

THE UNIVERSITY OF MICHIGAN  
OFFICE OF RESEARCH ADMINISTRATION  
ANN ARBOR

SOME COMPUTER EXPERIMENTS WITH PREDICTIVE  
CODING OF SPEECH


Technical Report No. 132

4251-1-T

Cooley Electronics Laboratory

Department of Electrical Engineering

By: M. P. Ristenbatt  
T. Felisky

Approved by:   
T. G. Birdsall

ORA Project 4251

CONTRACT NO. DA-36-039 sc-87172  
U. S. ARMY SIGNAL SUPPLY AGENCY  
U. S. ARMY SIGNAL RESEARCH AND DEVELOPMENT LABORATORY  
Fort Monmouth, New Jersey

March 1962

## ACKNOWLEDGMENTS

The authors gratefully acknowledge the work of Mrs. Marcia Feingold who wrote the complex computer program required for this work. Thanks is due to the Research Department of the Bell Telephone Laboratories for giving this Laboratory an analog-to-digital converter suitable for use with a digital computer. The authors also acknowledge Mr. A. Boniello of the Signal Corps Communications Department for suggesting these experiments and calling attention to pertinent related work on delta modulation.

## TABLE OF CONTENTS

	<u>Page</u>
LIST OF ILLUSTRATIONS	iv
LIST OF SYMBOLS	v
ABSTRACT	vi
1. INTRODUCTION	1
1.1 Nature of the Problem	1
1.2 Prediction Theories	2
1.3 Delta Modulators	7
1.3.1 Delta Modulation--Single Integration	7
1.3.2 Delta Modulation--Double Integration	8
1.3.3 Delta Modulation Using Exponential Decay	10
1.3.4 Delta Modulation--Single Integration, n-Digit	12
1.3.5 Delta Modulation--Double Integration, n-Digit	13
1.3.6 Delta PCMM	13
1.3.7 Log Differential PCM	15
1.3.8 Some Comments on Delta Modulation	16
2. OBJECTIVE OF EXPERIMENTAL STUDY	19
3. EXPERIMENTAL PROCEDURE	22
3.1 System Block Diagram	23
3.2 Basic Analog-to-Digital Converter	25
3.3 Record-Playback Electronics	26
3.4 Computer Program	29
4. EXPERIMENTAL RESULTS	34
4.1 Sine Wave Results	35
4.2 Experimental Speech Results	38
5. CONCLUSIONS	40
APPENDIX A: DATA FOR SINE WAVE EXPERIMENTS	42
APPENDIX B: SPEECH DATA	46
APPENDIX C: DETAILED COMPUTER PROGRAM	49
REFERENCES	74
DISTRIBUTION LIST	75

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Block diagram of theoretical prediction transmission system .	1
2	"Slope" prediction ( $M = 2$ ).	5
3	Single-digit $\Delta M$ -SI.	7
4	"Staircase" approximation resulting with $\Delta M$ -SI.	8
5	Block diagram of single digit $\Delta M$ -DI.	9
6	"Slope" approximation resulting with $\Delta M$ -DI.	9
7	Illustration of waveform approximation with single exponential network and single-sided pulse.	11
8	Block diagram of $\Delta M$ -SI, n-digit.	13
9	Block diagram of $\Delta PCM_M$ system.	14
10	Block diagram of log differential PCM.	15
11	Theoretical quantization S/N versus "bit rate" for delta modulation.	17
12	Block diagram of predictive coding system used in experiments.	19
13	Block diagram of experiments showing operations and notation.	22
14	Block diagram of experimental predictive coding system.	24
15	Block diagram of analog-to-digital converter.	27
16	Block diagram of record/playback electronics associated with AD converter.	28
17	Block diagram of computer program for predictive coding experiments.	30
18	Experimental quantization S/N versus bit rate for 800 cps sine wave and linear quantization.	36
19	Synoptic flow chart.	55
20	Flow chart for <u>QUAN1</u> (linear quantizing) procedure.	56
21	Flow chart for <u>QUAN2</u> (companded quantizing) procedure.	57
22	Flow chart of part of TR 200 subroutine which computes E.	58
23	Part of TR 200 subroutine.	59
24	Part of computation of TR 200 subroutine.	60
25	Recording part of TR 200 subroutine.	61
26	Control part of TR 200 subroutine.	62
27	Iteration part of TR 200 subroutine.	63

### LIST OF SYMBOLS

- $e_M$  = the difference between the predicted and the actual input signal samples (before going into the error quantizer)
- EM = computer program term for the above  $e_M$
- $e_Q$  = the quantized value of  $e_M$ , as quantized by the "error quantizer"---may be either "linearly quantized" or "companded"
- $f_{in}$  = the value of the input signal sample
- $f_p$  = the value of the predicted signal sample, as calculated by the "predictor"
- FP = the computer program term for  $f_p$
- $f_{ro}$  = the value of the output signal sample, computed by the receiver
- FRO = the computer program term for  $f_{ro}$
- M = the number of past samples used to compute  $f_p$  (at the sample rate of 10 kc)
- n = number of "bits" required to specify the  $e_Q$  signal; n is related to the number of quantizing levels. Hence n = number of bits required for the transmitted signal (n = Q for linear quantizing;  $n = \log_2 2N$  for companded quantizing).
- N = the number of quantizing levels (on a single side) for companded quantizing. Hence  $n = \log_2(2N)$  for the companded error quantizing.
- OL1 = a computer program term which specifies the smallest quantizing interval when companding error quantizing is used (see Table V in Appendix A)
- Q = a computer program term, where  $2^Q$  = the total number of quantization levels for linear error quantizing. Hence  $Q = n$  for the linear error quantizing.
- $Q_{MAX}$  = a computer program term which is related to the maximum range for the linear error quantizing. Thus the maximum range in units is given by  $\pm 2^{Q_{MAX}-1}$ .

## ABSTRACT

This report presents the results of experiments in which linear predictive coding was applied to speech. Predictive coding permits reducing the digit rate of high quality speech encoded by PCM. Phonetically balanced speech was used as input; the experiments were implemented by a special-purpose analog-to-digital converter and a digital computer.

With the linear predicting law used here, it was found that one should not use more than two past samples in making a prediction. With both one and two past samples, it was found that one could reduce the PCM digit rate from the presently-used 48 kc to 40 kc with no easily discernible decrease in quality, using either linear or log quantizing. With experiments run at 30 kc, a discernible increase in noise was found, but the quality was still good.

The predictive coding used here is an alternative to delta modulation systems, which are most appropriate for medium or low quality applications, and to Vocoder codings, where one can expect relatively large equipment for high quality speech. On the basis of experiments to date it appears that one should be able to obtain digit rates of 30 kc with no discernible loss in quality from that of 48-kc companded PCM. The additional equipment required should be modest.

## 1. INTRODUCTION

The objective of this report is to describe the results obtained thus far from experiments using predictive coding on speech. It was the objective of these experiments to reduce the digit rate of sampled speech from that required by PCM at the Nyquist sampling rate, while retaining suitable quality.

### 1.1 Nature of the Problem

Considerable attention has been given in the literature to the theory of "prediction" as applied to communication systems, in particular to linear prediction. The idea is that if there is correlation present in a signal (or intersymbol influence) then it should be possible to make a prediction of the future of the signal, based on its past and present values. This is simply one procedure for reducing redundancy in the input signal, since if both the receiver and transmitter can predict successfully, all that need be transmitted is the discrepancy between the predicted value of the signal and the actual value at the moment of transmission. This "error" signal, then, contains all of the information present in the original signal. A block diagram of such a theoretical system is shown in Fig. 1. In the ideal case, the transmitted "error" signal should have much less power or less entropy (or both) than the original signal, but still contain all the information in the original signal.

Our object here will be to apply a particular type of "prediction" (linear prediction) to speech signals. The objective is to determine experimentally whether this prediction enables one to reduce digit rates of speech without seriously affecting quality.

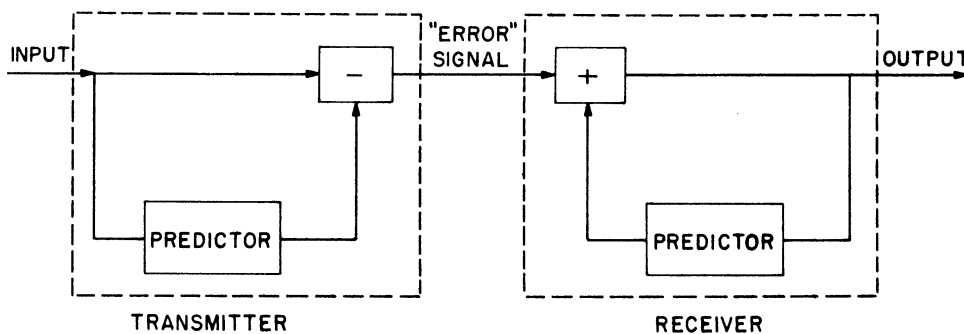


Fig. 1. Block diagram of theoretical prediction transmission system.

As will be noted below, predictive codings have been studied theoretically under various assumed conditions. On the other hand, various experimental tests (see Section 1.3) have been run on both speech and television signals in which "predictions" have been used in some way (Ref. 1). The names "feedback coders" and "differential quantizers" have sometimes been used to refer to these predictive codings. Also, the "delta modulations" may be regarded as a special form of predictive coding. In delta modulation the "prediction" is based on the error between the previous actual value and the previous predicted value.

In addition to the general prediction theories, and the various work on delta modulations and feedback coders, there has been some theoretical work (Ref. 2) which suggests that, under certain conditions, using a linear predicting law on speech with more than one past sample results in an improved quantization signal-to-noise ratio.

Based on the above, the objective of the work reported here is to determine experimentally the effects of using a linear predicting law to reduce the digit rate of speech without greatly deteriorating quality.

Although certain conclusions can be made, based on the experiments to date (see Section 4), it is clear that many more experiments have to be done before final conclusions are available.

## 1.2 Prediction Theories

In predictive coding for digital (or sampled analog) signals the essential idea is to predict the next (future) value of the input signal, and then send only the difference between the actual value and the predicted value. For this to operate successfully, of course, the receiver must be able to predict in the same fashion as the transmitter predictor. Predictors can be considered as being either: (1) statistical predictors, in which one attempts to make predictions according to the "long-time" statistics of the signal (based on some criterion), or (2) nonstatistical predictors, which are chosen on some reasonable basis (such as "linear extrapolation," constant curvature, etc.), and where the criterion is ultimately the terminal (or listener).

Statistical predictors have been described in the well-known prediction theory of Wiener (Ref. 3), and in a paper by Elias (Ref. 4).

For orientation it is worth reviewing the status of these two types of classical prediction. In general the prediction of any signal depends on the "intersymbol" influence. In other words, if there is no intersymbol influence then no prediction is possible.



Wiener's prediction theory is as follows: assume that the intersymbol influence is of a linear type (in other words, the most probable future value is determined by a linear combination of past values). Then if one assumes that the signal is ergodic of a known powerspectrum and if one uses the criterion of least mean-square error, then a prediction filter can be found. It should be noted that Wiener's filter will be optimum in the least mean square sense if the intersymbol influence is linear and if  $s(t)$  is ergodic. One might be able to do better with a nonlinear predictor but this cannot be assured except for the Gaussian case (where one knows that a linear filter is as good as the best nonlinear one).

In the statistical prediction theory of Ref. 4 it is assumed that the error signal is going to be coded (i. e. , a Shannon-Fano code). Therefore it is not sensible to design in terms of least mean-square error since the coding may change the power properties of the error signal. This predictive coding theory uses the criterion of "that predictor is best which leads to an average 'error term' distribution having minimum entropy" (best information-theory predictor). In general, this theory enables one to find the best "information-theory predictor" for some very special cases---first-order Markov processes. In practical situations it is virtually impossible to ascertain the necessary statistics for determining the best predictor (using this theory) so that an experimental variational approach is suggested.

When considering the use of nonstatistical predictors, there is no esoteric theory available and the analysis must be carried out in a tailored manner. Usually the predictor will still have to be evaluated in some statistical fashion, and some criterion (or measurement) must be used for comparison. One general criterion (suggested for television signals in Refs. 1 and 5) would be to use that predictor which gives best results in critical areas of the "terminal" (i. e. , observer, listener) and poor results in uncritical areas. This predictor may well not coincide with the best statistical predictor, but appears very sensible.

In the work here we take the viewpoint that the ultimate criterion is the "listener." In other words, we expect that any comparisons of quality will be made in terms of the judgments of subjective listeners. However, in the tests conducted, we have used the measure of quantization signal-to-noise ratio based on rms difference as a quantitative guide.

For the experiments here, a linear (nonstatistical) predictor is used. A linear

law implies that the predicted value of the next signal is simply the weighted sum of previous signal samples. This method of prediction is of particular interest when the signal is already sampled (as in a PCM communication system) and is also of interest because of its simplicity and relative ease of instrumentation. Thus, if  $f(t_i)$  are the previous values of the signal at sampling times  $t_i$ , the predicted value is given in general by:

$$f_p(t_i) = \sum_{i=-M}^{-1} a_i f(t_i) \quad (1)$$

where:  $M$  = number of past samples used in prediction.  
 $a_i$  = weighting of the samples in the prediction law.

For this work a piecewise "polynomial fitting" procedure is employed, resulting in the  $a_i$ 's of Eq. 1 becoming the binomial coefficients with alternating sign. Also, the  $a_i$ 's for  $|i| > M$  are zero if we use only the past  $M$  samples for the prediction. Then the  $f_p(t_i)$  can be written:

$$f_p(t_i) = \sum_{j=0}^{M-1} (-1)^{M-j-1} \binom{M}{j} f_{in}(t_{i+j-M}) \quad (2)$$

where:  $M$  = the number of samples to be used in the "estimation" of  $f_p(t_i)$   
 $f_p(t_i)$  = value predicted for sample time  $t_i$ , which is made at time  $t_{i-1}$   
 $f_{in}(t_k)$  = input sample values at sample time  $t_k$ .

This equation would be applicable to the system depicted in Fig. 1. The actual equation employed in our experiments will be a slight modification of Eq. 2---the prediction will be made on "quantized" input samples, so that the  $f_{in}$  will change to a quantized  $f_{ro}$ .

The "polynomial fitting" properties of this "linear law" may be seen by noting a few cases:

$$\underline{M = 1}$$

In this case, it is simply assumed that the predicted value is the value of the sample at the previous sample time. Thus, a "constant" is the "polynomial fit":

$$f_p(t_i) = f_{in}(t_{i-1}) \quad (3)$$

M = 2

Here the process is stepped back and it is assumed that the slope remains constant thus (see Fig. 2),

$$\text{slope at } t_i = \text{slope at } t_{i-1} \quad (4)$$

Since a slope can be computed from the two previous sample values:

$$\text{slope} = \frac{1}{\Delta t} [f_{in}(t_{i-1}) - f_{in}(t_{i-2})] \quad (5)$$

where:  $\Delta t = t_{i-1} - t_{i-2}$ , the sampling interval.

The predicted value  $f_p(t_i)$  can be obtained by

$$\begin{aligned} f_p(t_i) &= f_{in}(t_{i-1}) + \Delta t (\text{Slope}) \\ &= f_{in}(t_{i-1}) + f_{in}(t_{i-1}) - f_{in}(t_{i-2}) \\ &= 2 f_{in}(t_{i-1}) - f_{in}(t_{i-2}) \end{aligned} \quad (6)$$

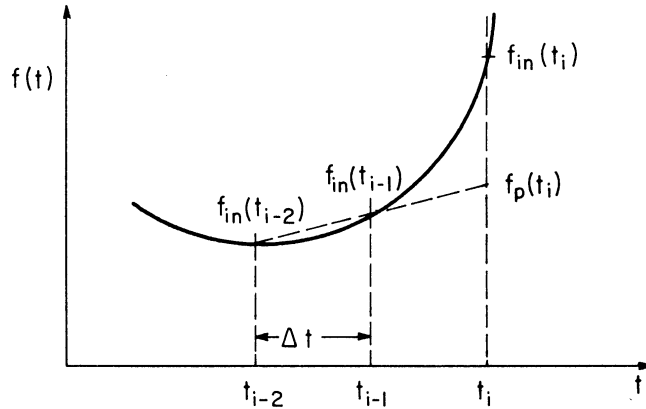


Fig. 2. "Slope" prediction (M = 2).

So, by knowing the previous two samples, at  $t = t_{i-1}$  and  $t = t_{i-2}$ , an estimate may be made of  $f(t)$  at  $t_i$ . This form of prediction is sometimes called "slope" prediction, and corresponds to a linear polynomial fit.

M = 3

In this case, a "curved" approximation is made for  $f(t)$  at  $t_i$ , based on the assumption that the "curvature" remains constant. Thus

$$\text{curvature at } t_i = \text{curvature at } t_{i-1} \quad (7)$$

Now, the curvature at  $t_{i-1}$  may be obtained by finding the "slope of the slope," thus,

$$\text{curvature at } t_{i-1} = \frac{1}{\Delta t} [\text{slope at } t_{i-1} - \text{slope at } t_{i-2}] \quad (8)$$

$$\text{where: slope at } t_{i-1} = \frac{1}{\Delta t} [f_{in}(t_{i-1}) - f_{in}(t_{i-2})]$$

$$\text{slope at } t_{i-2} = \frac{1}{\Delta t} [f_{in}(t_{i-2}) - f_{in}(t_{i-3})]$$

Assuming a constant curvature, then, the slope at  $t_i$  can be estimated as,

$$\text{slope at } t_i = \text{slope at } t_{i-1} + \Delta t (\text{curvature at } t_{i-1}) \quad (9)$$

Substituting, one obtains

$$\begin{aligned} \text{slope at } t_i &= \frac{2}{\Delta t} [f_{in}(t_{i-1}) - f_{in}(t_{i-2})] \\ &\quad - \frac{1}{\Delta t} [f_{in}(t_{i-2}) - f_{in}(t_{i-3})] \end{aligned} \quad (10)$$

Using this, the predicted value of  $f_p(t_i)$  can be obtained as before,

$$\begin{aligned} f_p(t_i) &= f_{in}(t_{i-1}) + \Delta t (\text{slope}) \\ &= 3 f_{in}(t_{i-1}) - 3 f_{in}(t_{i-2}) + f_{in}(t_{i-3}) \end{aligned} \quad (11)$$

This prediction corresponds to a quadratic or parabolic polynomial fit.

#### General M

By continuing the above process, the  $f_p(t_i)$  for M-sample prediction can be found to be (as was noted in Eq. 2)

$$f_p(t_i) = \sum_{j=0}^{M-1} (-1)^{M-j-1} \binom{M}{j} [f_{in}(t_{i+j-M})] \quad (12)$$

Note that, as M increases, the "prediction" is weighted more heavily on samples further back. Hence this particular prediction law would not be expected to be suit-

able for M greater than 3 or 4 unless one uses samples closer together (higher sampling rate).

### 1.3 Delta Modulators

The linear law predictive coding schemes being tested here are somewhat related to the various "delta-modulation" schemes which have been proposed and tested in the past 10 years, since they both employ feedback.

For this reason the results in delta modulation have pertinence here. In this section we will quickly review the various delta modulation schemes.

Delta-modulation ( $\Delta M$ ) systems transmit an "error signal" (or difference signal), similar to the predicting systems (Fig. 1). However, in  $\Delta M$ , the "predicted" (or subtracted) signal is based on the difference between the previous (sample) signal value and predicted value. Hence, the feedback loop in  $\Delta M$  acts primarily as a control loop to cause the receiver output signal to "follow" the input signal. In predictive coding, on the other hand, the predicted value is based on actual (but quantized) past samples.

While the original delta modulation schemes had "simplicity" as their chief objective, later systems seek lower digit rates at the expense of some complexity. When this is the objective, it is sensible to regard  $\Delta M$  as one form of predictive coding.

The "difference" signal in  $\Delta M$  may be unquantized or quantized, coded or uncoded, giving rise to the various systems described briefly below. In the systems to be described we will first note the "uncoded" ones, then the coded ones. The last system described is the most recent, the log differential PCM system.

1.3.1 Delta Modulation--Single Integration ( $\Delta M$ -SI). The block diagram of the one digit, single integration delta modulation system is shown in Fig. 3 (Refs. 6, 7, 8, 11, and 14). The difference between the present input signal and the (integrated) previously

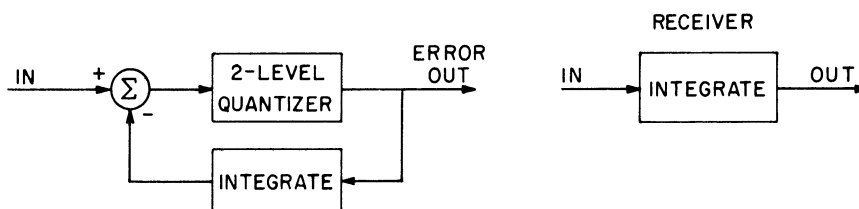


Fig. 3. Single-digit  $\Delta M$ -SI.

transmitted signals is quantized into two (positive or negative) pulses, according to whether the difference was positive or negative. At the receiver, an integrator simply adds up the incoming pulses to give an approximation to the original signal. This system is the original delta modulation as it was first conceived. The output pulses do not have to be strictly two-level; they may be PAM type, giving the magnitude of the difference, but the transmission advantages of uniform pulses are then lost. They may, however be encoded as in the system of Section 1.3.4.

The chief idea in such a system is to approximate the actual wave with a series of "equal step" staircase waves. This is illustrated in Fig. 4 for an arbitrary curve.

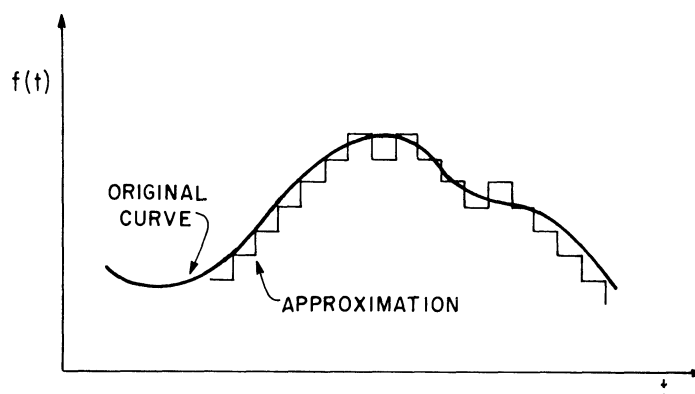


Fig. 4. "Staircase" approximation resulting with  $\Delta M$ -SI.

As would be expected, the quality from this type of system improves as the sampling rate increases. However, this would require an increase in bandwidth. Comparing  $\Delta M$ -SI with PCM, the  $\Delta M$ -SI has an  $(S/N)_q$  about equal to that of PCM for low bit rates ( $< 24$  kc)---corresponding to 3-bit PCM. However, for better quality speech bit rates ( $> 24$  kc), the  $\Delta M$ -SI has a lower  $(S/N)_q$  for a given bandwidth (or bit rate). Conversely, at these higher quality bit rates,  $\Delta M$ -SI requires more BW for a given  $(S/N)_q$  than does PCM. (See Fig. 11.) Nevertheless, the advantages to be gained in sending only 1 signal may make it worthwhile for certain applications.

1.3.2 Delta Modulation--Double Integration ( $\Delta M$ -DI). A method which improves the quantizing signal-to-noise ratio from that above is to use two integrators in the feedback loops of Fig. 3. Thus the block diagram would appear simply as shown in Fig. 5.

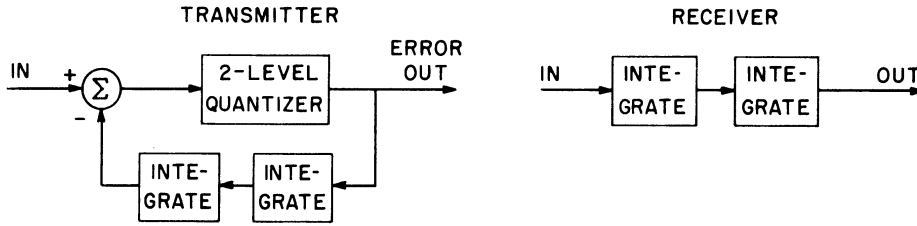


Fig. 5. Block diagram of single digit  $\Delta M$ -DI.

Accordingly, the receiver would need two integrators in the demodulating circuit, as shown in Fig. 5. The double integration system amounts to approximating the input signal with a series of short lines of varying slope rather than a staircase curve as is done by the single integration systems. Thus the approximating curve is a series of slopes, and adjacent slopes may differ by only one unit. This is depicted in Fig. 6.

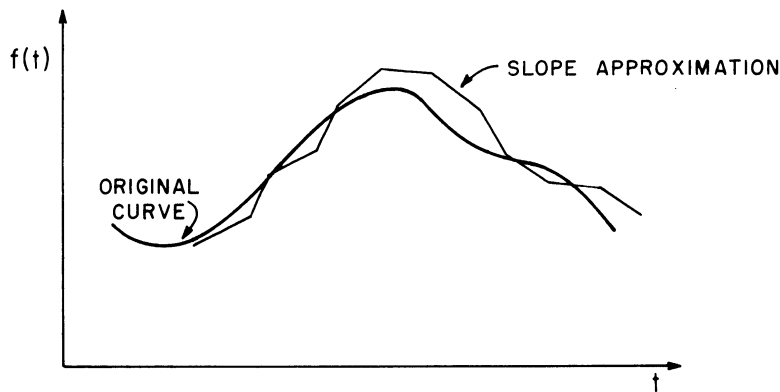


Fig. 6. "Slope" approximation resulting with  $\Delta M$ -DI.

Such a system will overload when the second derivative of the signal gets too large, and has an improved quantizing noise from that of the system above. Since there is a tendency for such a system to "hunt" (Ref. 9), a judicious choice of single integration and double integration is desirable. As far as we know no delta modulations of this type with 3 or 4 stage integration have been tried.

Substantial experiments with a modified  $\Delta M$ -DI are reported in Ref. 14. It is found that "fairly good" quality is obtained for a 32 kc bit rate, and "good" quality for 40 kc bit rate. These results are probably representative of "the best one can do" with  $\Delta M$ -DI.

1.3.3 Delta Modulation Using Exponential Decay. Delta modulation with exponential decay essentially is a modification of either of the previous two systems, where a "decaying exponential network" is used instead of an ideal integrator in Fig. 3 or Fig. 5. The network thus appears as an integrator for short periods of time, but allows the capacitor voltage to decay over long periods of time. The "exponential network" may be a simple RC condenser network. There are three advantages in using an exponential network instead of an ideal integrator: (1) the effect of channel errors disappears after some period; (2) one can transmit dc waveforms; and (3) one can operate with "single-sided" pulses, as opposed to two-valued pulses needed with an integrator.

Since using an RC exponential network instead of an ideal integrator is really a modification of the "ideal integrator" type, it is questionable whether one should treat this case separately. However, there are sufficient differences to warrant a separate section.

In the experiments of Ref. 14, a double-exponential circuit with single-sided pulses was used. Here we will briefly describe a single-exponential circuit with single-sided pulses, to note the results of using an exponential decay network instead of an ideal integrator. This system was studied and experimentally tried (Ref. 10).

With a single exponential network (and single-sided pulses) the signal voltage is approximated by a series of triangular-appearing pulses (actually small portions of exponential rise and decay curves) rather than by a staircase function as previously. The transmitted pulses are uniform positive if the signal voltage is increasing, or zero if the voltage is decreasing. Thus, a rising voltage is formed by a series of triangles added together, when the input signal is a series of pulses, while a decreasing voltage is formed by the decay of the condenser voltage when no pulses are transmitted (see Fig. 7).

The use of exponential decay exhibits several interesting properties. First of all, due to the capacitor charging-discharging action, a series of input pulses of given magnitude will have more effect on increasing the output voltage at a small output voltage level, than when the capacitor voltage is large (with single-sided pulses the waveform must be biased so that it is always positive). Similarly, the output voltage will decay more quickly when it is large, than when it is small, if the input pulses are absent. This action is desirable for any "oscillating" amplitude-limited wave, since it is the "expected" behavior of the waveform itself.

Another advantage of exponential networks concerns dc stability. A long series



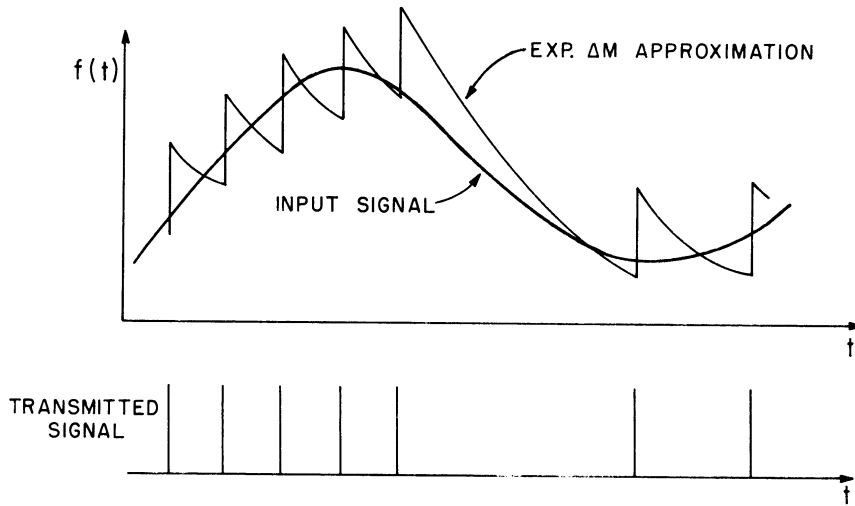


Fig. 7. Illustration of waveform approximation with single exponential network and single-sided pulse.

of transmitted "ones" will simply cause the output voltage of the RC network to reach a maximum (determined by the pulse shape and the RC network), whereas a long series of "ones" with ideal integrators will cause the output voltage to increase indefinitely, until the integrator saturates. A series of erroneous "ones" (such as might be caused by errors in transmission) will cause a permanent change in the dc level of an ideal integrator output---perhaps even close enough to the saturation level so that subsequent signal voltage outputs are clipped or distorted. However, because of the exponential decay characteristics of the exponential network, the output error due to an erroneous pulse sequence will gradually die out and the output voltage will return to the proper level.

This can be illustrated by considering the transmission of a fixed dc level. With ideal integrators, the dc level to be transmitted must be approached by a gradual curve (assuming a known starting point) which must be described by the transmitter, and then an alternating sequence transmitted to hold the output at that level. Thus, the transmission sequence might be:

. . . 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 . . .

that is, a sequence of "ones" until the level is reached, and then an oscillating signal with average change of zero about that point. Obviously, if errors occur in the above sequence of pulses upon transmission, a permanent change will occur in the receiver's average output voltage.

However, with exponential networks, all possible repetitive pulse sequences of reasonably short duration, from all "zeros" (minimum voltage) to all "ones" (maximum output voltage), correspond to a fixed average voltage level. Thus a . . . 1 0 1 0 1 0 1 0 1 0 . . . sequence corresponds to a dc average voltage midway between the largest and smallest. Choosing this sequence and transmitting it will assure that the receiver output is equal to that dc voltage (or rather, the average value of the output is that dc value). Since a repetitive signal is sent, the output voltage will fluctuate about the dc average. Any transmission errors (such as a burst of "ones") will perturb the output; but because the discharging characteristics of the RC network depend upon the output level, the output voltage will gradually settle back to the required value.

Except for the above characteristics, the single exponential  $\Delta M$  is quite similar to ordinary  $\Delta M$ -SI. A rough calculation (Ref. 10) of the quantizing signal-to-noise ratio shows that it is slightly more than that of  $\Delta M$ -SI, 1-digit. Reference 10 also gives a design procedure for exponential  $\Delta M$ , including how to choose the time constant of the RC network.

Although the above description of a single-exponential case illustrates the properties of exponential networks (in  $\Delta M$ ), a better  $(S/N)_q$  is obtained if one uses a double-exponential network (as in Refs. 7 and 14). All of the above advantages are gained, plus a closer approximation to the input signal. A double exponential case simply uses two RC networks instead of the integrators in Fig. 5.

With a single exponential network the  $(S/N)_q$  can probably be improved by use of a double-sided pulse rather than single-sided. For a double exponential network, however, a single-sided pulse appears satisfactory (Ref. 14).

1.3.4 Delta Modulation--Single Integration, n Digit ( $\Delta M$ -SI, n Digit). One of the possible attractions of the above systems is their simplicity. We now consider modifications of those systems which, although they increase complexity, improve the quality of the transmission. The first of these consists of a simple modification of Fig. 3---the adding of multiple levels to the single integration case.

This system is the same as that shown in Fig. 3, except that a multilevel quantizer is used, and the quantized (PAM) pulses are binary coded into a PCM code for transmission (see Fig. 8).

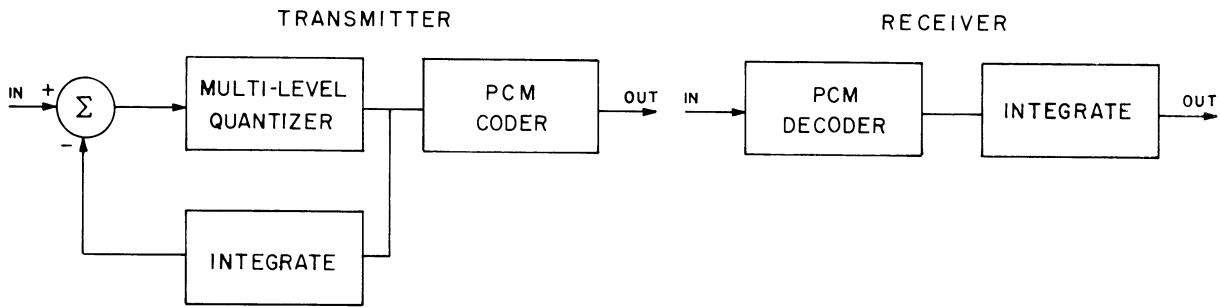


Fig. 8. Block diagram of  $\Delta M$ -SI, n-digit.

The chief idea now is that one has available steps of various amplitudes with which to make the staircase approximation to the actual signal. Thus a closer approximation to the input signal is made with this system compared with the case where fixed positive or negative pulses, and consequently only two quantization steps, are used.

1.3.5 Delta Modulation--Double Integration, n-Digit. The "delta-modulation, double integration, n-digit" system is simply the addition of a PCM coder and multilevel quantizer to the double-integration system of Section 1.3.2. Now, instead of the "sloping line" approximation to the actual curve being limited to "unit" changes in slope, various integer changes in slope can be accounted for with the use of the multilevel signals.

It may be noted, interpreting the previous systems in light of "prediction," that the predicted value is related to the difference between the previous signal and the previous predicted value.

1.3.6  $\Delta PCM_M$ .<sup>1</sup>  $\Delta PCM_M$  is the name used in Ref. 2 when a linear predicting law is used in the scheme of Fig. 1. Reference 2 describes theoretical calculations, using a "full-load sine wave" (which is suitable for evaluating S/N of speech) dealing with the quantization signal-to-noise ratio. The linear "predicting law" assumed is the alternating binomial series (or expansion in a geometric series) as described in Section 1.2. The manner in which this would be used is shown in Fig. 9.

The theoretical curves reported in Ref. 2, using the full-load sine wave, show that the quantization signal-to-noise ratio is improved as one increases the number of past samples used (M) above 1 in the range where the number of digits (n) exceeds 6. In other words, the quantization signal-to-noise ratio is improved (for  $M > 1$ ) if one already has a

<sup>1</sup>According to the terminology adopted in this report (Section 1.3) the  $\Delta PCM_M$  system is not strictly a "delta-modulation," rather it is one form of the more general "predictive coding."

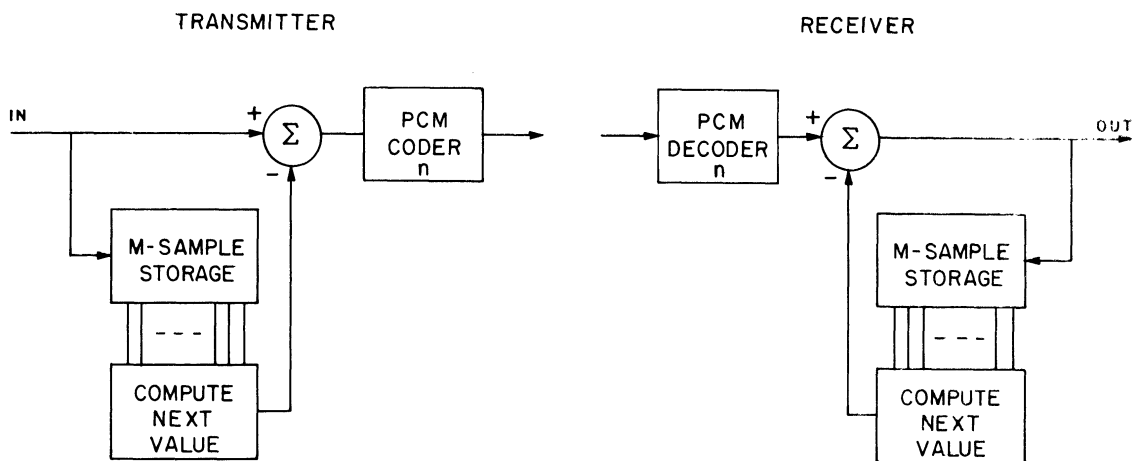


Fig. 9. Block diagram of  $\Delta\text{PCM}_M$  system.

good signal-to-noise ratio. However when one gets above 6 digits (at an 8-kc sampling rate) one already has sufficient voice quality for telephone communications. Hence any increase in quality obtained at the cost of complexity appears to be of doubtful value. Our interest in the work reported here will be to use a circuit similar to Fig. 9 to attempt lowering the digit rate for acceptable quality.

Note that a  $\Delta\text{PCM}_0$  system, where no previously transmitted terms are used in estimating the signal ( $M = 0$ ) is identical to an ordinary PCM system. The transmitted error is simply the actual value of the signal at the sampling time. The  $\Delta\text{PCM}_1$  system is similar to a "feedforward" version of the n-digit  $\Delta\text{M-SI}$  system. Here the predicted value of the input signal at the next sampling instant is the same as at the previous instant. Hence the error signal is simply the difference between successive samples.

Consider the comparison of a  $\Delta\text{PCM}_2$  with a  $\Delta\text{M-DI}$  case. For  $\Delta\text{PCM}_2$ , the two previous samples are used (to determine the slope) for predicting the next value. In  $\Delta\text{M-DI}$  the double integration also determines a slope, but here the slope is determined by the difference between the previous actual and the previous predicted value. Thus, although both cases involve "slope," they can be expected to behave somewhat differently.

Note also that in Fig. 9 the transmitter predictor is operating from actual samples, whereas the receiver predictor is having to base its values on quantized (since the transmission is PCM coded) values.

The predicting scheme which we will use is very similar to Fig. 9 except that a different feedback situation will be used. The "summer" will be at the input so that the

transmitter signal and receiver predictor will be working on the same values if there is no channel noise. It appears that this "predicting" difference would cause difficulty in operating a system such as Fig. 9, even in the absence of channel noise.

1.3.7 Log Differential PCM. A differential scheme recently reported which is being considered for use in Unicom is termed "log differential PCM" (Ref. 12). A block diagram of this scheme is shown in Fig. 10. As in the delta modulation schemes described above, this system bases its "predicted" value on the difference between the previous actual

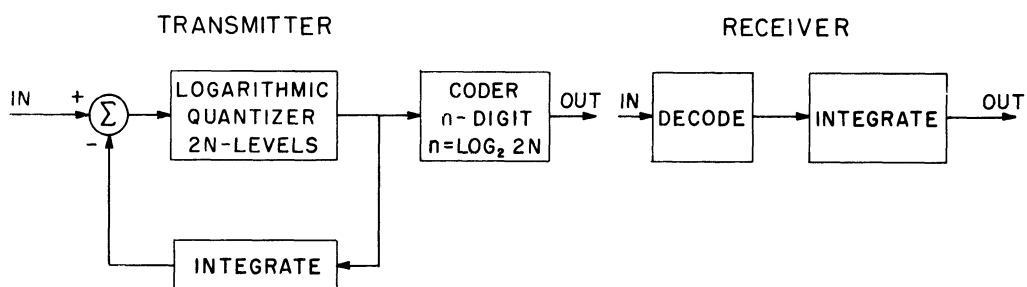


Fig. 10. Block diagram of log differential PCM.

and predicted values. There is one large difference, however, in that the error samples are quantized logarithmically (or companded). As with any companding, the chief value of this companding is to provide suitable quantization signal-to-noise ratios over a range of signal volumes. For example, companding is a necessary alternative to putting such things as a long-time AGC circuit on a given speech source.

The basic idea in companding the error signal is that both large and small step sizes can be utilized at any point on the signal wave. Thus if the prediction is in error by a large amount, one sends an error signal corresponding to this large error, but quantized very roughly. On the other hand, for all cases in which the "prediction" is reasonably correct, the quantization is rather fine to take advantage of the fact that one hopes to predict correctly most of the time.

Some of the chief results obtained with experimenting with this system (based primarily on sine wave calculations with a computer) are that 4-digit log differential PCM sampled at 9600 kc (38.4 kc bit rate) is about equivalent to 6-digit (48 kc bit rate) companded PCM. Thus one has been able to save roughly 10 kc in bit rate. Also reported is

the fact that 3-digit log differential PCM is theoretically calculated to be somewhere between 4-digit and 5-digit companded PCM. Consequently it appears that, as the digit rate is reduced, the advantage of log differential becomes smaller. For this reason it is believed that 4-digit log differential PCM sampled at 9600 kc is the most reasonable rate for PCM speech transmission.

When taking into consideration the fact that any system has to operate suitably over a range of (long-time average) volumes, the companding in the feedback loop as shown in Fig. 10 is a very sensible operation. For this reason experiments with this type of companding were tried in this work and reported below.

1.3.8 Some Comments on Delta Modulation. In general the early motivation for the delta modulation schemes described above was to obtain simplicity over PCM. In other words it was regarded as an alternative to digitally encoding by PCM methods. The basic unmodified systems require more bandwidth for a good quality signal-to-noise ratio (bit rate  $> 24$  kc) than does PCM. However, all of the modifications of the basic systems tend to change this. The later tendency has been to combine "delta" methods with PCM in order to obtain good quality for lower bit rates. For medium quality, however, it is found that a basic system with double exponential networks is suitable (Ref. 14). When adding PCM to "delta" methods it is clear that the delta methods are not an alternative to PCM, but rather an efficient coding for PCM (predictive coding). This is especially true of the  $\Delta\text{PCM}_M$  and the log differential method.

For any delta modulation system it is necessary to keep in mind the effect of even infrequent channel errors. At best, the system must be arranged so that channel errors quickly die out in the receiver signal. It can be safely concluded that practically all delta modulation systems are constricted to those areas where the channel errors are fairly infrequent.

Such signals as speech (with a drooping frequency characteristic above 800 cps) are especially appropriate for the differential (or delta) methods. Also, as would be expected, the signal-to-noise ratio of all delta modulations improves as the sampling rate goes up. However it is not in our interest to increase the sampling rate since we wish to decrease the "bit" rate for good quality. Therefore our objective is to look for better "predictors" which may be used at the Nyquist sampling rate, but which will require very few amplitude levels for good speech.

Figure 11 below is a summary graph of the theoretical signal-to-noise ratios of the various exponential delta modulations discussed above. Theoretical curves are shown for the following cases: (1)  $\Delta\text{PCM}_M$ , (2) PCM, (3)  $\Delta\text{M-SI}$ , n-digit, (4)  $\Delta\text{M-DI}$  (1-digit), (5)  $\Delta\text{M-SI}$  (1 digit), and (6) single exponential delta modulation. Quantization S/N is plotted versus "bit rate." For all n-digit cases, the sampling rate is 8 kc, so that the bit rate equals 8 kc times the number n. For all the 1-digit cases (shown in light lines), the bit rate equals the sampling rate.

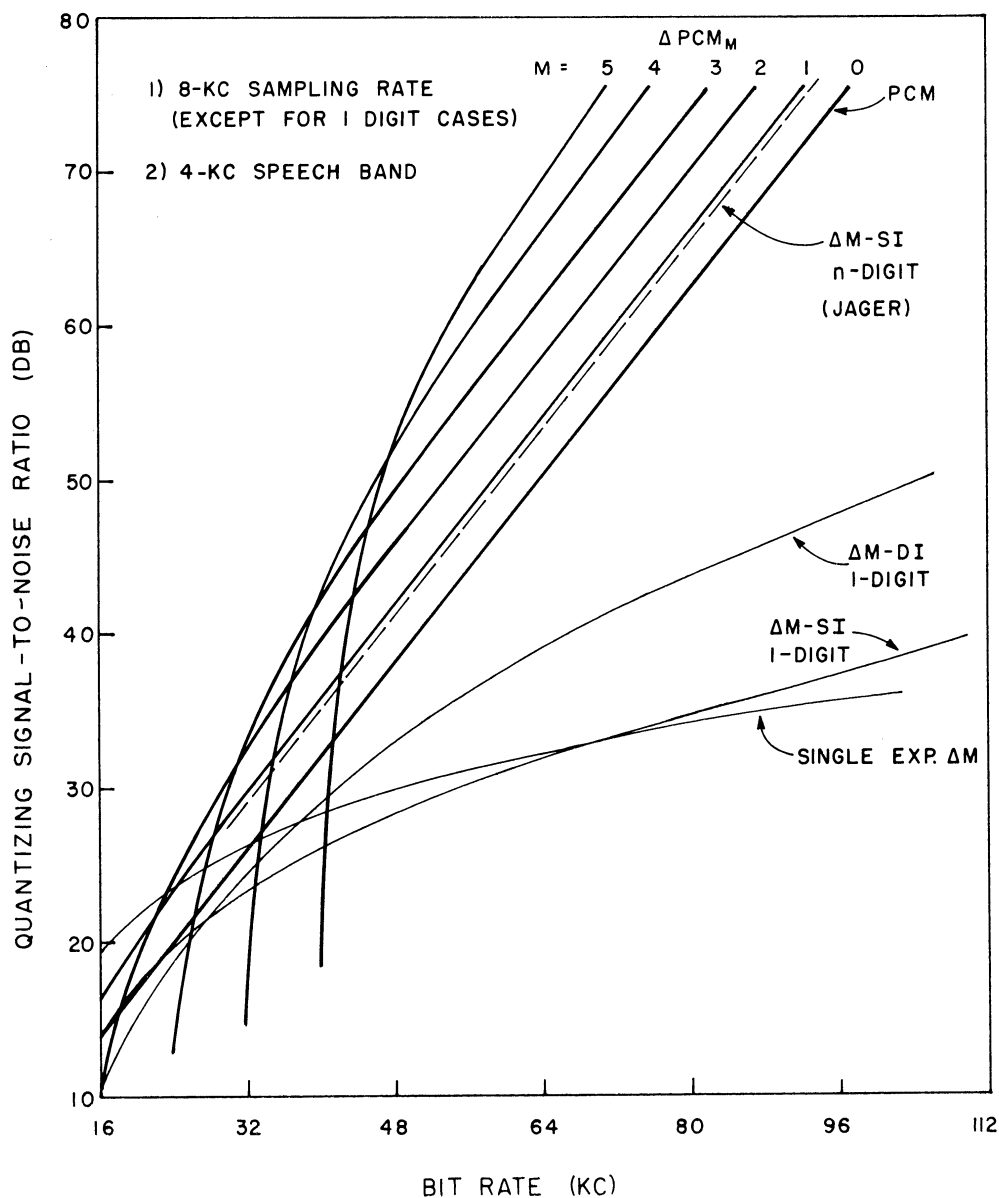


Fig. 11. Theoretical quantization S/N versus "bit rate" for delta modulation.

The  $\Delta\text{PCM}_M$  curves of Fig. 11 were obtained by plotting Eq. 3-11a of Ref. 2. It may be noted that these curves are not the same as those plotted in Ref. 2 (at the lower bit rates), since the approximation involved was not used. The  $\Delta\text{M-SI}$ , n-digit curve was taken from Ref. 7. The  $\Delta\text{M-DI}$  and  $\Delta\text{M-SI}$  were taken from the full-load sine wave calculations of Ref. 2.

The curve for the single exponential delta modulation was approximated by assuming a worst-case shape for the error voltage and determining its rms value. Using the results of Ref. 10, an estimate can be made of error rms value for a given sine wave, as a function of the sampling rate. The error is a function not only of the sampling frequency and the input signal, but also of the time constant of the RC network. Thus, families of curves may be drawn for different RC values. The curve indicated in Fig. 11 is a representative one drawn for the "optimum" value of RC time constant, determined by using a typical speech amplitude vs. frequency plot as given in Ref. 10. The procedures described in Ref. 10 were used to determine the frequency of the reference sine wave and its amplitude, as well as the RC time constant of the decoding network. Thus, the ratio of the rms of the full-load sine wave to the rms error can be plotted as a function of sampling frequency.

It must be cautioned that the curves of Fig. 11 are theoretical calculations based on the various assumptions used. Actual conclusions as to the comparison of the various systems for speech would probably have to rest on extensive subjective listening tests.

Based on the curves, however, the  $\Delta\text{M-SI}$ , n digit is consistently better than PCM at all rates. For very high bit rates,  $\Delta\text{PCM}_M$  shows very high S/N ratios---too high for most purposes. At lower bit rates (about 32 kc), it appears that single exponential  $\Delta\text{M}$ , and  $\Delta\text{M-SI}$ , n digit are slightly better than  $\Delta\text{M-DI}$ ,  $\Delta\text{M-SI}$ , or PCM. For the very low bit rates (16 kc), exponential delta modulation appears better than the others. The experimental results of Ref. 14 also indicate that, at the low bit rates, a double exponential network results in  $(S/N)_q$  superior to that of PCM.



## 2. OBJECTIVE OF SIMULATION STUDY

The major objective pursued in the experimental study being conducted here is to determine whether linear predictors which use one or more past samples are of value in retaining good quality speech while reducing the digit rate. From the above description of delta modulation systems it is clear that a great many results and conclusions are known for this special form of "predicting coding." Our objective then, in light of the known results, is to seek improvements in quality for low digit rates with linear predictive coding.

Based on the theoretical results (and other considerations) for  $\Delta\text{PCM}_M$  noted above in Section 1.3.6, it was decided to begin with a linear predicting law where the prediction equation is given by Eq. 2 and Eq. 12. This was chosen because of its simplicity, and because such a linear prediction law has a "curve extending" interpretation: if one sample is used, one is making a prediction that the next sample will be the same as the past sample; if two past values are used, one is making the prediction that the slope between the past two samples will remain in force for the next sample; if three samples are used, one is predicting that the curvature between the last three samples will be retained for the next sample, etc.

Because of the necessity of eventually running a receiver predictor in synchronism with the transmitter predictor, a feedback loop of the type shown in Fig. 12 was chosen as the scheme to be used with the prediction law. This is also identical to the prediction experiments run on television signals in Ref. 1.

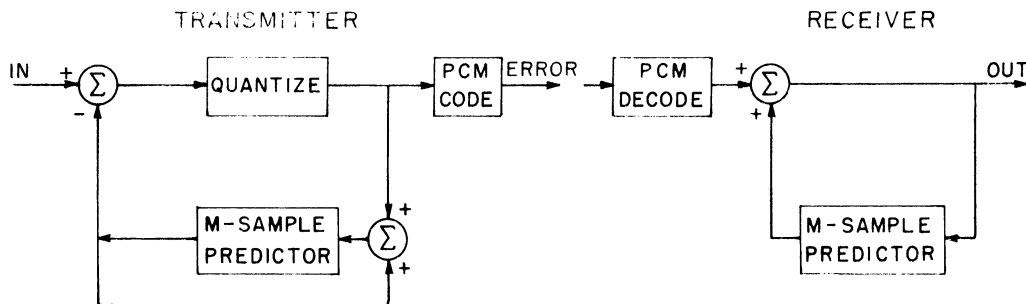


Fig. 12. Block diagram of predictive coding system used in experiments.

Assuming that sampled speech (analog or very finely quantized samples) is put into the input, the first operation is to subtract the predicted value from the present sample. This predicted value is calculated from past (quantized) samples using various values of  $M$  as denoted in Eqs. 2 and 12. The error signal is then quantized according to some characteristic which is directly related to the number of PCM digits which will be required to transmit the error signal. Experiments were run in which the quantizing of the error was done according to an ordinary PCM method and also according to a companded method (as in Section 1.3.7 above).

Note that there are some fundamental differences between the operation of Fig. 12 and that of Fig. 9 ( $\Delta\text{PCM}_M$ ). Most of these differences center about the fact that the transmitter of Fig. 12 involves a feedback loop, whereas that of Fig. 9 does not. First of all, this means that the receiver predictor of Fig. 12 can operate on the same (quantized) values as the transmitter predictor (in the absence of channel noise). In Fig. 9, the receiver samples are quantized while the transmitter ones are not.

The two systems also differ in their reaction (or transient response) to occasional limiting (or clipping). The feedback situation of Fig. 12 permits design of "stability" between transmitter and receiver, whereas that of Fig. 9 does not. Since any "efficient" operation will permit overloading a small amount of time, this is a most important consideration.

When investigating alternatives for experimentally conducting the above desired prediction studies, three methods were considered: (1) simulation (of the system) on a digital computer, (2) simulation on an analog computer, or (3) building an actual analog prediction circuit either with digital components or by means of passive filters. It was concluded that simulation on a digital computer was the most attractive because of the inherent flexibility in the various kinds of signal processing which may be performed, and the ease with which system parameters may be changed. It was seen rather quickly that the major drawback to digital computer equipment simulation is the rather complex equipment needed to get an analog signal into a form suitable for operations on a digital computer. Since the ultimate requirement in any speech transmission system is intelligibility and recognizability of the actual speech signal, it was desirable to perform the evaluation tests with actual speech inputs and outputs; hence the computer input-output problem.

Another problem was the necessity for writing an "efficient" program, to keep the computer time-to-real time ratio reasonable. This requires writing in machine language instead of more simple translator programs.

After investigating the three methods it was decided that the computer simulation was clearly the best choice if the input-output problem could be overcome. After some search it was found that a special purpose input-output converter was available to be modified for use in our experiments.<sup>2</sup> The equipment converts ordinary analog speech (on tape or spoken into a microphone) into high quality binary PCM (recorded on magnetic tape). This tape can then be operated upon by the computer in a number of ways to simulate various communications systems. In the reverse process, the digital output from the computer may be converted back to analog audible speech. The equipment is described in more detail in Section 3.

This equipment, plus a highly efficient computer program, allows one to perform a wide variety of experimental tests on digital encoding techniques. We are convinced that the computer technique is the proper one for experiments on digital encoding of this type. Reference 13 is an excellent source for a description of the computer simulation method and its advantages.

As mentioned above the major objective of the experimental studies conducted here was to determine the value of using a linear prediction law on speech. Towards this end, some initial full-load sine wave experiments were run in order to determine the qualitative behavior of the various  $M$  predictions.

The actual method used was first to run full-load sine waves through the computer and have it compute an rms signal-to-quantization-noise ratio. Based on these results and various behaviors in the loop shown in Fig. 12, the "ground rules" were learned concerning using one or more past samples ( $M \geq 1$ ). With such ground rules established, the actual speech experiments were run using digit rates lower than the currently used 48 kc. In addition to calculating the rms quantization signal-to-noise ratio on the speech, some qualitative aspects of the speech quality were ascertained from listening to the tapes.

The first speech tests concerned bit rates of 41.6 kc, an  $M = 1$ , and companding quantization. The motive here was to compare these results with those of Ref. 12, which use a "delta" method under similar circumstances. Then a variety of speech experiments were run using a 30-kc bit rate, and attempting to find the better methods (see Section 4).

<sup>2</sup>This equipment was originally built under the auspices of the Research Department of the Bell Telephone Laboratories, for use in conjunction with research on speech.

### 3. EXPERIMENTAL PROCEDURE

The essential method used in these experiments was to have a 709 digital computer simulate the actual digital equipment necessary for a "predictive coding" transmitter and receiver. Note that the equipment is simulated; the test is actually an experiment (and not a simulation) since actual speech was used.

Figure 13 shows a block diagram of the functional operation of the test. The initial input is very finely quantized speech samples (which can be regarded as analog). The computer then performs the indicated "loop" operation on the samples. Note that the "predictor" is utilizing sample values which are quantized by the "error" quantizer.

In the experimental tests the transmitter and receiver predictor were identical (corresponding to no channel noise). This means that the receiver samples would be identical to the inputs to the predictor (called  $f_{ro}$  on Fig. 13). In this way the tests could skip the operation of binary encoding and decoding which would be required in practice.

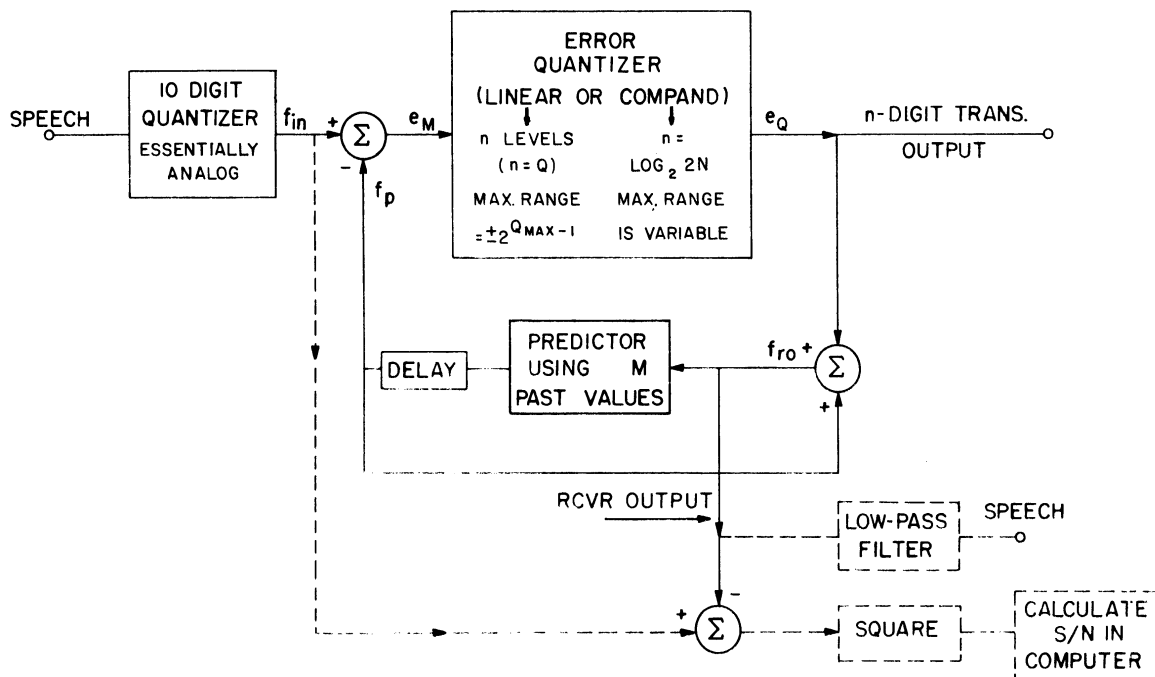


Fig. 13. Block diagram of experiments showing operations and notation.

The dashed lines in Fig. 13 indicate the  $(S/N)_q$  calculation and the "listening" aspect of the tests. Figure 13 should be referred to when dealing with the computer program of Section 3.4.

In addition to calculating the S/N ratio (and processing the signals), the computer also noted the number of times that either the error quantizer  $e_M$ , the predictor output  $f_p$ , or the receiver output  $f_{r0}$  exceeded their limits. In addition, the largest value for each of these items was noted.

### 3.1 System Block Diagram

Figure 14 shows the operational block diagram of the entire system, showing each of the steps in the experimental procedure. The equipment consists of an audio tape recorder, a low-pass filter, an analog-to-digital (or digital-to-analog) sampler and converter, and a digital tape recording unit. In addition, the necessary synchronizing and control circuits are provided, along with circuits to check the completed digital tapes. The basic equipment was obtained from the Research Department of the Bell Telephone Laboratories, after which modifications were made to bring the equipment into conformity with the IBM 709 tape characteristics and the necessary operating characteristics of The University of Michigan Computing Center. A brief description of the entire sequence of operations will be given below after which some of the individual steps in sequence will be discussed in further detail.

In the recording mode an audio input signal is first recorded on the audio tape recorder. This recording is done at 15" per second speed. This recording is then played at half-speed into the analog-to-digital converter. This 2:1 speed change is necessitated by the fact that the maximum sampling rate available with the existing tape unit and recording equipment is 5 kc. To achieve a real time sampling rate in the neighborhood of 10 kc, the input signal is played in at half-speed. The "sampler and converter" samples the input signal from the audio tape recorder and converts it into a 10-digit binary code representation of the value of the sample voltage. These 10 digits are transmitted in parallel fashion to the digital recording electronic circuits. There the 10-digit binary sample is gated in two groups of 5 digits to the 7-track digital tape unit, where it is recorded on standard IBM tape. The sixth track of the 7-track tape is recorded with a "0-1" sequence to indicate to which portion of the 10-digit sample the accompanying 5 digits belong, i. e., the "front" or the "back" half of the 10-digit word. The seventh track contains the parity check digit across

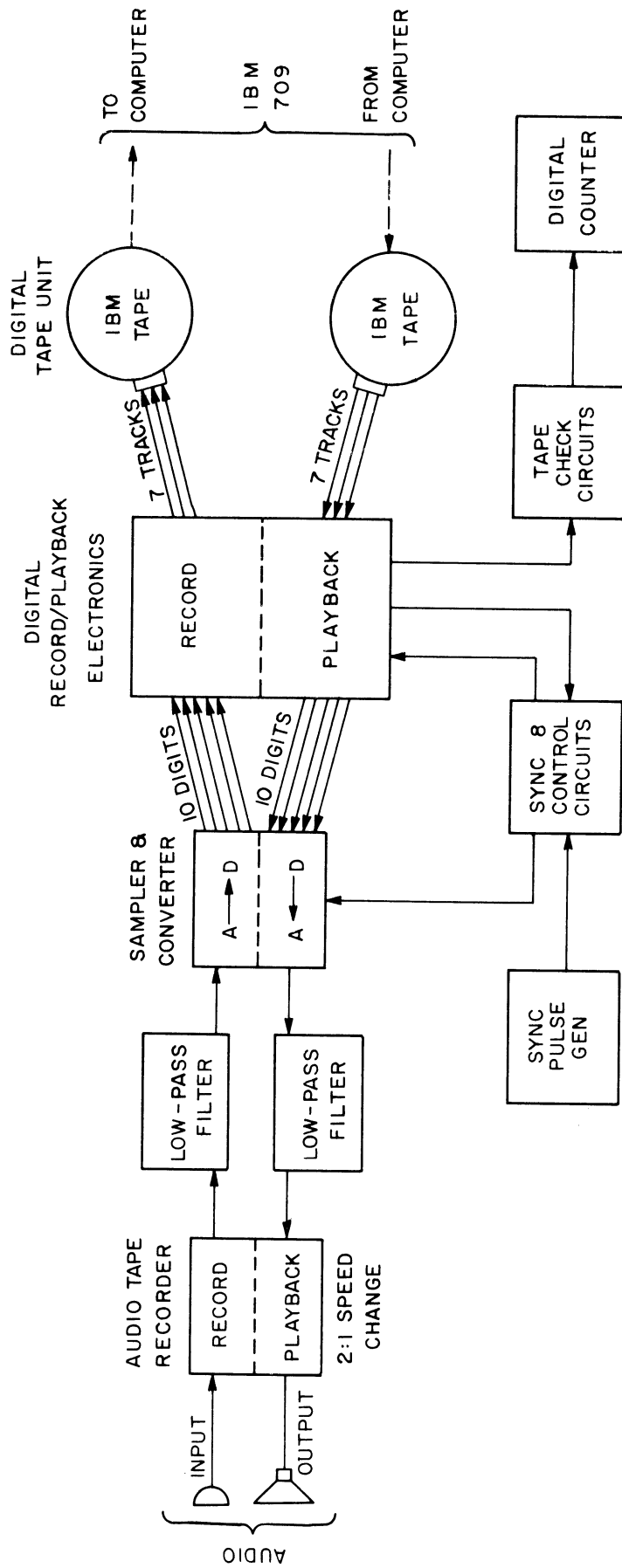


Fig. 14. Block diagram of experimental predictive coding system.

the other six tracks in the standard IBM binary mode. This procedure makes the tape compatible to the IBM tape units used with the 709 computer. The completed digital tape is then taken to the Computing Center where, in conjunction with the written program, it is placed on the 709 computer. The computer then performs the indicated computations to simulate the desired predictive coding system.

Upon completion of the computer calculations, the computer records the output signal on the IBM tape. This is the processed and reconstructed "receiver" output. The reverse procedure then is followed in the playback mode. The IBM tape is placed on the tape unit and played into the digital playback electronics equipment. There the two 5-digit half-words from two recorded lines of tape are re-assembled into the corresponding 10-digit binary sample. This sample is then transmitted in parallel form to the digital-to-analog converter. The digital-to-analog converter produces a voltage value corresponding to the binary number represented by the 10-digit binary sample. The sequence of voltage pulses is passed through a low-pass filter and recorded on the audio tape recorder at the 7-1/2" per second speed. Then, in order to listen to the results of the test, the audio tape is played at the 15" per second speed through an audio amplifier and speaker.

The synchronization and control circuits perform, as their name implies, the proper controlling and timing functions for both record and playback cycles. In addition, circuits are available for checking the recorded digital tape by counting the number of records in the tape and the number of recorded tape characters in each tape record.

### 3.2 Basic A to D Converter

Figure 15 shows the simplified block diagram for the analog-to-digital converter. This is an Epsco Digital Voltage Translator, B-DATRAC. In the analog-to-digital mode it accepts input voltage samples and then produces 10 dc output voltages representing the binary equivalent of the value of the sampled voltage. The external trigger starts the sampling and binary countdown cycle to produce the 10-digit binary representation of the voltage. In brief, the operation of the analog-to-digital converter is as follows: the sampled input voltage is compared to the voltage output of the binary switch tubes. By means of fixed resistors, each binary switch tube can add a fixed increment of dc voltage in proportion to its position in the 10 tube chain, i. e., in powers of two. An error in the voltage sum produced by the ten-switch tubes as compared to the input sample voltage causes a chain of events which results in the switching of the appropriate binary switch tubes until the

voltage output of the 10 binary switch tubes most closely matches the input sample. The resulting states of the 10 binary tubes, "off" or "on," represent the 10-digit binary number corresponding to the input sample. These output states then are transmitted in parallel form to the record electronics circuits.

In the playback mode, or the digital-to-analog mode, input dc voltages from the playback electronics circuits are fed directly to the 10 binary switch tubes and are used to set the switch tubes to the particular state representing the input binary number. The input amplifier and the sample-and-hold circuit are disabled. The voltage established across the fixed resistors switched by the binary switch tubes is the output sample voltage in the digital-to-analog voltage mode. This sequence of pulse-amplitude-modulated voltages is then smoothed by passing through a low-pass filter.

### 3.3 Record-Playback Electronics

Figure 16 shows the simplified block diagram for the record-playback electronics circuits. In the recording mode the circuits accept the 10-voltage outputs from the 10 binary switch tubes on the analog-to-digital converter and gate these, in groups of 5 digits, into the record triggers. This must be done because the IBM tape format allows only 7 tracks, one of which must be a parity check track. So the recording circuits gate five digits at a time into the recording triggers, along with a "0" or "1" into the sixth trigger (or the corresponding track "B"). This "0" or "1" code indicates which five digits of the entire 10 digit binary word are recorded on a particular line of tape, as it takes two lines of tape to record one 10-digit sample. At the same time a lateral check digit computer computes the value of the digit to be recorded in the seventh track, or track "C". This check digit is required by the IBM 709 tape units, which automatically check for its presence. A "1" must be recorded in this track whenever there is an even number of ones recorded in the other six tracks. This seventh digit constitutes an odd parity check across the tape.

In addition, at the end of each record the computer tape units require that an additional word be recorded a specified distance from the last recorded line on the tape. This word must be an even parity check down the length of the tape. That is, for each track of the tape a "1" is recorded if the total number of "1's" recorded down the whole record for this track is odd, and a "0" is recorded if the total number of "1's" recorded



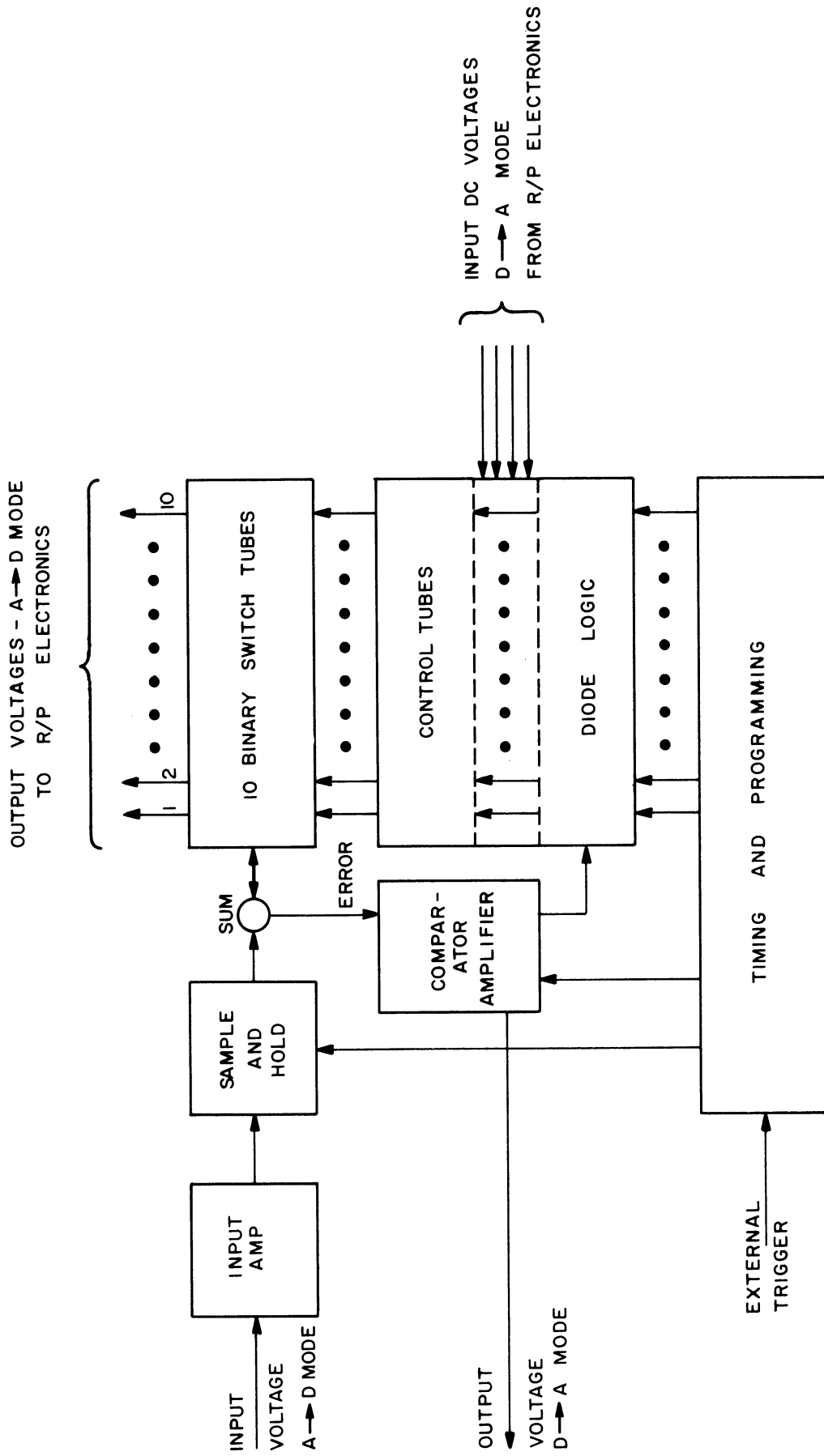


Fig. 15. Block diagram of analog-to-digital converter.

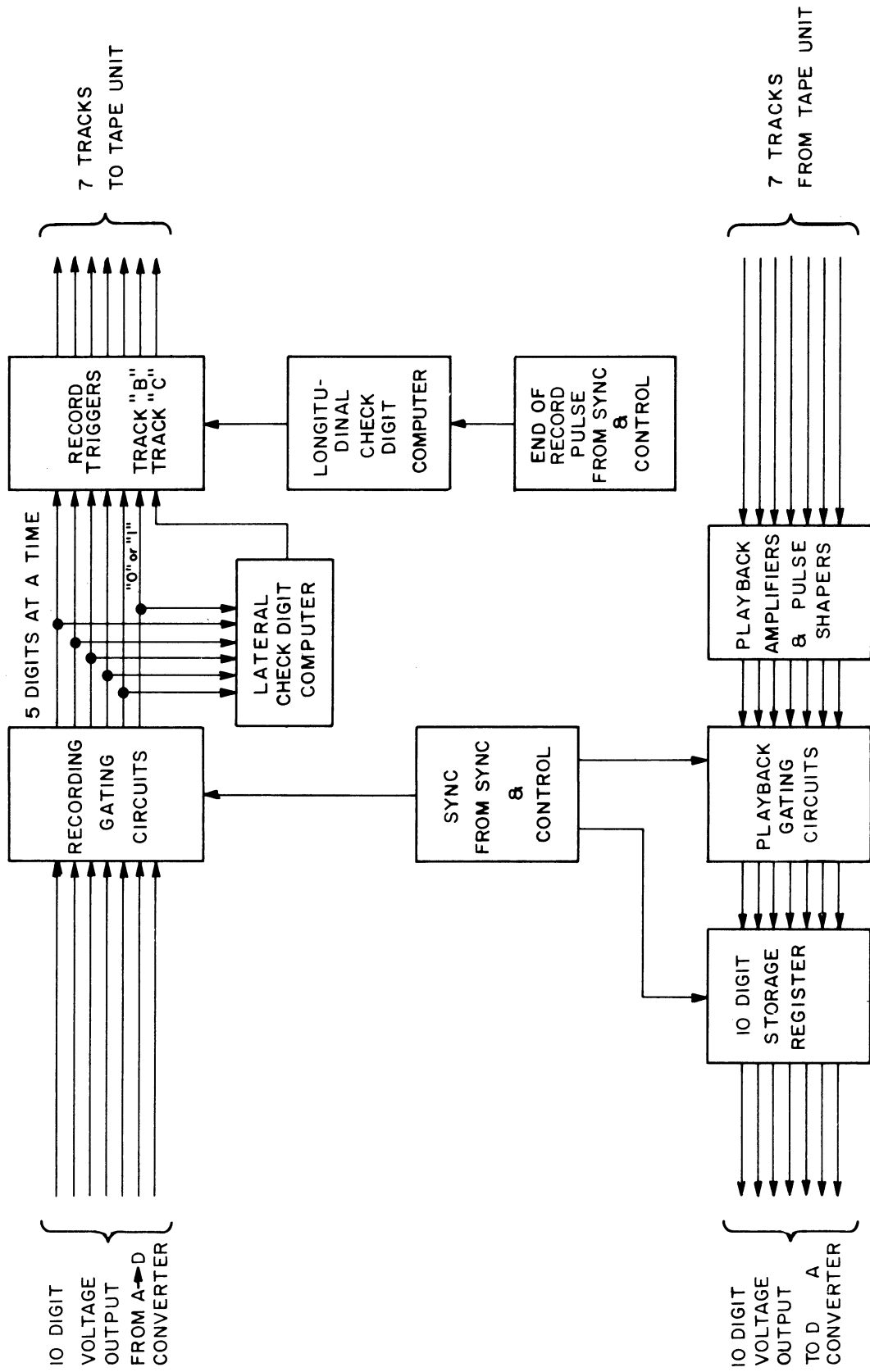


Fig. 16. Block diagram of record/playback electronics associated with A-D converter.

down the length of the record for this particular track is even. This longitudinal check word is recorded by the "end of record" pulse obtained from the synchronization and control circuits. Since the recording for this equipment is the "nonreturn to zero" form (a change in the direction of magnetization is a "1" and no change is a "0") the recording triggers changed state with each "1" input. Consequently, to provide the longitudinal parity check one need only set these triggers to the "0" state at the end of the record. This automatically provides the proper longitudinal check digits.

In the playback mode, the outputs of the seven tracks from the tape unit are fed into the playback amplifiers and pulse shapers. At this point the pulses from the seventh track "C" are simply dropped since they are of no importance in the rest of the cycle. The playback gating circuits gate the five information digits into the proper places in the 10 digit storage register, as determined by the "0" or "1" pulses in the sixth track. When two lines of input have been read from the tape and the complete 10 digit binary word assembled in the 10 digit storage registers, a gate pulse transfers these 10 digits to the digital-to-analog converter, which, in turn, produces the voltage amplitude corresponding to the binary number represented by the 10 digits.

#### 3.4 Computer Program

A simplified block diagram of the computer program is shown in Fig. 17. The program has been written to simulate the predictive coding scheme where a function of the output signals at M previous sampling times is computed by the "transmitter" and used to predict the value of the input signal at the next sampling time. The signal which is then transmitted is a function of the difference between the predicted and the actual signal values. The "receiver" performs the inverse operations and computes the values of the output signal samples. This operation is described in Fig. 13.

Table I summarizes the notation used in the computer program discussions. It should be noted here that all quantities involved in these computations in the computer are integers less than 512. That is, the entire range of the sample values (plus or minus 512) is represented by a 10 digit binary number ( $2^{10} = 1024$ ).

The program accepts the data input samples as recorded by the recording equipment onto the standard IBM tape. The input signal samples are grouped into "records" on magnetic tape. For each sample, the following set of computations is performed (the notation is defined in Table I). The predicted value of the input signal is formed by the linear

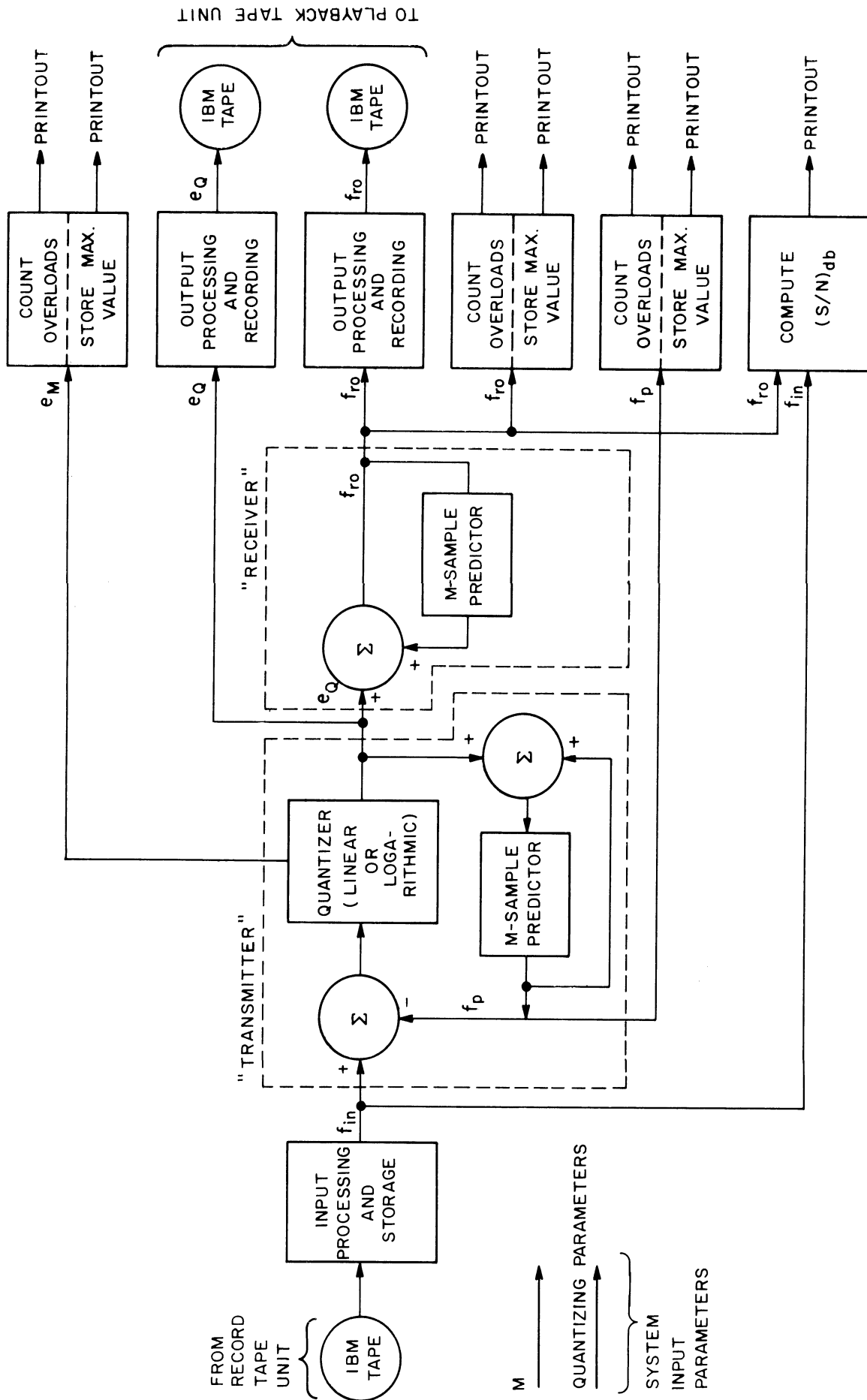


Fig. 17. Block diagram of computer program for predictive coding experiments.

$t_i$	the instant of time at which the input signal is sampled ( $t_i - t_{i-1}$ is the sampling interval and is constant for all $i$ ).
$f_{in}(t_i)$	the input signal sample at time $t_i$ .
$f_p(t_i)$	the predicted input signal.
$f_{ro}(t_i)$	the output signals computed by the receiver.
$e_M(t_i)$	the difference between the predicted and the actual input signals.
$e_Q(t_i)$	the quantized value of $e_M(t_i)$ .
$M$	the number of points used to compute $f_p(t_i)$ .

Table I. Computation notation.

law (discussed in Section 1.2):

$$f_p(t_i) = \sum_{j=0}^{M-1} (-1)^{M-j-1} \binom{M}{j} [f_{ro}(t_{i+j-M})] \quad (13)$$

except that, for the first  $M$  samples on each tape "record,"  $f_p(t_i)$  is set equal to  $f_{in}(t_i)$ . (This is a necessary "starting" method for the feedback loop.) The error term,  $e_M$ , is the difference between the predicted and the actual values:

$$e_M(t_i) = f_{in}(t_i) - f_p(t_i)$$

Before transmittal (or feedback),  $e_M$  is quantized into the term  $e_Q$ . Either of the methods (linear or companded) described below may be used here. The receiver then computes the output signal from the transmitted signal and the previous output signals.

$$f_{ro}(t_i) = e_Q(t_i) + f_p(t_i) \quad (14)$$

If any  $f_p(t_i)$ ,  $e_M(t_i)$ , or  $f_{ro}(t_i)$  exceeds its maximum allowable absolute value, that quantity will be replaced by its maximum before computation proceeds. The maximum value for  $|f_p|$  and  $|f_{ro}|$  is 511; the maximum for  $e_M$  is defined below.

These computations are then performed for the next instant of time,  $t_{i+1}$ . The  $e_Q$ 's and  $f_{ro}$ 's are accumulated and written by records onto separate tapes. When all the input has been processed, the signal-to-noise ratio is printed.

$$(S/N)_{db} = 20 \log_{10} \sqrt{\frac{S^2}{N^2}} \quad (15)$$

where: 
$$S^2 = \sum_i f_{in}^2(t_i)$$

$$N^2 = \sum_i [f_{ro}^2(t_i) - f_{in}^2(t_i)]^2$$

The  $e_Q$ 's and  $f_{ro}$ 's are then copied onto the input tape immediately after the  $f_{in}$ 's. The S/N ratio is (for speech) computed over an entire test length which includes a number of "records" (see above).

The program is set up so that methods of quantizing  $e_M$  into  $e_Q$  may easily be changed. At the present time, two methods for quantizing  $e_M$  are available, in subroutines named QUAN1 and QUAN2. The equations below are only for positive values of  $e_M$ , since all quantizing schemes must be symmetric around the origin.

QUAN1 (linear): Given the parameters QMAX and  $Q = n$ , such that  $2 \leq Q \leq$   
QMAX  $\leq 10$ ,

$$e_Q = \left[ \frac{e_M}{2^{QMAX-Q}} + \frac{1}{2} \right] 2^{QMAX-Q} \quad 0 \leq e_M \leq 2^{QMAX-Q} (2^{Q-1} - 1) \quad (16)$$

where square brackets indicate the greatest integer in the quantity enclosed, and

$$e_Q = 2^{QMAX-Q} (2^{Q-1} - 1) \quad \text{otherwise} \quad (17)$$

This is a step function where the height of each step is  $2^{QMAX-Q}$  and the maximum value of  $e_Q$  has no more than QMAX binary digits (including the sign bit).

QUAN2 (Companded): In this method, the height of the first two steps is approximately equal, the third step is the sum of the heights of the first two, and each subsequent step is twice the height of the preceding one. The parameters to be specified are 0L1, the

value of  $e_Q$  for the first step, and  $N$ , the maximum number of such steps (on each side of the origin). The restrictions on the parameters are:  $1 \leq 0L1 \leq 511$ , and  $1 \leq N$ .

$$e_Q = 0L1 \cdot 2^n \quad 0 \leq e_M < \left[ \frac{3 \cdot 0L1}{2} \right]$$

$$e_Q = 0L1 \cdot 2^n \left[ \frac{3 \cdot 0L1 \cdot 2^{n-1}}{2} \right] \leq e_M < \left[ \frac{3 \cdot 0L1 \cdot 2^n}{2} \right] \quad (18)$$

That is, where  $e_M$  is in the interval  $< [A]$ ,  $e_Q = \frac{2}{3} A$ . Two limitations apply to the above quantizing method:

$$e_Q = 0L1 \cdot 2^{N-1} \quad e_M \geq \left[ \frac{3 \cdot 0L1 \cdot 2^{N-1}}{2} \right] \quad (19)$$

except that, for any  $n$  such that  $0L1 \cdot 2^n \geq 512$ ,

$$e_Q = 0L \cdot 2^{n-1} \text{ for all } e_M \geq \left[ \frac{3 \cdot 0L1 \cdot 2^{n-1}}{2} \right] \quad (20)$$

Each of the QUAN subroutines also computes a quantity called EMMOST, which represents a maximum allowable value of  $e_M$ . These two methods are summarized below:

<u>Method</u>	<u>EMMOST</u>
QUAN1	$2^{QMAX-1-1}$
QUAN2	lowest of $511$
	$\left[ \frac{3 \cdot 0L1 \cdot 2^{N-1}}{2} \right] - 1$
	$\left[ \frac{3 \cdot 0L1 \cdot 2^{n-1}}{2} \right] - 1$ where $0L1 \cdot 2^n \geq 512$

(21)

This number, plus the number of computed values of  $e_M$  which exceed EMMOST in each record, are printed after computation for each record is finished.

In later discussion, QUAN1 will be referred to as "linear quantization" and QUAN2 as "log quantization" (or companding).

The detailed program is described in Appendix C.

#### 4. EXPERIMENTAL RESULTS

In the experiments run thus far the general procedure has been to run full-load sine wave experiments initially, and then to run the speech experiments. The purpose of the sine wave experiments was to enable us to study the operation of the predictive coding system under various conditions (such as infrequent clipping), and also to provide an interesting comparison with the theoretical results of Section 1.3.6 for  $\Delta\text{PCM}_M$ . Since time permitted the running of only a few speech experiments, these concentrated on evaluating the usefulness of linear predictive coding at a bit rate (30 kc) appreciably lower than the 48 kc bit rate of current systems. Also, an experiment at 41.6 kc was run to compare with the results of Ref. 12.

Two types of "error" quantizing were tried with the predictive coding scheme depicted in Fig. 13. The one type we call a "linear" type, and this amounts to simply dividing the total error range into equal quantization steps. The other type is a "companded" or logarithmic type where the quantization step size increases (as one moves away from zero) according to integer-exponentials (i. e. , 1-2-4-8, etc.). In both the sine wave and the speech tests both the linear and companded quantizing were used.

The quantization signal-to-noise ratio which was calculated in these experiments was calculated by use of Eq. 15. It may be noted that this calculates "rms signal power" divided by "rms error." Note that the "error" is the difference between the finely quantized input samples (essentially analog) and the experimental output samples. Thus the ratio is related to an "rms" criterion of quality of the waveform. The signal-to-noise ratios must be taken with a certain amount of caution since this calculation involves all the implications of "rms criterion"---that is, it weights large errors heavily. Thus signals having equal signal-to-noise ratios could have substantially different "listening" properties. Consequently, we feel that the signal-to-noise ratios must be used as a quantitative guide, and any final conclusions would have to rest on elaborate, subjective listening tests.



#### 4.1 Sine Wave Results

"Full-load" sine wave results were first taken for linear error quantizing with values of M from 0 to 4 (see Fig. 13 for block diagram). The calculated signal-to-noise ratio results (in db) from these tests are depicted in Table II. For these tests, precalculated values for the amplitudes of an 800-cycle sine wave were inserted as data for the computer. Thus, the analog-to-digital converter was not used for these tests. The frequency of 800 cycles per second was selected because this frequency should give a fair representation of the speech signal-to-noise ratio for differential systems (Ref. 7). In computing the sine wave values, 61 samples were computed in a 5-cycle portion. Thus, the actual period of the "sampled sine wave" was 5 cycles. It would be an improvement to have this "sampling rate" be incoherent with the sine wave, or to use sine waves over a range of frequencies and average the results. However, there was no time to perform this modification.

Bit Rate	30 Kc	40 Kc	50 Kc	60 Kc	70 Kc	80 Kc
M = 0 (PCM)	14.5	21.7	28.4	35.4	42.6	47.7
M = 1	20.33	26.3	31.8	37.6	—	49.01
M = 2	25.6	34.2	—	47.5	53.0	63.0
M = 3	27.7	37.8	44.7	49.2	53.45	(> 70 db)
M = 4	—	38.72	44.1	49.6	62.9	(> 70 db)

Table II.  $(S/N)_q$  in decibels for linear quantized sine wave of 800 cps.

The results of Table II can best be viewed by plotting the  $(S/N)_q$  ratios versus bit-rate for the various M's. This is done in Fig. 18.

These sine wave results suggest that, with the linear predicting coding and operating conditions chosen here, there should be good improvement in going to M=1 (over PCM), and that further improvements are available for using M=2. Also, it is suggested that increasing M above 2 (with the unattenuated linear law) is not very profitable. These generalities were found to hold for the few speech experiments which were run.

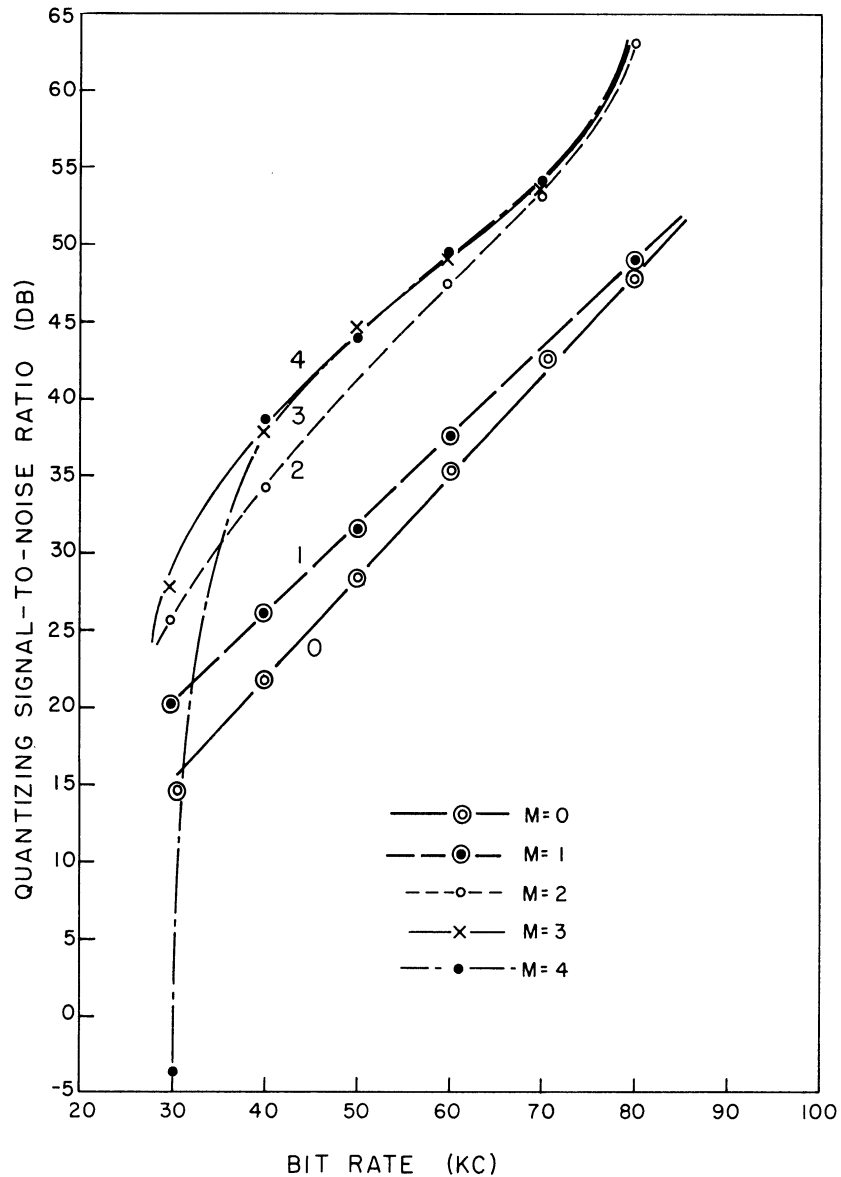


Fig. 18. Experimental quantization S/N versus bit-rate for 800 cps sine wave and linear quantization.

It is interesting to compare these experimental sine wave values with the theoretical sine wave values of Ref. 2 (noted in Section 1.3.6) for  $\Delta\text{PCM}_M$ . First of all, it must be noted that the feedback coding used here differs from that of  $\Delta\text{PCM}_M$ , as may be seen by comparing Fig. 9 and Fig. 12. In Fig. 9, the receiver predictor has quantized samples, while the transmitter predictor has analog samples. Also, in the experiments, occasional limiting of  $e_M$ ,  $f_p$ , and  $f_{ro}$  was permitted. It was found, generally, that some clipping of  $e_M$  resulted in a higher S/N than if the quantization was made "more gross" to prevent any clipping.

Comparing the  $\Delta\text{PCM}_M$  curves of Fig. 11 with Fig. 18 shows that the experimental curves are consistently lower than the theoretical ones, and that the improvement at  $M=3$  and  $M=4$  begins to disappear. The "consistently lower" experimental values can be attributed to the following reasons: (1) the feedback loop and occasional limiting is different from  $\Delta\text{PCM}_M$ ; (2) the "sampling" in the experiments may not have been sufficiently "incoherent" (or aperiodic) to obtain an "equally-likely distribution" of error signals within a quantization step (which is assumed in theoretical derivations); and (3) the theory of Ref. 2 reduces the rms error noise by a "bandwidth factor," which of course results in a higher S/N.

In the matter of lack of improvement for  $M=3$  and  $M=4$ , it is felt that the experiments suggest that the assumptions made in the theoretical calculations are not valid for these  $M$ 's.

For the purpose of establishing the "ground rules" of operating a feedback loop with companded error quantizing, some sine wave results were also run for this case. The results of these tests are shown in Table III below.

Bit Rate	30 Kc	35.8 Kc	38 Kc	40 Kc	41.6 Kc
M = 1	15.4	—	14.05	21.0	21
M = 2	17.2	29.44	25.9	30.1	—
M = 3	7.0	30.0	33	33.4	—

Table III.  $(S/N)_q$  in decibels for companded quantizing of 800 cps sine wave.

Comparing Table II and Table III it may be noted that the values for the companded situation consistently fall below those of Table II (for linear quantizing). This is to be expected for any "full-load" situation, since it is well known that the companding actually reduces the signal-to-noise ratio at full-load but has the advantage of maintaining a good signal-to-noise ratio over a range of volumes.

As regards the effect of increasing  $M$  for the logarithmic companding, these experimental sine wave results suggest that, at low (30 kc) bit rates,  $M=2$  or  $3$  is not profit-

able. If one allows as much as 40 kc, however,  $M=2$  is 9 db better, and 3 more db can be obtained with  $M=3$ . This should be verified with actual speech signals. It should be emphasized that the values reported here are "reported as taken." We had no time to assure, either analytically or experimentally, that the highest possible values were obtained for each case. However, the results shown here do afford enough over-all consistency to indicate the general trends.

#### 4.2 Experimental Speech Results

When conducting the speech experiments, sentences made up of phonetically balanced words were used. The first operation consisted of finding a suitable level of speech to put into the converter. This level was found subjectively by trying various levels and selecting that level which sounded best after repeated subjective listening. As would be expected, this level accounted for a small number of clippings of certain points in the speech waveform. The data sent to the computer consisted of about 10 records with approximately 13,000 samples per record; hence each signal-to-noise ratio reported below is based on about 130,000 samples.

As mentioned before, the major objective in the speech experiments was to determine whether, with linear predictive coding, one can reduce the bit-rate without seriously affecting quality. For this reason, a number of experiments (as time permitted) were run at 30 kc. Also, one experiment at 41.6 kc was run in order to compare with the "log differential" results of Ref. 12 (see Section 1.3.7) using actual speech.

The 41.6 kc bit-rate accounts for a sampling rate of 10 kc and 9 (positive and negative) levels. A companded error quantization and an  $M=1$  were used for this experiment. With this companded quantization (and speech input) a signal-to-noise ratio of 18.0 db resulted. This compares with the value of 21 db obtained in the sine wave experiment shown in Table II. Also, when listening to this tape, it was found that the quality is quite good. We feel that it would take some "training" to distinguish this case from the 50 kc PCM (which is approximately the value used in present military systems). Also, a signal-to-noise calculation for  $M=0$  (or linear PCM) with speech resulted (at 40 kc) in a signal-to-noise ratio of 13.7 db. Thus it appears that, even at the full-load condition, the  $M=1$  with companded quantizing appears to be quite profitable. When one further considers that the companded case can be expected to do better over a varying volume than the linear PCM case this adds emphasis to the value of using  $M=1$ .

In addition, the results here agree with the results found in Ref. 12 (where "log differential" is used) in that quality is not essentially reduced if one operates at 41.6 kc under these conditions as opposed to 48 kc. It may be noted that the actual values of signal-to-noise ratio here can not be compared directly with those of Ref. 12 because a different method of evaluating signal-to-noise ratio was used there.

The experimental results for speech at 30 kc are depicted in Table IV below.

Quantizing Type	Linear	Compond
PCM (M = 0)	9.5	—
M = 1	10.17	10.63
M = 2	11.57	9.67

Table IV.  $(S/N)_q$  for speech at 30 Kc bit rate.

At 30 kc it was found that linear PCM (M=0) resulted in a signal-to-noise ratio of 9.5 db which we will use as reference here. Based on the values shown in Table IV it appears that under the conditions of these experiments there is little to be gained by using either M=1 or M=2. However it must be remembered that if one considers varying volumes, the companded error signals can be expected to do a good deal better than either the PCM or the linear. Consequently if one takes the expected varying volume situation into consideration, one can conclude that companded M=1 appears to be the best method.

From the data taken on actual speech thus far, there is no good reason to believe that significant improvement can be obtained by increasing M from 1 to 2 at 30 kc. Furthermore, there seems to be no hope of gaining improvements with M=3 or 4 with the unattenuated linear predicting law used in these experiments.

## 5. CONCLUSIONS

The experiments reported on here used a linear prediction law, where the number of (10 kc) samples used in the prediction was varied from 0 to 4, with the objective of trying to reduce bit rates without seriously affecting quality. For the experiments reported here it was necessary to make a number of engineering judgments, especially as concerns the maximum levels for the errors ( $e_M$ ), the prediction ( $f_p$ ), and the receiver output ( $f_{ro}$ ). Based on all the experiments run, it was generally found that one should allow some clipping or limiting of the error signal to obtain the best signal-to-noise ratio. In other words, if one quantized more grossly so that the error never exceeded its maximum one obtained poorer signal-to-noise ratios than otherwise. However, it was found necessary to keep the error clipping at a relatively low level or else the signal-to-noise ratio collapsed rapidly. Furthermore, it was found that the range of "good" clipping levels for EM varied with M. Thus there are many variables in these experiments.

The clipping level for FP directly affects the transient response of the loop. If FP is not limited one can obtain the fastest "rise time" (step function response) for a given M---the rise time improves as M increases with the linear law used here. However, if one puts in some clipping of the prediction, then one essentially puts in a nonlinearity which is effective in reducing possible oscillations---especially for M=3 and 4.

In these experiments the receiver output ( $f_{ro}$ ) was always limited to the maximum full-load sine wave value (511). This was done because the predictor uses the  $f_{ro}$  values to find the estimate. Although  $f_{ro}$  exceeded its maximum only infrequently (see data in Appendixes A and B), having  $f_{ro}$  unlimited should improve S/N slightly.

It should also be remembered that the experiments here used an unattenuated linear prediction law on the speech as given by Eq. 2, in the "coding" circuit of Figs. 12 and 13.

Based on the experiments run thus far, and keeping the above items in mind, the following conclusions are suggested:

- (1) For linear quantizing it appears that using either  $M=1$  or  $M=2$  allows one to decrease bit-rate from 48 kc (presently used) to 30 kc, without seriously affecting quality. Also, the  $M=1$  and 2 case at 30 kc is noticeably better (for listening) than PCM. It also appears, under the conditions of these experiments, that  $M=3$  or 4 is not desirable.
- (2) For companded quantizing, we could get desirable listening results (at 30 kc) only for  $M=1$ . The  $M=2$  case provided unpleasant noise. However, more experiments must be run before ruling out  $M=2$ .

In general, then, it has been found that using  $M=1$  and 2 under the conditions used here offers good potential for reducing bit-rate while retaining acceptable quality. The  $M=3$  and 4 cases were not found desirable under the conditions here.

Considering the varying volume situation, it appears that the use of companded quantizing is the most profitable---although no varying volume tests were conducted so far.

Finally it must be emphasized that all the above conclusions must be regarded as tentative, and based on the relatively few experiments which were run thus far. The major portion of the time available for this work was consumed in writing the computer program and modifying the available analog-to-digital converter.

In any future experiments it is suggested that: (1) the computer find the error-term distribution, so that desirable limiting conditions can be prescribed; (2) some attenuating factors on the prediction law used here be tried; (3) based on such studies, a combination of linear and nonlinear (clipping) properties be selected so as to optimize conditions for a given bit-rate.

## APPENDIX A: DATA FOR SINE WAVE EXPERIMENTS

The following are the data resulting from the full-load sine wave experiments (see Fig. 13 for notations). The data for linear quantizing is given first, followed by that for log quantizing. The data for each case is arranged by bit rate, starting with low bit rates and proceeding to the higher ones. The data listed here correspond to Table II of the report. The items given are: (1) the value of  $M$ ; (2) the  $(S/N)_q$ ; (3) the quantizing parameters, and (4) "limiting conditions."

For linear quantizing  $Q$  ( $Q=n$ ) equals the number of digits required by the error signal.  $Q_{MAX}$  refers to the maximum range of the error, which is given by  $\pm 2^{Q_{MAX}-1}$ . Note that a given range can be quantized more grossly or less grossly by adjusting  $n$  relative to  $Q_{MAX}$ . If  $n$  equals  $Q_{MAX}$  the quantization is the "finest" possible.

For log quantizing, the number of error steps (above zero) is given by  $N$ . Then the number of digits required is  $n = \log_2 2N$ . The "0L1" refers to the value of the smallest step in the log arrangement. In effect, it is the base of the log. The "compand type A" is (1, 2, 4, 8, ...), while the compand type B is (1, 4, 16, 64, ...). The combination of  $N$ , 0L1, and "A" or "B" determines the maximum range and the "grossness" of steps in the log case. A table of output-levels and decision levels versus different 0L1 and for type A and B is given on the following page (Table V).



Type A

Type B

N	Basic Output Levels	Basic Decision Levels	(QUAN2)			Basic Output Levels	Basic Decision Levels	(QUAN2 + 2005)	
			0L1 = 2	0L1 = 3	0L1 = 4			0L1 = 2	0L1 = 3
9	(511) 256	383	(511)						
8	128	191	383 256	384	(511)				
7	64	95	191 128	287 192	383 256				
6	32	47	95 64	143 96	191 128				
5	16	23	47 32	71 48	95 64	256	383	(512)	
4	8	11	23 16	35 24	47 32	64	95	191 128	287 192
3	4	5	11 8	17 12	23 16	16	23	47 32	71 48
2	2		5 4	8 6	11 8	4	5	11 8	17 12
1	1			4 3	5 4	1		3 2	
0				zero is not used					

Table V. Companded quantizing levels.

(All entries are symmetric about zero.)

In the charts below, the "limiting conditions" refer to three items: (1)  $FP = f_p$ ; (2)  $EM = e_M$ ; (3)  $FRO = f_{ro}$  (see Fig. 13). The tables below give the following for each of these items: (1) the maximum value for that item; (2) the number exceeding the maximum value; (3) the largest value of that item.

For all of this data the number of samples involved is 14,820. Consequently, the "number exceeding maximums" should be referred to this number.

DATA FOR SINE WAVE EXPERIMENTS WITH LINEAR QUANTIZING

(14,820 Samples)

Bit Rate	M	(S/N) <sub>q</sub>	Q = n	Q <sub>MAX</sub>		Maximum	No. Exceeding Maximum	Largest
30 Kc	1	20.33	3	10	FP	511	2429	513
					EM	511	0	306
					FRO	511	2430	513
	2	25.69	3	8	FP	511	6315	641
					EM	127	1943	161
					FRO	511	0	511
	3	27.71	3	8	FP	511	4612	603
					EM	127	970	150
					FRO	511	243	513
40 Kc	1	26.34	4	10	FP	511	2429	513
					EM	511	0	280
					FRO	511	2429	513
	2	34.22	4	8	FP	511	5829	637
					EM	127	729	139
					FRO	511	0	511
	4	38.72	4	8	FP	511	1213	591
					EM	127	2	138
					FRO	511	5	519
50 Kc	1	31.79	5	10	FP	511	2429	513
					EM	511	0	274
					FRO	511	2429	513
	3	44.72	5	8	FP	511	3887	573
					EM	127	0	97
					FRO	511	0	511
	4	44.11	5	8	FP	511	487	537
					EM	127	0	93
					FRO	511	242	514
60 Kc	1	37.57	6	10	FP	511	1457	513
					EM	511	0	266
					FRO	511	1457	513
	2	47.43	6	8	FP	511	5830	629
					EM	127	486	130
					FRO	511	0	511
	3	49.19	6	8	FP	511	3644	575
					EM	127	0	85
					FRO	511	0	511
	4	49.61	6	8	FP	511	240	523
					EM	127	0	81
					FRO	511	246	513
70 Kc	2	52.99	7	8	FP	511	5830	629
					EM	127	243	128
					FRO	511	0	511
	3	53.45	7	8	FP	511	3644	565
					EM	127	0	89
					FRO	511	0	511
	4	54.17	7	8	FP	511	0	511
					EM	127	0	66
					FRO	511	242	512
80 Kc	1	49.01	8	10	FP	511	485	513
					EM	511	0	259
					FRO	511	485	513
	2	62.99	8	8	FP	511	5830	629
					EM	127	243	129
					FRO	511	0	511
	3	(> 70 db)	8	8	FP	511	3887	565
					EM	127	0	86
					FRO	511	0	511
	4	(> 70 db)	8	8	FP	511	0	511
					EM	127	0	65
					FRO	511	0	511

DATA FOR SINE WAVE EXPERIMENTS WITH COMPANDED QUANTIZING

(14,820 Samples)

Bit Rate	LOG				Compand Type		Maximum	No. Exceeding Maximum	Largest
	M	(S/N) <sub>q</sub>	N	OL1					
30 Kc	1	15.41	4	3	B	FP	511	244	573
						EM	287	1698	335
						FRO	511	244	573
	2	17.17	4	2	B	FP	511	5344	791
						EM	191	484	322
						FRO	511	244	527
	3	7.01	4	3	B	FP	511	3888	1195
						EM	287	3152	1022
						FRO	511	406	629
35.8 Kc	2	29.44	6	4	A	FP	511	6072	653
						EM	191	0	155
						FRO	511	0	507
	3	29.91	6	4	A	FP	511	3160	639
						EM	191	0	127
						FRO	511	243	519
38 Kc	1	14.05	7	3	A	FP	511	727	545
						EM	287	3649	374
						FRO	511	727	545
	2	25.96	7	3	A	FP	511	6315	654
						EM	287	0	173
						FRO	511	0	500
	3	33.12	7	3	A	FP	511	3035	650
						EM	287	0	125
						FRO	511	121	524
40 Kc	1	20.98	8	2	A	FP	511	1456	531
						EM	383	0	314
						FRO	511	1456	531
	2	30.07	8	1	A	FP	511	5830	645
						EM	191	0	152
						FRO	511	0	510
	3	33.44	8	1	A	FP	511	3644	626
						EM	191	0	133
						FRO	511	242	515
1	21	9	1	A	FP	511	1456	531	
					EM	383	0	314	
					FRO	511	1456	531	

APPENDIX B: SPEECH DATA

		Avg. No. Samples	Max.	Avg. No. Exceeding Max.	Avg. Value of Largest
SPEK 1	M = 1	13,621	FP 511	0	511
	Q = 8		EM 127	719	738
	Q <sub>MAX</sub> = 8		FRO 511	0	511
	(S/N) <sub>db</sub> = 9.0476 Bit Rate = 80 Kc				
SPEK 2	M = 2	11,176	FP 511	78	611
	Q = 8		EM 127	307	408
	Q <sub>MAX</sub> = 8		FRO 511	0	376
	(S/N) <sub>db</sub> = 14.734 Bit Rate = 80 Kc				
SPEK 3	M = 3	11,176	FP 511	3885	1108
	Q = 8		EM 127	4889	754
	Q <sub>MAX</sub> = 8		FRO 511	0	429
	(S/N) <sub>db</sub> = -7.3016 Bit Rate = 80 Kc				
SPEK 4	M = 1	13,621	FP 511	43	564
	OL1 = 1 (1-2-4-...)		EM 383	26	576
	(S/N) <sub>db</sub> = 18.273		FRO 511	41	564
	Bit Rate = 41.6 Kc				
SPEK 5	M = 2	11,211	FP 511	83	609
	Q = 3		EM 127	396	444
	Q <sub>MAX</sub> = 8		FRO 511	4	424
	(S/N) <sub>db</sub> = 11.572 Bit Rate = 30 Kc				

SPEECH DATA (Cont.)

		Avg. No. Samples	Max.	Avg. No. Exceeding Max.	Avg. Value of Largest
SPEK 6	M = 2	11,176	FP 511	94	688
	OL1 = 2		EM 191	331	496
	N = 4, (2, 8, 32, 128)		FRO 511	9	467
	(S/N) <sub>db</sub> = 9.6371 Bit Rate = 30 Kc				
SPEK 7	M = 2	11,445	FP 511	65	778
	OL1 = 3		EM 287	124	520
	N = 4, (3, 12, 48, 192)		FRO 511	4	486
	(S/N) <sub>db</sub> = 6.0893 Bit Rate = 30 Kc				
SPEK 8	M = 0	10,796	FP 511	0	0
	Q = 3		EM 511	0	385
	Q <sub>MAX</sub> = 10		FRO 511	0	307
	(S/N) <sub>db</sub> = 9.4699 Bit Rate = 30 Kc				
SPEK 9	M = 0	10,796	FP 511	0	0
	Q = 4		EM 511	0	385
	Q <sub>MAX</sub> = 10		FRO 511	0	352
	(S/N) <sub>db</sub> = 13.729 Bit Rate = 40 Kc				
SPEK 10	M = 1	10,796	FP 511	10	401
	Q = 3		EM 511	.2	323
	Q <sub>MAX</sub> = 10		FRO 511	10	401
	(S/N) <sub>db</sub> = 10.172 Bit Rate = 30 Kc				
SPEK 11	M = 1	10,796	FP 511	8	463
	OL1 = 2		EM 287	14	333
	N = 4, (2, 8, 32, 128)		FRO 511	8	463
	(S/N) <sub>db</sub> = 10.451 Bit Rate = 30 Kc				

SPEECH DATA (Cont.)

		Avg. No. Samples	Max.	Avg. No. Exceeding Max.	Avg. Value of Largest
	M = 1	11,445	FP 511	4	463
	OL1 = 3		EM 191	91	388
SPEK 12	N = 4, (3, 12, 48, 192)		FRO 511	4	463
	(S/N) <sub>db</sub> = 10.634				
	Bit Rate = 30 Kc				
SPEK 13 (Intentionally omitted)					
	M = 1	11,445	FP 511	.7	395
	Q = 3		EM 127	291	429
SPEK 14	Q <sub>MAX</sub> = 8		FRO 511	.7	395
	(S/N) <sub>db</sub> = 10.035				
	Bit Rate = 30 Kc				

## APPENDIX C: DETAILED COMPUTER PROGRAM

by Marcia G. Feingold

### I. COMPUTER PROGRAM OPERATING INSTRUCTIONS

#### A. The Program Deck\*

A complete program deck for the computer simulation on any input tape contains the following set of cards:

1. Two identification cards
2. A specification card: "\$EXECUTE, DUMP, I/O DUMP, BINARY".
3. Binary program cards labeled TR 000-015
4. Binary program cards labeled QUAN1 or QUAN2 000-004
5. Binary program cards labeled TR 200-238, 300-307  
No. 239 results in tape reading and writing ten times in case of error, and must be included until an error in the present I/O supervisor is corrected.  
Nos. 400 and 401 must also be included until the new I/O supervisor is completed.
6. Binary program cards labeled SLDMP 000-003
7. A specification card: "\$DATA".
8. Data card: "TAPEX = 4, TAPEY1 = 2, TAPEY3 = 3 \*".  
These numbers should be changed, for greater program speed, if the present assignment of work tapes to data channels is changed. TAPEX should be on one channel, and the other tapes should be on the other channel.

---

\*Some familiarity with the Michigan Executive System and UMAP is assumed.

9. Data card: "M = x, RECCT = x, IDENT = \$ x \$", where M is the number of previous input samples used to compute  $f_p(t_i)$ , RECCT is the number of records of  $f_{in}$  on tape, and IDENT is a label of six or fewer characters to be put on the magnetic tape case and on the identification cards.
10. Data card:
  - for QUAN1 -- "QMAX = x, Q = x \*"
  - for QUAN2 -- "OL1 = x, N = x \*"

### B. Sine Wave Program Deck

For the experiments with sine waves, a precalculated sine wave was used. A complete program deck for the computer simulation on an internally-generated sine wave contains the following set of cards:

1. Two yellow ID cards
2. One specification card: "\$EXECUTE, . . ."
3. The SINEF routine, cards 000-008
4. A binary transition card (the first column is punched in rows 12, 7, and 9; all other columns are blank)
5. The TR routine, cards 000-015, plus a TR 016 card, plus TR 017-018, if no tape is to be used
6. A QUAN routine, cards 000-004, and 005 if used in the log case
7. The TR 200 routine, the same cards as listed in Section A
8. The SLDMP routine, cards 000-003
9. A binary transition card
10. One specification card: "\$DATA"
11. A data card giving the identification of the tape reel to be mounted, the number of records of sine wave to be written, and the number of the input tape (almost always, 4), i. e. ,  
"IDENT = \$ x \$, RECCT = x, TAPEX = 4 \*"
12. Seventeen data cards with the sine wave in Mode II form
13. The data cards used regularly with the main computation routine.



### C. Input Parameters

These parameters have been described and defined in Section 3. 4.

Their ranges are given below:

$$\begin{aligned}
 0 &\leq M \leq 6 \\
 1 &\leq \text{RECCT} \leq 60 \\
 2 &\leq Q \leq \text{QMAX} \leq 10 \} && \text{for QUAN1} \\
 1 &\leq \text{OL1} \leq 511 \} && \text{for QUAN2} \\
 1 &\leq N
 \end{aligned}$$

IDENT = six or fewer characters

RECCT must not exceed the actual number of records on the input tape.

### D. Input Tape Layout

The  $f_{in}(t_i)$  are characters in Mode II form, grouped into no more than 60 records of less than 14,850 characters (29,700 lines on tape) each. If more characters occur, only the first 14,850 of each record will be used for computation. The data recorded on tape is assumed to be in the following form: the bits are recorded (from left to right) in tracks C, B, A, 8, 4, 2, and 1; the bit "p" in track C is an odd parity bit, as the tapes are read in binary mode.

#### Recorded Values

Decimal Value	Converted binary digits		Leading half-character			Trailing half-character		
	$a_1$	$\dots a_{10}$	p 0	$a_1 \dots a_5$	p 1	$a_6 \dots a_{10}$		
-511	10000	00000	00	10000	01	00000		
.	.	.	.	.	.	.		
-3	11111	11100	00	11111	11	11100		
-2	11111	11101	00	11111	01	11101		
-1	11111	11110	00	11111	01	11110		
-0	11111	11111	00	11111	11	11111		
+0	00000	00000	10	00000	01	00000		
+1	00000	00001	10	00000	11	00001		
+2	00000	00010	10	00000	11	00010		
+3	00000	00011	10	00000	01	00011		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
+511	01111	11111	10	01111	11	11111		

It is assumed that the high order bits,  $a_1 \dots a_5$ , of any character always precede the low order bits of that character. However, provision is made in the program for records in which the first half-character of the first character was not recorded. In this case, the odd half-character is simply ignored.

#### E. Output Tape Layout

Let  $r$  (called RECCT on the data cards) be the number of records of  $f_{in}(t_i)$  to be used in the computation. Immediately following the  $r$  record on the input tape the following information will be given:

1. An end of file mark.
2.  $r$  records of  $e_Q(t_i)$ .
3. An end of file mark.
4.  $r$  records of  $f_{ro}(t_i)$ .

Corresponding records of  $e_Q$  and  $f_{ro}$  will always have the same number of words (the line count may differ by 1 due to a zero longitudinal check mark).

Normally, a record of  $e_Q$  (and  $f_{ro}$ ) will have 0-3 fewer characters than the corresponding record of  $f_{in}$  [since the last IBM 709 word (3 characters) or portion thereof of the  $f_{in}$  record is not considered for computation]. However there are certain exceptions to this:

1. No output record ( $e_Q$  and  $f_{ro}$ ) will have more than 14,847 characters.
2. In case some input information is mistaken for an end of file mark, it will be counted as an input record and one-word records will be written for  $e_Q$  and  $f_{ro}$ .
3. If a record has only one word on tape, this, as mentioned above, will not be considered for computation. However, one-word records will be written for  $e_Q$  and  $f_{ro}$ .
4. In case of error in the format of word  $w$  of an input record, if  $w > 0$ ,  $w$  words will be written on the  $e_Q$  and  $f_{ro}$  records; if  $w = 0$ , one-word records will be written.

#### F. Printed Output

Each input record will cause one of the following lines of print:

- 1) "RECORD x WAS AN EOF MARK" means that some information on the input tape was interpreted as an end of file mark.

- 2) "RECORD x WAS TOO SHORT" means that the input record had only one word or part of one word.
- 3) "RECORD NO. x HAS x WORDS" The number of words given here refers to the number of IBM 709 words, which are the length of three Mode II characters. This word count is the number of words on the input record minus the last word or part of a word.  
  
If the odd parity bits or the longitudinal check marks are not correct, the last line will be preceded by
- 4) "RECORD x HAD REDUNDANCY ERRORS" Line 3) may also be followed by one or both of the following lines:
- 5) "LOW ORDER BITS ARE LEADING," which refers to the situation mentioned in the last paragraph of Section II. C;
- 6) "RECORD x HAD FORMAT ERRORS IN WORD w. A MEMORY DUMP FOLLOWS." Line 6) will be followed by a memory dump of 4951 words, (10-1/2 pages) the first part of which consists of record x.

After computation for each record is completed, a table is printed giving the largest computed values of  $f_p$ ,  $e_M$ , and  $f_{r0}$ , plus the number of times in which the maximum allowable value for each quantity was exceeded in absolute value (511, EMMOST, and 511, respectively). The value of EMMOST is also printed.

After computation for the entire input tape has been completed, the signal-to-noise ratio is computed and printed. If  $N^2 = 0$ , this will be indicated.

### G. Timing

A workable approximation to the time needed to run this program, assuming records of maximum length (4950 words), is

$$T = 4 \text{ minutes} + 50 \text{ seconds/record}$$

More precisely, let

$r$  = the number of records of input,

$t_c$  = the time to change a tape,

$t_r$  = the time to rewind a tape an unknown distance.

Then, for  $12 \leq r \leq 36$  (and, roughly, for other values), again assuming maximum sized records,

$$T = 2t_c + 2t_r + 80 + r(44 + 1.5 M) \text{ seconds}$$

Still more precisely, letting  $w$  = the average number of words per record, we have

$$\begin{aligned} T = & 2t_c + 2t_r + \frac{3w}{2500} \\ & + \text{lesser } \frac{6w \cdot r}{2500}, 144 + \frac{4w \cdot r}{2500} \\ & \left( + \text{lesser } 72, \frac{3w \cdot r}{2500} \right) \\ & + r[1 + w(.006204 + .000144 M + .000072 f(M))] \quad \text{seconds} \end{aligned}$$

where  $f(M)$  is defined as:

<u>M</u>	<u>f(M)</u>
0	2
1	2
2	4
3	7
4	8
5	10
6	14

## II. DETAILS OF THE PROGRAM

The program (Fig. 19) consists of the main TR (000-015) routine, two QUAN subroutines (Figs. 20 and 21), the TR (200-242) subroutine (Figs. 22--27), and the SLDMP subroutine.

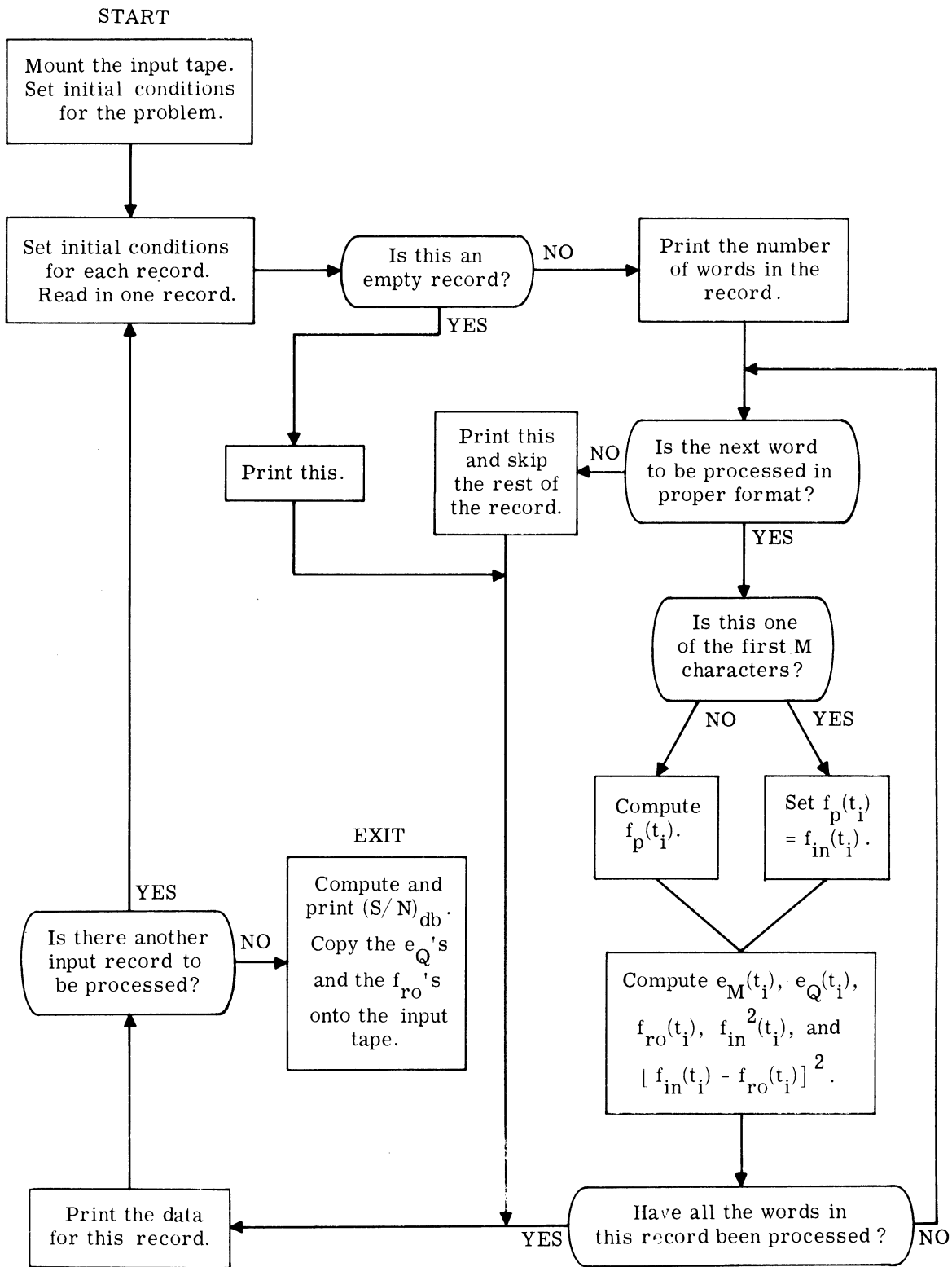


Fig. 19. Synoptic flow chart.

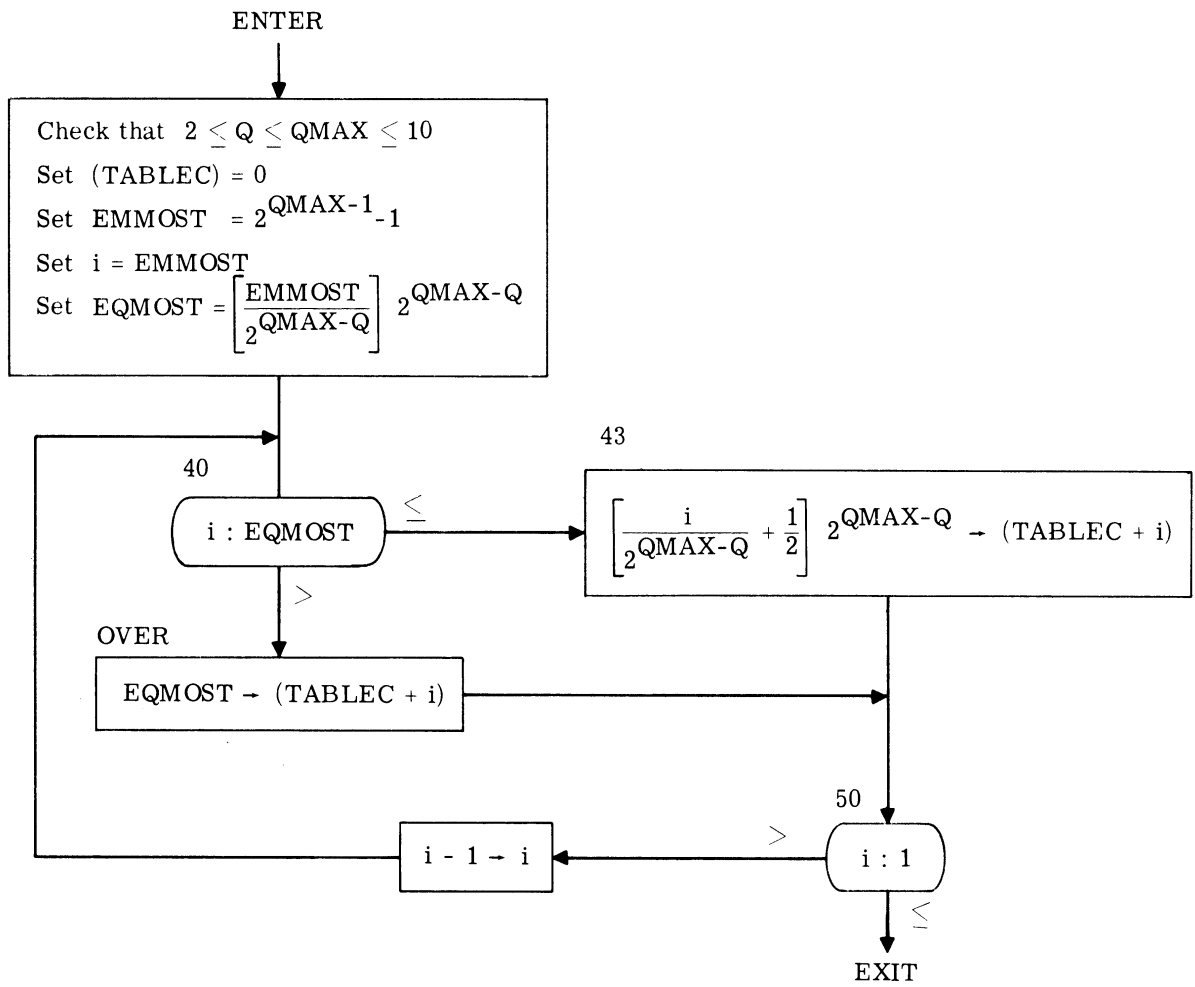


Fig. 20. Flow chart for QUAN1 (linear quantizing) procedure.

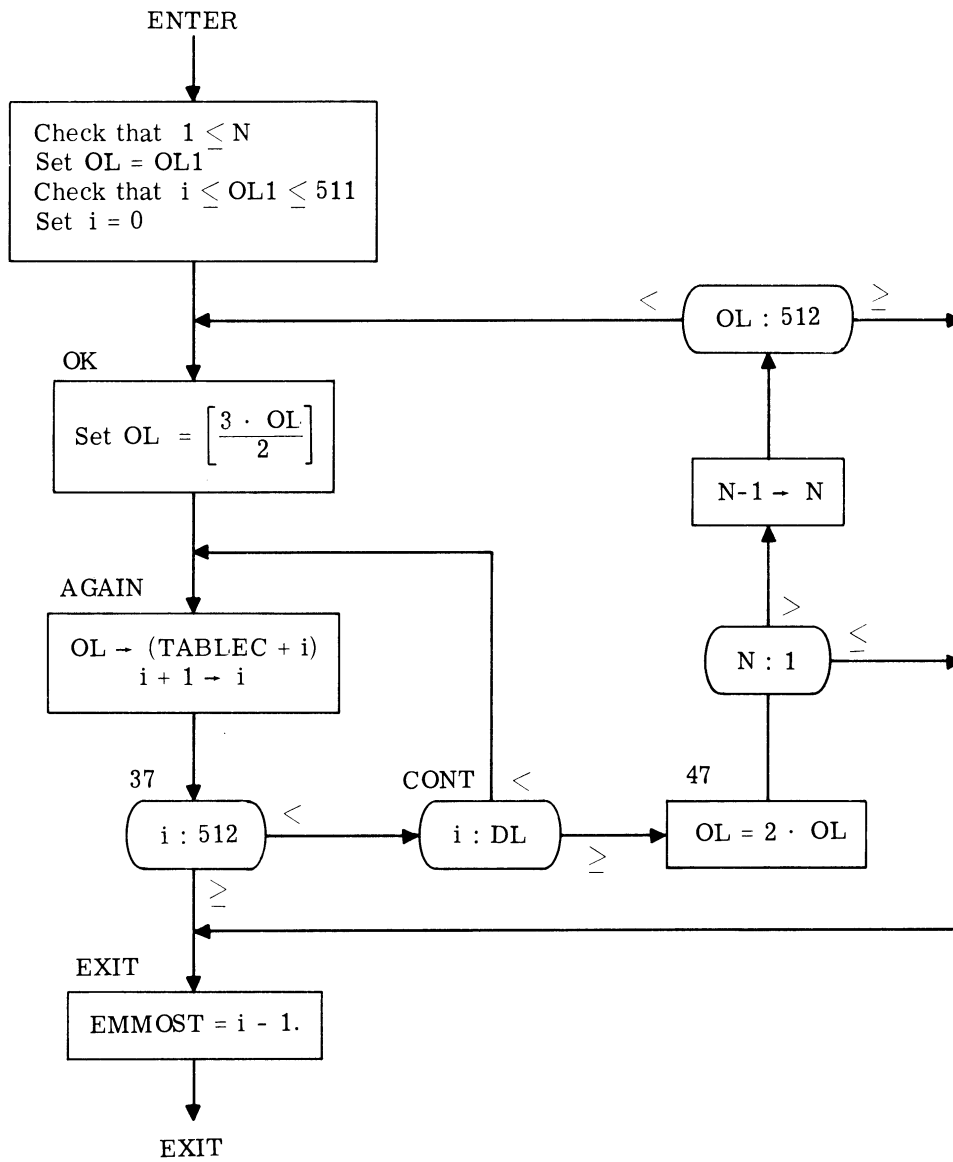


Fig. 21. Flow chart for QUAN2 (companded quantizing) procedure.

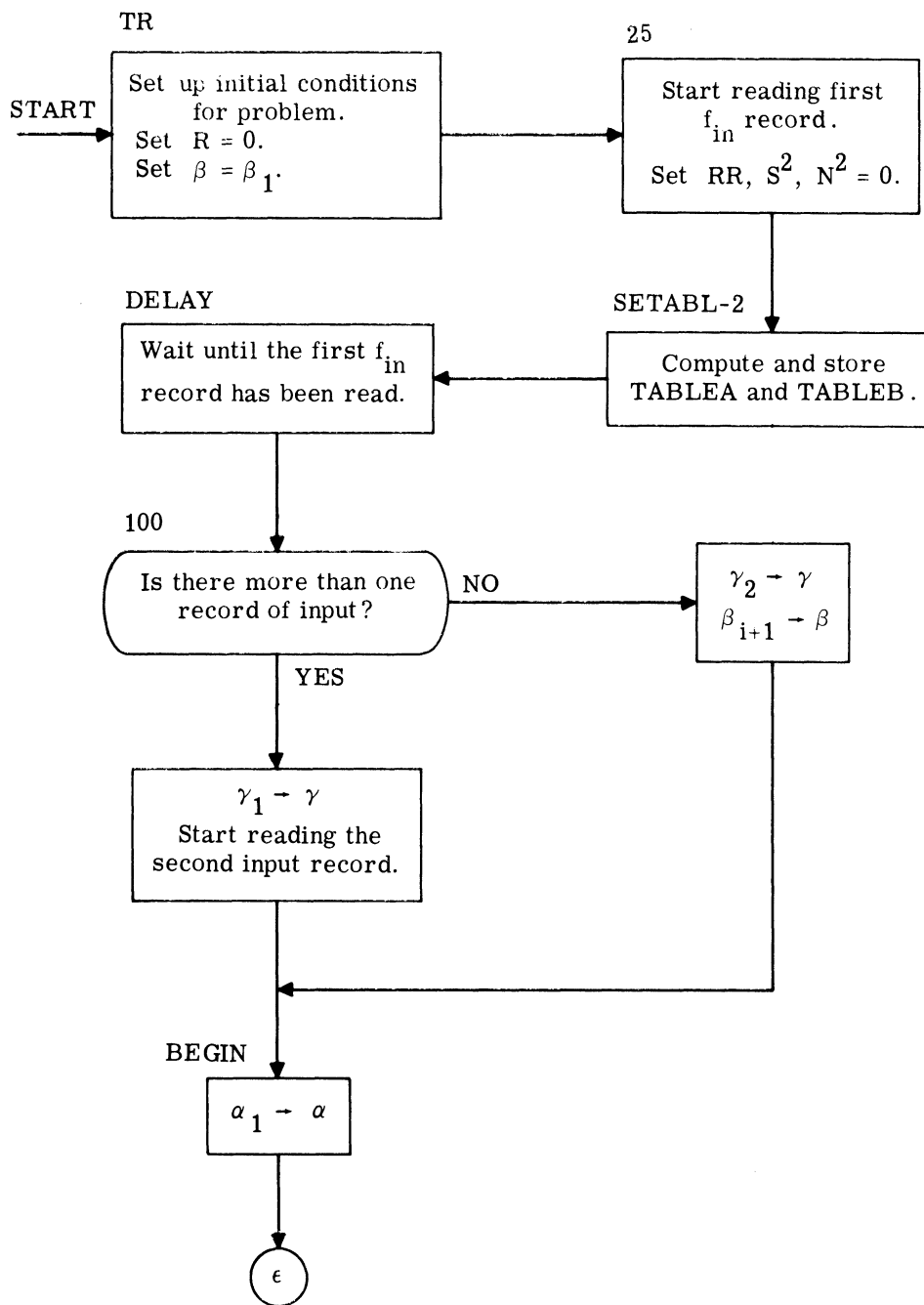


Fig. 22. Flow chart of part of TR 200 subroutine which computes E.



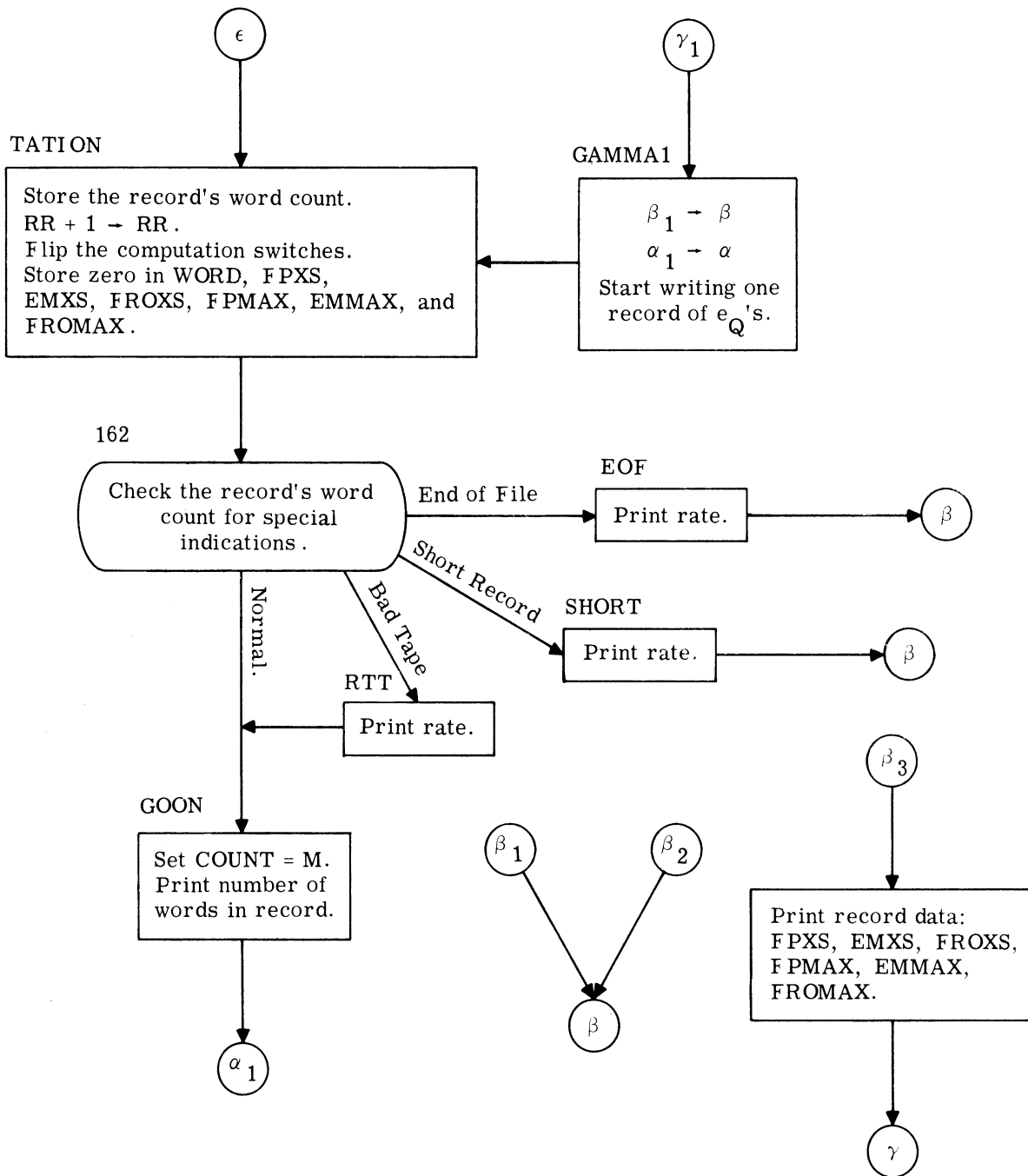


Fig. 23. Part of TR 200 subroutine.

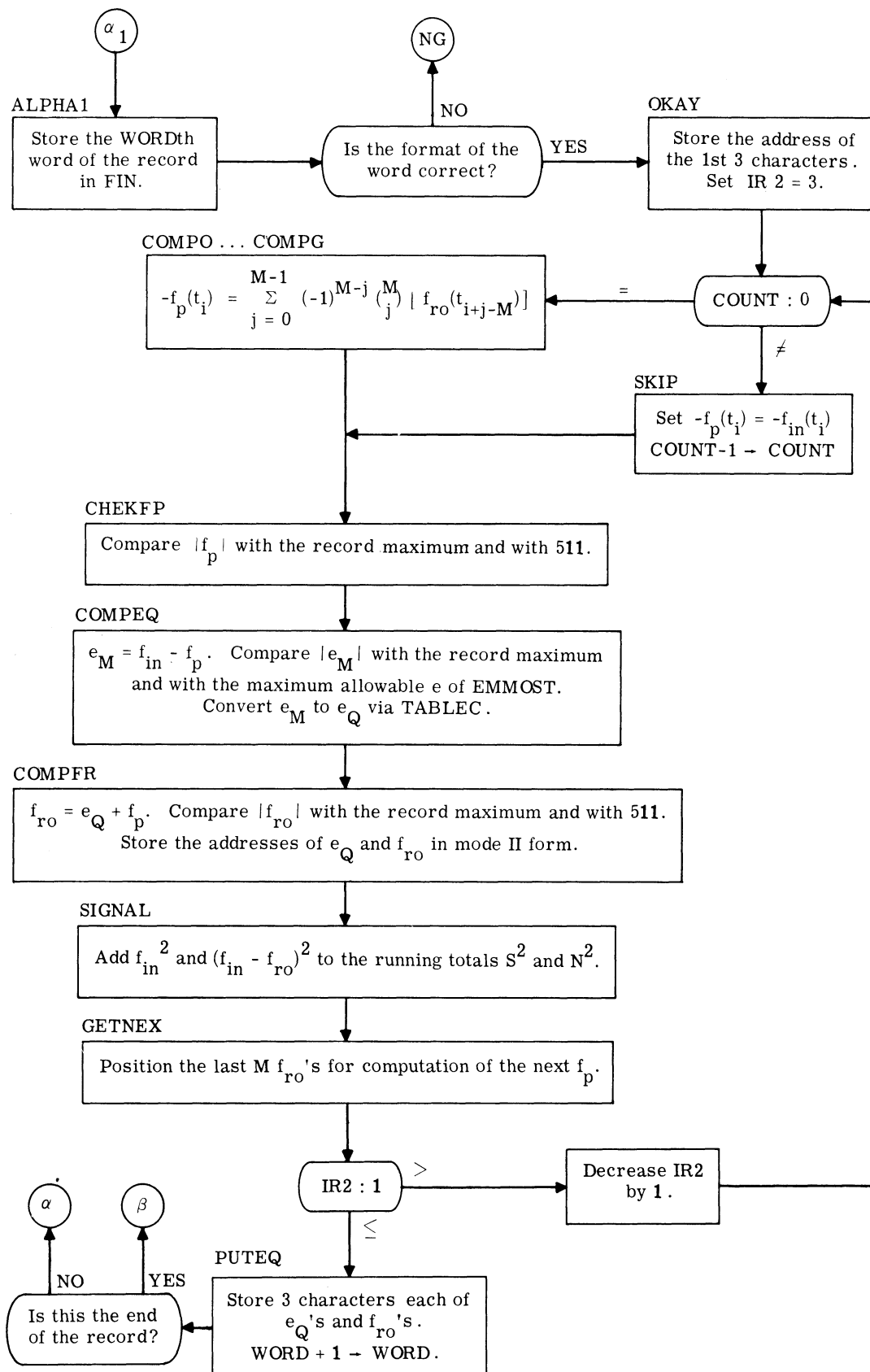


Fig. 24. Part of computation of TR 200 subroutine.

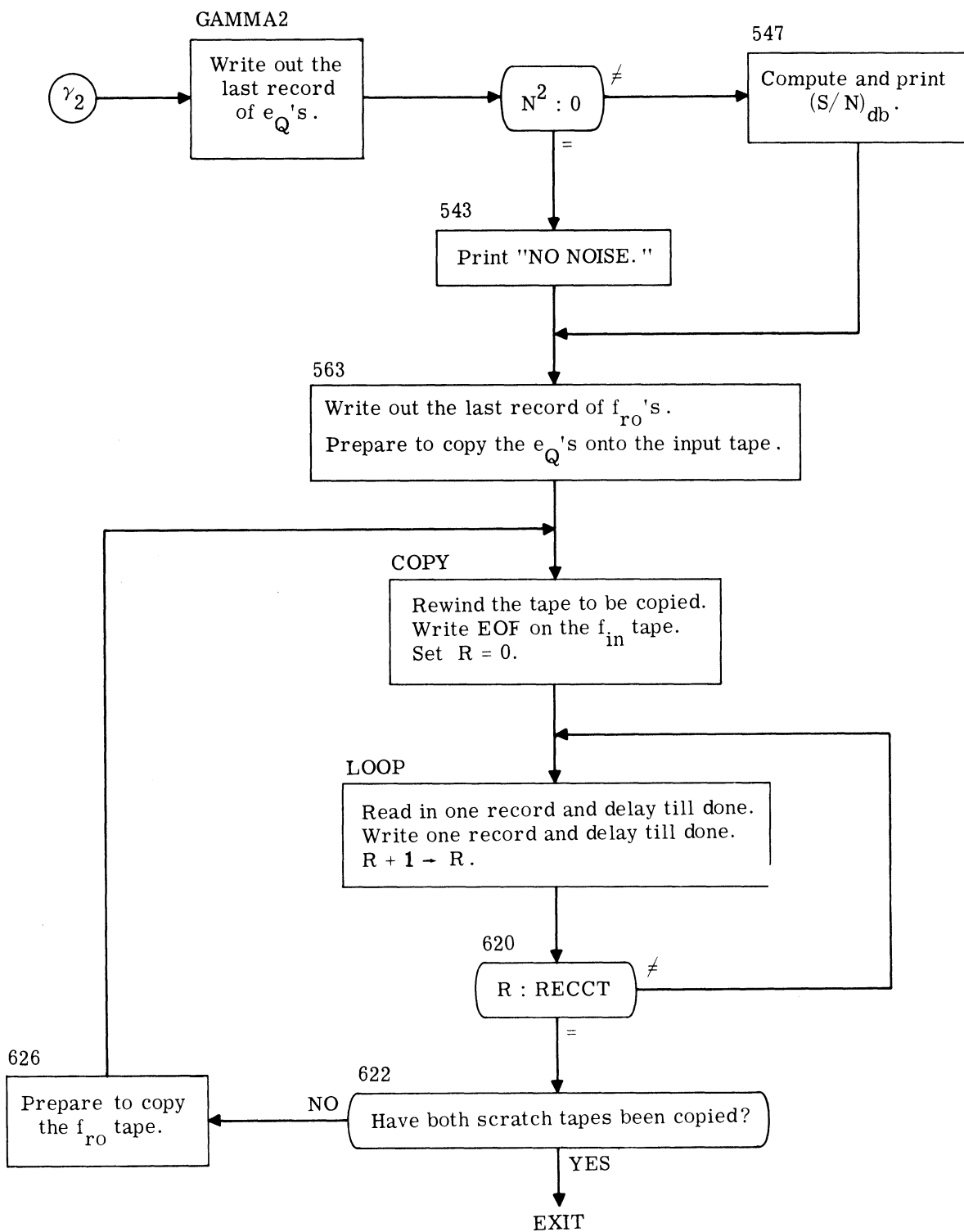


Fig. 25. Recording part of TR 200 subroutine.

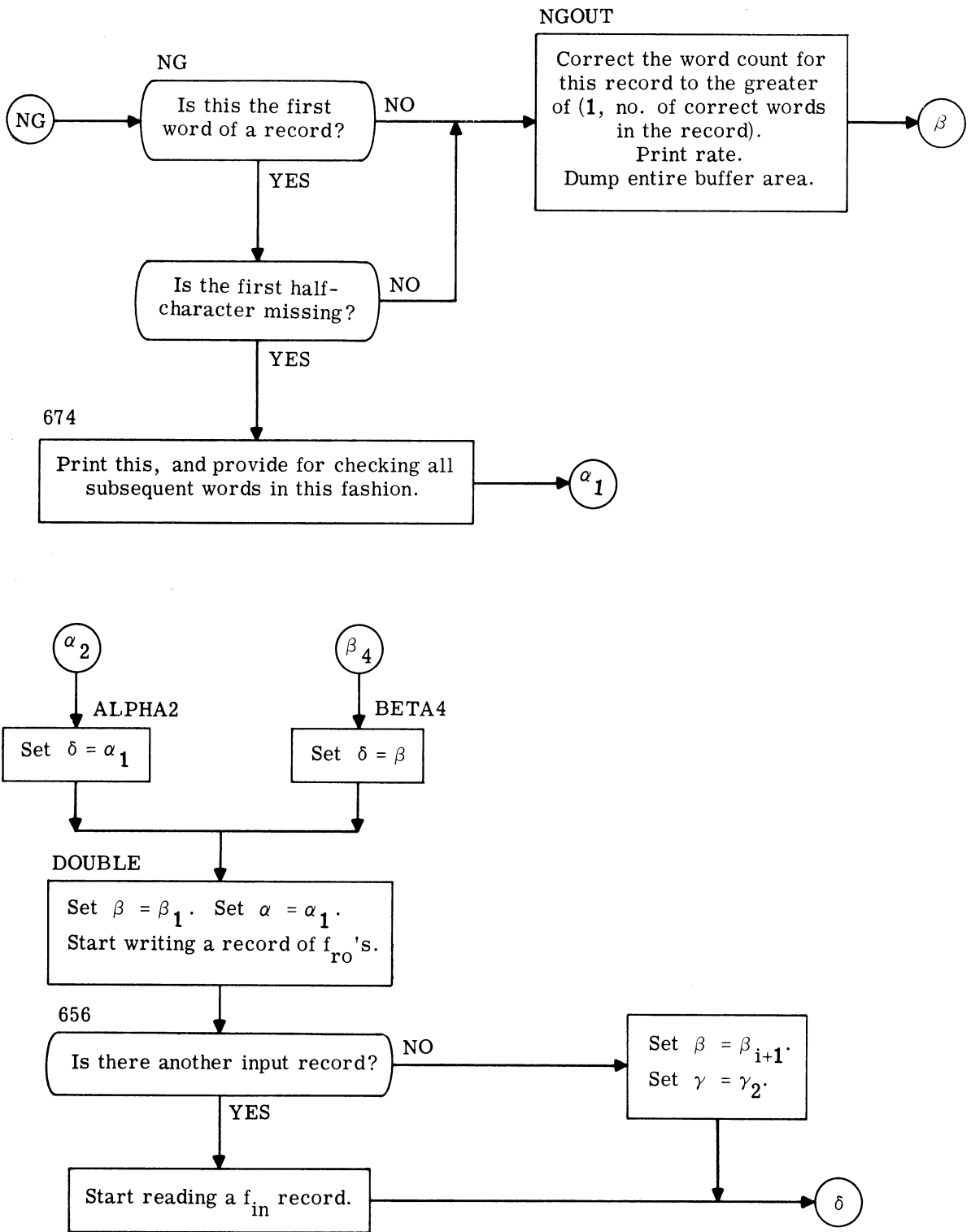


Fig. 26. Control part of TR 200 subroutine.

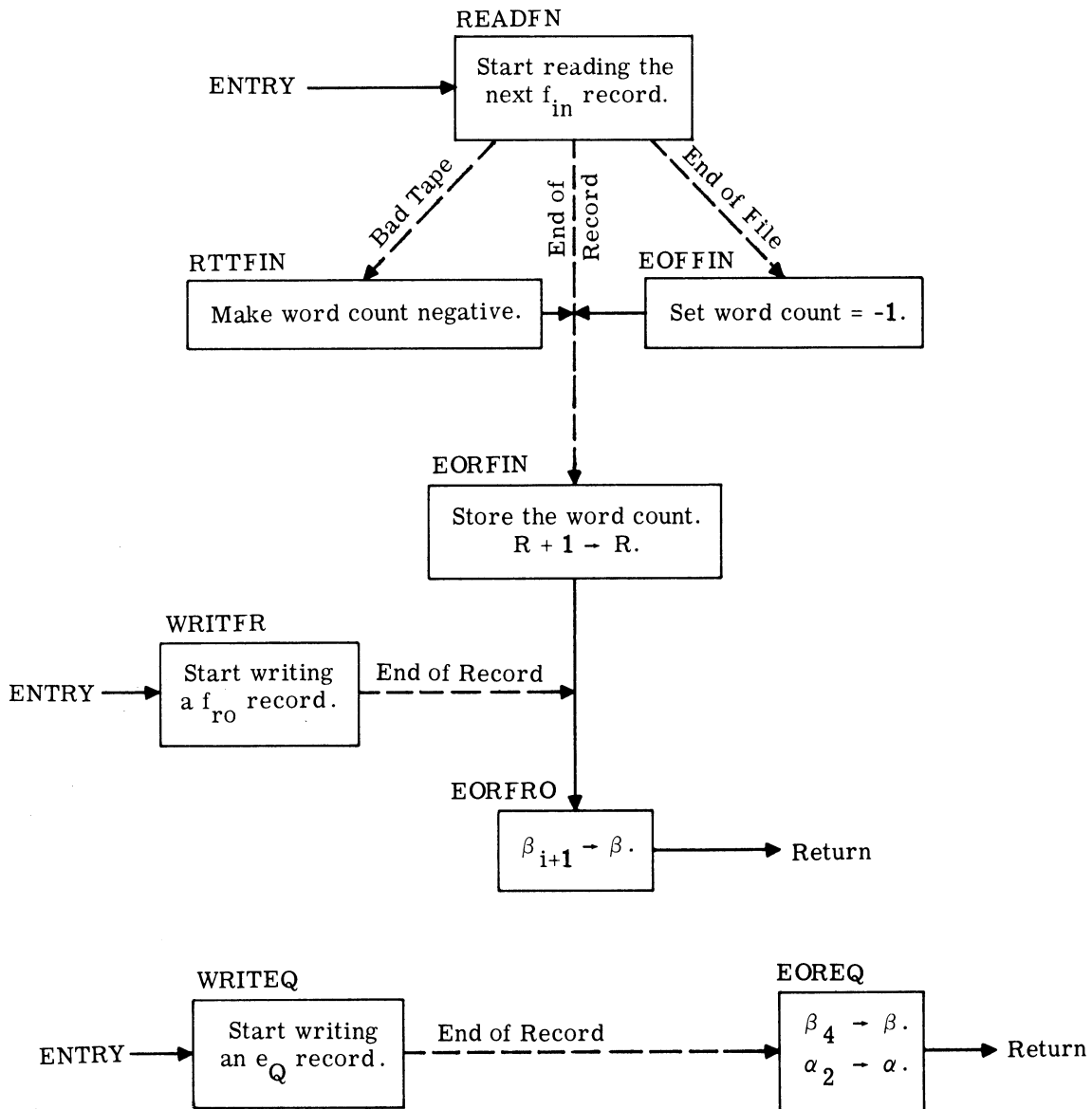


Fig. 27. Iteration part of TR 200 subroutine.

A. Buffering

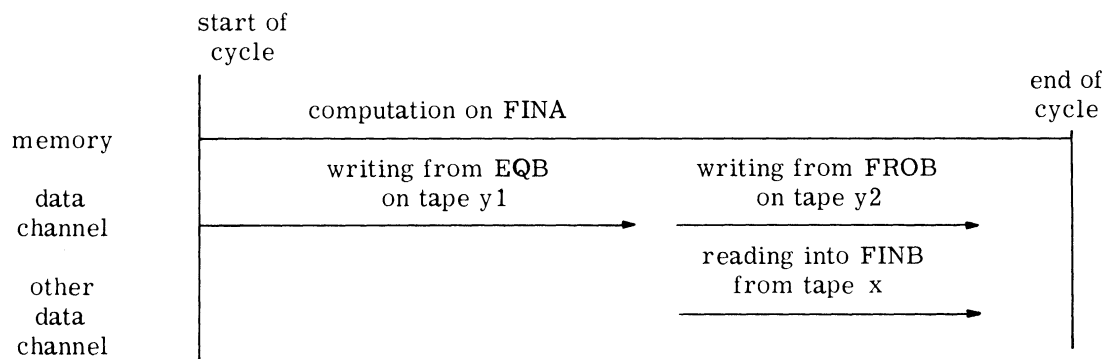
Three of the four available scratch tape units are needed for this program, and it is assumed that all of them will never be assigned to the same data channel. (If they are, the program is slower in operation by approximately the time it takes to read the input tape.) The input tape of  $f_{in}$ 's should be mounted on tape x, and two work tapes for the  $e_Q$ 's and  $f_{ro}$ 's should be on units y1 and y2, where x is on one data channel and y1 and y2 are on the other. x, y1 and y2 must be specified by a data card (Section I. A).

Memory is buffered also, into two sets of A and B storage areas. There are two areas of 4950 words each, FINA and FINB, for the  $f_{in}$ 's, and two similar areas, FROA and FROB, for the  $f_{ro}$ 's. The  $e_Q$ 's occupy the same place as the  $f_{in}$ 's. That is, after a word (3 characters) of  $e_Q(t_i)$  is computed, it is stored directly on top of the  $f_{in}(t_i)$ 's which generated the  $e_Q(t_i)$ 's. To summarize:

	<u>Tape</u>	<u>Memory</u>
$f_{in}$	x	FINA, FINB
$e_Q$	y1	EQA, EQB $\equiv$ FINA, FINB
$f_{ro}$	y2	FROA, FROB

Initially, one record of  $f_{in}$  is read into the B storage area, FINB. Computation on that record starts while the second  $f_{in}$  record is being read into FINA. At the conclusion of both the reading and the computation, the steady-state situation is started.

Computation is started on the  $f_{in}$ 's in FINA, and writing onto tape y1 is started from the EQB (or FINB) area. When the writing is completed, writing onto tape y2 is started from the FROB area, and reading from tape x is started into FINB. When reading, writing, and computation are all complete, regardless of the order in which they are completed, the cycle is repeated with the A's and B's reversed.



After the last EQ and FRO areas have been written on tape, tapes y1 and then y2 are rewound and written onto tape x, following the last-used  $f_{in}$  record. This copying is not buffered.

B. Storage Table

Following is a partial list, with explanations, of variables used in the program. Vectors (noted with the subscript v) are considered as extending in the opposite direction from those of MAD. That is, if FRO is location 1235, then FRO(1) is location 1236 and FRO (-2) is location 1233. All addresses are in octal. Absolute addresses are noted with the subscript a.

Program Parameters

TAPEX	10000 <sub>a</sub>	the tape from which the $f_{in}$ 's are read.
TAPEY1	10001 <sub>a</sub>	the tape on which the $e_Q$ 's are written temporarily.
TAPEY2	10002 <sub>a</sub>	the tape on which the $f_{ro}$ 's are written temporarily.
M	10003 <sub>a</sub>	the number of past samples used to compute $f_p(t_1)$ .
RECCT	10006 <sub>a</sub>	the number of records to be read from the input tape.
WRDMAX	10007 <sub>a</sub>	the maximum number of words allowed in any one input record. This is in the TR 000 routine, and is implied by BSS pseudo ops in the TR 200 subroutine.

Switches

FINXY	1205	} switches designed to alternate reading and writing into and from different buffer areas.
EQXY	1206	
FROXY	1207	
FINSW	1210	} switches designed to alternate computing on different buffer areas.
FROSW	1211	
ALPHA	511	a switch which is set to ALPHA2 at the end of writing a record of $e_Q$ 's onto tape. At ALPHA2, writing the previous $f_{ro}$ record and reading the next $f_{in}$ record are started. At this point, ALPHA is set back to ALPHA1.
BETA	512	a switch which is also a delaying mechanism. At the end of writing a record of $e_Q$ 's, BETA is set to BETA4, an entry which is equivalent to the ALPHA2 entry.

Regardless of the entry which starts the writing of the  $f_{ro}$ 's, when this does occur, BETA is set to TRA BETA (a one-word loop). The order after BETA is always TRA BETA. At the end of computation on one record, control is transferred to BETA. At the end of writing the  $f_{ro}$ 's, the address part of BETA is increased by one. At the end of reading the next  $f_{in}$  record, BETA is also increased by 1. This insures that the 3rd order after BETA is not entered until a complete cycle of input, output, and computation (Section II. A) has been made.

GAMMA 537 a switch which is set to GAMMA2 when there are no more input records to be read.

Conversion Tables

TABLEA <sub>v</sub>	50121	}	used for conversion of input and output between Mode II and IBM 709 integer forms. If $m$ is a quantity in Mode II form and $i$ is the same quantity expressed as a 709 integer, then the value at TABLEA + $m = i$ , and the value at TABLEB + 512 + $i = m$ .
TABLEB <sub>v</sub>	54121		
TABLEC <sub>v</sub>	10011 <sub>a</sub>		used for quantizing $e_M$ . $ e_Q $ is located at TABLEC + $ e_M $ . TABLEC is computed by the QUAN subroutines.

Computation Storage

FIN <sub>v</sub>	1274	FIN(-3), FIN(-2), and FIN(-1) hold the addresses of three $f_{in}$ 's in IBM 709 integer form (as opposed to Mode II form). FIN holds 3 $f_{in}$ 's in Mode II form.
EQ <sub>v</sub>	1270	EQ(-3)...EQ(-1) contain the addresses (in TABLEB) of the Mode II values of three $e_Q$ 's. EQ holds the last-computed $e_Q$ .
FRO <sub>v</sub>	1235	FRO(-3)...FRO(0) are used in a similar fashion to the corresponding elements of the EQ vector. FRO(1)...FRO(6) hold the last $M f_{ro}$ 's for use in the computation of $f_p$ .



FINA <sub>v</sub>	1371	} are buffer areas for records. The EQ areas are the same as the FIN areas.
FINB <sub>v</sub>	13117	
EQA <sub>v</sub>	1371	
EQB <sub>v</sub>	13117	
FROA <sub>v</sub>	24645	
FROB <sub>v</sub>	36373	
ESSQ	1257	holds the running total of $f_{in}^2(t_i)$ .
ENSQ	1260	holds the running total of $[f_{in}(t_i) - f_{ro}(t_i)]^2$ .
FPMAX	1252	} contain, respectively, the maximum values of $f_p$ , $e_M$ , and $f_{ro}$ in each record.
EMMAX	1253	
FROMAX	1254	

### Counting Mechanisms

R	1255	counts the number of records which have been read into memory.
RR	1256	counts the number of records on which computation has been started.
WC <sub>v</sub>	1275	contains the number of words in each record of input which are to be used for computation. This is always adjusted so that it is never less than 1 in absolute value.
COUNT	1261	is set to M at the beginning of each record and is decremented until it reaches zero. This determines the computation of $f_p(t_i)$ .
WORD	1245	counts the number of words in each record, starting at 0.
WORDCT	1246	holds the number of words in the $f_{in}$ record presently being used for computation.
FPXS	1247	} count the number of $f_p$ 's, $e_M$ 's, and $f_{ro}$ 's in each record that exceed the maximum allowable value.
EMXS	1250	
FROXS	1251	
EMMOST	10010 <sub>a</sub>	a number which is considered the maximum allowable value of $e_M$ .

### C. Explanation of Program Cards

TR 000-015. This is the main routine of the deck described in Section I. A. The following steps are performed in the order given.

1. reads in the data card with the scratch tape assignments
2. reads in the data card with the computation parameters,

- tape identification, and record count
3. prints a note to the operator to mount the tape, and pauses
  4. rewinds all tapes
  5. prints the title
  6. executes the QUAN subroutine
  7. executes the TR subroutine
  8. rewinds the tape
  9. prints a note to the operator to dismount the tape, and pauses
  10. dumps memory and exits to the system.

TR 016. This is an override which eliminates Step 3, above. This is to be used when it is unnecessary, for any reason, to mount a special tape.

TR 017-018. This is an override which eliminates Step 9, above. It is to be used when one does not need the output tape. Also, this eliminates Step 10, above, and causes a transfer at this point to Step 2.

TR 200-238. (See Figs. 22--27.) This is the principle computation subroutine, where the  $e_Q$ 's and the  $f_{rO}$ 's are computed and written onto the scratch tapes and then the end of the input tape. The following steps are included:

1. A conversion table is prepared, so that numbers in Mode II form can be converted to IBM 709 integer form, and vice versa.
2. To assist in debugging, the contents of certain registers are dumped at various points in the computation.
3. When reading or writing binary tapes, rereading or rewriting is not retried in case of error.
4. When a binary block with wrong parity or redundancy bits is read into memory from the  $f_{in}$  tape, the word count is recorded, a comment is printed, but computation tries to proceed, nevertheless.

Figures 22--27 show the flow diagrams for the TR 200-238 portion of the program.

TR 239. This is an override which was necessitated by an error in the present I/O supervisor routine. This overrides Step 3, above, so that reading and writing is tried up

to ten times in case of error.

TR 240-241. This is an override which affects Step 1, above. This allows a scale factor (less than one) to be applied to the input. This override is on top of the debugging dump which is overridden by TR 303. In other words, when TR 240 and TR 241 are being used, TR 303 must be used also.

TR 242. This is the identification which should be given to the scale factor mentioned above. This is an override to location 503 (octal). The form is:

```

                $Assemble,      Punch Object      TR 242
                END PGM
                SAK
503 OCT          XXXXXXXXXXXXX
                END
```

where XXXXXXXXXXXXX is a 12-digit octal fraction (all 12 digits required) representing the desired input scale factor times 1/2. If  $s$  is the decimal scale factor to be applied to the input samples:

$$\left(\frac{1}{2} s\right)_{10} \longrightarrow \left(\frac{1}{2} s\right)_8 \longrightarrow \text{location 503 (as above).}$$

TR 300-304. These are overrides which eliminate the printing mentioned in Step 2, above. These should be used for all non-debugging runs.

TR 305-307. This override changes the "Record No. X has XXX Words." print-out to read "Record No. X has YYY Samples.", where the number of samples is the number of words times 3, or the actual number of Mode II samples in the record.

TR 310. This overrides the selective dump in case of format error in a particular record. The dump is eliminated.

TR 400-401. These are overrides which change Step 4, above, so that computation is not tried on bad binary blocks. These cards are necessary because the present I/O supervisor does not read in the entire record when wrong parity or redundancy bits are detected, and even a partial word is not given.

QUAN1 000-004. This subroutine prepares a table for quantizing  $e_M$  by the linear method. (See Fig. 20.)

QUAN2 000-004. This subroutine prepares a table for quantizing  $e_M$  in a logarithmic manner where the output levels increase in powers of 2. (See Fig. 21.)

QUAN2 005. This is an override which changes the logarithmic quantizing so that the output levels increase by powers of four, rather than two.

SINEF 000-008. This routine reads in data in Mode II form, duplicates the string of data to form a repetitive string of about 4950 words, and writes a series of identical records on tape. The following steps are performed in the order given.

1. reads a data card giving the record count, the tape identification, and the input tape assignment
2. prints a note to the operator to mount the tape, and pauses
3. reads in the basic string of data
4. duplicates the data in a block image in memory
5. writes a given number of blocks onto the tape, inverting the word order. Note that the order of IBM 709 words is inverted, not the order of the Mode II characters. To read in  $n$  words, a series of  $3n$  Mode II characters  $a_1 a_2 a_3 a_4 \dots a_{3n}$ , one must punch the characters in the following order; grouping 3 characters to a 709 word:  $a_{3n-2} a_{3n-1} a_{3n}$ ,  $a_{3n-5} a_{3n-4} a_{3n-3}$ ,  $\dots$ ,  $a_1 a_2 a_3$ .
6. executes the subroutine SEQPGM, which calls in the next program (actually, the next core load in a ping-pong deck) from tape. This should be the deck listed in Section I. A.

SINEF 009. This override eliminates Step 2, above. It is used if a special tape is not used. Card TR 016 eliminates the tape mounting instruction in the main program and must be used if the SINEF routine is used. TR 017 and TR 018 must also be used to eliminate the tape dismounting instruction.

## D. Areas of Code Improvement

### 1. Possible sources of bad runs

No satisfactory provision has been made for the possibility that one of the tapes is too short. In case the temporary storage tape for either the  $e_Q$ 's or the  $f_{ro}$ 's is too short, the program exits to the system's ERROR subroutine. If the input tape is too short to carry the  $e_Q$ 's and the  $f_{ro}$ 's, copying stops and a normal exit is forced.

There is also no provision made for an RECCT than the actual number of records on the input tape. If this is the case, the program exits to the ERROR subroutine after trying to read the RECCT + 1st record, and all output computed up to that point is lost.

### 2. Increasing program speed

There are three obvious areas of savings:

- a) One can compute  $S^2$  and  $N^2$  for each record in integer mode, adding these in floating point mode to grand totals only at the end of computation on each record.
- b) One can buffer the final copying loop.
- c) It may not continue to be necessary to check computed values against a maximum for each record.

### 3. Rescaling output:

1. If one rescales at 370 (after storing FRO), and if the rescaled number exceeds the maximum, should FRO (original value) be truncated also?
2. Should  $e_Q$  be rescaled also? If so, question 1 applies here also.
3. If one rescales by changing TABLEB, what does one do then with too large values? Also, this means that  $e_Q$  is automatically rescaled along with  $f_{ro}$ .

### 4. Introducing noise:

It would be convenient and/or necessary to introduce some new variables, which would have functions analogous to the variables with similar names, excluding the terminal N (for noise):

MN would be used to count from M to 0 at the beginning of each record.

FRON(1)...FRON(6) would be used to hold the M previous output values

computed by the receiver. FRO(1)...FRO(6) would then hold the M previous output values computed by the transmitter.

FPN is the value of  $-f_p$  computed by the receiver, from the FRON vector.

FRON would be the newest output value computed by the receiver.

EQN would hold EQ with noise.

1. One would compute as before up to, but not including, 376. This computes FP, EQ, and FRO, checking each value with the maximum to date in the record and the maximum allowable value.
  2. At this point, one must introduce a piece of coding similar to that from SKIP-2 up to COMPEQ, computing and checking FPN, and doing something special for the first M values.
  3. Noise would be added to EQ, forming EQN. This must be checked with maximum values (?).
  4. FRON is computed from EQN and FPN, and checked.
  5. FRON must be used in the S/N computation.
  6. The vector FRON must be moved up at GETNEX similarly to the vector FRO.
  7. Registers holding the number of values exceeding the maximum allowable values, and the maximum for each record, must be zeroed, used, and printed.
5. Changing to Mode I operation:
1. TABLEA and TABLEB must be changed (the SETABL section of code).
  2. Format checking at GETFIN must be eliminated.
  3. At OKAY,  $f_{in}$  should be shifted left only 6 bits at a time. To effect this, one must have FIN BES 6.
  4. The SKIP-3 order must be an AXT 6, 2.
  5. One must also expand the EQ and FRO vectors by BES 6 pseudo ops.
  6. One must change the PUTEQ and PUTFRO sections, shifting only six bits each time.

7. The maximum allowable values of  $f_p$  and  $f_{r0}$  must be changed from 511 to 63.
8. In the QUAN routines, all references to 511 must be changed to 63.

It would be feasible to do away with TABLEA and TABLEB entirely, but this would necessitate more drastic code changes.

## REFERENCES

1. R. E. Graham, "Predictive Quantizing of Television Signals," Bell Telephone System Monograph 3272, 1958 IRE Wescon Convention Record, Pt. 4, pp. 147-157.
2. E. Pieruschka, "The Signal-to-Noise Ratio in Delta Modulation Systems," Technical Report 2R27F, Research Laboratories, Ordnance Missile Laboratories, Redstone Arsenal, March 10, 1958 ASTIA No. 156 165.
3. Davenport and Root, An Introduction to the Theory of Random Signals and Noise, McGraw-Hill Book Co. , 1958.
4. P. Elias, "Predictive Coding," Part 1 and Part 2, IRE Transactions on Information Theory, March 1955.
5. R. E. Graham, "Communication Theory Applied to Television Coding," Bell Telephone System Monograph 3096.
6. H. van der Weg, "Quantizing Noise of a Single Integration Delta Modulation System with an N-Digit Code," Phillips Research Report No. R223, 1953, pp. 367-385.
7. F. de Jager, "Delta Modulation, A Method of PCM Transmission Using the One Unit Code," Phillips Research Report 7, No. R203, 1952, pp. 442-466.
8. L. H. Zetterberg, "A Comparative Study of Delta and Pulse Code Modulation," Third London Symposium on Information Theory (Edited by Colin Cherry), 1956.
9. S. K. Bowers, "What Use is Delta Modulation to the Transmission Engineer?" Transactions of AIEE (Communication and Electronics), No. 30, May 1957, pp. 142-147.
10. J. Holzer, "Exponential Modulation for Military Communications," Technical Memorandum No. M-1777, SCEL, Fort Monmouth, New Jersey, June 1, 1956.
11. Schouten, Jaeger and Greefkes, "Delta Modulation," Phillips Technical Review, March 1952.
12. R. L. Miller, "The Possible Use of Log Differential PCM for Speech Transmission in UNICOM," Paper presented at GLOBECOM V, May 22-24, 1961, Chicago, Illinois.
13. David, Mathews and McDonald, "Experiments With Speech Using Digital Computer Simulation," Bell Telephone System Monograph 3405, Proceedings of 1958 National Electronics Conference, pp. 766-775, and IRE Western Joint Computer Conference, 1959, pp. 354-357.
14. A. Lender and M. Kozuch, "Single Bit Delta Modulating Systems," Electronics, November 17, 1961, pp. 125-129.



DISTRIBUTION LIST

(1 copy unless noted otherwise)

OASD (R&E), Rm 3E1065  
ATTN: Technical Library  
The Pentagon  
Washington 25, D. C.

Chief of Research and Development  
OCS, Department of the Army  
Washington 25, D. C.

Chief Signal Officer  
ATTN: SIGRD-6  
Department of the Army  
Washington 25, D. C.

Chief Signal Officer  
ATTN: SIGOP-5  
Department of the Army  
Washington 25, D. C.

Chief Signal Officer  
ATTN: SIGAC  
Department of the Army  
Washington 25, D. C.

Chief Signal Officer  
ATTN: SIGPL  
Department of the Army  
Washington 25, D. C.

Director, U. S. Naval Research Laboratory  
ATTN: Code 2027  
Washington 25, D. C.

Commanding Officer and Director  
U. S. Navy Electronics Laboratory  
San Diego 52, California

U. S. National Bureau of Standards  
Boulder Laboratories  
ATTN: Library  
Boulder, Colorado

Commander  
Aeronautical Systems Division  
ATTN: ASAPRL  
Wright-Patterson Air Force Base, Ohio

Commander, Air Force Cambridge Research  
Laboratories  
ATTN: CRO  
L. G. Hanscom Field  
Bedford, Massachusetts

Commander, Air Force Command and  
Control Development Division  
ATTN: CCRR & CCSD  
L. G. Hanscom Field  
Bedford, Mass.

Commander, Rome Air Development  
Center  
ATTN: RAALD  
Griffiss Air Force Base, N. Y.

Commanding General  
U. S. Army Electronic Proving Ground  
ATTN: Technical Library  
Fort Huachuca, Arizona

Commander, Armed Services Technical  
Information Agency  
ATTN: TIPCR  
Arlington Hall Station  
Arlington 12, Va. (10 copies)

Chief, U. S. Army Security Agency  
Arlington Hall Station  
Arlington 12, Va.

Deputy President  
U. S. Army Security Agency Board  
Arlington Hall Station  
Arlington 12, Va.

Commanding Officer  
U. S. Army Signal Equipment Support  
Agency  
ATTN: SIGMS-ADJ  
Fort Monmouth, N. J.

U. S. Continental Army Command LnO  
USASRD  
Fort Monmouth, N. J. (3 copies)

Corps of Engineers Liaison Office  
USASRD  
Fort Monmouth, N. J.

AFSC Liaison Office  
Naval Air R&D Activities Command  
Johnsville, Pa.

Marine Corps Liaison Office  
USASRD  
Fort Monmouth, N. J.



DISTRIBUTION LIST (Cont.)  
(1 copy unless noted otherwise)

Commander, Air Force Command and  
Control Development Division  
ATTN: CRZC  
L. G. Hanscom Field  
Bedford, Mass.

Commanding Officer  
USASRDL  
ATTN: Dir of Research/Engineering  
Fort Monmouth, N. J.

Commanding Officer  
USASRDL  
ATTN: Tech Documents Center  
Fort Monmouth, N. J.

Commanding Officer  
U. S. Army Signal R&D Laboratory  
ATTN: SIGRA/SL-NDM  
Fort Monmouth, N. J.

Commanding Officer  
USASRDL  
ATTN: Logistics Division  
For: B. J. Keigher  
Fort Monmouth, N. J. (9 copies)

Commanding Officer  
U. S. Army Signal R&D Laboratory  
ATTN: Tech Info Div  
Fort Monmouth, N. J. (3 copies)

Chief of Naval Operations  
U. S. Navy Department  
ATTN: E. H. Stuermer (OP-07T1C)  
Washington 25, D. C.