branch path is chosen as survivors along with the best path. Thus there is at least a delay of L sample times in the (M,L) tree search algorithm. Binary path decisions are made on the basis of examining at most M most likely candidate paths of length L at any given time.

Figure 3 illustrates an example of an M = 4 and L = 3 tree search algorithm. Beginning at the starting node all paths for L = 3 branches is considered. Only the top M = 4 of the 8 possible paths are selected. The end nodes of these surviving paths are circled with the single best path shown with a solid circle. Next only those surviving paths on the same half of the tree as the one best path is extended by one branch. Among these 6 paths only the top M = 4 are selected as survivors with the best path again shown with a solid circle on the end node. Now only those surviving paths sharing a common node L = 3 branches back with the best path are extended. This process results in a path sequence being selected.

The (M,L) tree search algorithm was investigated for values of M = 2,4,8,16,32 and values of L = 8,16,32. The resulting binary sequence $\underline{r}$ represented the binary sequence into the receiver's cascade of biquad filters that results in a output sequence that is "close" to the actual speech.

Up to this point we had a 9600 bits per second voice compression system, 8000 bits per second of excitation and 1600 bits per seconds for parameters. M = 8 and L = 32 was adequate but the compressed speech sounded rather noisy

Figure 3. Full Tree M=4, L=3



and there was occasional distortions.

To reduce the data rate to 5600 bits per second, the next step was to do a 2:1 compression of $\underline{r}$. We found that any compression algorithm that used more memory or large block codes tended to sound worse than those simpler shorter codes. In general, any compression algorithm has the same effect as transmission errors on the uncompressed sequence and short codes tended to "localize" this error.

Using various ad hoc simple short block codes for data compression, and reducing the parameter quantization to 800 bits per second, we found that the 4800 bps speech was much more noisy than the 9600 bps speech. This was expected. However, the resulting speech had a natural sounding quality to it compared to conventional 4800 bps LPC speech compression. The conclusion was that conventional LPC speech was relatively noise free but the speech itself had an "electronic accent." Our approach resulted in natural sounding speech but with considerable background noise. This was where we were at the end of the first three month period of this contract.

Punctured Tree Search Algorithms

During the first three months we discovered the now obvious result that better overall performance could be achieved if the (M,L) tree search took into account the impact of data compression. This led to the concept of punctured tree search algorithms that combine tree search and data compression into a single algorithm. This algorithm turns out to be the natural source coding dual to punctured convolutional codes used in channel coding [4]. Hence we call these algorithms punctured (M,L) tree search algorithms.

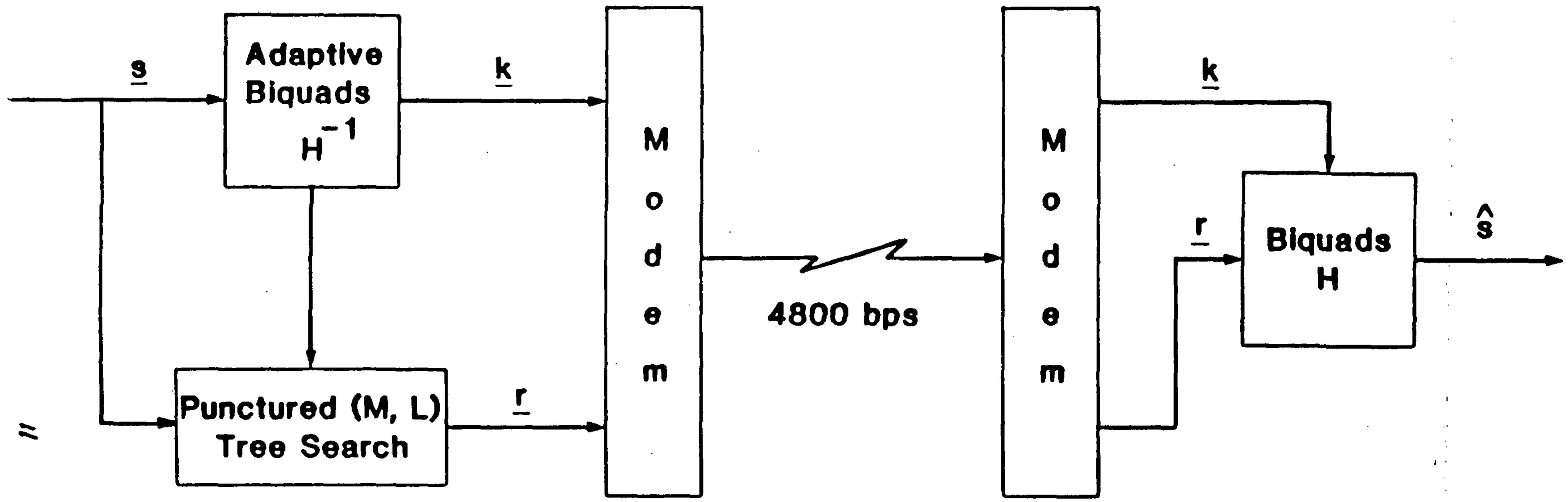
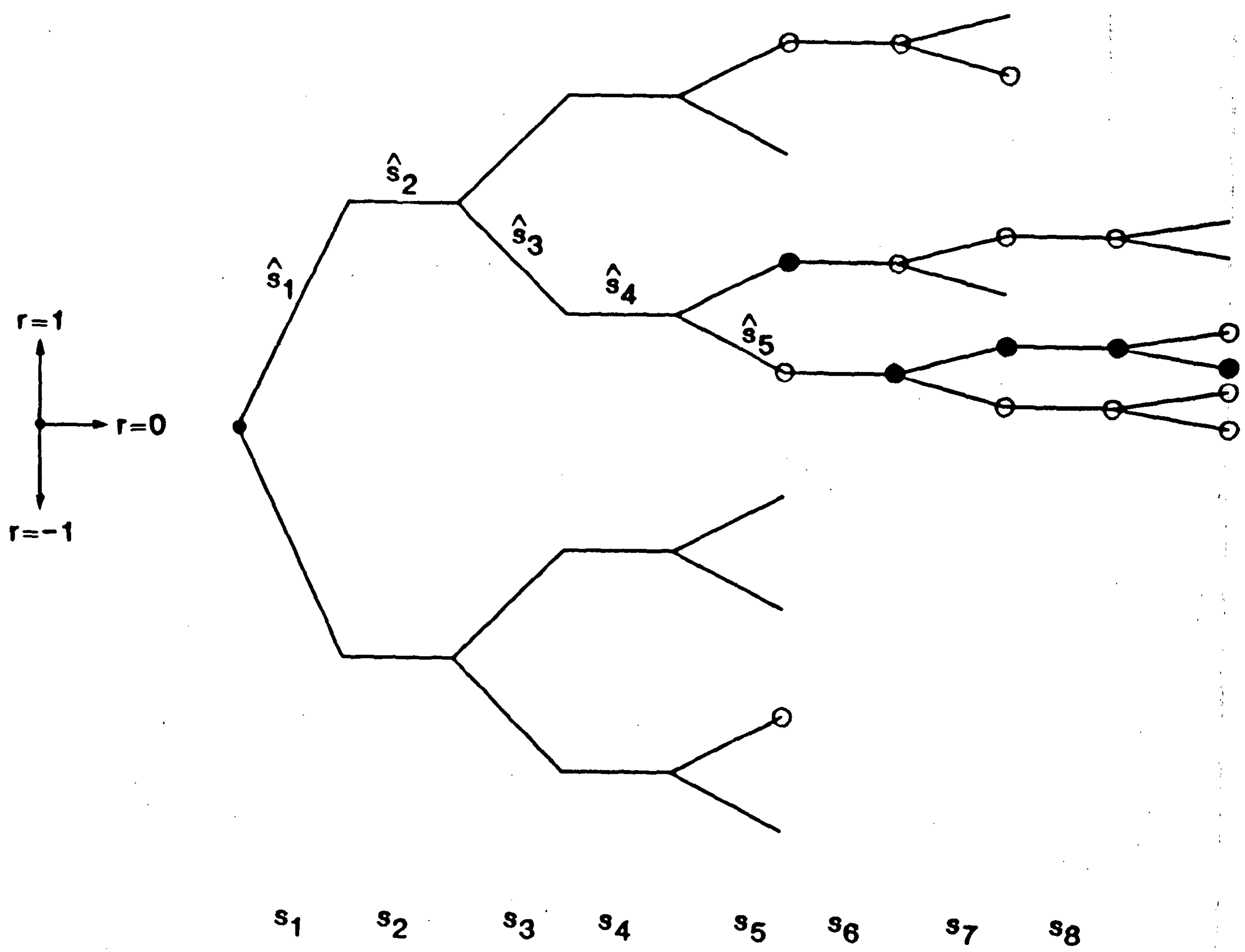


Figure 4. Voice Compression System

Figure 5. Punctured Tree M=4, L=5



The new system is sketched in Figure 4. To illustrate the punctured tree algorithm consider the example illustrated in Figure 5 where we choose M = 4 and L = 5. Here we assume initially that every other bit transmitted over the channel is now eliminated. This results in a 2:1 compression. The punctured tree algorithm takes this into account by constructing a new tree shown in Figure 5 where there is only one branch leaving each node corresponding to those cases where nothing enters the receiver's biquad filters. Essentially the same basic (M,L) algorithm is used except now the tree diagram that models the receiver's speech generation process is modified by the various data compression algorithms.

In this research various punctured tree search algorithms were examined. To achieve 4000 bps for the residual, we first tried eliminating half the transmitted bits in a binary transmitted sequence. This is essentially the type shown in Figure 5. Another example of 4000 bps is to send two bits (one of four amplitudes) one out of every four sample times. This results in a punctured tree with a repeated pattern of one branch leaving each node for three nodes followed by four branches leaving the next node.

Using the punctured tree algorithms, we obtained better compressed speech quality. There seemed, however, a limit on further improvement due to some instabilities of the adaptive algorithm for finding biquad filter coefficients.

Stabilizing the Adaptive Biquad Filger Algorithms

The adaptive biquad filter algorithm of Martin and Sun [1,2] has the form

$$K(n + 1) = K(n) - uS(n)d(n)$$

where

$$d(n) = \text{residual signal at time } n$$

$$S(n) = \frac{\delta\ d(n)}{\delta\ K}$$

$$u = \text{positive constant.}$$

This is a gradient tracking method. Here K(n) is a typical filter coefficient. There are two such coefficients for each of four biquads used in this study.

Occasionally we observed instabilities in the adaptive algorithm and tried modifications

$$K(n + 1) = K(n) - u\ \mathrm{sgn}[S(n)]\ d(n)$$

and

$$K(n + 1) = K(n) - u\ \mathrm{sgn}[S(n)d(n)].$$

The first had the advantage of small step size when d(n) is close to zero. While the second approach limits the maximum step size. The best compressed speech was obtained by using both of these in the form

$$K(n + 1) = \begin{cases} K(n) - u_1\mathrm{sgn}[S(n)d(n)] \text{ if } |d(n)| \geq \gamma \\ K(n) - u_2\mathrm{sgn}[S(n)]d(n) \text{ if } |d(n)| < \gamma \end{cases}$$

This requires careful selection of parameters u_1, u_2, and

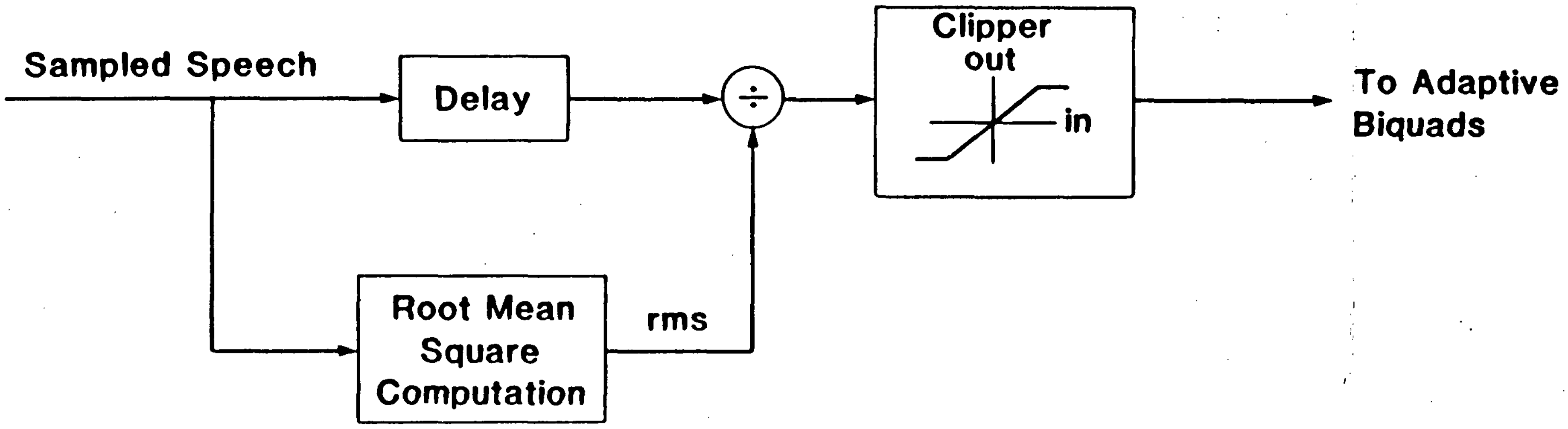


Figure 6. Normalization and Clipping

Unfortunately, good choices of parameters u_1, u_2, and varied with the speech samples used in our tests. In particular good choices of these parameters depended on the sampled speech power level. This led to the root mean square (rms) normalization scheme illustrated in Figure 6. In addition to this normalization we found that stability of the adaptive algorithm was established for all our sampled speech by clipping very large speech samples after the normalization. The clipping threshold introduced another parameter to be selected for the 4800 bps voice compression system.

Conclusions

The voice compression system shown in Figure 4 together with the rms normalization and clipping process shown in Figure 5 is the final 4800 bps voice compression system that evolved in this contract research. Our estimate of the required computation speeds indicate that this voice compression system can be implemented on a single IBM PC board using two Texas Instruments TMS 32010 digital signal processor chips. Also some general control processor chip such as a Motorola 68000 may be needed.

The simulation results at 4800 bps had very natural sounding speech compared to LPC synthesis techniques. It has, however, much more quantization noise. To test the robustness of the system we considered voice with background interference. Since this system is basically a waveform tracking approach, as expected, it is very robust to background interference. This may be the system's most important property.

This work represents an initial investigation of the application of two new concepts in voice compression:

1. Biquad Filters
2. Punctured Tree Search

In the 9 month contract period we feel that we have illustrated the practical feasibility of these new concepts and recommend that further work be conducted on this system. Specifically, we recommend developing a single board prototype implementation of the system for further testing. For the mobile satellite service applications where robustness is important, the 4800 bps voice compression system developed here appears to be a good candidate. More work, however, is required. Our contract research work was limited by slow processing where two seconds of speech took approximately 20 minutes of time on the time shared MASSCOMP computer. This makes it difficult to do more extensive testing of the many variations of the system.

V. References

[1]. K. Martin and M. T. Sun, "A State-Variable Digital Biquad," IEEE International Conference on Circuits and Systems, May 1984.

[2]. K. Martin and M.T. Sun, "Adaptive Filters Implemented by a Cascade of State-Variable Digital Biquads, "IEEE International Conference on Circuits and Systems, May 1984

[3]. J.B. Anderson and J.B. Bodie, 'Tree Coding of Speech," IEEE Trans. Inform. Theory, vol. IT-12, pp. 379-387, July 1975.

[4]. Y. Yasuda, Y. Hirata, K. Nakamura, and S. Otani, 'Development of Variable-Rate Viterbi Decoder and Its Performance Characteristics," Sixth International Conference on Digital Satellite Communications, Phoenix, Arizona, September 19-23, 1983.

PART II

REPORT ON SIMULATION

OF

LOW COST VOICE COMPRESSION FOR MOBILE DIGITAL RADIOS

1. INTRODUCTION

This report presents details of the computer simulations of the voice compression system. It is assumed that readers understand the basic performance characteristics of the digital biquad filter and the conventional (M,L) tree encoding scheme[1,2,3,4].

The simulations have been conducted on the MASSCOMP computer system, where program language C has been used. The voice compression system has two types of information to be transmitted through a noisy channel to a receiver; the residual signal, and parameters representing the biquad coefficients and two root-mean-square(r.m.s.) values. Based on the transmission rate for the information on the residual signal, the voice compression system is called sys-16k, sys-12k, sys-8k, or sys-4k, where the numerical digits denote the residual signal transmission rate. For example , sys-8k needs 8000 bits per second for the residual signal. For sys-16k, sys-12k, and sys-8k an additional 1600 bits per second was used for transmitting system parameters resulting in total data rates of 17.6kbps, 13.6k bps, and 9.6kbps respectively. For sys-4k, 1600 bps and 800 bps were used for system parameters. There was little difference in subjective speech quality between these two cases in sys-4k so that 800 bps was used in the final 4800 bps system.

We divide the voice compression system into five subsystems; speech source, input normalization, speech analysis, tree search algorithm, and speech reconstruction. The block diagram of the system is sketched in Figure 1.

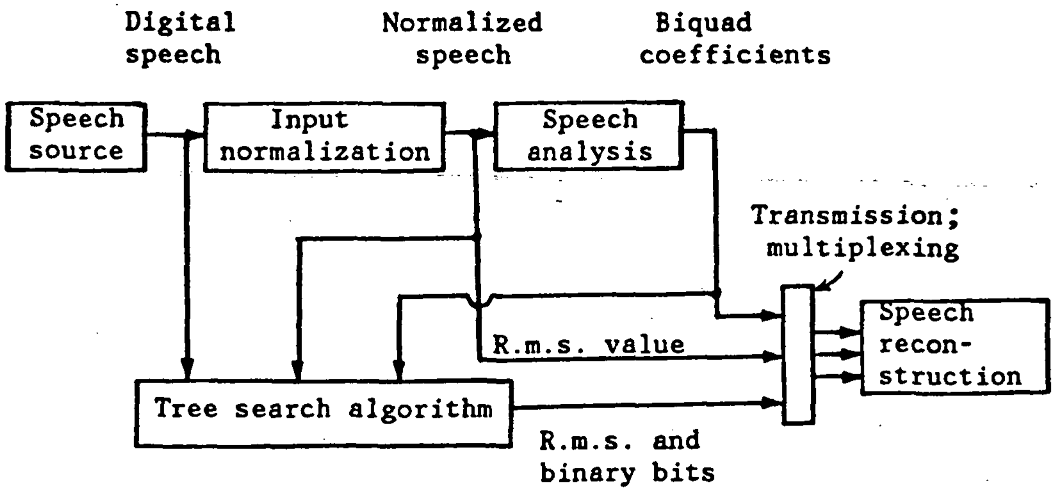


Figure 1: Block diagram of the system

The following sections examine the simulation behaviors of individual subsystems in detail. Signal processing is based on a frame, where the length of frame is usually 20ms. Thus, the time index n, denotes the n-th sample of the current frame.

2. SPEECH SOURCE

There are two different types of speech files in the MASSCOMP computer system. The first type is the original test set of 16-bit quantized speech sampled at 8000 times per second which was obtained from Professor Tom Barnwell of Georgia Tech. Since the A/D and D/A converters of the MASSCOMP can handle only 12-bit quantized samples, to convert this digital speech into an analog signal, the digital samples of this original test file must be divided by a number higher than 2^4 before entering it into the D/A converter. Appendix I describes this original test set of quantized speech used for most of this contract work.

The second type is the set of 12-bit quantized speech that we generated ourselves at various sampling rates. The generating process is illustrated in Figure 2.

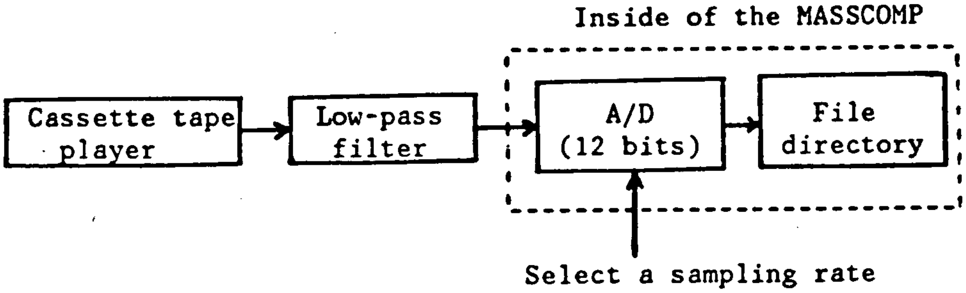


Figure 2: Generating process of a speech file

We first record the segment of a voice on a cassette tape, where the maximum number of samples for the MASSCOMP is 32000, i.e. 4 seconds at the sampling rate of 8000 per second. When we replay the segment through the low-pass filter into the MASSCOMP, we can choose a specific sampling rate by modifying an integer of the computer command statement. Since the A/D converter of the MASSCOMP was used, these speech files are 12-bit quantized samples.

The low-pass filter in Figure 2 is an active filter using switched capacitors. The bandwidth is contolled by the selection of the clock frequency. The clock oscillator operates the switched capacitors. For a specific sampling rate and a bandwidth, a very narrow-band tone is generated and added to the original segment of voice. The cause of this is due to subharmonic components of the clock signal. One way to reduce such a undesired noise is to change the clock frequency until the noise meets a desired level. Because of this undesirable tone due to our active filter we used a relatively wide front end bandwidth. Thus, the actual bandwidth we had for the second type of files is much wider than 4.4kHz , where the controllable minimum bandwidth of the low-pass filter is 4.4KHz.

If any amplitute of the signal out of the low-pass filter is greater than a voltage level of 2^{12}, the A/D converter changes it to zero instead of truncating to 2^{12}. This is also true for the negative amplitudes. Thus, a voice segment should be recorded on a cassette tape so that the amplitudes from the low-pass filter voltage range between -2^{12} and 2^{12}. Otherwise, the resulting quantized speech has large discontinuities. We took care of the above problems when we generated the speech files. The noise in these speech files is negligible.

3. INPUT NORMALIZATION

It has been observed that the adaptive estimator for biquad coefficients works well, when input amplitudes entering the inverse biquad filter(speech analysis filter) are less than a voltage level of about 1.5. Under this condition, a good value of the gain factor in the recursive update formula is u = 0.0625. If either the input voltage amplitude is much greater than 1.5 or u >> 0.0625, the voice compression system can become unstable, in which case our computer simulation stops. To eliminate instability and to achieve a better estimation of the adaptive biquad coefficients, we need to use normalization prior to the speech anlysis.

We compute the r.m.s. of input samples by $\sqrt{\Sigma\ s^2(n)\ /\ \mathrm{FRAME}}$, where FRAME = frame length , the summation is over the frame, and s(n) is the n-th input speech sample of the current frame. The modified r.m.s. of the frame with a dc-bias has the form

$$\mathrm{rms1} = \beta_1 \ \left(\sqrt{\Sigma\ s^2(n)/\ \mathrm{FRAME}} + \beta_2\right) \ . \qquad (1)$$

The purpose of β_2 is to avoid the case that rms1 is equal to or close to zero. Our choice is

$$\beta_1 = 2.0 \quad \text{and} \quad \beta_2 = 0.1 \qquad (2)$$

Different values of β_1 and β_2 do not make much of a difference in subjective speech quality, while the value of u should be chosen to obtain good quality. Thus, we fixed these values throughout the simulation, and optimized other system parameters.

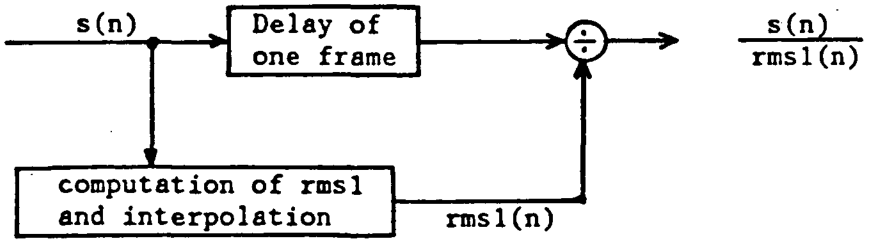


Figure 3: Block diagram of the normalization

The block diagram of the normalization is sketched in figure 3. Other more easily implemented forms of normalization were not examined here. Linear interpolation of rmsl is employed to avoid an abrupt change of envelop over the junction between two frames. Let rmsl(n) be the interpolated rmsl of the n-th component in the current frame. It is given by

$$\mathrm{rmsl}(n) = \mathrm{rmslp} + (n+1)\,\frac{(\mathrm{rmslc} - \mathrm{rmslp})}{\mathrm{FRAME}} \qquad (3)$$

where rmslc and rmslp are the r.m.s's of the current and the previous frames respectively. In the simulations, we implement this in the form of (4).

$$\mathrm{rmsl}(n) = \mathrm{rmsl}(\, n-1) + \Delta \qquad (4)$$

where $\Delta = (\mathrm{rmslc} - \mathrm{rmslp})/\ \mathrm{FRAME}$, and $n = 0, 1, \ldots, \mathrm{FRAME}-1$.

We compared two cases in quality ; with and without the interpolation. The case without the interpolation sounds like discontinious voice, while with interpolation there is no noticeable discontinuity. A simple graphic illustration is shown in figure 4, where FRAME = 10, and rmslp = 3 is assumed. Notice that the envelop without the interpolation has a different shape.

We also observed occasional instability in the voice compression simulations when employing the normalization. This means that there are still some of normalized voltage amplitudes that are larger than 1.5. To limit them to some level around 1.5, we add a clipping device

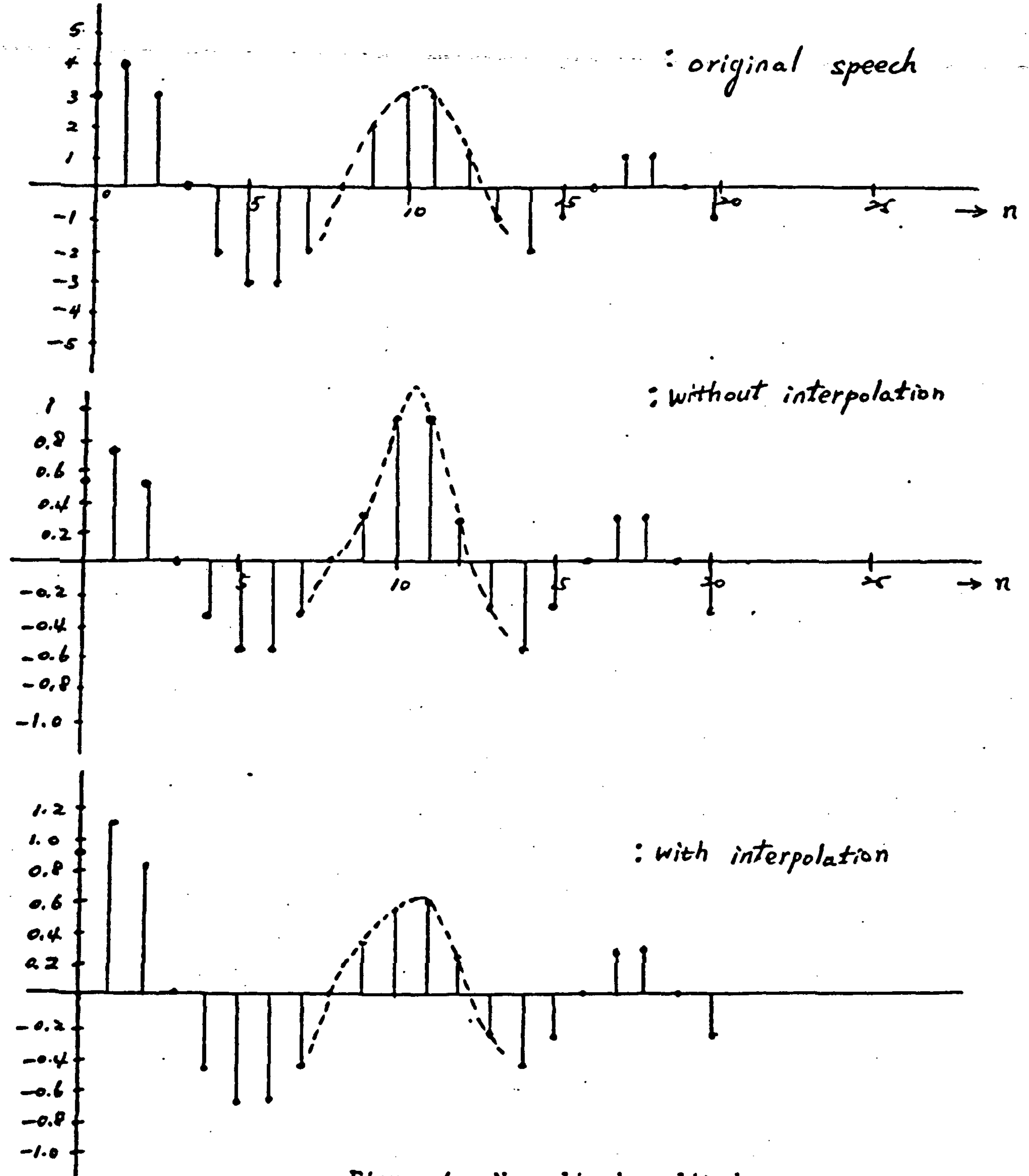


Figure 4: Normalized amplitudes

shown in figure 5. With this clipping we had no instability in any of our tests.

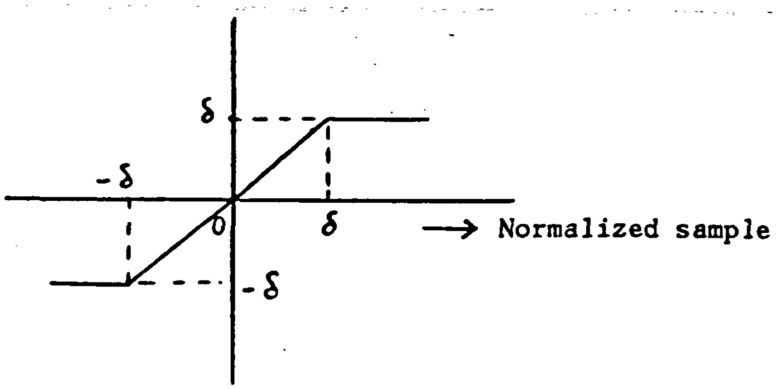
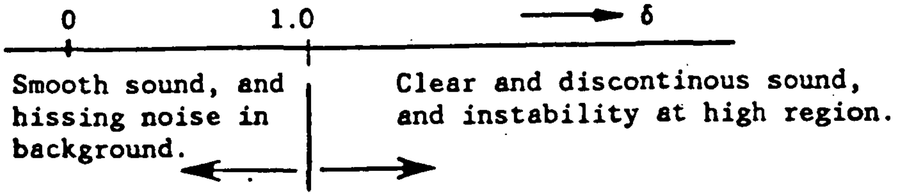


Figure 5: Clipping device in the normalization

Let $\bar{s}(n)$ be the final output from the normalization and clipping which is given by

$$\bar{s}(n) = \begin{cases} \delta & , \text{ if } \mid s(n) \,/\, rms1(n) \mid > \delta \\ s(n)/rms1(n) & , \text{ otherwise} \end{cases} \qquad (5)$$

As the value of δ varies, we have different voice qualities. One good choice of δ is around 1.0. The rough demonstration of voice quality with respect to δ is as follows.


0
1.0
→ δ
Smooth sound, and hissing noise in background.
Clear and discontinous sound, and instability at high region.

If we double β_1 , δ should beredyced by one half. However, both cases provide a similar voice quality.

4. SPEECH ANALYSIS

The inverse biquad filter analyzes a speech spectrum over 20 ms. The filter consists of four inverse biquads in cascade. The i-th inverse biquad estimates i-th formant frequency f_i and the sharpness Q, of its spectrum envelop around f_i. Finally it notches out the input spectrum in the sense of minimizing the residual power. Let k_1 and k_2 be the coefficients of the i-th biquad. The relationship between (f_i, Q) and (k_1, k_2) is roughly

$$f_i \cong k_1 f_s / 2\Pi \qquad (6)$$

where f_s is the sampling frequency[Hz].

$$Q \cong 1/k_2 \qquad (7)$$

The block diagram of the speech analysis system is sketched in figure 6. The transfer function of each inverse biquad is

$$H_i^{-1}(Z) = k_1^2 \, [1 - (2 - k_1 k_2 - k_1^2)Z^{-1} + (1 - k_1 k_2)Z^{-2}] \qquad (8)$$

where each biquad has a different pair (k_1,k_2).

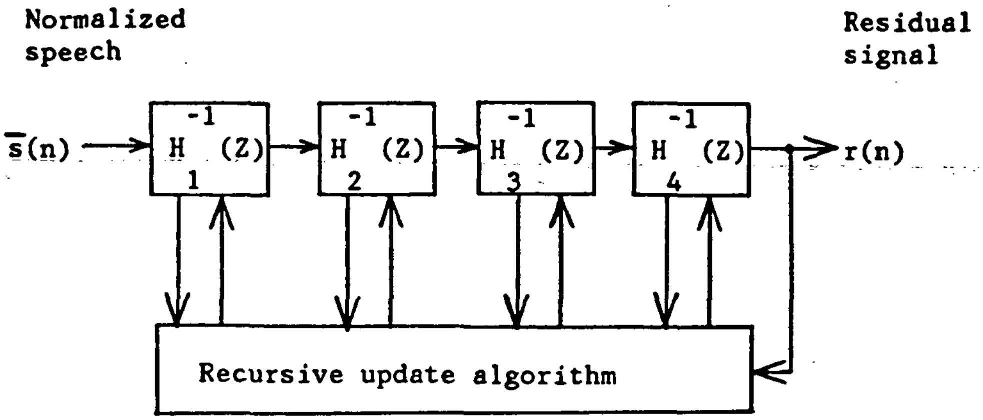


Figure 6: Block diagram of the inverse biquad filter

The main problems of this subsystem are how to build a recursive update algorithm to accurately estimate the coefficients k_1's, k_2's, and how to establish stability in the algorithm. It was observed that with a large value of u (>> 0.0625) the simulation program can stop due to instabilities.

4.1 Recursive update algorithm

A simplified gradient expression of the recursive update algorithm is

$$k_i(n+1) = k_i(n) - u\, s_i(n)\, r(n) \quad , \qquad i = 1,2 \qquad (9)$$

where r(n) is the output of the inverse biquad filter, and

$$s_i(n) = \frac{\partial\, r(n)}{\partial\, k_i(n)} \quad \text{(sensitivity term)} \qquad (10)$$

where $s_i(n)$ can be implemented by a second-order filter. The meaning of $2\, s_i(n)r(n)$ is , in fact, the slope of $r^2(n)$ with respect to $k_i(n)$, i.e.

$$\frac{\partial\, r(n)^2}{\partial\, k_i(n)} = 2\, \frac{\partial\, r(n)}{\partial\, k_i(n)}\, r(n) = 2\, s\, (n)\, r(n) \qquad (11)$$

We can control the tracking speed by the gain factor u. It also should be noticed that the update size $\Delta k_i(n) = k_i(n+1) - k_i(n)$, heavily depends upon the input power entering the inverse biquad filter. The gradient update formula is illustrated in Figure 7, where $k^*_i(n)$ is the minimizing coefficient.

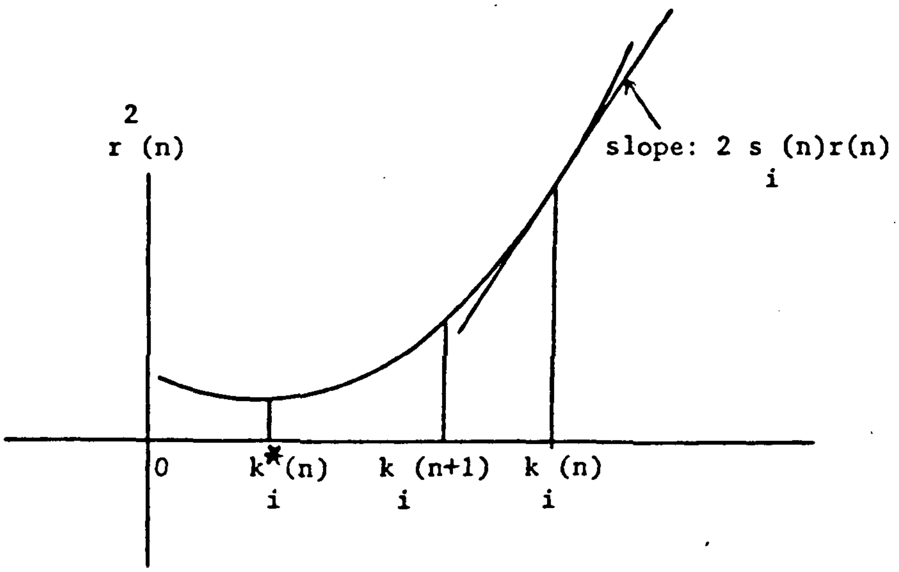


Figure 7: The gradient update algorithm

Recall that we already used a clipping device in the input normalization to avoid any high amplitudes. Even though the input levels are limited, the final residual signal r(n) is sometimes too large to keep a desired update size for a good estimation. To keep a robust update size, we employ a clipping device as shown in figure 8.

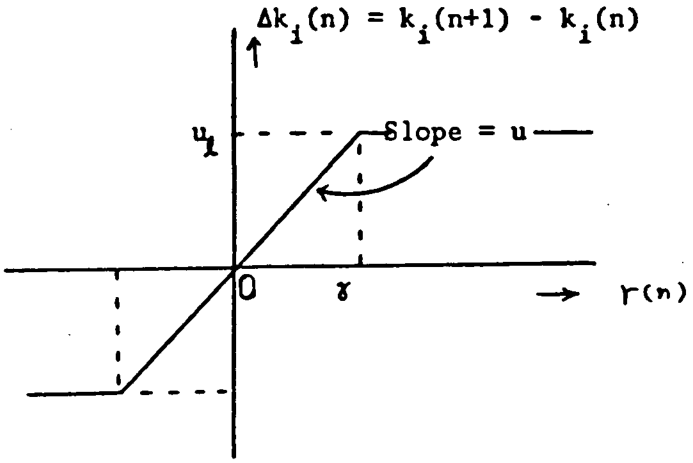
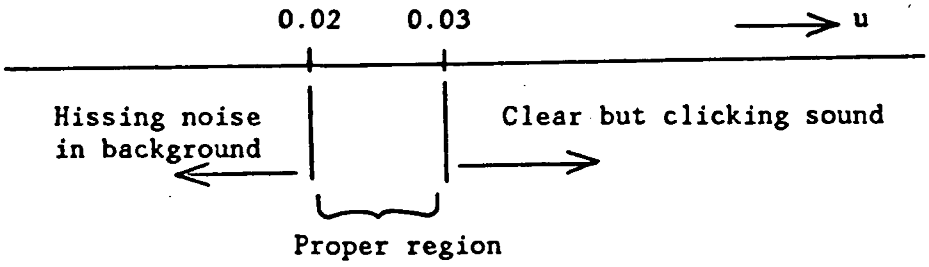


Figure 8: Clipping in the recursive update algorithm

Thus, our final version can be given by (12).

$$k_i(n+1) = \begin{cases} k_i(n) - u_\ell \text{ sign } [s_i(n)\ r(n)] & \text{, if } |r(n)| > \gamma \\ k_i(n) - u \text{ sign } [s_i(n)]\ r(n) & \text{, otherwise.} \end{cases} \tag{12}$$

A careful choice of u_ℓ and γ is required because they have an effect on quality and instablity. According to our tests, a good choice is $u_\ell = 0.018$ and γ is around 0.5, where $\delta = 1.0$ (see figure 5). Using these values, we have not yet found an unstable case, and we can obtain good quality compressed voice. However, quality varies as u changes. Figure 9 demonstrates a rough behavior of quality. Our simulations use a single value of u for all four biquads. We tested some different combinations, but they didn't make much of a difference in quality.



,where $\gamma = 0.5$ and $u_\ell = u \cdot \gamma$

Figure 9: Qulaity vs. u

It is necessary to compute the average of k_i over each frame, which is then transmitted over the channel. We use the average value

$$k_i = \sqrt{\frac{\Sigma\, k_i(n)}{\mathrm{FRAME}}} \qquad (13)$$

where the summation is over the frame. This requires a frame delay. averaged k_i is finally tested by a stability checker to determine whether or not the biquad $H_j(Z)$ (not $H_j(Z)^{-1}$) is stable. The next subsections consider the stability checker and the quantization process of the biquad coefficients.

4.2 Stability check of H(Z).

The transfer function of the biquad in the speech synthesis is

$$H_i(Z) = 1/ [1 - (2- k_1k_2 - k_1^2)Z^{-1} + (1 - k_1k_2)Z^{-2}] \qquad (14)$$

The necessary and sufficient condition for the stability of $H_i(Z)$ is

$$k_1 \, k_2 > 0 \quad \text{and} \quad \frac{k_1^2}{2} + k_1k_2 > 2 \qquad (15)$$

If the output ($k_1(n)$, $k_2(n)$) of the recursive update algorithm violates the constraint of (15), we make a modification. Instead of checking (15) directly, we use a look-up table. First we set the lower and upper bounds of formant frequencies of most practical utterences. Using the relationship between formant frequency and k_1 (see (7)), we can compute the corresponding bounds of k_1's. Next the bounds of k_2's are calculated by (15). The table 1 shows the bounds.

For example, suppose that we have $k_1(n) = 0.215$, $u = 0.025$, and $s_1(n) > 0$. Then $k_1(n+1) = k_1(n) - u \text{ sign}(s_1(n)) \, r(n) = 0.19$. Since $k_1(n+1) = 0.19$ is lower than the bound, we set $k_1(n+1) = 0.2$. It is obvious that the average of k_i's are in the stable region.

4.3 Quantization of biquad coefficients

Table 1. Lower and Upper bounds of k_1 and k_2

	1-st Biquad		2-nd biquad		3-rd biquad		4-th biquad	
	Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
k_1	0.2	0.8	0.2	1.15	0.6	1.64	0.6	1.87
k_2	0.1	1.312	0.05	1.16	0.01	0.398	0.01	0.13

Our voice compression system allocates 1350 bits per second (27 bits per 20 ms.) to transmit the 8 biquad coefficients. Appendix 2 gives a procedure for deriving an optimal bit allocation scheme for our system. Based on this anlysis, table 2 shows the bit allocation we use.

Table 2. Bit allocation for biquad coefficients

	1-st biquad	2-nd biquad	3-rd biquad	4-th biquad
k_1	4	4	4	4
k_2	3	3	3	2

We compared simulation results for two cases; with and without the quantization, where optimized parameters of both cases are different.

There was no apparent degradation for sys-12k, sys-8k and sys-4k. Since sys-16k provides very good quality, there was some noticable degradation caused by this quantization.

Three places in the voice compression system use the quantized coefficients; computation of rms2 (to be discussed), biquad filter of (M,L) tree search algorithm, and the speech reconstruction, where linear interploation is applied to the quantized biquad coefficients. The interpolation is

$$\bar{k}_i(n) = \bar{k}_i(-1) + (n+1)\,\frac{\bar{k}_i(\text{FRAME}) - \bar{k}_i(-1)}{\text{FRAME}} \qquad (16)$$

where

$\bar{k}_i(-1)$ = quantized coefficient for the previous frame,

$\bar{k}_i(\text{FRAME})$ = quantized coefficient for the current frame,

and n = 0, 1, 2, . . . , FRAME - 1. We implement the interpolation by using the same simple version as in the input normalization. When we did not apply the interpolation, we had some clicking sound.

Appendix 3 lists the quantized biquad coefficients of sys-4k. Notice that all the values meet the constraint for stability.

5. TREE SEARCH ALGORITHM

The (M,L) tree search algorithm searches through branches of the tree populated with outputs from the biquad filter. It searchs for the best input sequence of digits so that the corresponding outputs provide minimum distortion with respect to the original speech. The best sequence of input digits is encoded into a binary sequence. Then it is sent through a noisy channel. The transmission rate can be determined by both the encoding scheme and the populating method. The biquad filter here consists of four biquads in cascade.

Our simulation used $(s(n) - \hat{s}(n))^2$ as the distortion criterion, where s(i) is the original sample and $\hat{s}(i)$ is the corresponding output of the biquad filter. Other alternatives are $| s(n) - \hat{s}(n) |$, and $|s(n) - \hat{s}(n) |^{p}$, $p > 2$, etc. When we compared the squared error and the absolute error, we just felt that the squared error criterion is slight better.

If there is no restriction on the transmission rate, we can use enough bits to accurately represent residual samples from speech analysis, and send them to a receiver that can recover the original speech samples. Suppose that the sampling and transmission rates are both 8000 bit per second, where we represent each residual sample by either +1 or -1. In this case, we actually generate a constant c so that either +c or -c hits the biquad filter. We call the constant the exciting reference denoted by rms2. It is desired to generate rms2 so that the outputs of the biquad filter are close to the original ones.

5.1 Exciting reference and Multi-level assignment

One natural choice of rms2 is the r.m.s. of the residual signal from the inverse biquad filter. The computation of rms2 is illustrated in figure 10. It should be noticed that the inverse biquad is identical to the inverse of the biquad except the dc gain. Thus, there is a division on the residual output by $\Pi\, \bar{k}_{i1}^{2}(n)$, where the product is from i=1 to i=4, and $\bar{k}_{i1}(n)$ is the interpolated $k_1(n)$ of the i-th biquad.

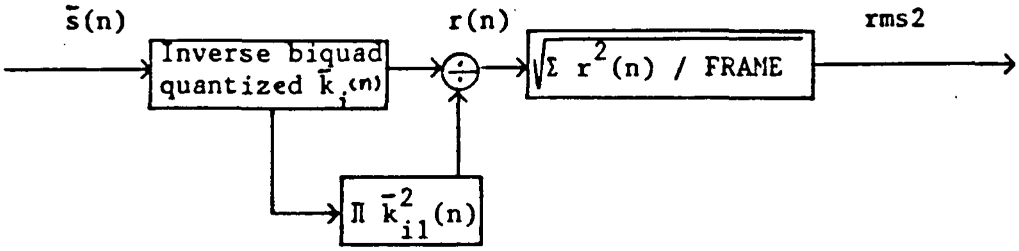
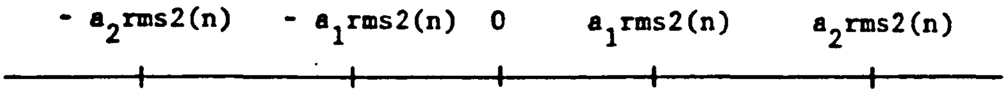
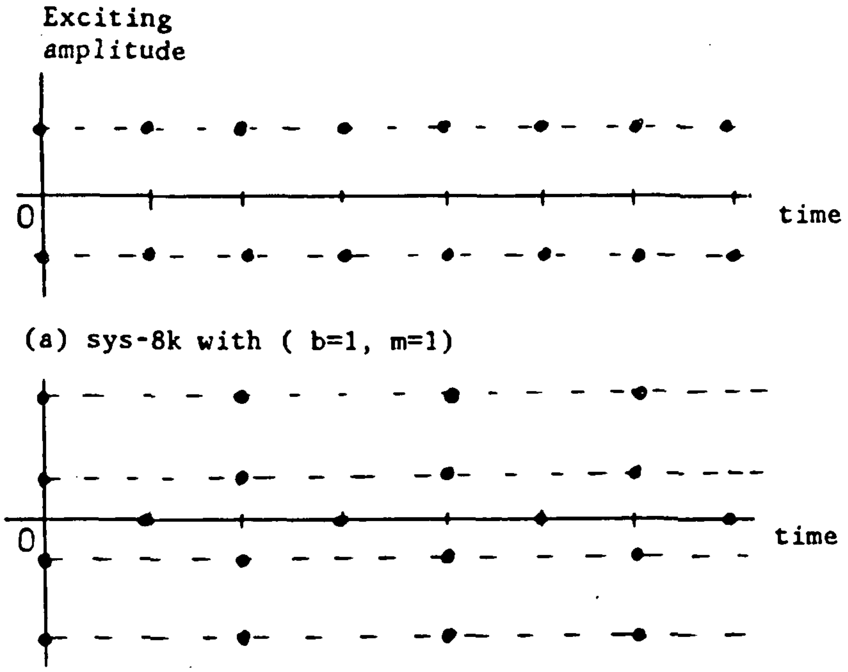

$\bar{s}(n)$
Inverse biquad
quantized $\bar{k}_i(n)$
r(n)
$\div$
$\sqrt{\Sigma\, r^2(n)\ /\ \text{FRAME}}$
rms2
$\Pi\, \bar{k}_{i1}^{2}(n)$

Figure 10: Computation of rms2

Let b be the number of bits representing a residual sample. If the sampling rate is 8000 times per second, then the transmission rate is 8000 b bits/sec. One of 2^b amplitudes can be transmitted in a binary form. For the case of b = 2, the actual amplitudes entering the biquad filter are denoted by



The choice of (a_1, a_2) is very important to produce a good quality. Suppose that b bits represent m samples. We then have for the residual signal transmission rate (b/m) x sampling rate. With a combination of b and m, we can build a variety of voice compression systems. For example, we can construct two different sys-8k's, where one has (b=1, m=1) and the other has (b=2, m=2) , with every other sample punctured out to zero. These two systems are demonstrated in Figure 11.



(a) sys-8k with (b=1, m=1)

(b) sys-8k with (b=2, m=2; every other one punctured)

Figure 11: Example of two different sys-8k's.

We compared three sys-8k's; (b=1,m=1), (b=2,m=2) where every other one is punctured out to zero, and (b=3,m=3) where two other samples

Table 3. Quality comparison

System discription	Quality
b=1, m=1	smooth and heavy hissing noise
b=2, m=2	clear and light electronic accent
b=3, m=3	clear and heavy electronic accent

are punctured out to zeros. The rough quality judgement is shown in table 3. The quality between (b=1,m=1) and (b=2,m=2) has a different aspect. It is not easy to conclude which one is better. The (b=2,m=2) case seems to be good for specifically male voices, while the other is good for female voices. When we take 80 samples as the frame length instead of 160 samples, the (b=2,m=2) provides better overall quality for female and male voices. Thus, we selected the (b=2,m=2) and tried to optimize its system parameters, where the frame length is 160. We ran several different utterances to find proper values of multi-levels for sys-8k with (b=2,m=2). Figure 12 illustrates the effect of a_1 and a_2 on quality.

For sys-4k, we tried a generalization of the punctured system where, instead of eliminating every other bit by puncturing, we replace a short block sequence by another block sequence. This is essentially a simple block compression scheme where punctured systems are special cases. Our tests have shown that the use of simple block compression provides better quality than punctured systems, which have heavy electronic accent at 4kbps residual data rates. The block com-

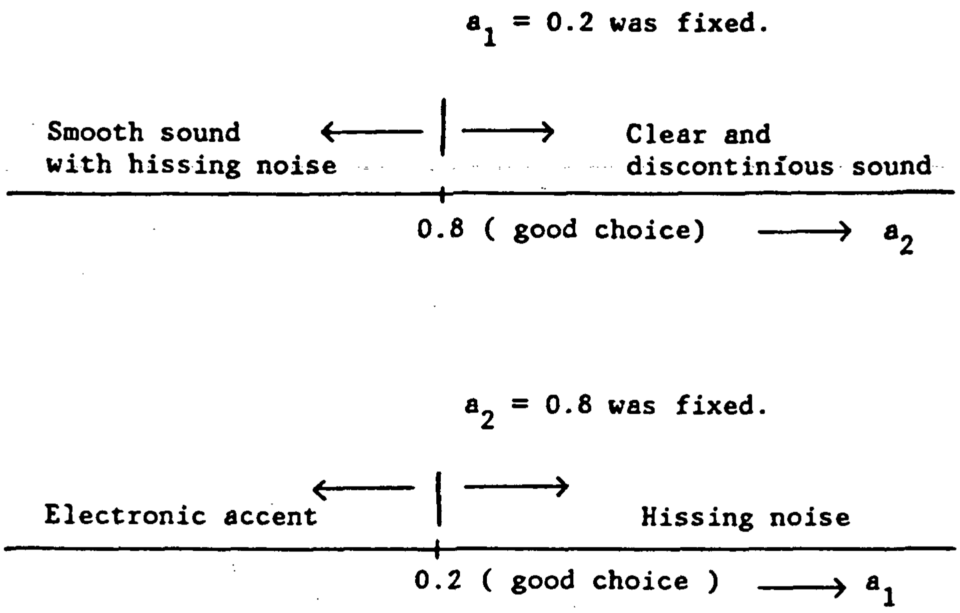


Figure 12: Effect of a_1 and a_2 of sys-8k

pression can be implemented by generating a code. A good code we found is shown in table 4, where the block length is 4 bits. Since 2 bits represent 4 samples, the transmission rate for the residual samples is 4000 bits per second. The tree search was done taking into account this block compression.

Except sys-4k, we employed only the punctured scheme for our voice compression systems. For sys-12k, we have b=3 and m=2, where every other sample is set to zero. For sys-16k, we have b=2 and m=1. Since sys-16k with (b=2,m=1) provided a good quality, we did not try any other combination of b and m. The choice of parameters of all the system are summaried in section 7.

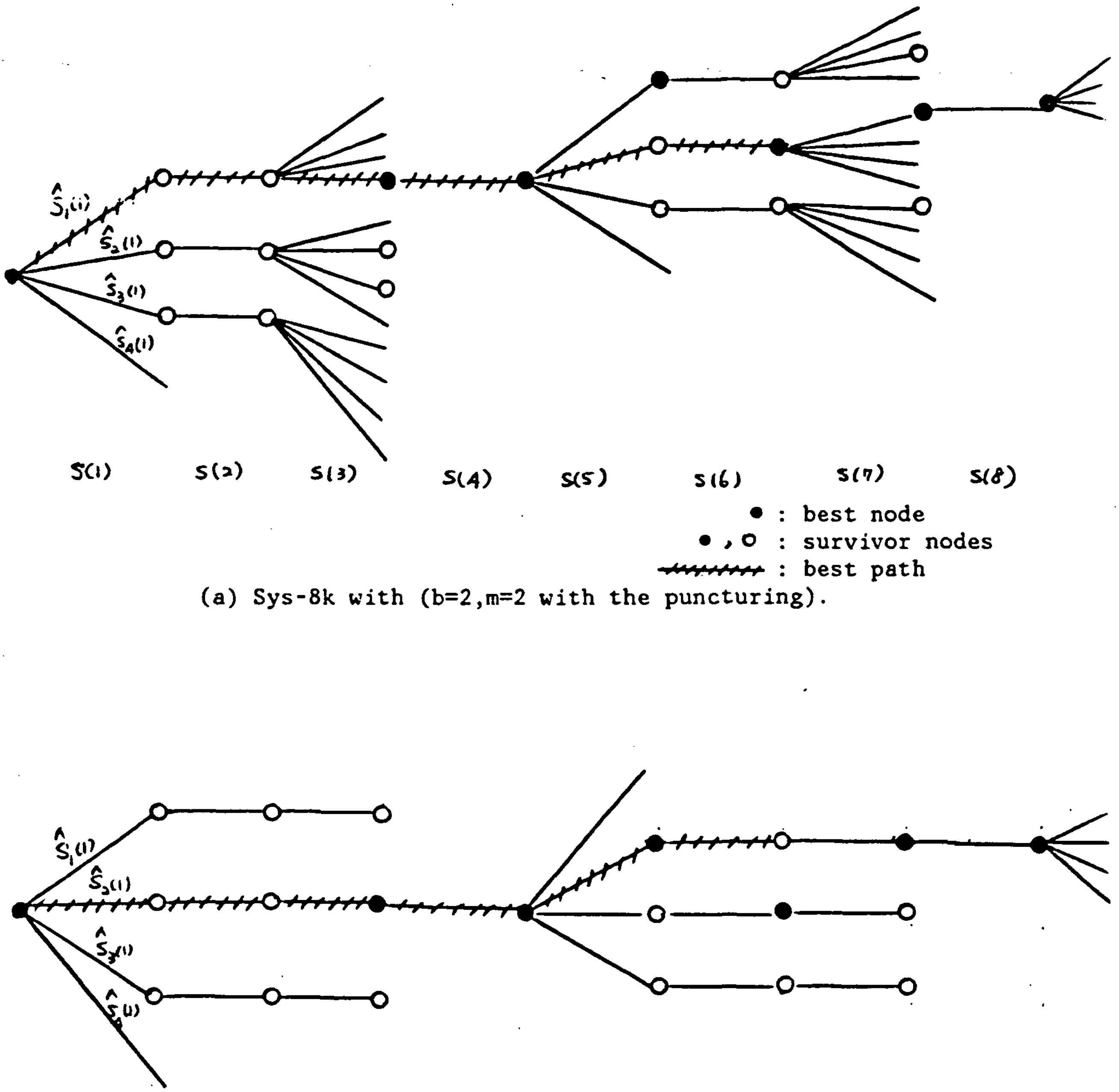
5.3 Effect of M and L

Table 4. Code for sys-4k ($a_1 = 0.3$, $a_2 = 0.65$)

Codeword	n=1	n=2	n=3	n=4
0 0	$a_2 \cdot rms2(n)$	0	$a_2 \cdot rms2(n)$	0
0 1	0	$a_2 \cdot rms2(n)$	0	$-a_1 \cdot rms2(n)$
1 0	0	$-a_2 \cdot rms2(n)$	0	$a_1 \cdot rms2(n)$
1 1	$-a_2 \cdot rms2(n)$	0	$-a_2 \cdot rms2(n)$	0

The (M,L) tree search algorithm keeps track of only M best paths in the populated tree. The decision of a best branch is made on a previous one of L branchs in depth from the current node having the smallest accumulated error. The number of extension branches at each survivor node is 2^b. For a punctured-out branch, just one branch is populated whose output is corresponding to the input value of zero. At each sample, we compute the accumulated errors of all extended nodes, and select the best M nodes. After making a decision of the best branch, we eliminate any of M nodes which does not have the same root as the best current node does of L branchs in depth. Thus, there are at most M survivor nodes. An example of (M=3,L=3) tree search algorithm for sys-8k and sys-4k are illustrated in figure 13.

Search time and voice quality depend upon M not L. The MASSCOMP computer system takes around 20 minitues a simulation of sys-8k for M



(a) Sys-8k with (b=2,m=2 with the puncturing).

(b) Sys-4k with the block compression of table 4.

Figure 13: Example of (M=3,L=3) tree search algorithm

= 7 and L = 32, where a 2 second utterance is tested. The simulation time increases exponentially with M. Differences in quality between M = 3 and M = 5 seem much larger than that between M = 7 and M = 9. M = 7 , however, provides good quality for sys-16k, sys-12k and sys-8k. As we can see in figure 13, sys-4k having M = 7 takes a much shorter time than other systems having the same M. Thus, we took M = 9 for sys-4k.

L represents the decision depth in the tree search. The simulation time actually does not depend upon L. L has an effect on smoothness in quality. According to our tests, the large value of L gives more smoothness but not much. The proper choice is either 16 or 32 for the sampling rate of 8000/sec. L = 32 or L = 64 might be good for the sampling rate of 16000/sec. We take L = 32 for all the systems.

6. SPEECH RECONSTRUCTION

The corresponding input sequence to the best outputs in the tree search is transmitted with biquad coefficients to a receiver. Copying the same process as used in the tree search algorithm, the receiver can reproduce the sequence of the best outputs. Since the D/A converter of the MASSCOMP is good for 12-bit quantized samples, we check the amplitudes of the final outputs by using a clipping device. It is shown in figure 14.

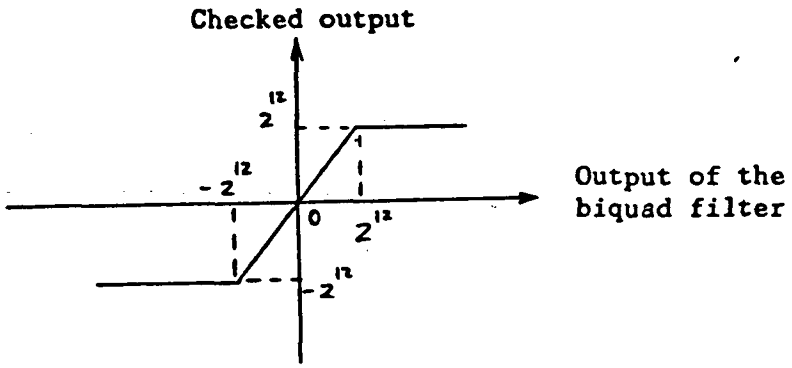


Figure 14: Amplitude checker in speech construction

7. SUMMARY AND DISCUSSION

7.1 Summary

The voice compression system has been divided into 5 subsystems; speech source, input normalization, speech analysis, tree search algorithm, and speech reconstruction. These individual subsystems have been investigated and optimization of parameters have been done.

Table 5 lists the main symbols used in this report. Table 6 summaries the choice of parameters which seems to be best for the speech files we used in our optimization. The block diagram of the simulation is sketched in figure 15. To understand the tracking behavior of outputs in the time domain, we drew the tracking curves of the 45-th frame of one of the speech files ("/usr/ee/moon/speech/sp1"). They are shown in Figure 16.

We recorded the simulated voice compression results on a cassette tape. There are 8 different types of utterances on the tape. The recording procedure and the tested contents on the tape are described in part III. The quantization of the biquad coefficients given by Table 2 and Appendix 2 has been applied to all the systems except sys-16k. Since sys-16k provides very good quality, the quantization causes a noticable degradation. For other systems, it is difficult to recognize any degradation due to the quantization.

Roughly speaking, the input normalization works well for both weak and strong voices, where the difference in power can be larger than 20 dB.

7.2 Discussions

For practical usuage, we have to test many types of utterences (specifically different pitch periods) under real situations in order to take a robustic choice of parameters. If there are several locally optimal parameter sets, we can implement an adaptive selection of parameter sets on a hardware product. For example, the set for a very clear background environment is different from that for a very heavy noise environment. Even though it seems that much improvement is not expected by changing of parameter values, they must be carefully selected.

If we can further encode the residual signal power distribution, there might be an improvement in quality. In our system, we sent one value of rms2 per frame. The technique which the system APC-4[5] uses might be useful here. It also seems that vector quantization is a useful tool to transmit the distribution with fewer bits. However, we tested short frame lengths, e.g. 20, 40, 80 samples and found that there was no big improvement, but we could feel a difference. Specifically, sys-8k was sensitive to the frame length.

The RELP system is known to provide good quality at higher data rates. If there is some way to combine the RELP system with vector quantization or with the tree search algoritm, it might be a good candidate. However, there is no specific idea for this combination right now.

Suppose we consider our system with a conventional LPC filter or a lattice filter instead of the biquads. Based on some preliminary tests we expect to have a similar result in quality and complexity. The biquad has a kind of pre-emphasis/de-emphasis perceptual weighting in it, but we can not apply to the biquad the usual noise-shaping technique, which is used in most of low-rate speech compression systems[6].

For sys-4k, we think that the system studied here works well. The use of both the block data compression and the puncturing scheme seems to work well. When we apply the block data compression to sys-8k, we did not notice any difference.

For sys-8k, the frame length of 80 gives a better sound (much better in some sense) than the length of 160. It is not true for other systems. Further investigation of this problem is recommended.

For sys-16k, the quantization on the biquad coefficients causes noticeable degradation. One way to reduce this loss is to rearrange the upper and lower bounds of k_1's and k_2's in table 1 so that we have a small quantization step-size.

The (M,L) tree search is the most time consumming part in our system. An efficient device (not the brute-force method) of the searching process will be a helpful for implementing a real-time system.

To achieve less complexity with the same quality, we might use the rms2 of the residual signal of the speech analysis rather than adding the filtering process for the rms2 computation in the tree search algorithm (see figure 16). A modification of parameter values will be needed.

Table 5. Symbol list

Symbol	Discription	Remarks
$s(n)$	Original speech sample	$0 \leqslant n < FRAME$
rms1	R.m.s. of $s(n)$	
rms1(n)	n-th interpolated rms1	
$\bar{s}(n)$	Normalized sample of $s(n)$	
$H_j(Z)$	Transfer function of j-th biquad	$1 \leqslant j \leqslant 4$
$H(Z)$	$H_1(Z)H_2(Z)H_3(Z)H_4(Z)$	
$r(n)$	Residual signal of $H^{-1}(Z)$	
$k_i(n)$	Coefficients of a biquad	$i = 1, 2$
k_i	Averaged $k_i(n)$ over a frame	
$\bar{k}_i(n)$	n-th interpolated k_i	
$s_i(n)$	Biquad sensitivity w.r.t. $k_i(n)$	
u	Gain factor in the update formula	
γ	Clipping threshold of $r(n)$	
rms2	R.m.s of the residual signal	
rms2(n)	n-th interpolated rms2	
b	bits representing a residual	
a_i	Exciting level	$1 \leqslant i \leqslant 2^b$

Table 6. System parameters

System	a_4	a_3	a_2	a_1	u	δ	γ	M	L
Sys-4k#	-	-	0.65	0.3	0.02	1.0	0.5	9	32
Sys-8k	-	-	0.8	0.2	0.02	1.0	0.5	7	32
Sys-12k	1.2	0.8	0.4	0.2	0.025	1.2	0.6	7	32
Sys-16k	-	-	0.9	0.2	0.025	1.2	0.6	5	32

# : the look-up table is shown in the table 4.

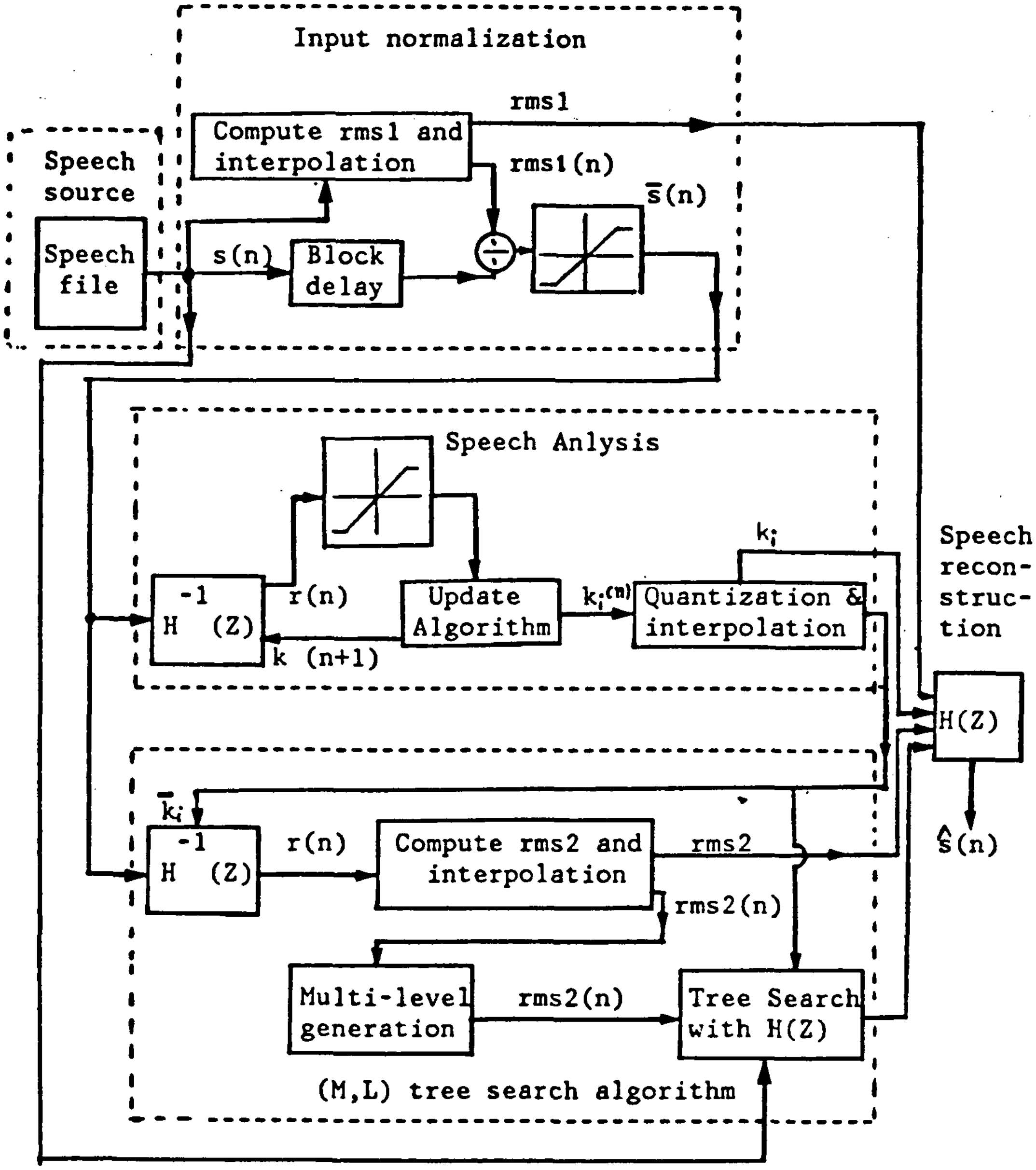


Figure 15: Block diagram of the simulation

Figure 16: Tracking curves

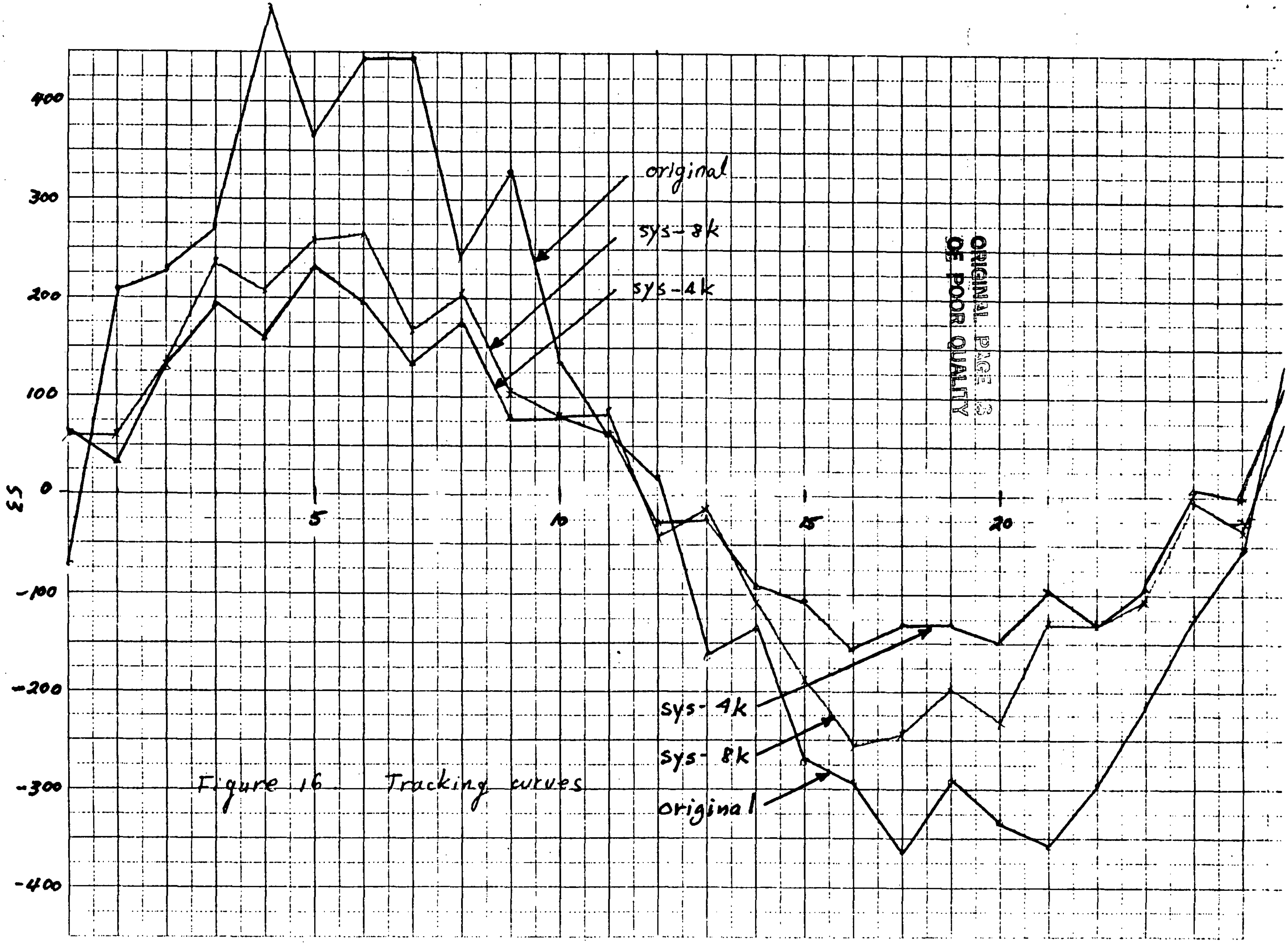


Figure 16 Tracking curves

REFERENCES

1) K. Martin and M. Sun, "A State Variable Digital Biquad," IEEE International Conference on Circuits and Systems, May 1984.

2) K. Martin and M. Sun, "Adaptive Filters Implemented by a Cascade of State Variable Digital Biquads," IEEE International Conference on Circuits and Systems, May 1984.

3) J. B. Anderson and J. B. Bodie, " Tree Coding of Speech," IEEE Trans. on Inform. Theory, Vol. IT-12, pp. 379 - 387, July 1975.

4) J. K. Omura, " Second Technical Progress Report on Low Cost Voice Compression for Mobile Digital Radios", Report to JPL, March, 1985.

5) The government Standard Adaptive Predictive Coding Algorithm: APC - 04.

6) B. Atal, "Predictive Coding of Speech at Low Bit Rates," IEEE Trans on Comm., Vol. COM-30, No. 4, pp. 600 - 614, April, 1982.

Appendix 1: Description of Speech Files

This appendix describes a set of files of six speech utterances and their pitch estimates generated by Professor T. Barnwell of Georgia Tech. The MASSCOMP computer stores the utterances and the pitch estimates under the file directory of /usr/spc_smp/, where the utterances are labeled by S1, S2, ..., S6, and the pitch estimates are labeled by PP1, PP2, ..., PP6.

Speech Data Base

A set of files of speech utterances is labeled S1, S2, ..., S6. The files contain 24,576 samples of 12-bit samples taken at a sampling rate of 8000 samples/sec. Each 12-bit sample is stored in a 16-bit integer word. Waveform plots of these utterances are attached.

The files PP1, PP2, ..., PP6 contain accurate estimates of pitch for files S1, S2, ..., S6 respectively. The estimates are obtained every 10 msec, i.e., every 80 samples of the waveform. The 307 pitch estimates are the first 307 numbers in the file. The remaining numbers are zero.

The numbers in the pitch files are the period of the speech waveform in samples where the sampling rate is 8000 samples/sec. A zero pitch period indicates unvoiced speech. Plots and listings of the pitch files are attached.

Catalog of Utterances

S1: "The pipe began to rust while new" (female speaker)

S2: "Thieves who rob friends deserve jail" (male speaker)

S3: "Add the sum to the product of these three" (female speaker)

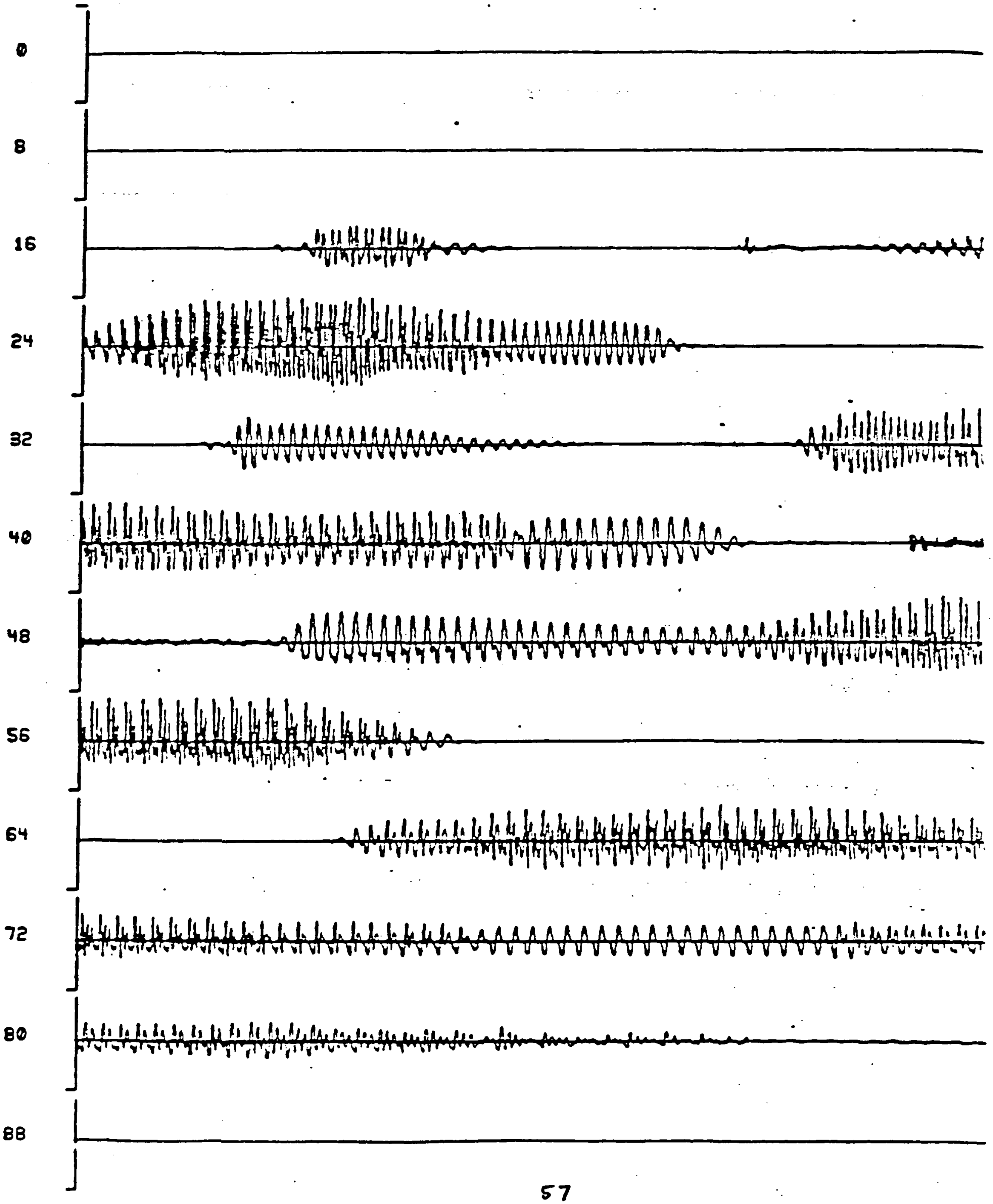
S4: "Open the crate but don't break the glass" (male speaker)

S5: "Oak is strong and also gives shade" (male speaker)

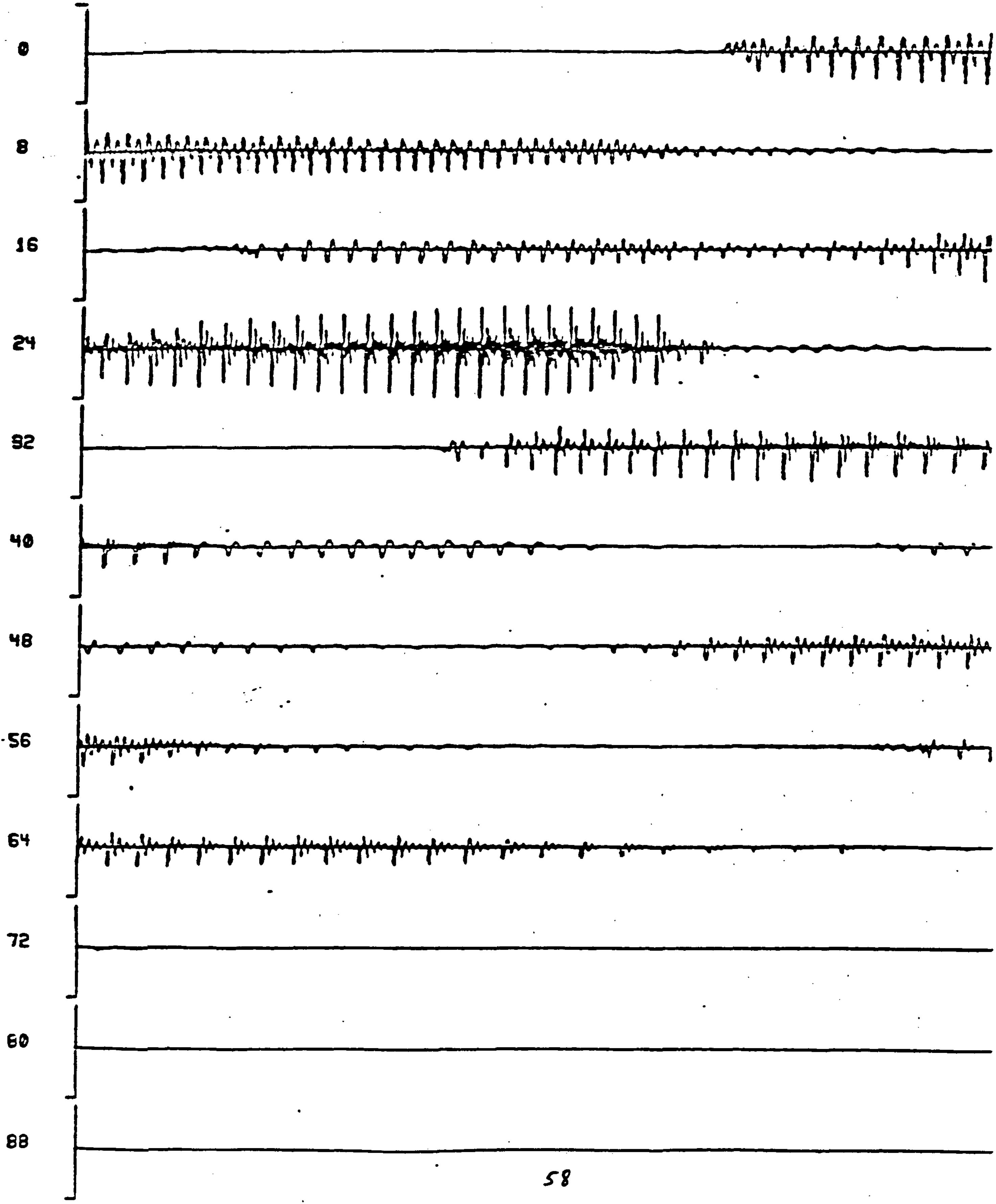
S6: "Cats and dogs each hate the other" (male speaker)

S1 THE PIPE BEGAN TO RUST WHILE NEW.

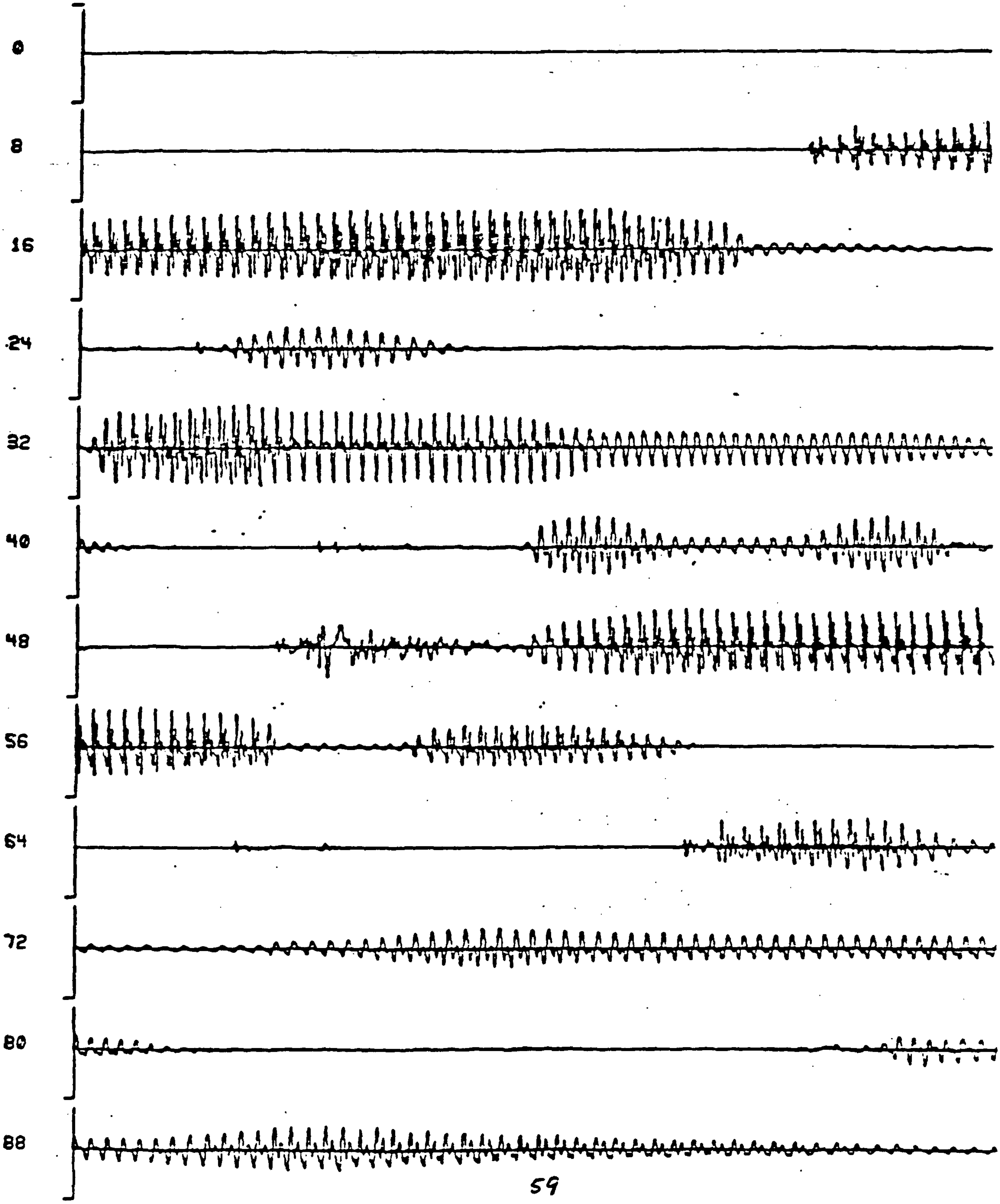
ORIGINAL PAGE IS OF POOR QUALITY



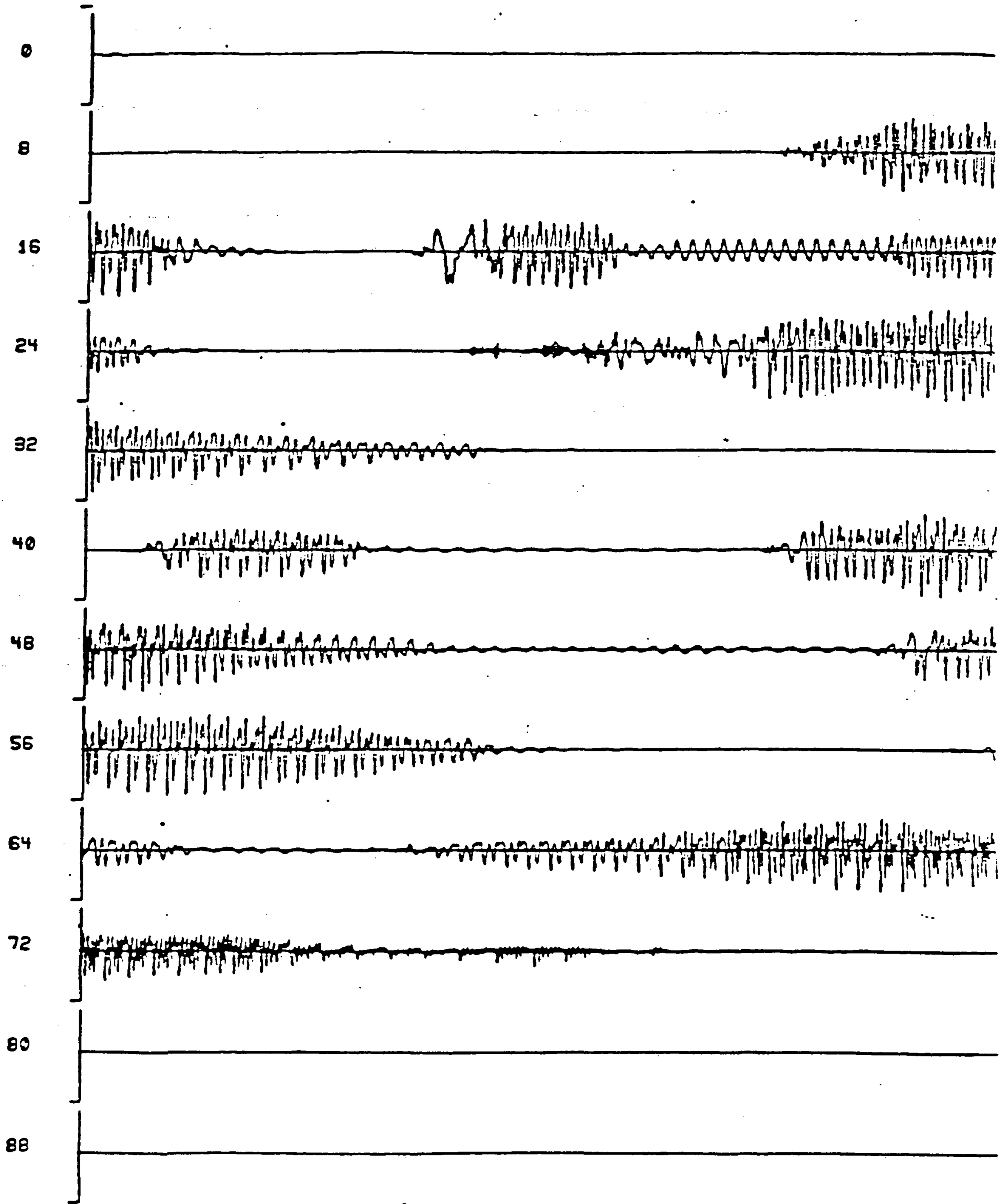
S2 THIEVES WHO ROB FRIENDS DESERVE JAIL.



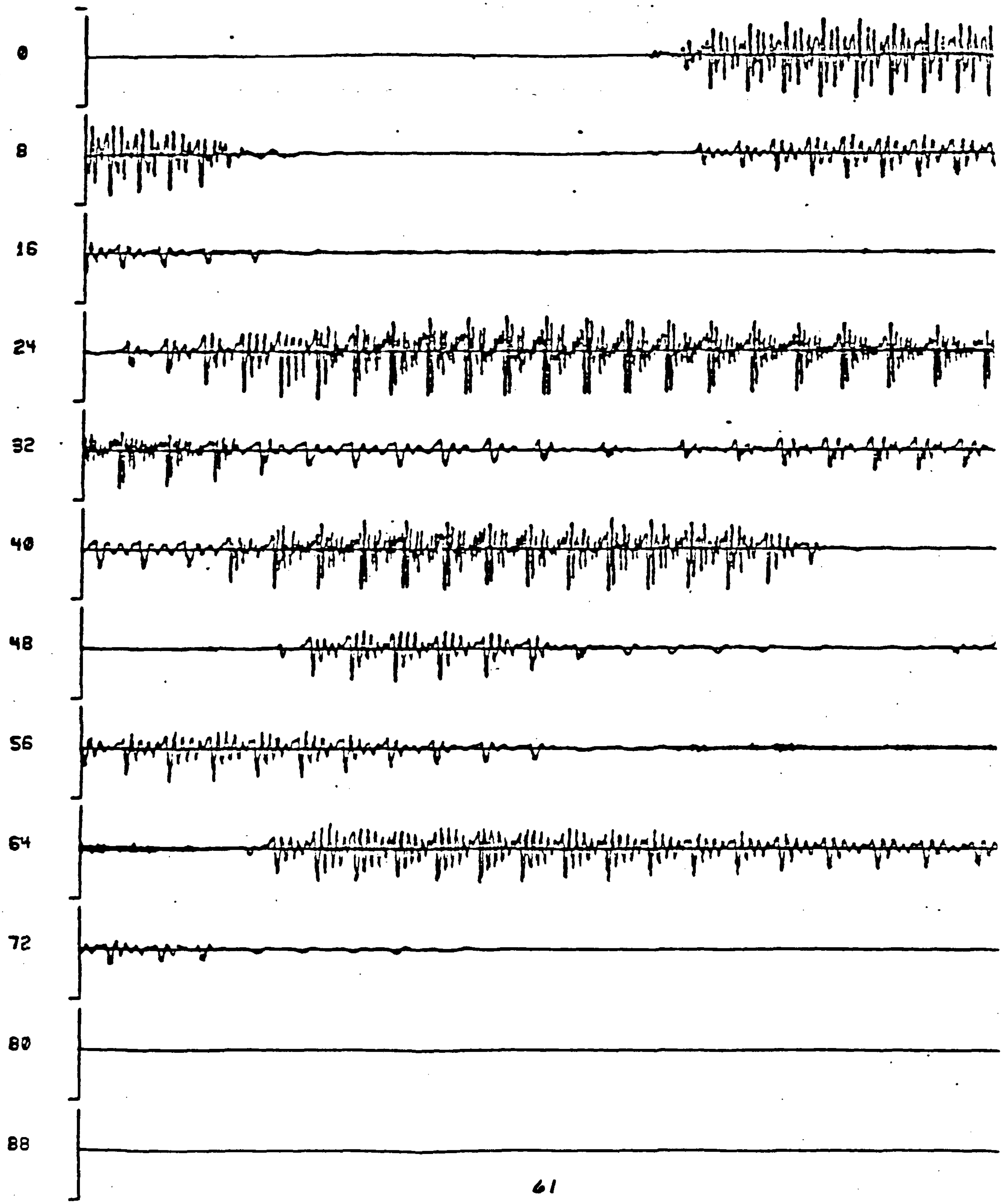
S3 ADD THE SUM TO THE PRODUCT OF THESE THREE.



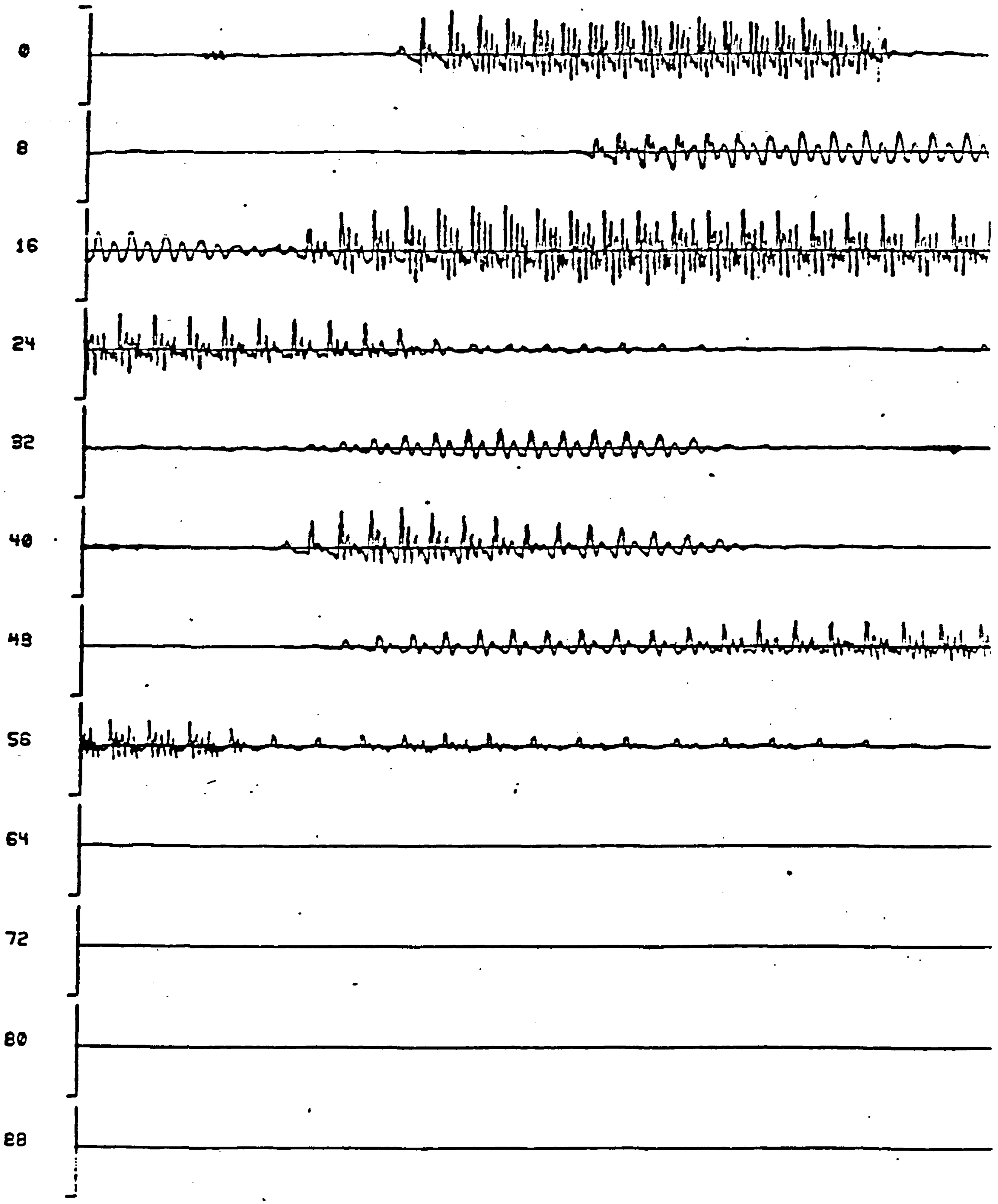
S4 OPEN THE CRATE BUT DON'T BREAK THE GLASS.

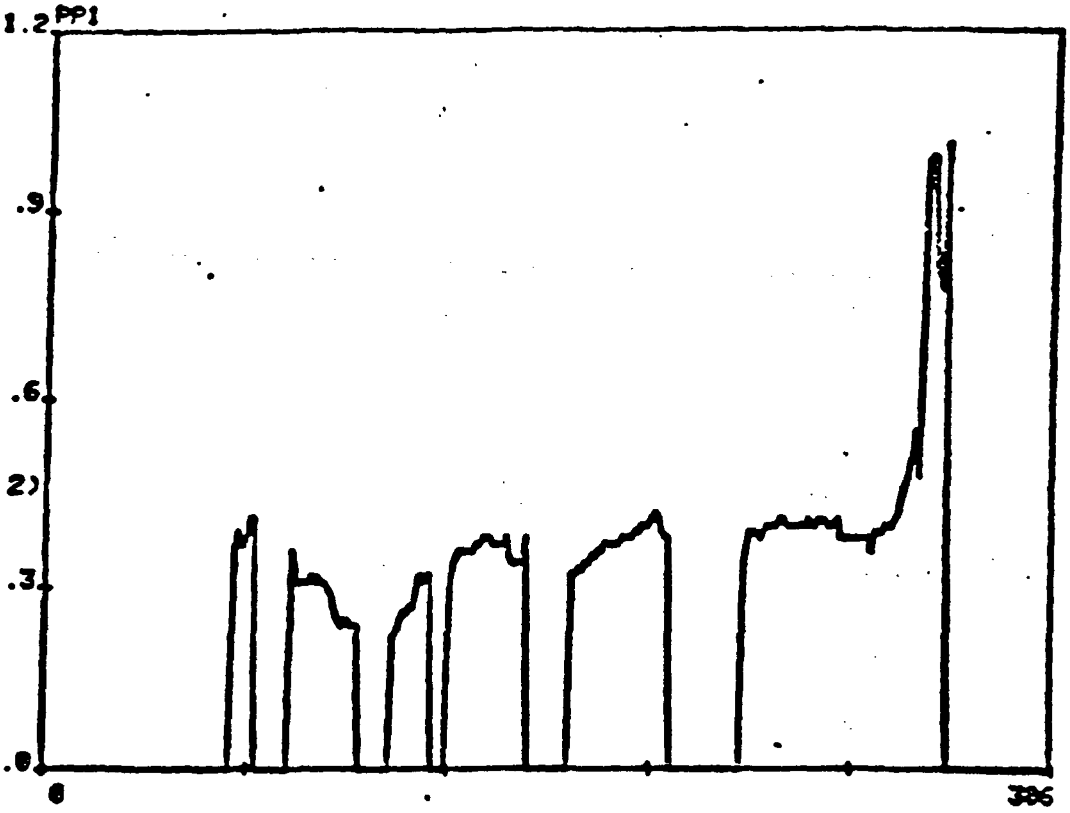


SS OAK IS STRONG AND ALSO GIVES SHADE.

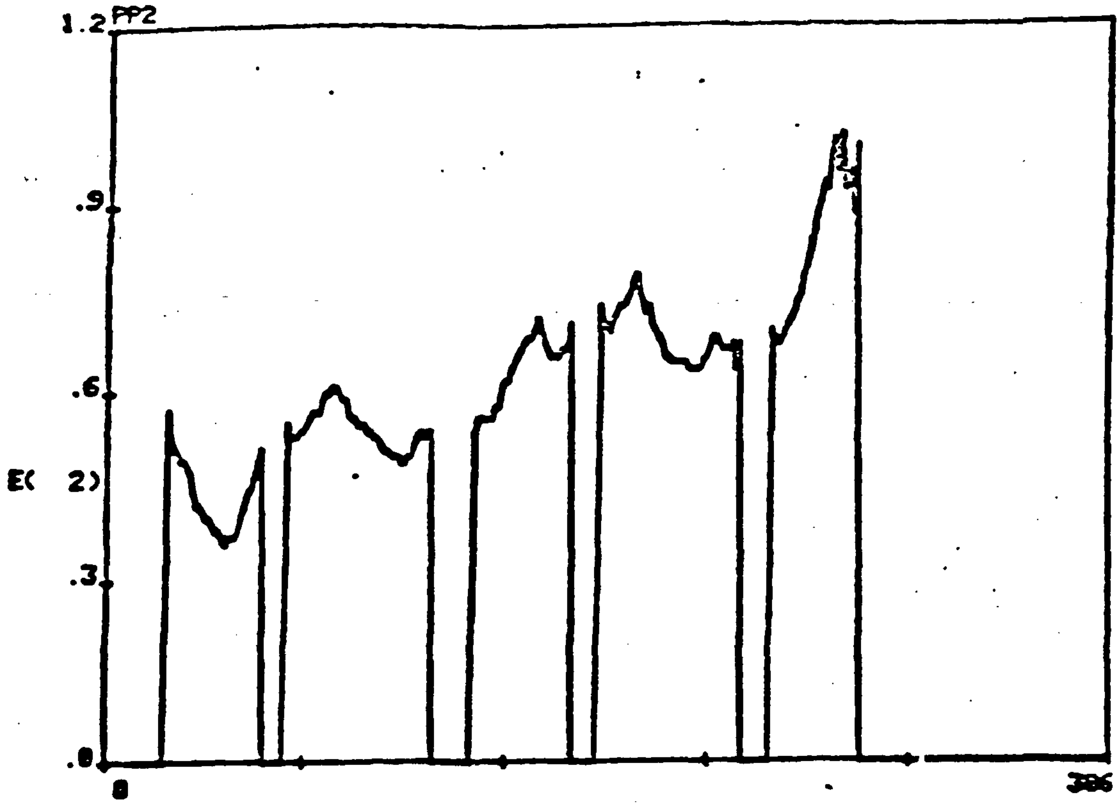


S6 CATS AND DOGS EACH HATE THE OTHER.

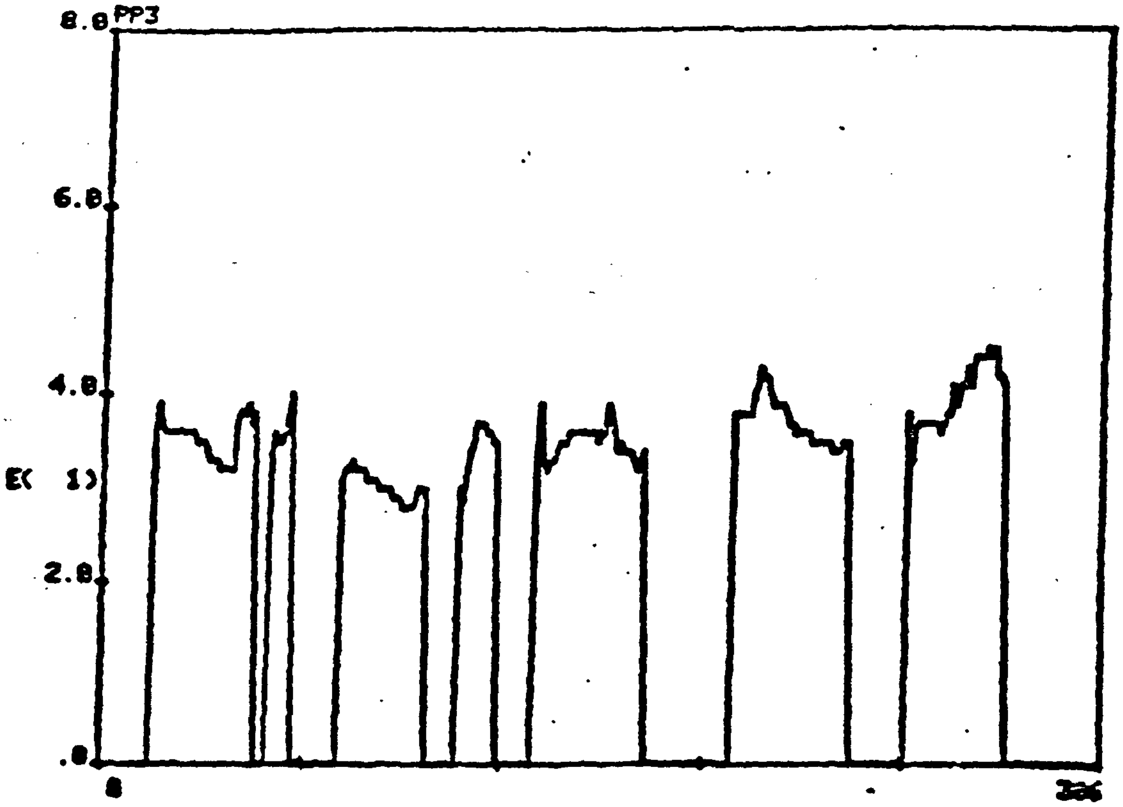




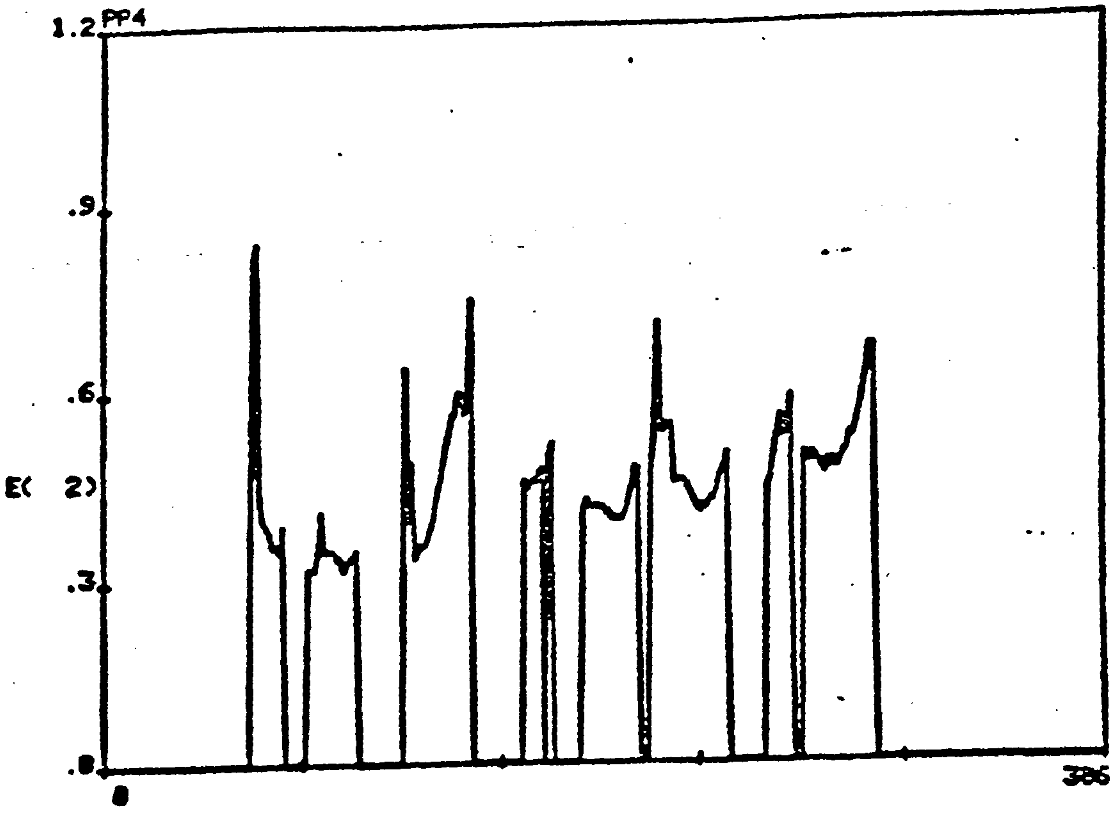
```
PP1
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0    36    39    37
    37    39    41    41     0     0     0     0     0     0
     0     0     0     0     0    36    32    31    31    31
    31    32    31    32    31    31    30    29    26    25
    24    25    25    24    24    24     0     0     0     0
     0     0     0     0     0     0    22    23    25    26
    26    27    27    30    32    31    32    32     0     0
     0     0     0    30    34    35    36    36    36    36
    36    37    37    38    38    38    37    37    37    37
    38    35    34    34    34    34    38     0     0     0
     0     0     0     0     0     0     0     0     0     0
    32    32    33    33    34    34    35    35    36    36
    37    37    37    37    37    38    38    38    38    39
    39    40    40    40    41    42    41    39    38    38
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0    34    37    39    39    39    38    39    40
    40    40    40    41    41    40    40    40    40    40
    40    40    41    40    40    41    41    41    40    40
    40    41    38    38    38    38    38    38    38    38
    38    36    39    39    39    40    40    40    41    42
    44    46    48    51    55    48    68    99    99    99
    81    80    78   101     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
```



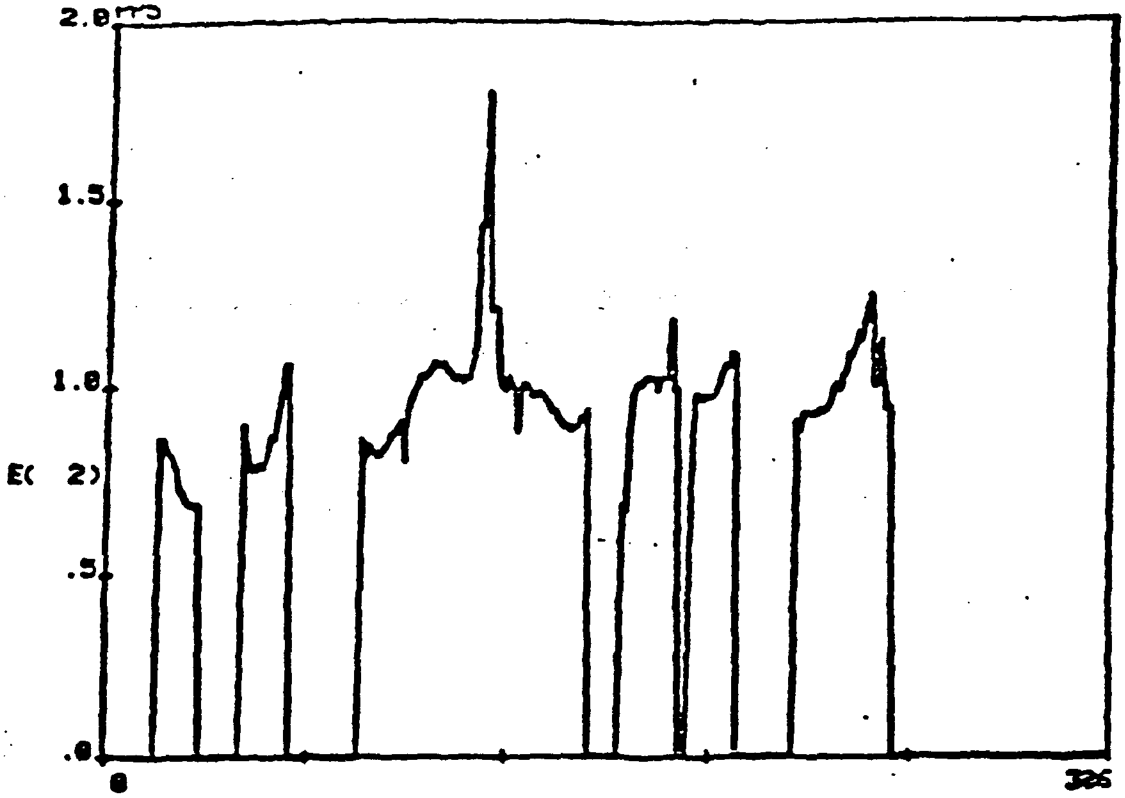
```
PP2
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0   57
   52   51   50   49   49   48   47   44   42   42
   41   40   40   39   38   38   37   36   37   37
   37   38   40   42   44   45   47   49   51    0
    0    0    0    0    0    0   55   53   53   53
   54   54   55   56   57   57   57   59   60   60
   61   61   60   59   59   57   56   56   55   55
   55   54   54   53   53   52   51   51   50   50
   50   49   49   50   50   52   53   54   53   54
   54    0    0    0    0    0    0    0    0    0
    0    0    0   54   56   56   56   56   56   58
   58   61   62   62   64   65   66   67   68   69
   69   70   72   70   68   67   66   66   66   67
   67   68   71    0    0    0    0    0    0    0
    0   74   71   71   70   72   73   74   74   75
   76   78   79   76   75   73   74   71   70   69
   68   66   66   65   65   65   65   65   64   64
   64   64   65   66   67   69   69   68   67   67
   67   68   64   68    0    0    0    0    0    0
    0    0    0   70   68   68   69   70   71   72
   73   75   76   79   81   83   86   89   91   94
   93   98  101   96  102   93   94   96   89  100
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0
```



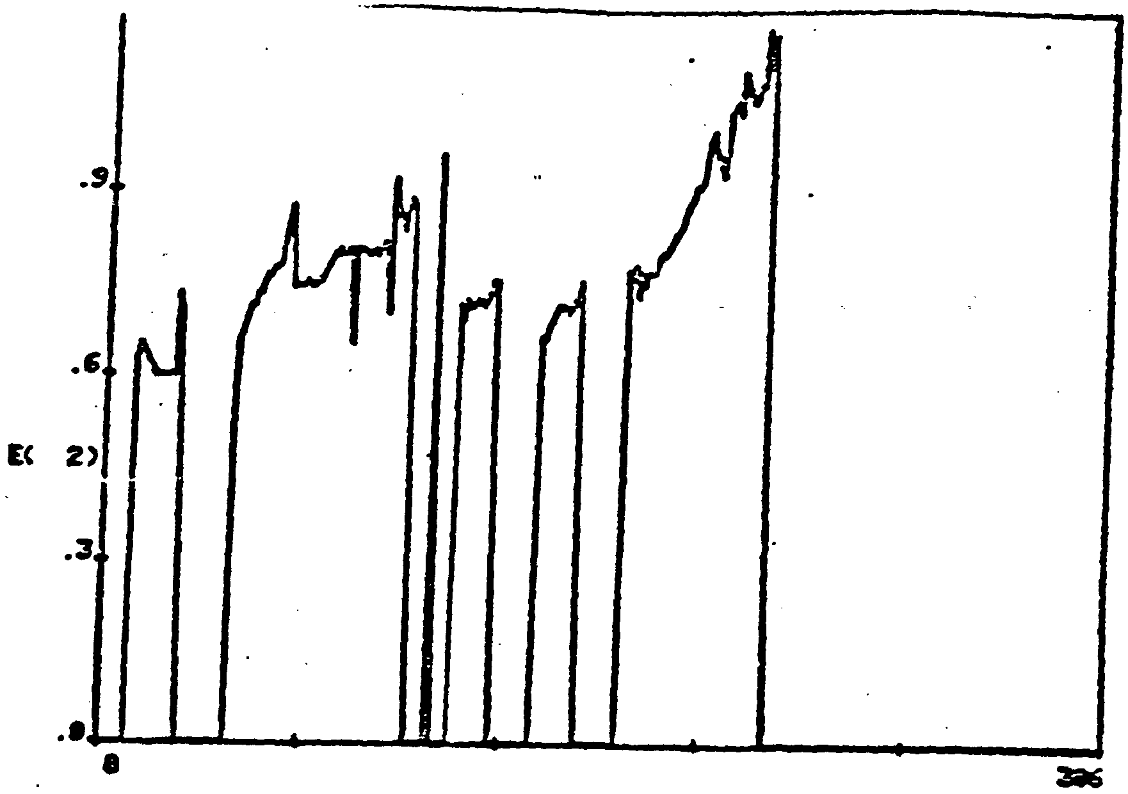
```
PP3
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0    35    39    37    36
     36    36    36    36    36    36    36    36    36    35
     35    35    34    33    33    33    32    32    32    32
     32    37    38    38    39    37    38     0     0     0
      0    34    36    35    35    36    38    40     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0    31    32    32    33    32    32    32
     32    31    31    31    31    30    30    30    30    29
     29    29    28    28    28    28    29    30    30    30
      0     0     0     0     0     0     0     0     0     0
     30    29    32    34    35    37    37    37    36    36
     35    35     0     0     0     0     0     0     0     0
      0     0     0    34    39    33    32    33    33    34
     35    35    35    36    36    36    36    36    36    36
     36    36    35    36    36    39    38    36    34    35
     34    34    34    33    32    33    34     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0    38    38    38    38    38    38    40
     41    43    42    42    40    39    39    39    39    38
     37    37    36    36    36    36    36    35    35    35
     35    35    35    34    34    34    35    35    35     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0    38    33    36
     37    37    37    37    37    37    37    36    37    38
     38    41    39    41    41    43    41    44    44    44
     44    45    44    45    42    42    41     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0
```



```
PP4
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0    82    84    43    40
    39    38    36    36    35    39     0     0     0     0
     0     0     0    32    32    32    35    41    35    35
    35    35    34    34    32    33    34    34    35     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0    64    40    49    34    35    36    36
    37    39    41    45    47    51    53    56    57    60
    60    57    58    75     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
    46    45    46    46    47    48     0    50    52     0
     0     0     0     0     0     0     0     0    41    43
    42    42    42    42    42    41    41    40    40    40
    40    41    43    45    48    48     0     0     0    47
    53    71    54    54    55    55    46    46    46    46
    45    44    43    42    41    42    42    43    43    45
    47    48    50     0     0     0     0     0     0     0
     0     0     0     0    45    46    50    53    56    53
    55    59     0     0     0    50    50    49    50    49
    48    47    49    48    49    48    49    50    51    53
    53    55    57    60    63    67    67     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
```



```
PP5
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0     86     86     83     82
     80     78     73     72     70     69     69     68     69      0
      0      0      0      0      0      0      0      0      0      0
      0      0     90     81     78     78     80     79     81     84
     87     87     93     98    106    106      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0     86     82
     85     84     82     82     83     84     85     87     88     90
     91     81     94     97     99    101    103    103    104    105
    107    106    106    107    105    105    103    103    103    102
    103    103    107    111    144    144    180    121    121    121
    105    101    100    103    100    100     89    101    101    100
     98     98     99     98     97     96     94     94     92     91
     90     90     89     90     90     92     92     94      0      0
      0      0      0      0      0      0      0      0     69     68
     89     99    101    101    102    103    103    103    100    103
    103    103    118    100    100      0      0      0     80     98
     98     97     98     98     98     99    101    104    106    106
    109    109      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
     91     89     93     93     93     93     93     94     94     95
     96    100    101    100    103    105    109    109    111    115
    113    118    122    125    101    102    113    101     95     95
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
```



```
PP6
      0      0      0      0      0      0      0      0      0     63
     65     64     63     62     61     60     60     60     60     60
     60     60     73     69      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
     51     65     67     69     71     71     73     73     75     76
     76     77     77     78     82     85     87     74     74     74
     74     75     74     74     75     75     76     77     78     79
     79     80     79     80     80     65     80     80     80     79
     79     80     79     80     81     81     70     91     86     85
     83     87     88     87      0      0      0      0      0      0
      0     95      0      0      0      0      0      0     71     69
     72     71     71     72     72     71     72     73     75     75
      0      0      0      0      0      0      0      0      0      0
      0      0      0     66     66     68     69     70     71     71
     71     70     71     72     72     75      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
     77     76     78     73     77     75     77     77     77     79
     80     80     81     82     83     84     85     87     88     89
     91     91     92     96     99    100     96     96     93     97
    103    104    105    103    110    108    106    105    106    108
    108    117    111    116      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
```

Appendix 2: Quantization of the Biquad Coefficients

An optimal quantization of the biquad coefficients is discussed in this appendix. This analysis shows an optimal bit allocation based on the minimization of maximum spectral error. The bit allocation derived here is used in our simulations.

Quantization of the Biquads Coefficients

I. Introduction

Let H(Z) = transfer function of the biquads with coefficients $\{k_i(n)\}$, where $i = 1,2$ and n denotes the stage of the biquads. Instead of mean squared error, we employ the average of the area of difference between two log spectra as a measure.

$$\Delta S = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left| \log|H(e^{jw})|^2 - \log|\tilde{H}(e^{jw})|^2 \right| dw \tag{1}$$

where $\tilde{H}(Z)$ = transfer function with a perturbation in a particular coefficient $k_i(n)$ (for example, $k_1(2)+\Delta k_1(2)$). The spectral sensitivity with respect to $k_i(n)$ is defined by

$$\frac{\partial S}{\partial k_i(n)} = \lim_{\Delta k_i(n)\rightarrow 0} \frac{1}{\Delta k_i(n)} \cdot \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\log|H(e^{jw})|^2 - \log|\tilde{H}(e^{jw})|^2\right| dw \tag{2}$$

It has been shown that the spectral sensitivity is a good measure for judging a quantization scheme for coefficients in linear predictive systems [1]. This appendix investigates the quantization properties of the biquads coefficients and derives a procedure for the bit allocation by minimizing the maximum spectral error.

II. Spectral Sensitivity

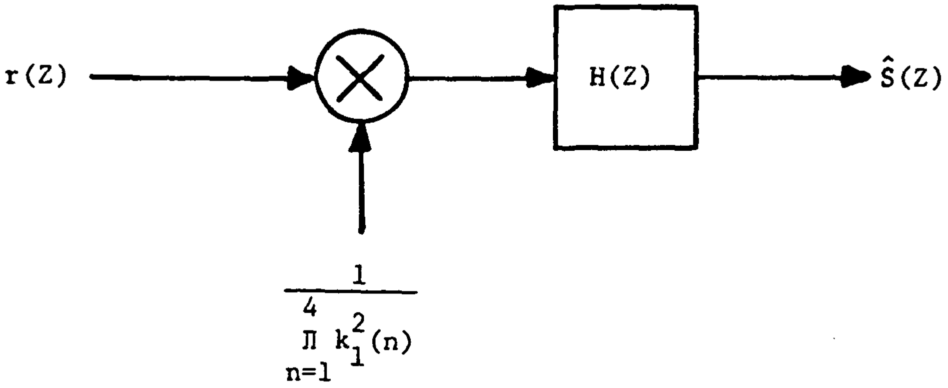
For simplicity, we take the 4 staged biquads.

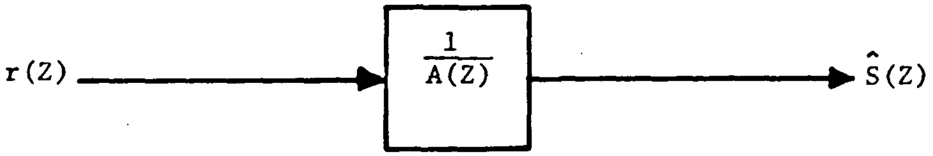
$$H(Z) = \prod_{n=1}^{4} H_n(Z) \tag{3}$$

where

$$H_n(Z) = \frac{-k_1^2(n)}{1-[2-k_1(n)k_2(n)-k_1^2(n)]Z^{-1}+[1-k_1(n)k_2(n)]Z^{-2}} \tag{4}$$

The decoder block diagram of our speech compression system is





where

$$\frac{1}{A(Z)} = \frac{1}{\prod_{n=1}^{4} k_1^2(n)} \cdot H(Z)$$

$$= \prod_{n=1}^{4} \frac{1}{A_n(Z)}$$

and

$$A_n(Z) = 1-[2-k_1(n)k_2(n)-k_1^2(n)]Z^{-1}+[1-k_1(n)k_2(n)]Z^{-2} \qquad (6)$$

The spectral sensitivity for our system can be written as follows.

$$\frac{\partial S}{\partial k_i(n)} = \lim_{\Delta k_i(n)\to 0} \frac{1}{\Delta k_i(n)} \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} |\log|A(e^{jw})|^2-\log|\tilde{A}(e^{jw})|^2|dw$$

$$= \lim_{\Delta k_i(n)\to 0} \frac{1}{\Delta k_i(n)} \cdot \frac{1}{2\pi} \sum_{n=1}^{4} \int_{-\pi}^{\pi} |\log|A_n(e^{jw})|^2-\log|\tilde{A}_n(e^{jw})|^2|dw$$

$$= \lim_{\Delta k_i(n)\to 0} \frac{1}{2\pi} \int_{-\pi}^{\pi} \cdot \frac{1}{\Delta k_i(n)} |\log|A_n(e^{jw})|^2-\log|\tilde{A}_n(e^{jw})|^2|dw$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\frac{\partial}{\partial k_i(n)} \log|A_n(e^{jw})|^2|dw$$

$$= \frac{1}{2} \int_{-1}^{1} |\frac{\partial}{\partial k_i(n)} \log|A_n(e^{j\pi\theta})|^2|d\theta$$

$$= \frac{1}{2} \int_{-1}^{1} a_i(n,\theta)d\theta \tag{7}$$

where

$$a_i(n,\theta) = \left| \frac{1}{|A_n(e^{j\pi\theta})|^2} \cdot \frac{\partial}{\partial k_i(n)} |A_n(e^{j\pi\theta})|^2 \right|, \tag{8}$$

and

$$|A_n(e^{j\pi\theta})|^2 = [k_1(n)-2(2-k_1(n)k_2(n))\sin^2 \tfrac{\pi}{2}\theta]^2 + [k_1(n)k_2(n)\sin n\pi\theta]^2 \tag{9}$$

The elimination of the summation in the above derivation is due to the fact that $|\tilde{A}_i(e^{jw})|^2$ has the same coefficients $\{k_i(n)\}$ as $|A(e^{jw})|^2$ except a particular single $k_i(n)$. The biquad is assumed to be stable, i.e., zeros of $A_n(Z)$ lie within the unit circle (not on the unit circle). Thus, $|\log|A_n(e^{jw})|^2|$ is bounded. Therefore we can take the derivative.

To compute $\frac{\partial S}{\partial k_i(n)} = \frac{1}{2}\int_{-1}^{1} a_i(n,\theta)d\theta$, we use the Gauss' formula, i.e.

$$\frac{\partial S}{\partial k_i(n)} \doteq \frac{1}{2} \sum_{m=1}^{L} w_m \, a_i(n,x_m) \tag{10}$$

where, for a fixed L, w_m and x_m are given for m= 1,2,...L.

Directly from (8) and (9), we have

i) for $k_1(n)$, n=0,1,2,3

$$a_1(n,x) = \left| \frac{2[k_1(n)-(4-2k_1(n)k_2(n))\sin^2 \frac{\pi}{2}x](1+2k_2(n)\sin^2 \frac{\pi}{2}x)+2k_1(n)k_2^2(n)\sin^2\pi x}{[k_1(n)-(4-2k_1(n)k_2(n))\sin^2 \frac{\pi}{2}x]^2+[k_1(n)k_2(n)\sin^2\pi x]^2} \right|$$

(11)

ii) for $k_2(n)$, n=0,1,2,3

$$a_2(n,x) = \left| \frac{2[k_1(n)-(4-2k_1(n)k_2(n))\sin^2 \frac{\pi}{2}x]2k_1(n)\sin^2 \frac{\pi}{2}x+2k_1^2(n)k_2(n)\sin^2 \pi x}{[k_1(n)-(4-2k_1(n)k_2(n))\sin^2 \frac{\pi}{2}x]^2+[k_1(n)k_2(n)\sin^2 \pi x]^2} \right| \tag{12}$$

The spectral sensitivity for a particular $k_1(n)$ does, in general, depend on the values of the other coefficients. A useful choice is the simple average of the sensitivity over many different sets of coefficients from a large number of different speech sounds,

$$\frac{\overline{\partial S}}{\partial k_i(n)} = \frac{1}{T} \sum_{t=1}^{T} \frac{\partial S}{\partial k_i(n,t)} \tag{13}$$

Figure I shows the $\frac{\overline{\partial S}}{\partial k_i(n)}$ of the 4 staged biquads. The average of the sensitivity was conducted over 10 sets of different coefficients (5 voiced, 5 unvoiced) from the sample speech S1. In Figure I, the smoothed values result in the curves where the exact sensitivity lies within ±1 dB around the curves respectively. The curves cover practical ranges of each $k_i(n)$ for the sample speech S1. S1 is "The pipe began .....". It can be noticed that the reconstructed speech quality is more sensitive to the quantization error around lower values of $k_1(n)$, n=0, 1, 2 and 3, while the sensitivities of $k_2(n)$ is more uniform.

III. Quantization Scheme

We define the optimal quantization as a quantization which provides a flat spectral sensitivity. Thus, the search for the optimal quantization scheme reduces to the search for a nonlinear transform that results in a flat spectral sensitivity, and then we employ the linear quantization for the transformed coefficients.

Let f(·) be the nonlinear transform such that

$$g = f(k) \quad , \quad k\in\{k_1(n)\} \quad . \tag{14}$$

Since $\frac{\partial S}{\partial g}$ is a constant for the optimality, we have

$$\frac{\partial S}{\partial g} = \frac{\partial S}{\partial k}.\frac{\partial k}{\partial g}$$

$$= c \qquad \text{(constant)}$$

Thus,

$$\frac{\partial f}{\partial k} = \frac{1}{c} \cdot \frac{\partial S}{\partial k} \tag{15}$$

If the expression of $\frac{\partial S}{\partial k}$ is given, we can obtain f(·) by integration. The sensitivity curves for $k_1(0)$ and $k_1(1)$ in Figure **|** can be approximately represented by

$$\frac{\partial S}{\partial k_1(n)} = 10\log_{10} \left. \frac{1}{1-(k_1(n)-\beta)^2} \right|_{\beta=0.85} \quad , \quad \begin{matrix} 0.1\le k_1(n)\le 0.8 \\ n = 0 \text{ and } 1 \end{matrix} \tag{16}$$

By (15), we obtain

$$f(k_1(n)) = \log_{10} \frac{(1-\beta)+k_1(n)}{(1+\beta)-k_1(n)}\Bigg|_{\beta = 0.85}$$

$$= \log_{10} \frac{0.15+k_1(n)}{1.85-k_1(n)} \quad , \quad 0.1 \leq k_1(n) \leq 0.8 \tag{17}$$

Figure 3 shows a plot of $f(\cdot)$. We have also plotted a line that provides close values over $0.1 \leq k_1(n) \leq 0.8$. Therefore, in practice, we could linearly quantize $k_1(0)$ and $k_1(1)$ as well as other $\{k_i(n)\}$ to obtain approximately flat sensitivity characteristics.

IV. Bit Allocation

We derive a procedure for binary bit allocation by minimizing the maximum spectral deviation. Let

M = the total number of bits for quantization

$\{q_i, i=1,2,...p\}$ = set of coefficients to be quantized

$N_i = 2^{M_i}$: number of levels for coefficient q_i

$\delta_i = \frac{\overline{q}_i - \underline{q}_i}{N_i}$: quantization size.

where

$\overline{q}_i$ = upper bound of q_i.

$\underline{q}_i$ = lower bound of q_i

For the linear quantization of q_i using round-off arithmetic, the maximum quantization error is

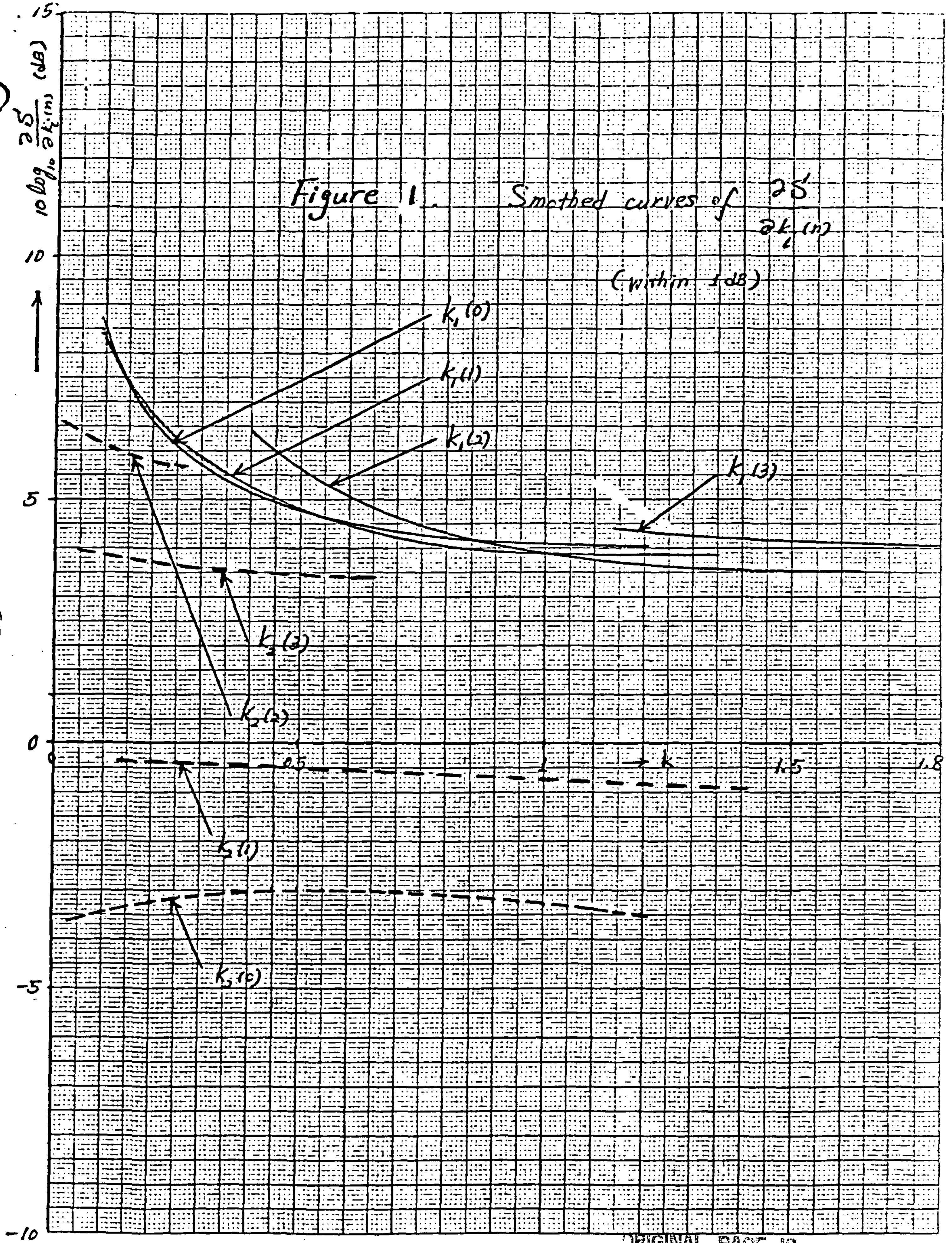


Figure 1. Smothed curves of $\frac{\partial \delta}{\partial k_l(n)}$

(within ±dB)

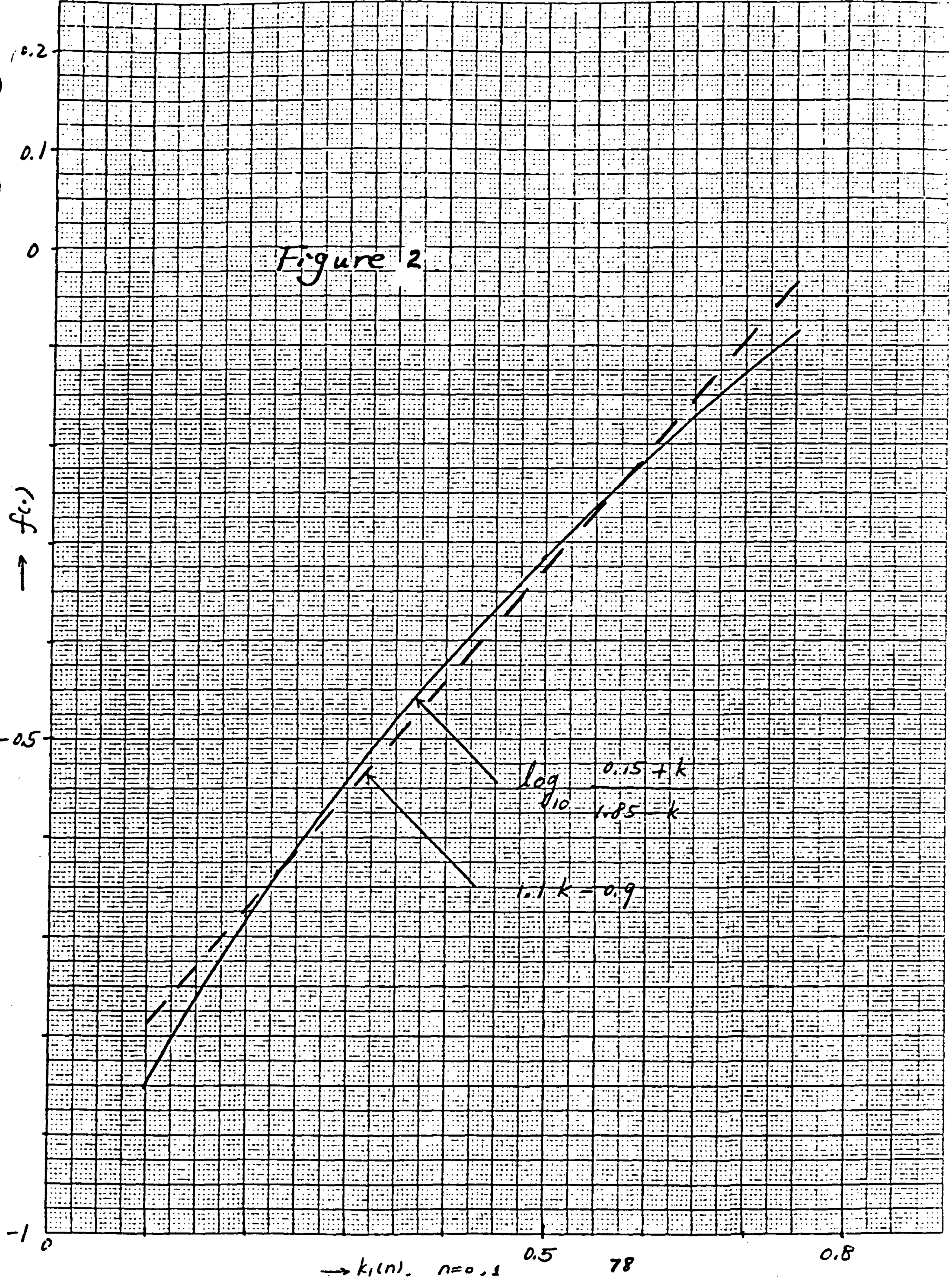


Figure 2

$$|\Delta q_i|_{max} = \frac{1}{2}\,\delta_i \tag{18}$$

The maximum total spectral deviation $(\Delta S)_{max}$ is given by

$$\begin{aligned}(\Delta S)_{max} &= \sum_{i=1}^{P} \left|\frac{\partial S}{\partial q_i}\right| \cdot (\Delta q_i)_{max} \\ &= \sum_{i=1}^{P} \frac{K_i}{N_i}\end{aligned} \tag{19}$$

where

$$K_i = \frac{\bar{q}_i - \underline{q}_i}{2} \left|\frac{\partial S}{\partial q_i}\right| \quad , \quad 1 \leq i \leq p \tag{20}$$

The problem is to find N_i, $i = 1, 2, \ldots p$, minimizing $(\Delta S)_{max}$ subject to the constraint $\sum_{i=1}^{P} \log_2 N_i = M$. The solution is given by [1],

$$N_1 = K_1 \left[\frac{2^M}{\prod_{i=1}^{P} K_i}\right]^{\frac{1}{P}}$$

and

$$N_i = \frac{K_i}{K_1} \cdot N_1 \quad , \quad 2 \leq i \leq p \tag{21}$$

For example, we use the 4-staged biquads, and can make a numerical table as follows. Here, we have 26 bits for quantization of the biquads' coefficients.

Table 1. Bit allocation with M=26 for sample speech S1

	$k_1(0)$	$k_1(1)$	$k_1(2)$	$k_1(3)$	$k_2(0)$	$k_2(1)$	$k_2(2)$	$k_2(3)$
pper bound	0.8	0.8	1.6	1.9	1.2	1.4	0.4	0.14
lower bound	0.1	0.1	0.8	1.6	0.02	0.2	0.02	0.02
$\frac{\partial S}{k_i(n)}$	3.46	3.16	2.8	2.6	0.48	0.9	2.3	4
$K_i(n)$	1.211	1.106	1.12	0.39	0.28	0.54	0.44	0.24
$_i(n)/K_i(0)$	1	0.91	0.92	0.32	0.23	0.45	0.36	0.2
$N_i(n)=2^{M_i(n)}$ efore truncation	20.66	18.8	19	6.6	4.75	9.3	7.44	4.13

After truncation of N_i and rearrangement of M bits, we obtain the bit allocation for our system for speech S1, as shown in Table 2.

Table 2. Bit allocation for coefficients

coefficient	bits
rms of residual	5
$k_1(0)$	4
$k_2(0)$	3
$k_1(1)$	4
$k_2(1)$	3
$k_1(2)$	4
$k_2(2)$	3
$k_1(3)$	3
$k_2(3)$	2
Total	30 bits

V. Computational Procedure

1. Using (10), compute $\frac{\partial S}{\partial k_i(n)}$ of different sets of coefficients from many different speech sounds, and take the average by (13).

2. Using (15), compute the nonlinear transform $f(\cdot)$ and apply the linear quantization scheme for the transformed coefficients.

3. As shown in Table 1, compute the bit allocation

Remark:

This report has considered the quantization properties of the biquads coefficients, and concluded that i) we could apply the linear quantization directly to $\{k_i(n)\}$ and ii) we have the bit allocation for speech S1, as shown in Table 2.

Reference:

[1] R. Viswanathan, and J. Makhoul, 'Quantization Properties of Transmission Parameters in Linear Predictive Systems," IEEE Trans. on ASSP, Vol. 23, June, 1975.

Appendix 3 : List of quantized biquad coefficients

This appendix shows the list of quantized biquad coefficients of sys-4k, where the utterence file of /usr/ee/moon/speech/spl was used. $k_i[j]$ denotes the coefficient k_1 of j-th biquad.

ORIGINAL PAGE IS
OF POOR QUALITY

FRAME	rms2	K1[0]	K1[1]	K1[2]	K1[3]	K2[0]	K2[1]	K2[2]	K2[3]
0	0.941	0.219	0.764	1.478	1.830	0.327	0.397	0.180	0.115
1	0.661	0.219	0.764	1.478	1.830	0.479	0.397	0.180	0.115
2	0.826	0.219	0.823	1.478	1.830	0.479	0.397	0.180	0.115
3	0.884	0.256	0.823	1.478	1.830	0.479	0.397	0.180	0.115
4	1.076	0.256	0.764	1.478	1.830	0.479	0.397	0.180	0.115
5	1.135	0.256	0.705	1.413	1.830	0.479	0.397	0.180	0.115
6	0.821	0.219	0.705	1.413	1.830	0.479	0.397	0.228	0.115
7	1.805	0.219	0.764	1.413	1.830	0.630	0.536	0.228	0.115
8	1.334	0.256	0.823	1.413	1.830	0.630	0.536	0.180	0.115
9	0.971	0.256	0.823	1.413	1.830	0.630	0.536	0.228	0.085
10	1.264	0.256	0.883	1.413	1.830	0.630	0.536	0.180	0.085
11	0.474	0.331	1.002	1.478	1.830	0.630	0.536	0.228	0.085
12	0.510	0.631	1.120	1.607	1.830	0.630	0.536	0.180	0.025
13	0.396	0.744	1.120	1.607	1.830	0.630	0.536	0.180	0.025
14	0.369	0.744	1.120	1.607	1.830	0.630	0.536	0.277	0.055
15	0.414	0.781	1.120	1.607	1.830	0.630	0.397	0.228	0.055
16	0.373	0.781	1.120	1.607	1.830	0.630	0.397	0.228	0.085
17	0.341	0.781	1.120	1.607	1.830	0.782	0.536	0.325	0.085
18	0.358	0.781	1.120	1.607	1.830	0.782	0.536	0.277	0.055
19	0.376	0.744	1.061	1.542	1.830	0.630	0.536	0.228	0.055
20	0.319	0.556	0.942	1.478	1.751	0.630	0.536	0.277	0.115
21	0.464	0.594	0.942	1.347	1.592	0.630	0.397	0.180	0.055
22	0.619	0.594	0.942	1.347	1.592	0.630	0.397	0.228	0.085
23	0.680	0.406	0.823	1.348	1.751	0.782	0.536	0.325	0.115
24	0.923	0.256	0.764	1.282	1.751	0.782	0.536	0.374	0.115
25	0.991	0.256	0.823	1.282	1.830	0.782	0.536	0.374	0.115
26	0.848	0.256	0.883	1.347	1.830	0.934	0.536	0.374	0.115
27	0.775	0.294	0.883	1.282	1.830	0.934	0.536	0.374	0.115
28	0.636	0.256	0.823	1.282	1.830	0.934	0.536	0.374	0.115
29	1.310	0.256	0.705	1.152	1.830	0.934	0.536	0.325	0.085
30	0.601	0.256	0.645	1.152	1.830	0.934	0.536	0.374	0.055
31	0.240	0.219	0.645	1.152	1.830	0.934	0.536	0.374	0.055
32	0.456	0.219	0.705	1.217	1.830	0.934	0.674	0.374	0.085
33	1.109	0.219	0.705	1.282	1.830	0.934	0.674	0.374	0.085
34	1.156	0.219	0.764	1.282	1.830	0.934	0.674	0.374	0.085
35	2.096	0.219	0.705	1.282	1.830	0.934	0.674	0.374	0.055
36	1.629	0.294	0.705	1.217	1.830	0.934	0.536	0.325	0.025
37	1.781	0.256	0.705	1.217	1.830	0.934	0.536	0.325	0.025
38	2.054	0.219	0.645	1.217	1.830	0.934	0.536	0.325	0.025
39	2.614	0.256	0.645	1.152	1.830	0.934	0.536	0.277	0.025
40	1.889	0.331	0.645	1.152	1.830	0.934	0.397	0.277	0.055
41	1.176	0.406	0.586	1.217	1.830	0.934	0.397	0.277	0.055

ORIGINAL PAGE IS
OF POOR QUALITY

42	0.672	0.444	0.586	1.282	1.830	0.934	0.397	0.325	0.055
43	0.543	0.444	0.527	1.347	1.830	0.934	0.397	0.374	0.025
44	0.575	0.406	0.467	1.413	1.830	0.934	0.397	0.374	0.025
45	0.632	0.369	0.408	1.413	1.830	0.934	0.397	0.374	0.055
46	0.444	0.331	0.348	1.413	1.830	0.934	0.397	0.374	0.055
47	0.107	0.331	0.348	1.413	1.830	0.934	0.397	0.374	0.085
48	0.671	0.406	0.348	1.413	1.830	0.934	0.397	0.325	0.085
49	1.301	0.481	0.348	1.413	1.830	0.934	0.397	0.325	0.115
50	1.383	0.519	0.348	1.413	1.830	1.085	0.536	0.325	0.115
51	1.214	0.519	0.408	1.348	1.830	1.085	0.536	0.277	0.115
52	0.599	0.519	0.408	1.348	1.830	1.085	0.536	0.277	0.115
53	1.173	0.481	0.348	1.347	1.830	1.085	0.536	0.277	0.115
54	0.974	0.519	0.289	1.413	1.830	1.085	0.536	0.325	0.085
55	0.559	0.481	0.289	1.478	1.830	1.085	0.536	0.374	0.055
56	0.763	0.444	0.289	1.478	1.830	1.085	0.536	0.374	0.055
57	0.226	0.406	0.230	1.478	1.830	1.085	0.536	0.374	0.055
58	0.856	0.444	0.230	1.478	1.751	1.085	0.536	0.374	0.055
59	1.850	0.519	0.289	1.478	1.672	1.085	0.536	0.374	0.055
60	9.656	0.556	0.289	1.478	1.672	1.085	0.536	0.374	0.085
61	1.432	0.556	0.289	1.478	1.672	1.085	0.536	0.374	0.115
62	1.086	0.519	0.289	1.478	1.672	1.085	0.536	0.374	0.115
63	0.823	0.519	0.348	1.478	1.672	1.085	0.536	0.374	0.115
64	0.783	0.481	0.408	1.478	1.672	1.085	0.536	0.374	0.115
65	0.772	0.444	0.408	1.478	1.672	1.085	0.536	0.374	0.115
66	1.080	0.444	0.408	1.478	1.672	1.085	0.536	0.374	0.115
67	1.192	0.444	0.408	1.478	1.672	1.085	0.536	0.374	0.115
68	1.236	0.406	0.408	1.478	1.672	1.085	0.536	0.374	0.115
69	1.093	0.406	0.408	1.413	1.672	1.085	0.536	0.325	0.115
70	1.159	0.369	0.408	1.348	1.672	1.085	0.536	0.277	0.115
71	1.154	0.331	0.408	1.347	1.672	1.085	0.536	0.277	0.115
72	0.788	0.294	0.408	1.347	1.672	1.085	0.536	0.277	0.115
73	0.568	0.294	0.408	1.348	1.751	1.085	0.536	0.325	0.115
74	0.565	0.294	0.467	1.348	1.751	1.085	0.536	0.374	0.115
75	4.964	0.294	0.467	1.348	1.751	1.085	0.536	0.374	0.115
76	3.807	0.406	0.586	1.478	1.751	1.237	0.674	0.374	0.115
77	0.680	0.444	0.586	1.413	1.751	1.237	0.674	0.374	0.085
78	0.927	0.444	0.586	1.348	1.751	1.237	0.674	0.325	0.085
79	0.563	0.406	0.527	1.282	1.751	1.237	0.674	0.325	0.085
80	0.796	0.331	0.467	1.217	1.751	1.237	0.674	0.325	0.115
81	0.998	0.294	0.408	1.152	1.751	1.237	0.674	0.325	0.115
82	1.977	0.294	0.408	1.152	1.751	1.237	0.674	0.325	0.115
83	0.672	0.256	0.408	1.152	1.751	1.237	0.674	0.325	0.115
84	1.531	0.256	0.408	1.152	1.751	1.237	0.674	0.325	0.115
85	6.036	0.256	0.408	1.152	1.751	1.237	0.674	0.325	0.115
86	5.969	0.256	0.408	1.152	1.751	1.237	0.674	0.325	0.115
87	2.658	0.256	0.408	1.087	1.751	1.237	0.674	0.277	0.115
88	3.008	0.294	0.408	1.087	1.672	1.237	0.674	0.228	0.085
89	4.378	0.331	0.408	1.087	1.672	1.237	0.674	0.277	0.115
90	2.529	0.369	0.467	1.087	1.672	1.237	0.674	0.228	0.115
91	2.297	0.369	0.467	1.087	1.672	1.237	0.674	0.277	0.115
92	1.588	0.369	0.467	1.087	1.751	1.237	0.674	0.277	0.115
93	0.968	0.369	0.467	1.152	1.751	1.237	0.674	0.277	0.115
94	0.423	0.331	0.467	1.152	1.751	1.237	0.674	0.277	0.115
95	0.509	0.331	0.527	1.217	1.751	1.237	0.613	0.374	0.115

```
 96    1.208    0.331   0.586   1.217   1.751   1.237   0.952   0.374   0.115
 97    1.446    0.369   0.645   1.282   1.751   1.237   1.091   0.374   0.085
 98    1.377    0.369   0.645   1.347   1.751   1.237   1.091   0.374   0.085
 99    0.730    0.369   0.705   1.347   1.751   1.237   1.091   0.374   0.085
100    0.851    0.369   0.705   1.347   1.751   1.237   1.091   0.374   0.115
101    1.092    0.369   0.764   1.348   1.751   1.237   1.091   0.374   0.115
102    1.161    0.369   0.764   1.347   1.751   1.237   1.091   0.374   0.115
103    0.176    0.406   0.764   1.348   1.751   1.237   1.091   0.374   0.115
104    0.713    0.331   0.705   1.282   1.751   1.237   1.091   0.374   0.115
105    0.915    0.331   0.705   1.282   1.751   1.237   1.091   0.374   0.115
106    1.000    0.294   0.645   1.217   1.672   1.237   1.091   0.374   0.115
107    1.439    0.256   0.586   1.152   1.672   1.237   0.952   0.374   0.115
108    3.404    0.256   0.527   1.087   1.672   1.237   0.952   0.374   0.115
109    2.091    0.256   0.527   1.022   1.672   1.237   0.952   0.374   0.115
110    2.891    0.294   0.527   0.957   1.672   1.237   0.952   0.325   0.115
111    4.520    0.331   0.527   0.957   1.672   1.237   0.952   0.277   0.115
112    6.510    0.369   0.527   0.957   1.672   1.237   0.952   0.277   0.085
113    3.420    0.369   0.527   1.022   1.672   1.237   0.952   0.325   0.085
114    3.369    0.369   0.527   1.087   1.672   1.237   0.952   0.325   0.115
115    1.694    0.369   0.527   1.087   1.751   1.237   0.952   0.277   0.115
116    1.173    0.369   0.467   1.087   1.751   1.237   0.952   0.277   0.085
117    1.369    0.369   0.467   1.022   1.751   1.237   0.952   0.228   0.115
118    1.434    0.331   0.467   0.957   1.751   1.237   0.952   0.228   0.115
119    1.421    0.331   0.467   0.957   1.751   1.237   0.952   0.228   0.115
120    0.962    0.331   0.467   0.957   1.751   1.237   0.952   0.228   0.115
121    0.830    0.294   0.467   0.957   1.751   1.237   0.952   0.228   0.115
122    2.764    0.294   0.467   0.957   1.751   1.237   0.952   0.228   0.115
123    2.911    0.294   0.467   0.957   1.751   1.237   0.952   0.228   0.115
124    3.468    0.294   0.467   0.957   1.751   1.237   0.952   0.228   0.115
125    2.984    0.294   0.467   0.957   1.751   1.237   0.952   0.277   0.115
126    1.845    0.294   0.467   1.022   1.751   1.237   0.952   0.325   0.115
127    1.664    0.294   0.467   1.087   1.751   1.237   0.952   0.325   0.115
128    1.697    0.294   0.408   1.152   1.751   1.237   0.952   0.325   0.115
129    2.330    0.294   0.408   1.152   1.751   1.237   0.952   0.277   0.115
130    3.277    0.294   0.408   1.152   1.751   1.237   0.952   0.228   0.115
131    4.725    0.294   0.408   1.152   1.751   1.237   0.952   0.228   0.115
132    4.345    0.294   0.408   1.087   1.751   1.237   0.952   0.180   0.115
133    6.793    0.294   0.408   1.087   1.751   1.237   0.952   0.180   0.115
134    4.167    0.294   0.408   1.087   1.751   1.237   0.952   0.180   0.115
135    1.819    0.294   0.408   1.022   1.751   1.237   0.952   0.180   0.115
136    2.126    0.294   0.408   1.022   1.751   1.237   0.952   0.180   0.115
137    0.830    0.294   0.408   1.022   1.751   1.237   0.952   0.228   0.115
138    2.515    0.294   0.408   1.087   1.751   1.237   0.952   0.277   0.115
139    1.365    0.294   0.408   1.087   1.751   1.237   0.952   0.325   0.115
140    2.137    0.294   0.408   1.087   1.751   1.237   0.952   0.374   0.115
141    1.194    0.331   0.467   1.152   1.830   1.237   0.952   0.374   0.115
142    1.264    0.331   0.467   1.217   1.830   1.237   0.952   0.374   0.115
143    1.313    0.331   0.527   1.282   1.830   1.237   0.952   0.374   0.115
144    1.044    0.294   0.467   1.282   1.830   1.237   0.952   0.374   0.115
145    1.197    0.294   0.467   1.282   1.830   1.237   0.952   0.374   0.085
146    1.396    0.256   0.527   1.347   1.830   1.237   1.091   0.374   0.085
147    1.431    0.294   0.586   1.348   1.830   1.237   1.091   0.325   0.115
148    0.940    0.294   0.586   1.348   1.830   1.237   1.091   0.325   0.115
149    0.000    0.294   0.586   1.348   1.830   1.237   1.091   0.325   0.115
file=::1a Acc. Error=4.673136e+03
```

PART III

AUDIO TAPE OF SIMULATIONS

The simulated results were recorded on an audio cassette tape. There are eight different types of utterences on the tape. Each of the following type of utterences are repeated several times:

1) "The pipe began to rust while new" (female speaker)

2) "Thieves who rob friends deserve jail" (male speaker)

3) "Add the sum to the product of these three" (female speaker)

4) "Open the crate but don't break the glass" (male speaker)

5) "Oak is strong and also gives shade" (male speaker)

6) "Cats and dogs each hate the other" (male speaker)

These six utterences are recorded in a clear background environment. The next two types have strong background interference. In type (7) there is another background voice while in (8) there is a white noise background.

7) "The pipe began to rust while new" (female speaker)

8) "Cats and dogs each hate the other" (male speaker)

For each type, the recording order is the original utterence, the output of sys-8k, and the output of sys-4k where each utterence is repeated twice. In all cases except the original quantization of the biquad coefficients has been applied.