General Disclaimer

One or more of the Following Statements may affect this Document

LINKABIT CORPORATION

10453 ROSELLE STREET
UNIVERSITY INDUSTRIAL PARK
SAN DIEGO, CALIFORNIA 92121

LINKABIT CORPORATION
10453 Roselle Street
San Diego, CA 92121

Final Report
for the
Design of a Digital Voice Data
Compression Technique for
Orbiter Voice Channels

28 April 1975

Contract No. NAS9-14415

DRL No. 2

Submitted to:

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0   Introduction

With the growth of digital space communications, the requirement for compressed digital voice transmission has assumed prime importance. Particularly in shuttle orbiter applications, where the majority of the digital transmission will be voice, the reduction of transmitted data rate below the presently planned 32 Kbps per voice channel would have major impact on the overall system design.

LINKABIT has performed a thorough investigation of candidate techniques for digital voice compression to a transmission rate of 8 Kbps. Besides the basic goal to achieve good voice quality and speaker recognition, considerable attention has been devoted to providing robustness in the presence of error bursts, as will occur when error-correcting coding is applied on the channel.

This report describes a new technique, delayed decision adaptive predictive coding, and demonstrates its potential advantages over conventional adaptive predictive coding (APC).

The main output of this study is a set of experimental simulations recorded on analog tape, which forms an integral part of this report. As discussed in Section 4.0, the tape demonstrates the potential improvement achievable with delayed decision APC over conventional APC, as demonstrated

on two FM broadcast segments. In addition, it shows that the performance of this new technique is virtually undegraded when the channel Viterbi decoder bit error rate is 10^{-3}, and the degradation is tolerable even at a bit error rate of 10^{-2}.

Preliminary estimates of the hardware complexity of this technique indicate the potential for practical implementation in space shuttle orbiter applications.

2.0 Background on Digital Voice Compression Techniques.

A variety of digital voice compression techniques have found application in digital communication systems over the past decade. These range in complexity from conventional PCM and simple delta modulation to sophisticated adaptive predictive encoders. Listed in approximate order of complexity, the six major categories of digital voice compression coding techniques are (References 1-11)

pulse code modulation (PCM)

delta modulation (ΔM)

differential PCM (DPCM)

adaptive delta modulation (AΔM)

adaptive DPCM (ADPCM)

linear predictive coding (LPC)

adaptive predictive coding (APC)

In fact, these various generally accepted techniques are not clearly distinct from one another. In the order given, from delta modulation through adaptive predictive coding, each technique represents an additional but moderate level of sophistication on one or more techniques higher on the list.

The last two compression techniques have the more
ambitious goal of speech analysis and synthesis, whose
classical predecessor is the channel vocoder. LPC attempts
to derive basic parameters of the speaker's vocal tract
and voice pitch and only these parameters are transmitted.
Though time-varying, these vocal tract and pitch parameters
have a bandwidth which is much lower than that of the voice
signal, thus affording a significant bandwidth compression
with consequent reduction in bit rate required for digital
transmission. At the receiver the voice is synthesized
by a filter model of the vocal tract driven by a pitch
generator and white noise for the voiced and unvoiced
sounds, respectively. Typically these vocal tract analysis-
synthesis techniques reduce the required transmission
rate to the order of between 2.4 Kbps and 10 Kbps, at a
significant cost in complexity, voice recognizability, and
susceptibility to channel errors. In contrast, the first
four techniques require transmission rates on the order
of 16 Kbps to 64 Kbps, the upper limit being typical of
that used by conventional PCM. The lowest rate speech
analysis-synthesis techniques (below 8 Kbps) would not
appear within the scope of the orbiter voice compression
study. However, it should be noted that some of the more
sophisticated techniques in the above list - notably adaptive
predictive coding - utilize approaches verging on vocal

tract analysis, and they approach the required bit rates
of the latter to within perhaps a factor of 2, with better
speaker recognition and immunity to channel errors.

Recent studies (Reference 12) have demonstrated
that many of the above techniques, ranging from delta
modulation through adaptive predictive coding, produce an
inherent tree-like code structure which is not fully
exploited in the conventional approaches. Multiple
simultaneous path searches through this code tree structure,
reminiscent of sequential decoding, appear to produce
improved performance.

In this section each of the basic conventional
digital compression techniques will be reviewed with
emphasis on their performance and implementation. Toward the
end of the section the multiple path search techniques will
be described. Existence of channel (error-correcting)
decoders of this type makes the implementation of such
techniques appear quite feasible with moderate complexity.
Furthermore, this is a natural extension to the APC techniques
considered the most promising of the classical approaches
for this application.

In Section 3 the details of the LINKABIT implementation
of this advanced APC technique will be described.

2.1   PCM, DPCM and ΔM

The oldest method for digital voice transmission is, of course, pulse code modulation (PCM) which consists of an analog-to-digital (A/D) converter employing a quantizer whose output is one of M levels of a "staircase" function, and a digitizer which assigns a binary codeword of length $\log_2 M$ bits to each of the levels. Much study (References 13, 14) has been devoted to optimizing the level spacing in the quantizer according to various performance criteria. For voice signals it was found that a compander, consisting of a memoryless nonlinearity used in conjunction with the quantizer, significantly improved voice quality. The most widely accepted such compander performs the logarithmic mapping

$$y = \frac{V \log (1 + \frac{\mu |x|}{V})}{\log (1 + \mu)} \ \text{sgn} (x)$$

where x and y are input and output, respectively, and V and μ are parameter constants (clearly as $\mu/V \to 0$, the function becomes linear). PCM with logarithmic companding is often cited as a standard of comparison for the evaluation of compression techniques. However, care must be taken to filter out any DC component for otherwise it will produce an undesirable distortion when this companding nonlinear function is used.

The next oldest voice digitization technique, of which a variant also has been used extensively with analog communication systems, is delta modulation (ΔM). Illustrated by the block diagram of Figure 2.1a, ΔM utilizes the coarsest possible quantizer, a hard limiter, to determine whether the present sample is greater or lesser than an estimate of the sample and correspondingly outputs either a $+\Delta$ or $-\Delta$. This estimate is just the sum of all previous hard limiter outputs. At the receiver this same estimate is formed and converted into reconstructed analog voice by a D/A converter.

The step size Δ can not be chosen too large, for otherwise the quantization noise, referred to in this case as "granularity noise" (Figure 2.1b), will be intolerable; on the other hand, too small a choice of Δ will result in an inability to track rapid variations in the voice signal, an effect called "slope overload noise" (Figure 2.1b). Conventional or linear ΔM design involves a compromise between granularity and slope overload, with recent studies (Reference 15) seeming to indicate that the former is more objectionable to voice quality than the latter. The advantage of ΔM, besides its simplicity, is that it requires transmission of only one bit per sample. However, to achieve high quality the sampling rate must be several times greater than the Nyquist rate.

Transmitter

Filtered Input → A/D → (+ −) → Hard Limiter → +Δ → Coder → 0110.. To Digital Channel

Unit Delay

Transmitter
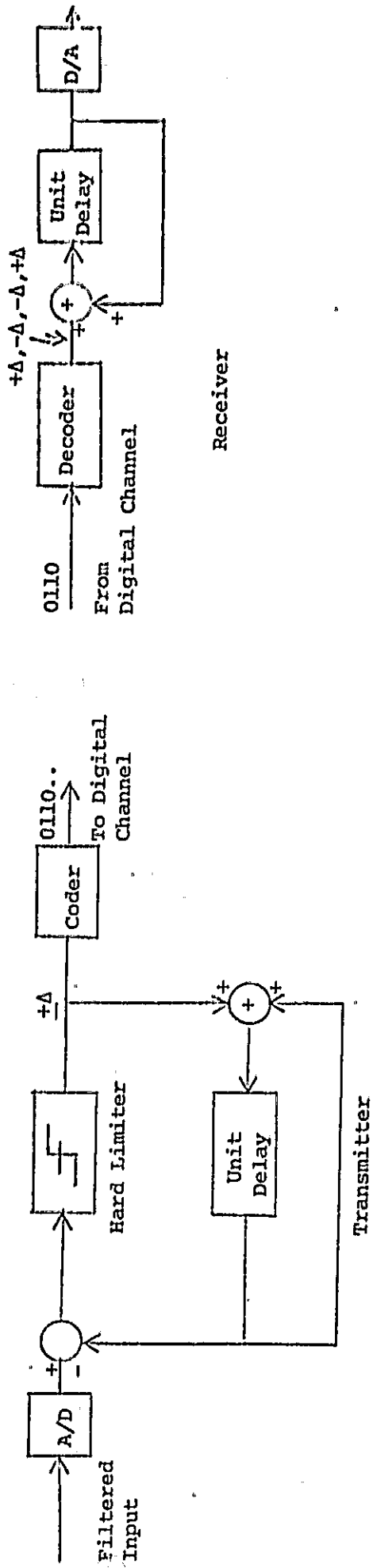
Figure 2.1a   Delta Modulation

Receiver

0110 From Digital Channel → Decoder → +Δ,−Δ,−Δ,+Δ → (+ +) → Unit Delay → D/A

Receiver

"Slope Overload" Noise

"Granularity" Noise

Δ
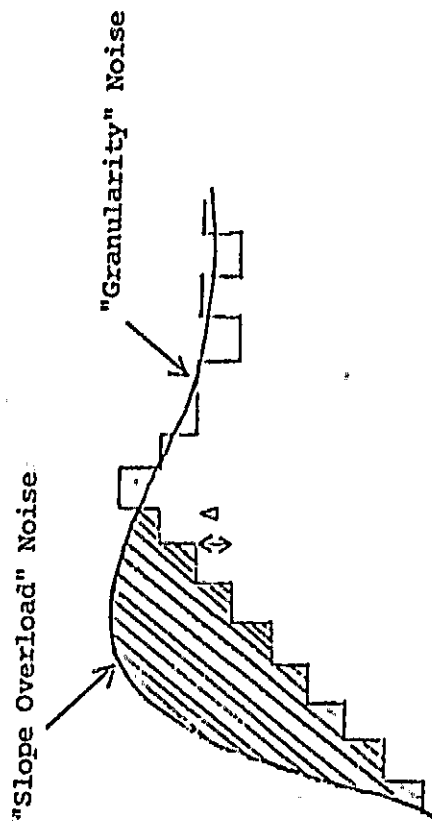
Figure 2.1b   Example of Delta Modulation Noise

Another approach, closely related to ΔM, is
differential PCM (DPCM) coding. In its simplest form the
encoder is the same as that for ΔM but with a multi-
level quantizer replacing the two-level hard limiter
(Figure 2.2). Thus this technique employs the quantizer
of conventional PCM on the difference between the present
sample and a quantized version of the last sample*. Use
of a more refined quantizer permits sampling to be performed
at the Nyquist rate or only slightly higher. However,
for a Q level quantizer the bit rate is now $\log_2 Q$ times
the sampling rate. Of course, the number of levels Q
is smaller than for conventional PCM, since the variance
of the sample differences is considerably less than that
of the samples. Relative performance of DPCM and ΔM
for the same bit rate is open to question, but ΔM is often
preferred for its simplicity. Both afford moderate
reductions in bit rate relative to PCM for the same
performance quality.

---

*A variation uses a linear prediction in place of the unit
delay, but this is relegated to Section 2.3 where the more
sophisticated technique of linear predictive coding is
discussed.
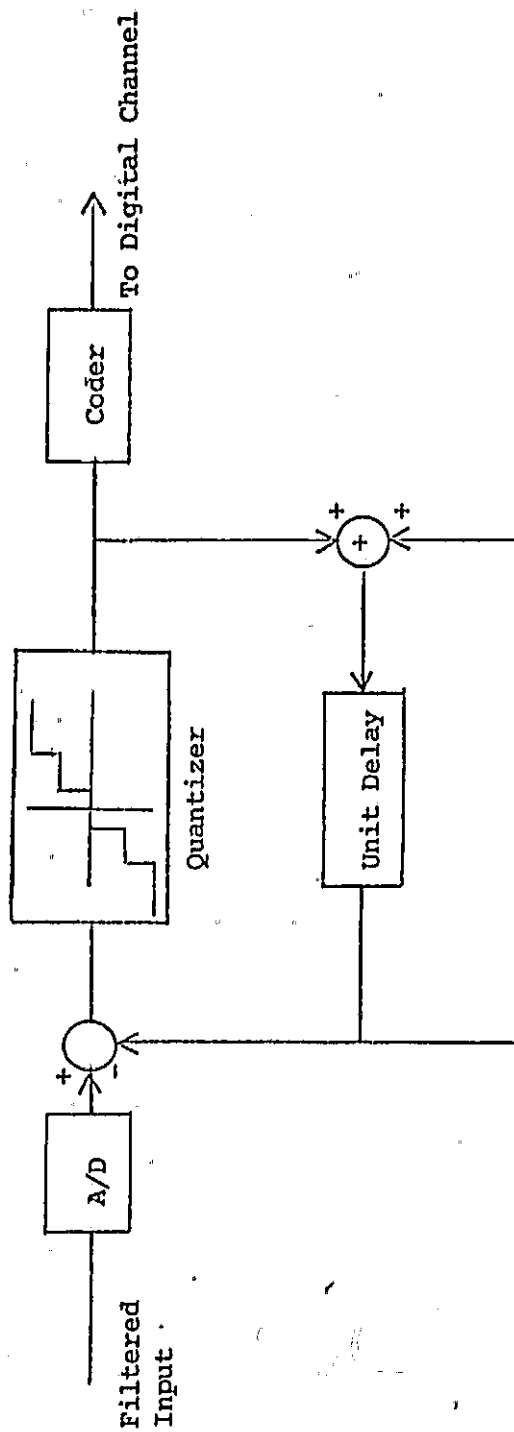
Figure 2.2   Differential Pulse Code Modulation (Transmitter)

2.2   Adaptive ΔM and DPCM

Adaptive variations on ΔM and DPCM, abbreviated AΔM and ADPCM, afford the possibility of varying step size Δ or quantizer level spacings based on the trends displayed by the last few quantizer outputs. First applied to ΔM (References 3, 16, 17), this has the advantage of reducing slope overload during periods of considerable signal variation while reducing granularity during periods of lesser variation and thus particularly reducing the idle noise.

Numerous formulas have been suggested for the variable step size as a function of previous quantizer outputs. Probably the simplest is the one which forms at the limiter output the present increment in terms of the last increment

$$\Delta_k = e_k \; \alpha^{e_k e_{k-1}} \; |\Delta_{k-1}|$$

where e_k and e_{k-1} are ± 1, the signs of the present and last quantizer outputs, Δ_k is the present increment, and $\alpha > 1$ is a constant. Thus if the limiter output changes sign, the increment is reduced, while if it remains the same indicating a potential slope overload condition, it is increased. Other more elaborate formulas have also been proposed (Reference 3).

Adaptive DPCM operates on the same principle as AΔM. Successive quantizer step increments are a function of the previous increment and the previous quantizer output. As an example, consider a 3-level quantizer with output (Figure 2.3)

$$y_k = \begin{cases} \sigma_k & \text{for } x_k > \sigma_k \\ 0 & -\sigma_k \le x_k \le \sigma_k \\ -\sigma_k & x_k < -\sigma_k \end{cases}$$

This can be made adaptive by varying its quantization level σ_k according to the formula

$$\sigma_{k+1} = \begin{cases} C_1 \sigma_k & \text{if } |x_k| \le \sigma_k \\ C_2 \sigma_k & \text{if } |x_k| > \sigma_k \end{cases}$$

where $C_1 < 1$, $C_2 > 1$

For more quantization levels more parameters are required. Empirically optimized values of these parameters are given in Reference 6 , as well as a measure of the performance improvement of this scheme over ordinary DPCM.

Receiver

Transmitter

$$\sigma_{k+1} = \begin{cases} c_1\sigma_k & |x_k| \leq \sigma_k \\ c_2\sigma_k & |x_k| > \sigma_k \end{cases}$$

$$C_1 < 1, \ C_2 > 1$$

Figure 2.3   Adaptive DPCM System

-13-

2.3   Adaptive Predictive and Linear Predictive Coding

Adaptive predictive coding (APC) is essentially a generalization on DPCM in which a linear predictor is used in the feedback path in place of the unit delay (Figure 2.4). This linear predictor can be modeled as a recursive or a nonrecursive (feed-forward) digital filter. In the simplest form (which is optimal for a first-order Gauss-Markov process), the predictor is simply an attenuated version of the previous increment, implemented by a unit delay followed by a scalar multiplier. More elaborate predictors (Reference 8) utilize a short-term predictor consisting of a linear function of the last few samples plus an attenuated replica of a sample M terms previous, where M represents the period of the quasi-periodic voice signal waveform. An example of such a predictor is shown in Figure 2.5.*

The limitation of predictive coding is that voice signals are basically nonstationary. Thus in particular the parameter M indicating the approximate period will vary from syllable to syllable and it will be inappropriate for unvoiced sounds. Similarly the short-term predictor coefficients provide accurate estimates only over a 5 msec to 10 msec interval. Thus for predictive coding to be useful for voice it must be made adaptive (APC). Techniques

---

*The simpler forms of such predictors are basically equivalent to the zero-order and first-order predictors often used in image data compression.

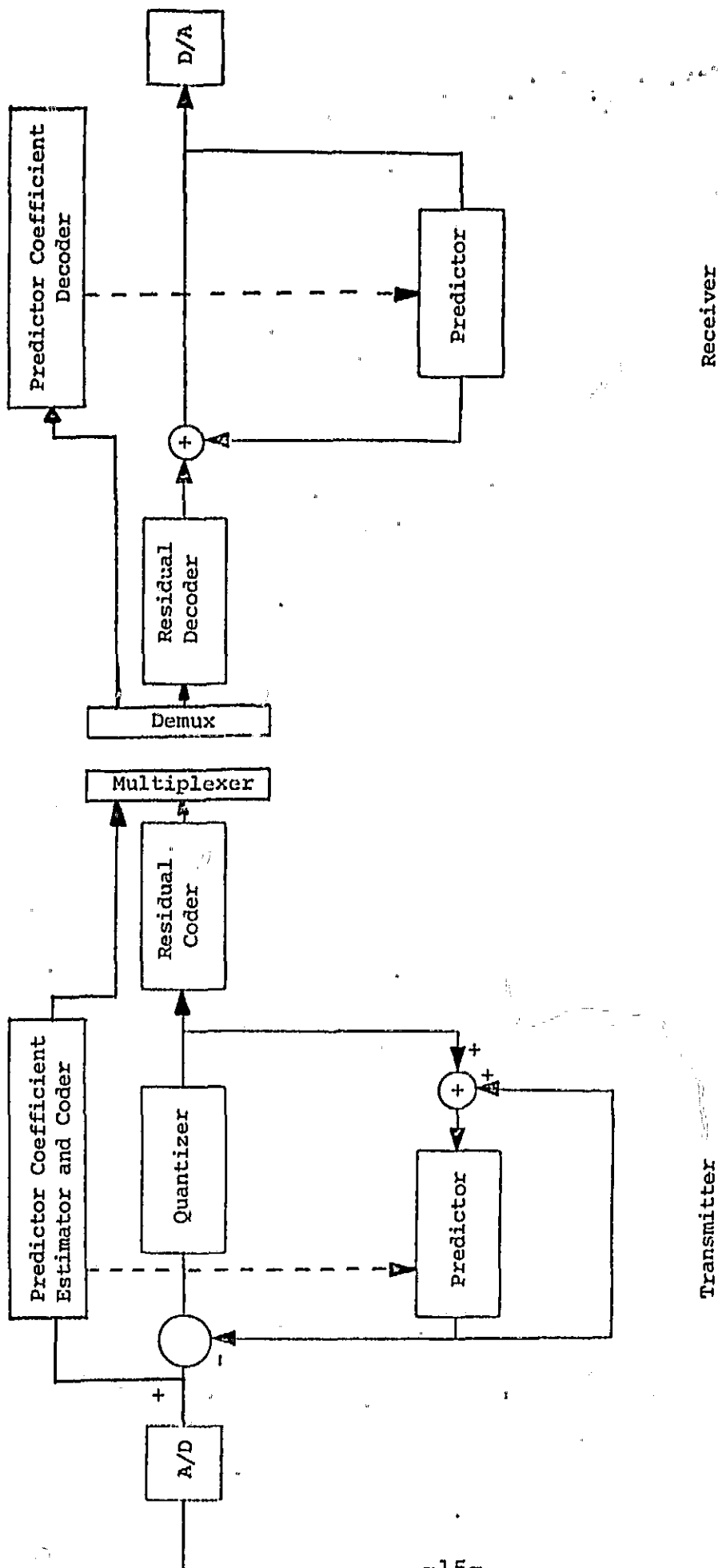Figure 2.4   Adaptive Predictive Coding

Input

Predictor
P₁(Z)

$P_1(Z) = \beta Z^{-M}$

Output

+

Predictor
P₂(Z)

$P_2(Z) = \sum_{n=1}^{8} \alpha_n Z^{-n}$

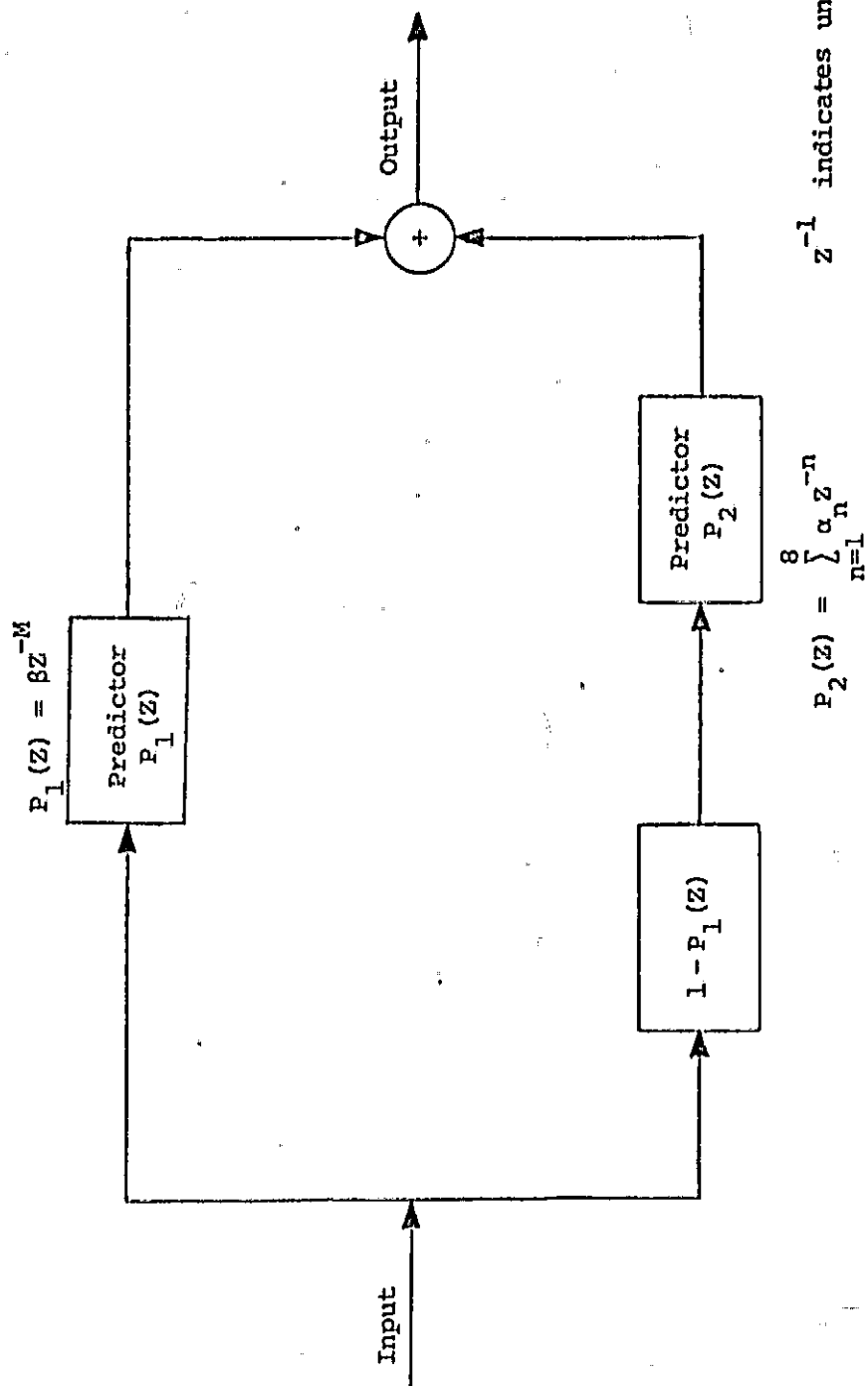1 − P₁(Z)

Z^{-1} indicates unit delay

Figure 2.5 Atal and Schroeder Predictor for Speech Signals

for measuring both the short-term and long-term predictor
coefficients generally involve measurements of the sequence
correlation function over the period in question (5 to 10
msec) followed by inversion of the correlation matrix to
solve the discrete (matrix) Wiener-Hopf equation. Eight-
tap adaptive predictors have been simulated with reasonably
good results (References 2, 8). Typically APC techniques
employ adaptive quantization, as used in ADPCM, as well as
adaptive adjustment of the predictor coefficients.

One problem with adaptive prediction is that the
transmitter must send the coefficients as well as the
quantizer outputs, sometimes called residuals. In one
implementation (Reference 2) speech is sampled at 8000 Hz
and a two-level quantizer generates an output at 8 Kbps.
The predictor coefficients are updated every 10 msec and
16 bits are used to transmit the parameters, requiring a
bit rate of 1.6 Kbps for parameter transmission and thus
a total bit rate of 9.6 Kbps.

On the other hand, a reasonable approximation to
the speech waveform can be obtained even without transmitting
the residuals. This is achieved by driving the receiver
digital filter (predictor in the feedback loop) by either
white noise - for unvoiced sounds - or a periodic pulse
train whose period, M, corresponds to the pitch period.

Thus in addition to the predictor coefficients, only this' pitch period parameter and a voiced/unvoiced decision needs to be transmitted (Figure 2.6). This technique known as linear predictive coding (LPC) requires only about one quarter to one half the transmission rate of APC, since residuals need not be sent, but it produces less acceptable performance and is more vulnerable to channel errors.
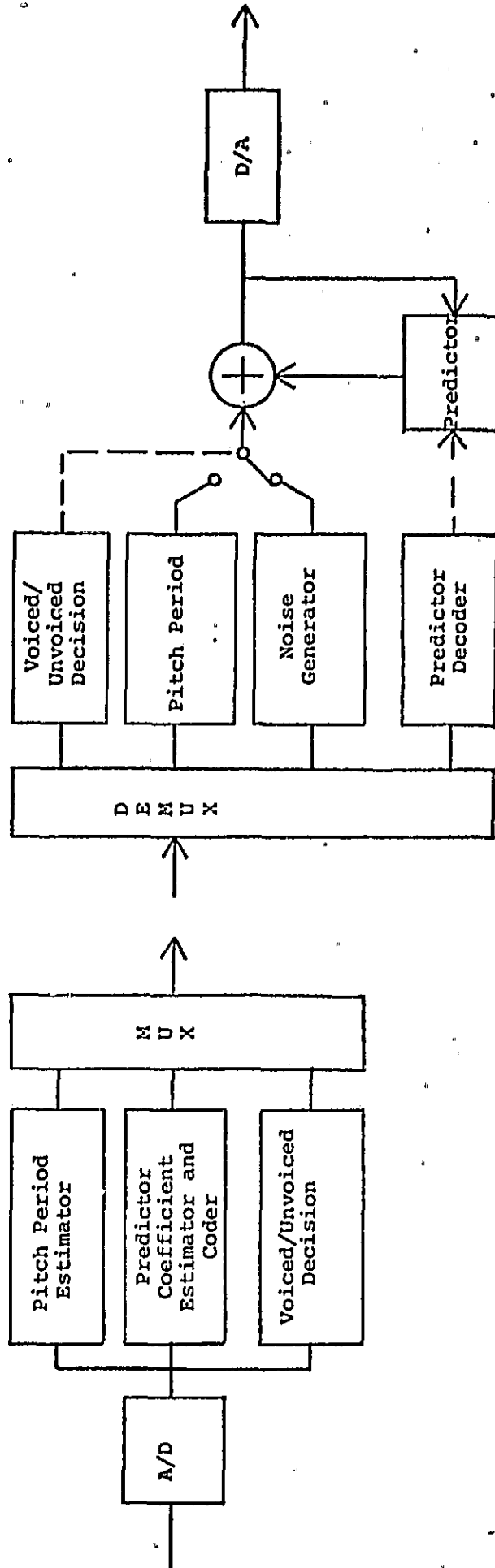
Figure 2.6   Linear Predictive Coding

2.4 Tree Structure of Digital Waveform Following Coding Techniques and More Elaborate Search Algorithms

All the techniques described thus far lend themselves to representation in terms of a code tree. The code tree of a single tap linear predictor*with two-level quantization is shown in Figure 2.7 with the hard-limiter quantizer step size normalized to unity. The conventional coding technique searches for a path through this tree, making decisions one branch at a time. That is, given that the search has led to a given node, the next node is chosen by comparing the two values of the branches stemming from this node with the input sample and choosing the best match. However, more elaborate tree searching techniques, common in channel decoding, may be employed to attempt to match longer segments of the input to the available codewords. By so deferring a decision it appears that better matches can be achieved overall than is possible by a series of decisions based on single branches. Such a source encoding algorithm can be implemented according to the block diagram of Figure 2.8. Storage must be provided for each of the multiple paths being searched simultaneously and for their distortion relative to the source. This distortion is updated at each node time and decisions made on which paths to pursue further.

---

*Similarly, code trees can be demonstrated for AΔM, ADPCM and other adaptive techniques.
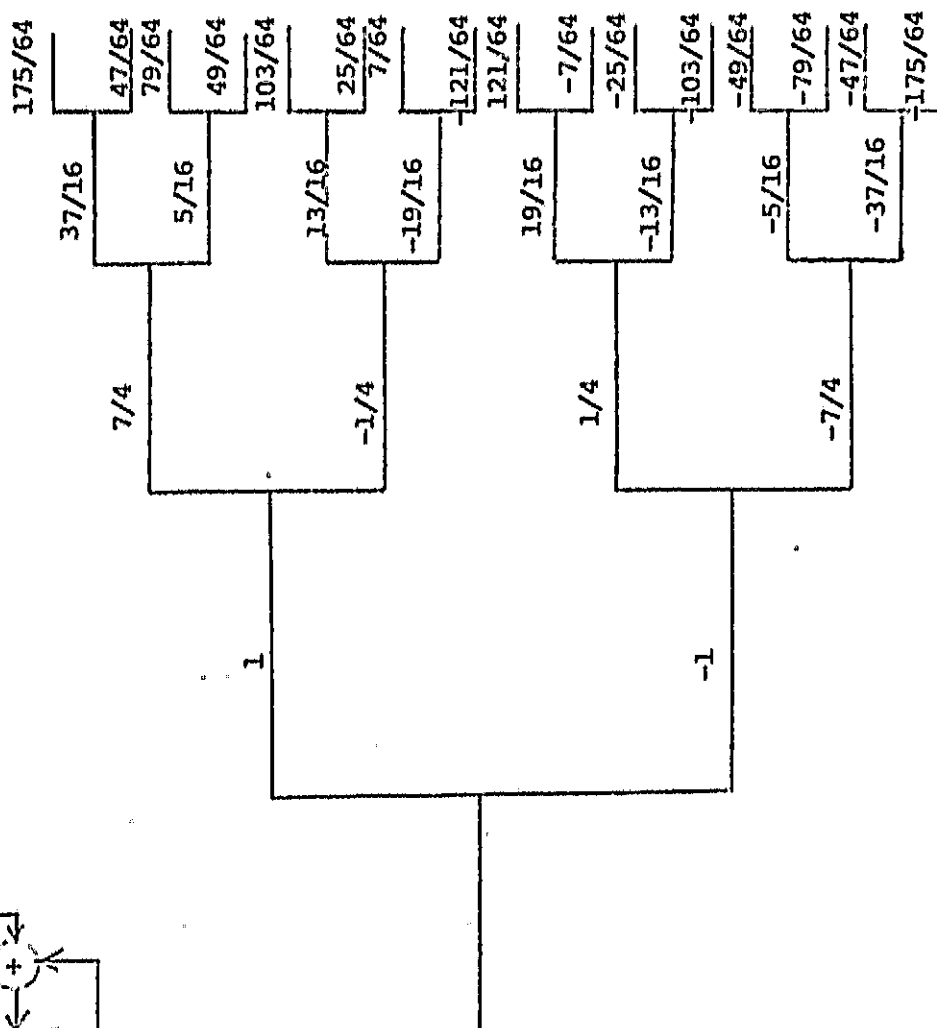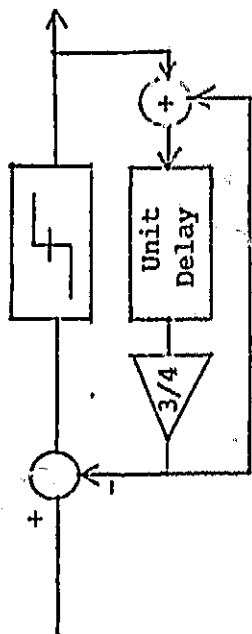
Figure 2.7 Code Tree for Predictive Coder with Hard Quantizer
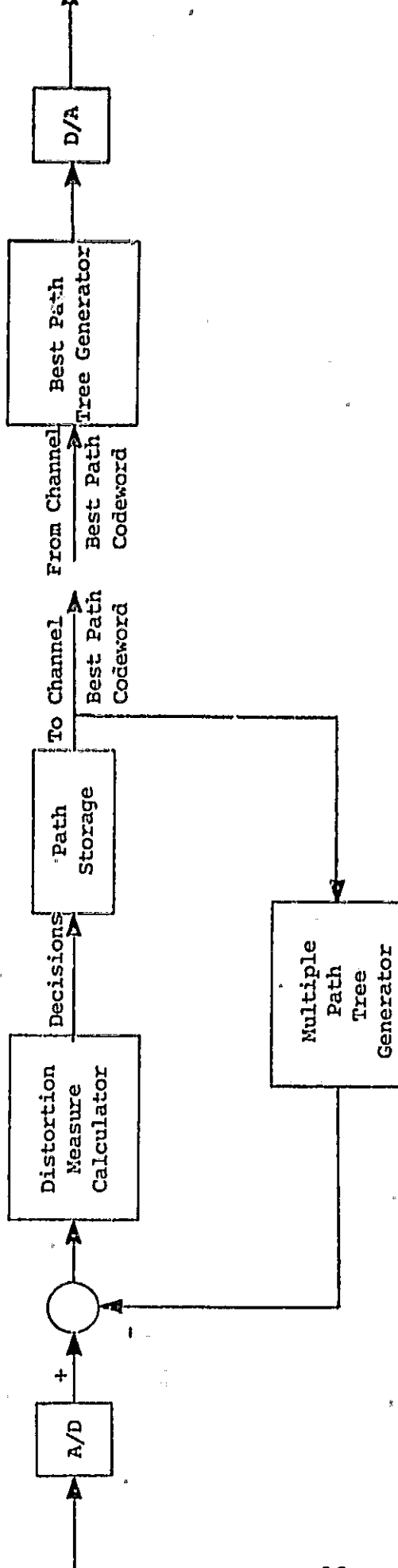
Figure 2.8 Source Tree Multiple-Path Encoder-Decoder

These multiple-path tree searching algorithms
are commonly used to decode convolutional codes transmitted
over a noisy channel. Recently these techniques have
also been proposed for source compression encoding (References
19 - 21). Two variations on sequential decoding searches
have been proposed (References 19, 21) and a direct analog
of a Viterbi decoding search has also been proposed and
analyzed for memoryless sources (References 20, 21). Most
of these studies have been either theoretical or based on
simulations with artificially generated source statistics.
On the other hand, very recently experiments have been
performed applying these techniques to voice. Using the so-
called M-algorithm (Reference 19) which preserves only the M
best paths in the sequential search, excluding all others,
Anderson and Bodie (Reference 12) have obtained considerable
improvement over DPCM at bit rates of 8 to 16 Kbps. Another
approach would be to preserve for each pair of paths
emanating from a given node the path which better matches
the source over the subsequent K branches; this approach
which corresponds essentially to the Viterbi algorithm,
requires the same storage as the M algorithm with $M = 2^{K-1}$ and
requires only about half as many comparisons per node.

While, as is shown in Reference 23, the Viterbi .
algorithm can be utilized for tree searching, even if the
tree does not have a finite-memory or remerging path
structure which reduces it to a trellis, there are two
advantages to be gained from assuming a trellis structure:

a) the predictor is a nonrecursive digital filter
(Figure 2.9) and consequently the tap coefficients
are less sensitive to quantization and to
approximation error,

b) channel errors have lesser effect since they
can influence the output over no more than
the memory (register length) of the predictor.

A finite memory linear predictive encoder, employing
three taps, with a hard quantizer, along with the
trellis structure of the code it generates, is shown in
Figure 2.9 as the simplest example of this coding technique.
The best path through the trellis is found by performing
pairwise comparisons, according to the Viterbi algorithm,
among all merging paths at each node level on the basis
of the distortion (mean square error or other convenient
measure) between the given path symbols and the digital
waveform to be encoded. These binary decisions only are
transmitted; at the receiver the closest matching path is
regenerated by passing the decision sequence through a
replica of the encoder nonrecursive digital filter (tapped
delay line). Note that for a K-tap filter, only 2^{K-1} states
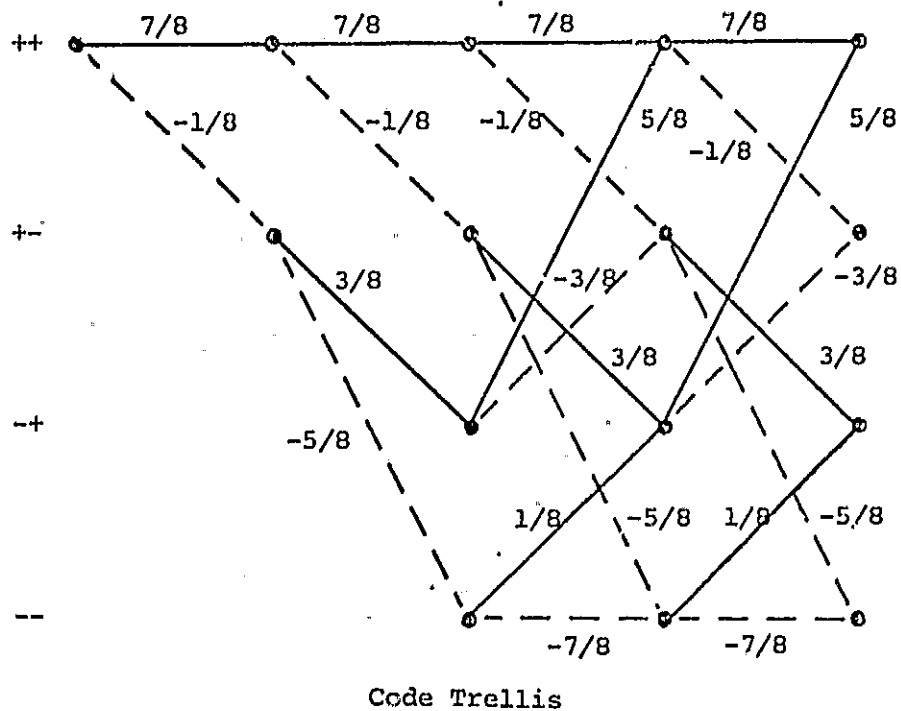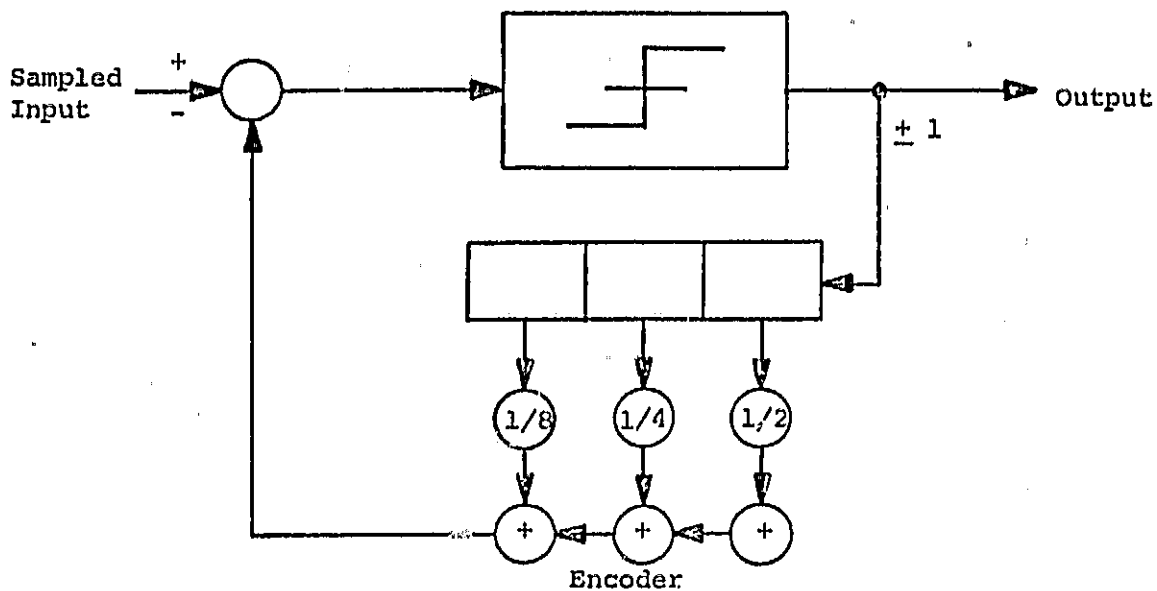
Figure 2.9   Simplified Predictive Coder Example

must be maintained and the path memory and metric for each state stored. Thus a 7 tap trellis source encoder is no more complex than the error control decoder employed in the orbiter communication system.

Adaptive adjustment of the tap coefficients in a manner quite similar to that used in APC is also possible using this scheme. Two approaches are suggested by existing APC techniques. In one case the nonrecursive filter coefficients are computed to best match the short term input statistics (autocorrelation function) over a syllabic period - 10 msec for example. These are transmitted separately by time division multiplexing with an additional data transmission overhead of 10% to 20%. A disadvantage of this adaptive approach is that each time the tap coefficients, and hence the trellis, is changed the previous trellis must be truncated with an additional overhead of K-1 bits. Because of the decision delay required for near-optimal Viterbi algorithm performance, tap adjustments must be delayed accordingly; however, this delay of a few samples is small compared to the "period" of the quasi-stationary voice signal. The advantage of this approach is twofold: not only is the additional transmission of tap coefficients avoided, but since the taps are adjusted continuously, and in the same way at both transmitter and receiver, no periodic trellis terminations are required.

3.0 APC with Delayed Decision Encoding

The LINKABIT speech compression experiments have focused on a variation of adaptive predictive coding (APC) (Reference 8) in which the usual memoryless predictor error signal quantizer is replaced by a delayed decision quantizer algorithm commonly known as the Viterbi Algorithm (VA). To simplify the discussion, the encoding and decoding techniques will initially be described for APC without "pitch prediction". The technique is later described for APC with pitch synchronous preprocessing.

3.1 General Description

The decoder for our VA APC encoding technique is illustrated in Figure 3.1. The 16 stage transversal filter shown is a nonrecursive approximation to a standard 4 pole APC decoding filter. The 16 tap weights represent the first 16 terms of the impulse response of the 4 pole APC decoder. The truncated impulse response is determined from the 4 lattice filter coefficients (Reference 24) that characterize the 4 pole prediction fitler. The adaptive nature of the coding technique is achieved by updating the predictor parameters periodically.

The speech encoder block diagram appears in Figure 3.2.
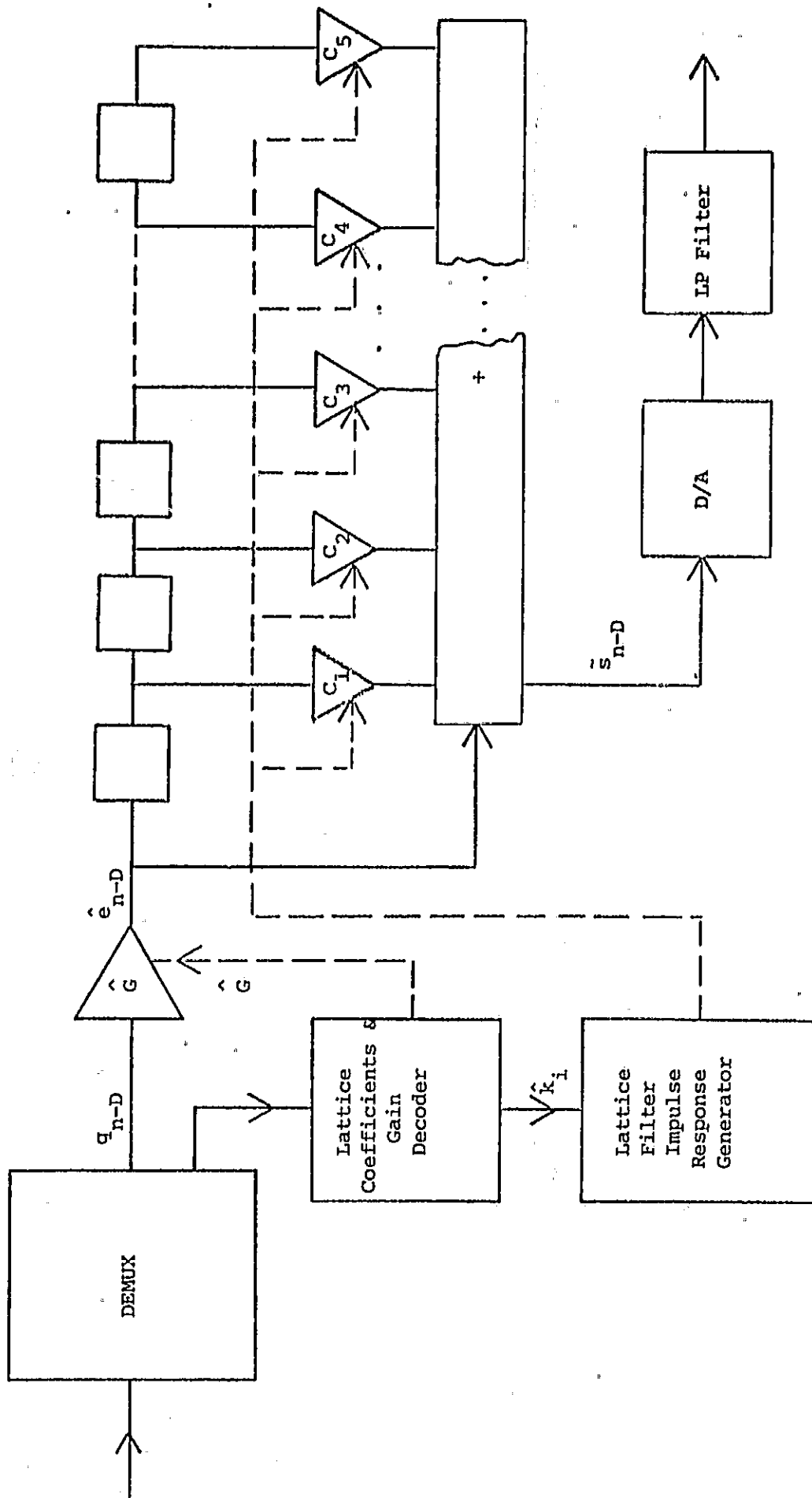
Figure 3.1   Delayed Decision APC Decoder Followed by D/A Conversion
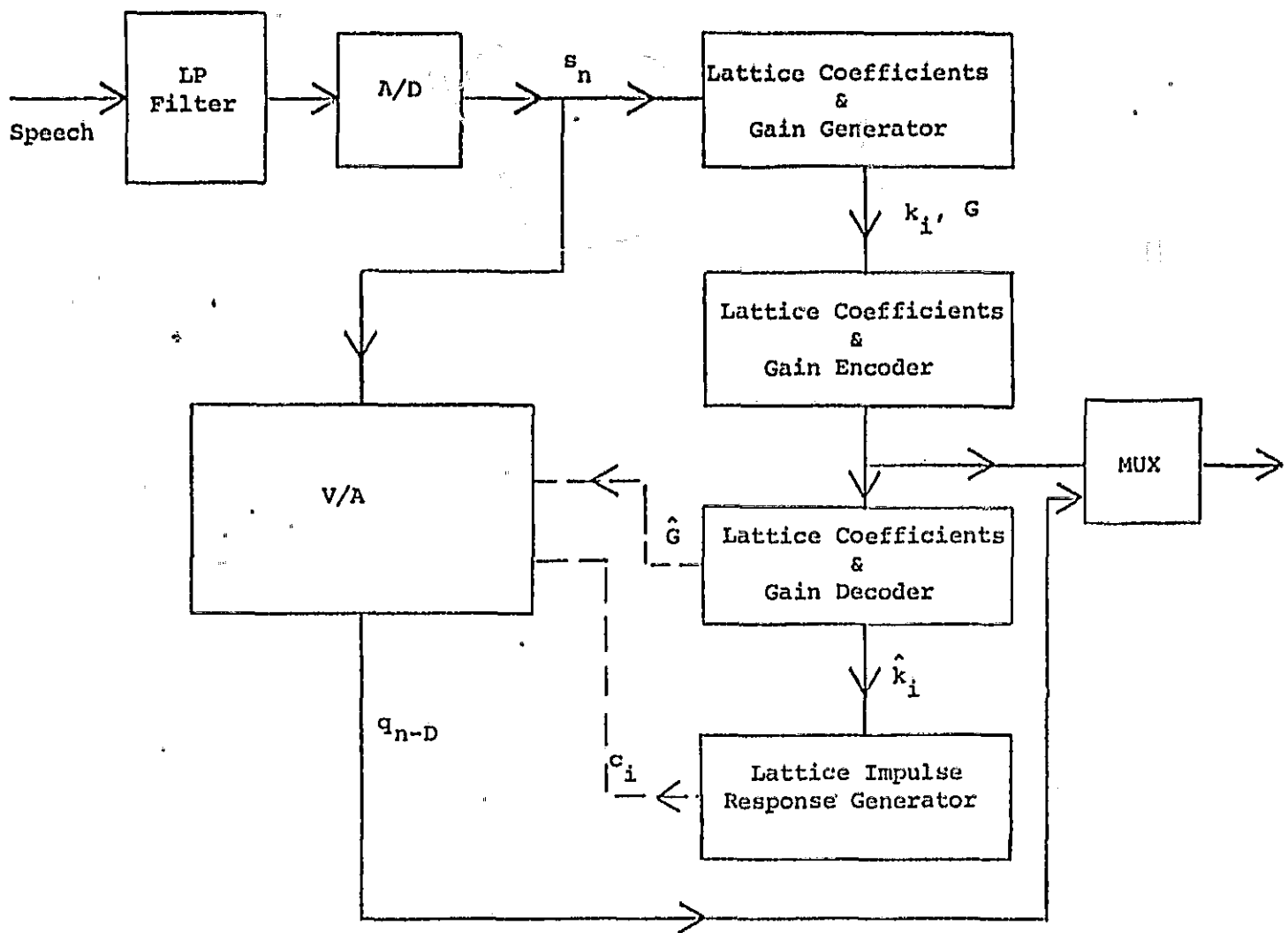
Figure 3.2   Delayed Decision APC Encoder Preceded by A/D Conversion

The Viterbi Algorithm inputs are digitized speech samples. The algorithm searches for the decoder binary driving sequence, q_n, that decodes into a speech sample sequence with a minimum-mean-square-error (MMSE) fit to the input speech sample sequence. Because the optimum decoder would be a cumbersome 2^{15} state machine we have used a more tractable 2^7 state suboptimum trellis search. The states of the trellis represent the possible states of the first seven stages of the decoder filter. Since the energy of the decaying impulse response of the decoder transversal filter is dominated by the leading 8 terms, the degradation in performance due to the reduced state search should be minimal. Some of the details of the VA appear in Section 3.2.

The predictor parameter selection algorithms are similar to those that might be employed for APC. The details and background appear in Section 3.3.

3.2   The Trellis Search (Viterbi) Algorithm

The Viterbi Algorithm trellis search as it is employed in the LINKABIT compression system is a $2^7 = 128$ state trellis search in which the states of the trellis represent the contents of the first seven decoder transversal filter cells. For each state, metrics are retained which indicate the quantization noise energy for that state relative to that of other states.

Trellis state transitions define only the contents
of the first 8 cells of the 16 stage decoder filter. Branch
metrics are therefore computed on the basis of the 8 bit
trellis state transitions bits and the most recent path
memory bits of the "from" state. Except that branch metrics
are determined from path memory contents as well as trellis
state transitions, the Viterbi Algorithm proceeds in the
normal fashion.

3.3   Predictor Parameter Generation and Coding

The tap weights of the decoder transversal filter
of Figure 3.1 are the first 16 terms of the impulse
response of the all pole APC decoder filter as described
by Atal and Schroeder (Reference 8). The poles, a_i, are
the solutions of

$$\sum_{k=1}^{p} a_k \, R(i-k) = R(i) \qquad 1 \le i \le p$$

where R(i) is the measured autocorrelation function of
the speech sample file. We have concentrated on the p = 4
model, since experimental results (Reference 25, page 3-15)
indicate that the residual error signal energy from a four
tap predictor is not much larger than that from a predictor
with 10 taps. The 4 pole APC decoder filter is shown in
Figure 3.3.
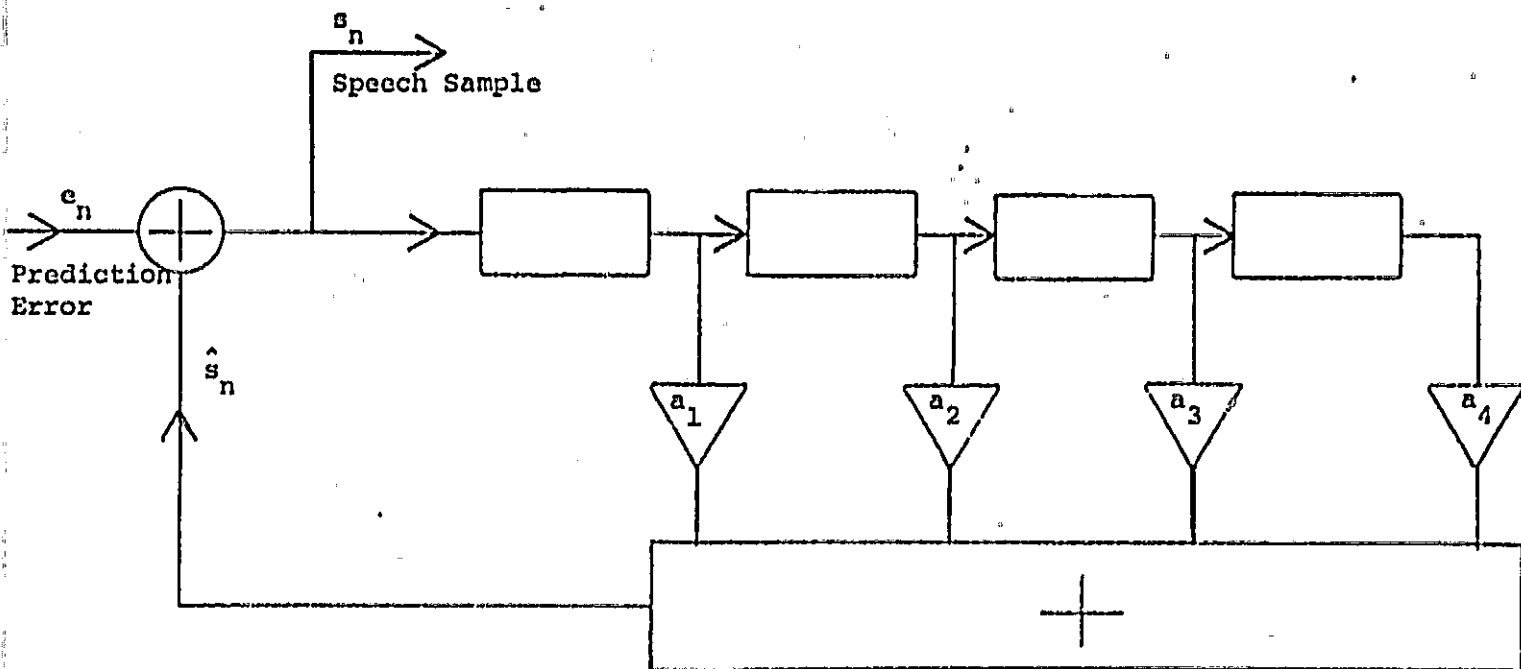
Figure 3.3   4 Pole APC Decoder
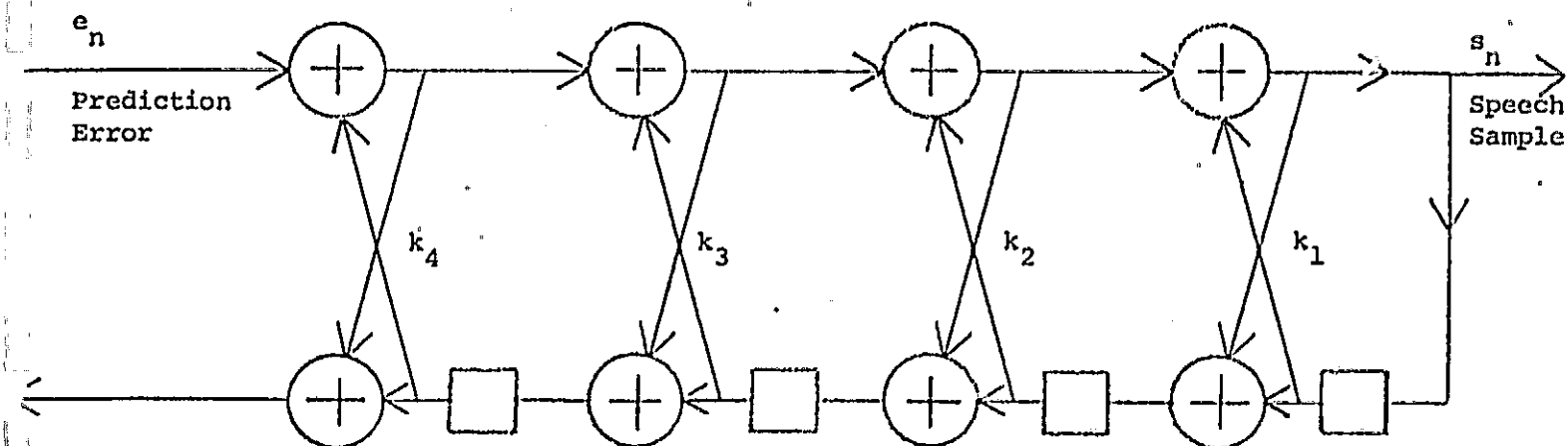
Figure 3.4   4 Stage Lattice Filter Equivalent to the Filter of
Figure 3.3

The recursive filter of Figure 3.3 has an equivalent lattice filter implementation which is illustrated in Figure 3.4. The lattice filter coefficients, k_i, possess many attractive properties (Reference 24), the following being of practical interest:

(1) The filter is guaranteed to be stable for $|k_i| < 1$; consequently saturation guarantees stability;

(2) The k_i may be derived recursively;

(3) The ratio of input to output energy of the i-th state is $[1 - k_i^2]^{-1}$; and

(4) The degradation in performance due to quantization errors is known and consequently optimal quantization procedures are known.

Because of these advantages we transmit the lattice filter coefficients, k_i, and determine the decoder tap weights from the k_i's.

The lattice coefficients are determined according to the algorithm of Figure 3.5, with R_ℓ the average ℓ-delayed speech sample product for the current block.

Logarithmic quantization of the k_i's is accomplished by linear quantization of

$$f(k_i) = g_i \, \log\!\left(\frac{1 + k_i}{1 - k_i}\right)$$
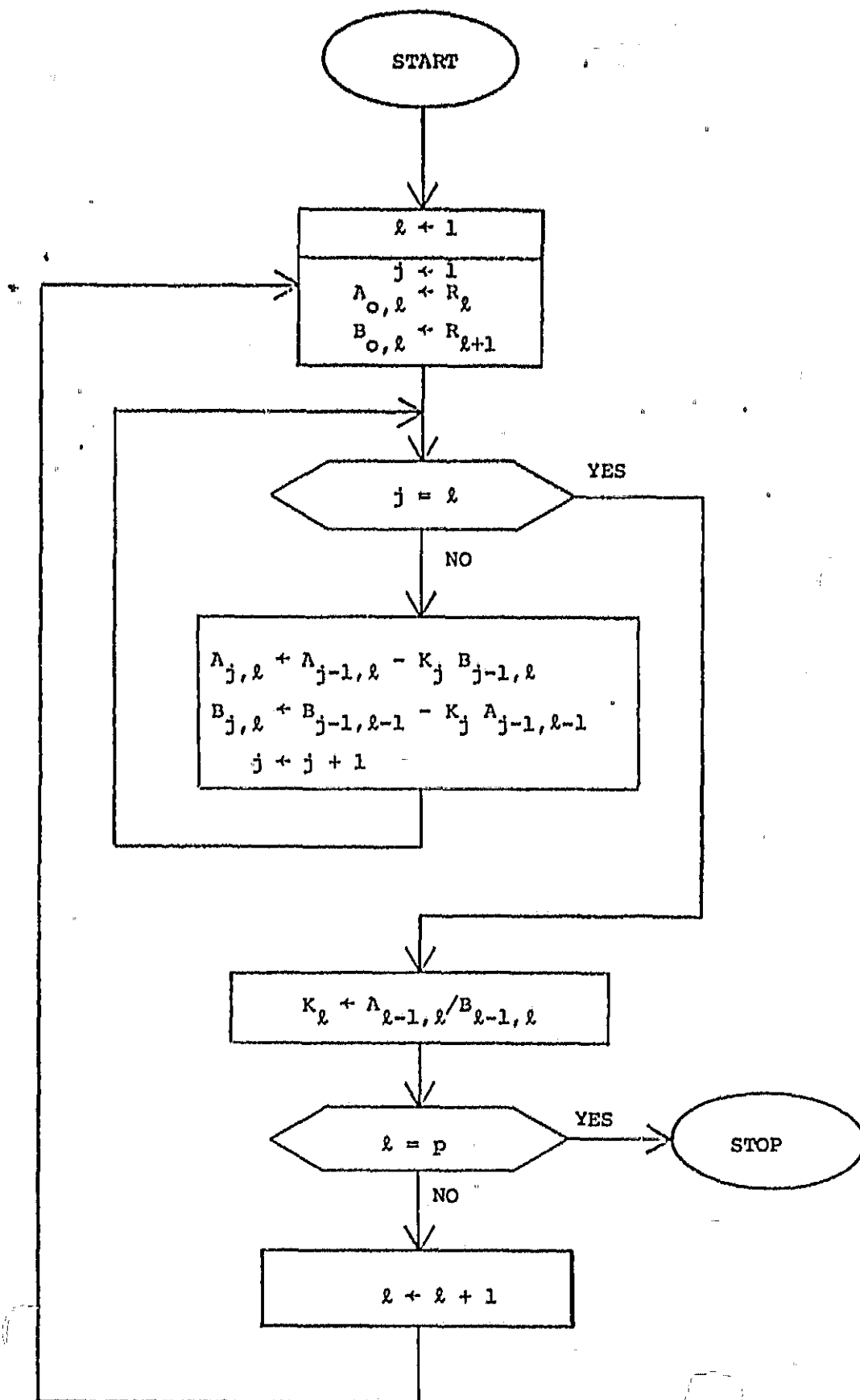
Figure 3.5 Lattice Coefficient Generator Algorithm

where g_i is a scaling factor which depends on the number of bits of quantization. The inverse function is

$$k_i = \frac{10^{f(k_i)/g_i} - 1}{10^{f(k_i)/g_i} + 1}$$

$f(k_i)$ is quantized by taking the integer part of $f(k_i)$. The absolute value of the quantized $f(k_i)$ is not allowed to exceed $2^{(\# \text{ bits of quant.})-1} - 1$, however. This provides an upper limit on $|\hat{k}_i|$ and assures that the impulse response of the lattice filter decays sufficiently fast. For our 8 Kbps compression results we used

$$g_i = \begin{cases} 10 & i = 1, 2 \\ 6 & i = 3, 4 \end{cases}$$

The 8 Kbps compression quantization of the k_i is summarized in Tables 3.1 and 3.2. If the sign of k_1 is negative, $\hat{k}_1$ is set to zero. Our experience and the k_1 histogram of Reference 25 suggest that restricting $\hat{k}_1$ to positive values has a minimal impact on distortion. Sign magnitude representation is used for k_2, k_3 and k_4.

The quantized gain term $\hat{G}$ is obtained by linear quantization of

$$G = \left[R_0 \prod_{i=1}^{4} (1 - \hat{k}_i^2) \right]^{1/2}$$

with $\hat{k}_i$ the quantized representative of k_i.

The lattice impulse response generator functions according to the algorithm of Figure 3.6.

Range of $|k|$		Quantized $|k|$
Lower Limit	Upper Limit	
0	.11461	0
.11462	.22626	.11462
.22627	.33227	.22627
.33228	.43050	.33228
.43051	.51948	.43051
.51949	.59847	.51949
.59848	.66731	.59848
.66732	.72638	.66732
.72639	.77636	.72639
.77637	.81817	.77637
.81818	.85281	.81818
.85282	.88129	.85282
.88130	.90453	.88130
.90454	.92342	.90454
.92343	.93868	.92343
.93869	∞	.93869

Table 3.1 k_i Quantization for $i = 1, 2$

Range of $|k|$		Quantized $|k|$
Lower Limit	Upper Limit	
0	.18955	0
.18956	.36596	.18956
.36597	.51948	.36597
.51949	.64548	.51949
.64549	.74400	.64549
.74401	.81817	.74401
.81818	.87242	.81818
.87243	∞	.87243

Table 3.2 k_i Quantization for i = 3, 4
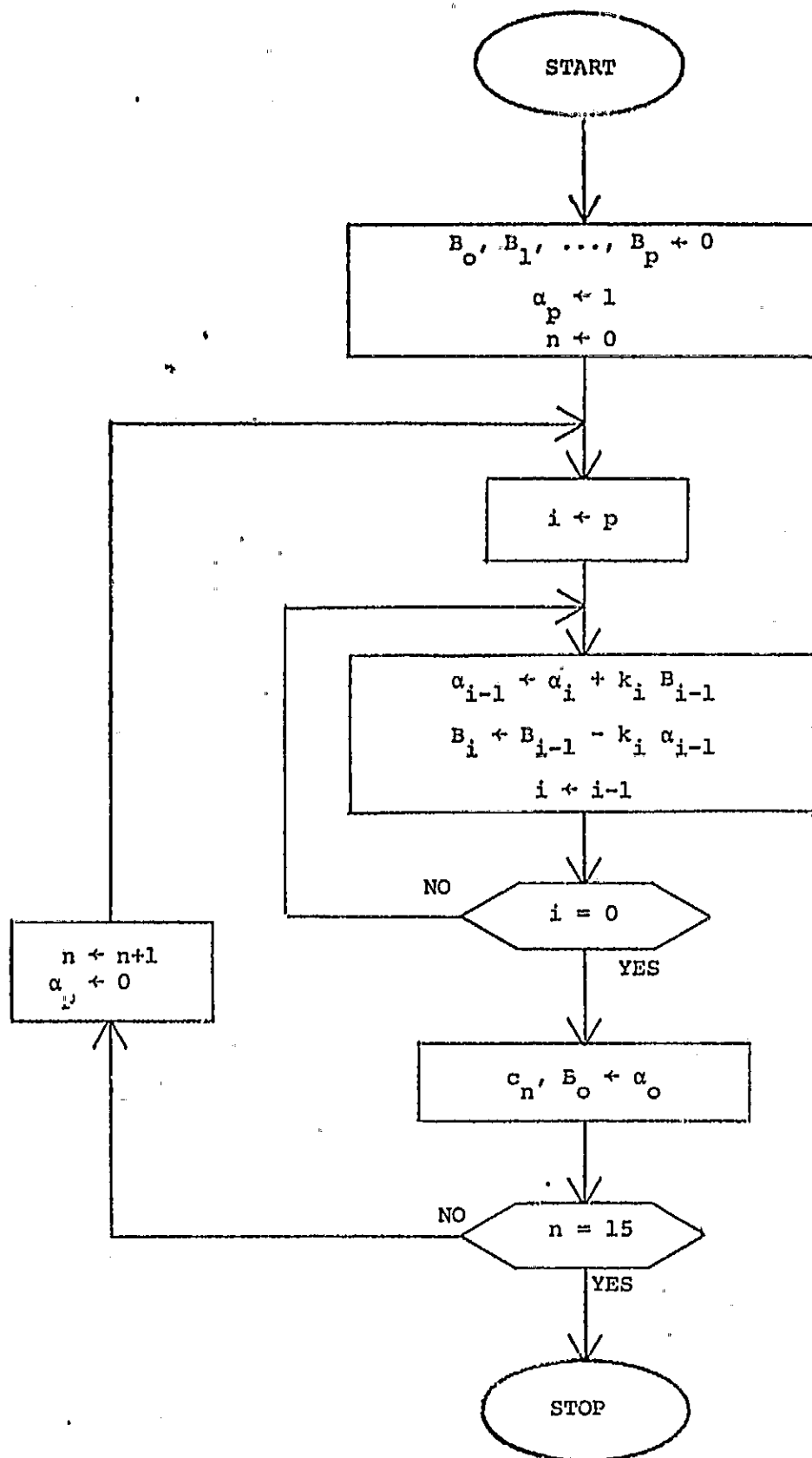
Figure 3.6   Lattice Filter Impulse Response Algorithm

3.4   Pitch Synchronous Preprocessing

In preliminary experiments we found that distortion is considerably reduced if the delayed decision APC techniques described in the previous sections are applied to a "preprocessed" speech file. The preprocessor that we have used is itself an adaptive predictive coder. The prediction is based on a single sample which occured M samples in the past, where M is selected for each block to minimize the prediction error. The "quantization" is performed within the preprocessor loop so that the pre-processor predictions are based on decoded speech sample estimations rather than the original speech samples. This constrains the VA, however, to a delay, D, of less than M. Our 8 Kbps compression results are for D = 32 and $33 \leq M \leq 160$.

Figure 3.7 illustrates the general encoding procedure, while Figure 3.8 describes the decoding operation. The delayed decision APC coding operations described previously are the heart of this technique. The peripheral tasks involve selecting a delay M and a weight b. To minimize the energy of the prediction error,

$$r_n = s_n - b \, s_{n-M}$$

M is selected within the range of allowable M so that
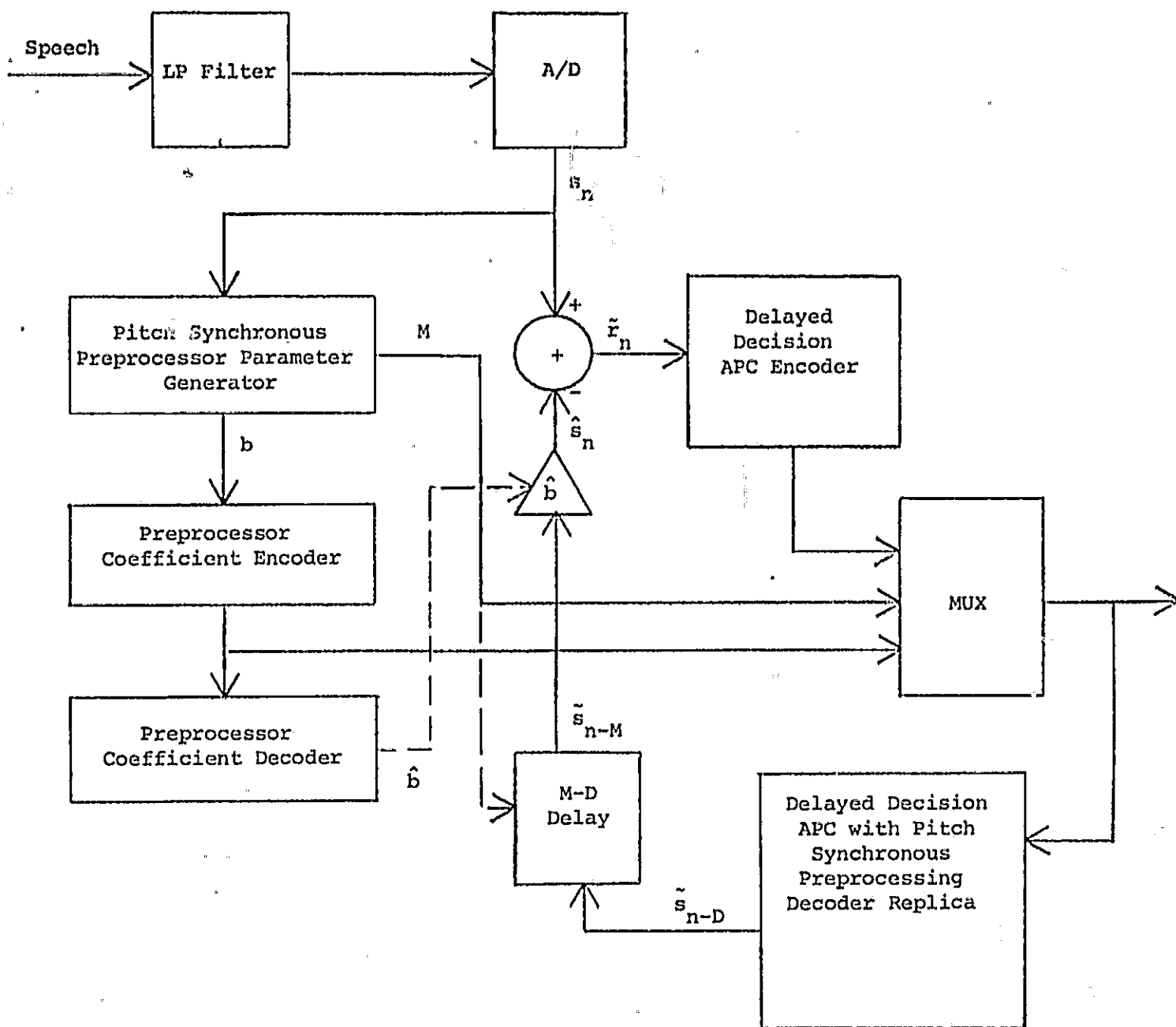
Figure 3.7   Encoder for Delayed Decision APC with Pitch Synchronous
              Preprocessing
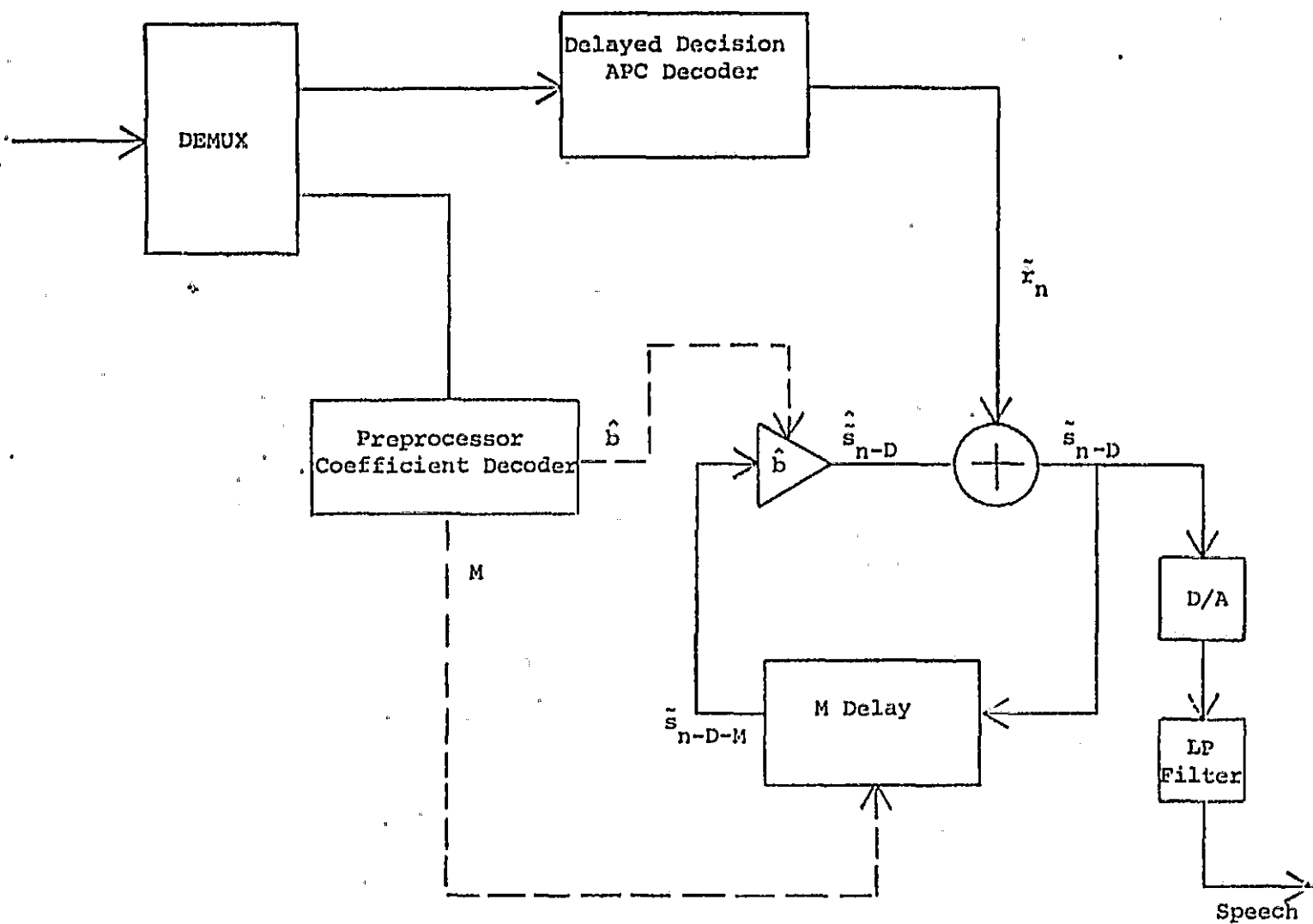
Figure 3.8   Decoder for Delayed Decision APC with Pitch Synchronous
             Preprocessing

$$\Sigma r_n^2 = \Sigma s_n^2 - 2b \; \Sigma \; s_n \; s_{n-m} - b^2 \; \Sigma \; s_{n-m}^2$$

is minimized. Since for a given M the minimizing value
of b is

$$b = (\Sigma \; s_n \; s_{n-m})/(\Sigma \; s_{n-m}^2) \qquad\qquad (*)$$

then

$$\Sigma r_n^2 = \Sigma s_n^2 - (\Sigma \; s_n \; s_{n-m})^2/(\Sigma \; s_{n-m}^2)$$

with the limits of the current block the limits of the
above sums. Equivalently, M can be selected to maximize

$$(\Sigma s_n \; s_{n-M})^2/(\Sigma s_{n-M}^2)$$

Once M is determined, b is calculated from (*). The
additional constraint that M be such that $b \geq 0$ is applied
to the M search algorithm, however.

Quantization and encoding of b is achieved by the
many to one mapping

$$B = \begin{cases} 1 & \text{for } b > 1 \\ \left\lfloor 2^{(\# \text{ of bits of quant})+1} \; [\sin^{-1}(b)]/\pi \right\rfloor & \text{for } 0 \leq b \leq 1 \end{cases}$$

with $\lfloor \cdot \rfloor$ indicating the integer part of the argument.
The decoding operation proceeds according to

$$\hat{b} = \sin\left(\frac{B\pi}{2^{(\# \text{ bits of quant})+1}}\right)$$

Table 3.3 summarizes the resulting quantization cut
points for 3 bit quantization used in our 8 Kbps compression
system.

Range of b		Quantized b
Lower Limit	Upper Limit	
0	.22251	0
.22252	.43387	.22252
.43388	.62348	.43388
.62349	.78182	.62349
.78183	.90098	.78183
.90097	.97492	.90097
.97493	.99999	.97493
1.00000	∞	1.00000

Table 3.3   Pitch Predictor Weight Quantization Levels

4.0   8 Kbps Compression Experiment Results

The recordings accompanying this report are processed samples of the FM news broadcast tape provided to LINKABIT by NASA-JSC on 1 April 1975.

The processing was accomplished with the LINKABIT data compression system configured for voice processing which is illustrated in Figure 4.1. The audio processing equipment includes the following:

> (a)   A high fidelity reel-to-reel tape deck - Tandberg 9000X with frequency response of 30 Hz - 24 Hz at 7.4 inches/sec with 68 dB signal-to-noise ratio.

> (b)   Krohn-Hite variable electronic filters Model 3343 with 48 or 96 dB/octave attenuation slope.

> (c)   Burr-Brown 12 bit A/D converter with sample-and-hold and conversion speed of 30 μ sec and 12 bit D/A converter with conversion speed of 7 μ seconds.

The LINKABIT dedicated in-house digital data compression processor consists of the following central processor and peripheral equipment:

> (a)   A Digital Scientific META-4 computer with 16K words of microsecond core memory, 2K words of 90 nanosecond Read-Only Memory, and 28 general purpose registers. The META-4 is also configured to emulate the IBM 1130 computer, thus utilizing the wide variety of 1130 software.
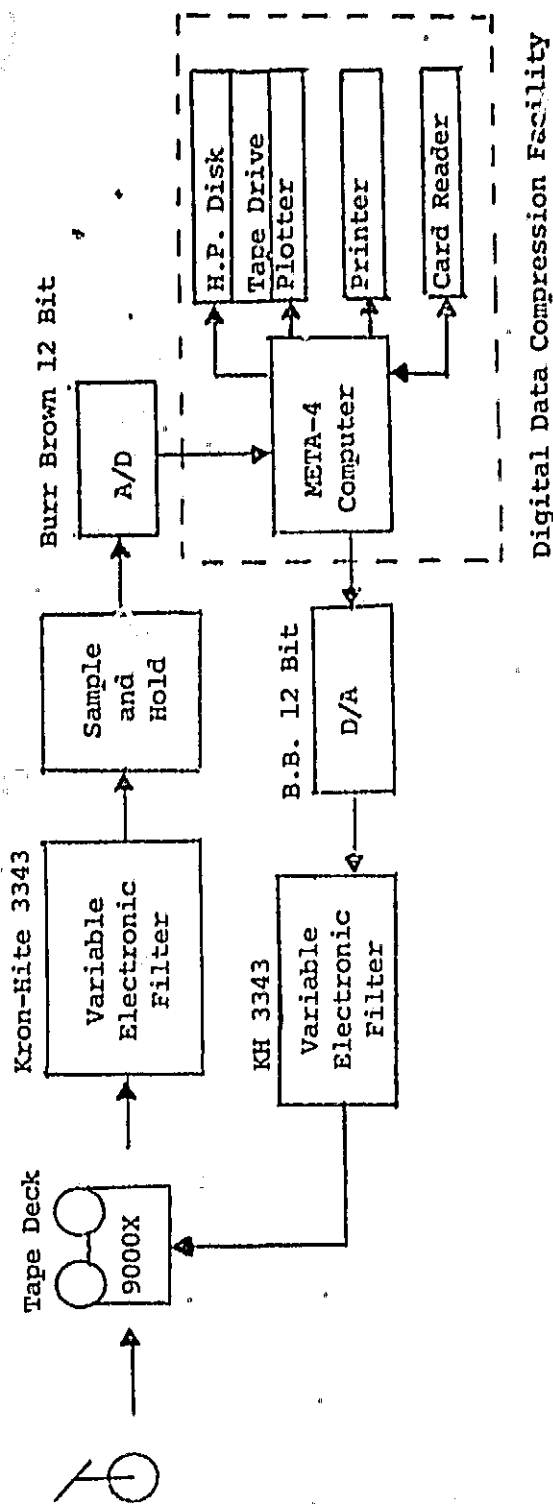
Figure 4.1   Digital Voice Processing Facility

(b) A 1000 card/minute card reader.

(c) A 600 line/minute line printer.

(d) An IBM Selectric keyboard-console printer.

(e) An HP disk memory system with 4 mega-bytes of on-line storage.

(f) A UCC Model 2000, 30-inch high speed digital plotter.

(g) A 25 ips digital tape drive.

Presently our system processes only one file of 12 bit speech sample data at a time. The file size is 51,200 samples. At the sampling rate of 6,660 samples/ second used for these recordings a single file contains 7.68 seconds of digitized uncompressed speech.

The recordings are based on two 7.68 second segments of speech selected at random from the FM broadcast tape. We refer to the 10 recordings as records 1 through 10 with the numbers indicating the relative record locations on the tape. Table 4.1 identifies the 10 records.

The first five records are the results of processing the first FM broadcast speech segment. Record 1 is the result of 79.92 Kbps PCM processing with no compression. Reocrds 2-5 involve 7.992 Kbps APC with pitch synchronous preprocessing. Record 2 processing is conventional APC with immediate decisions. The encoding operation is equivalent to the one diagrammed in Figure 3.8, with a

Record #	Speech Segment #	Transmission Rate (Kbps)	Processing Technique	Channel Bit Error Probability
1	1	79.92	PCM	0
2	1	7.992	APC with Pitch Prediction	0
3	1	7.992	APC with Pitch Prediction and Trellis Search	0
4	1	7.992	APC with Pitch Prediction and Trellis Search	.001
5	1	7.992	APC with Pitch Prediction and Trellis Search	.01
6	2	79.92	PCM	0
7	2	7.992	APC with Pitch Prediction	0
8	2	7.992	APC with Pitch Prediction and Trellis Search	0
9	2	7.992	APC with Pitch Prediction and Trellis Search	.001
10	2	7.992	APC with Pitch Prediction and Trellis Search	.01

Table 4.1   Summary of Recorded Speech Compression Results

one state, immediate decision VA. The decoding procedure
is that of Figure 3.7. Bit allocation to achieve a 7.992
Kbps transmission rate is summarized in Table 4.2 and
applies to all records except the first and sixth which
are uncompressed. Record 3 is the result of APC processing
similar to that used for Record 2, except that a 128
state VA is employed with a delay of 32. For Records 4
and 5, the processing is identical to that of Record 3,
except that Viterbi decoder output noise is added to the
decoder (binary) input. For Record 4, the probability of
a bit error is 10^{-3} while for Record 5 it is 10^{-2}. For
Records 6-10 the same sequence of processing was applied
to the second FM broadcast speech segment.

For conventional APC processing we observe two
classes of distortion. First and possibly least objectionable
is what may be termed granularity noise. Granularity noise
manifests itself in a steady level of "white" background
noise. The second form of distortion we term "loss of
track". Loss of track is similar in nature to "slope
overload" noise (Section 2.1) encountered in delta modulation.
Loss of track in ACP has a much more persistent and severe
effect, however, because of the relatively long memory of
the predictor - as much as 160 samples or 24 msec in our
implementation. Typically, an overload or loss of track

	Bits/Frame	
Lattice Coefficients (k_1, k_2, k_3, k_4)	17	
Pitch Predictor Coefficient (b)	3	
Pitch Period (M)	7	
Gain (g)	5	
Decoder Driving Sequence (q_n)	160	
	192	Total

160 Samples/Frame and 6.66k samples/sec ⇨ 7.992 Kbps

Table 4.2   Bit Allocation for 8 Kbps APC

condition requires a number of sample times equal to several predictor memory lengths to subside. For delta modulation this is only a few samples, but for APC with pitch prediction it is several pitch periods.

Our delayed decision APC procedure using the VA trellis search appears to anticipate potential loss of track problems quite well. On the APC recordings (2 and 6) we observe several occurences of loss of track, that is several short segments of rather severe distortion. These severly distorted segments were very much improved with delayed decision APC. The level of granularity noise also appears to be noticeably reduced with VA APC encoding.

Records 4 and 9 suggest that transmission errors, correlated as though they were produced from the output of Viterbi Decoder, cause an almost imperceptible effect on distortion if the channel error rate is 10^{-3} or less. Records 5 and 10, however, indicate that an error rate of 10^{-2} produces a noticeable increase in distortion, although the speech still appears to be intelligible.

5.0   Estimated Hardware Requirements

The decoding operation for delayed decision APC with pitch synchronous preprocessing (Figure 3.8) is readily accomplished with a microprocessor system requiring only a few chips. The more complicated encoding operation requires some additional high speed hardware for the Viterbi Algorithm and for the pitch synchronous preprocessor parameter calculation.

The 128 state Viterbi Algorithm is similar in structure to the LINKABIT LV7015 Viterbi decoder. The speech compression VA as it is simulated requires 16 bit arithmetic, however, whereas the LV7015 does not require such accuracy. We estimate that the chip count for the VA would be approximately 50 TTL chips.

The determination of the pitch period M requires high speed calculation of the autocorrelation function of the speech sample file. This sum of delayed products operation would require approximately 10 TTL chips.

To summarize, the decoder for delayed decision APC with pitch synchronous preprocessing (Figure 3.8), excluding the low pass filter and digital to analog conversion, can be implemented with a microprocessor system of not more than 10 chips. The encoding operation (Figure 3.7) can be implemented with approximately 70 chips by a microprocessor system with peripheral hardware for the Viterbi Algorithm

and high speed autocorrelator. A large scale integration implementation of the encoder would probably reduce the chip count by a factor of 5 or more.

6.0 Conclusions

It should be emphasized that the recordings provided
with this report do not represent the ultimate in 8 Kbps
delayed decision APC. Since we spent much of our efforts
in searching for a promising compression technique and
developing the necessary software, we had very little
opportunity to optimize bit allocation for the 8 Kbps
delayed decision APC scheme to which we eventually con-
verged. The bit allocation used and summarized in Table
4.2 represents an initial estimate based on the results
of previous APC experimenters and on present constraints
in our software.

It should also be noted that the rate of speech
on the FM broadcast tape provided LINKABIT on 1 April 1975
is considerably more rapid than that on the original four
test tapes provided. By reducing the sampling rate
slightly and thereby being able to shorten the block length
and make the system more adaptive, improved 8 Kbps performance
may be possible.

In conclusion we remark that we are persuaded that
delayed decision adaptive predictive coding is very
competitive with existing voice digitizing techniques.
At an 8 Kbps transmission rate intelligability as well as
speaker recognizability, appear good, even in the

presence of a transmission error rate of 10^{-3}. For a 10^{-2} transmission error rate intelligibility is reduced somewhat, but still may be judged adequate. In addition, a hardware implementation of the system appears to be within the complexity limitations on orbiter using state-of-the-art technology.

References

1. J. L. Flanagan, "Focal Points in Speech Communication Research," <u>IEEE Trans. on Comm. Tech.</u>, Vol. COM-19, pp. 1006-1015, Dec. 1971, Pt. I.

2. J. G. Dunn, "An Experimental 9600-Bit/s Voice Digitizer Employing Adaptive Prediction," <u>IEEE Trans. on Comm. Tech.</u>, Vol. COM-19, pp. 1021-1032, Dec. 1971, Pt. I.

3. C.-L. Song, J. Garodnick, and D. L. Schilling, "A Variable-Step-Size Robust Delta Modulator," <u>IEEE Trans. on Comm. Tech.</u>, Vol. COM-19, pp. 1033-1044, Dec. 1971, Pt. I.

4. S. J. Campanella and G. S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing," <u>IEEE Trans. on Comm. Tech.</u>, Vol. COM-19, pp. 1045-1050, Dec. 1971, Pt. I.

5. N. S. Jayant, "Adaptive Delta Modulation with a One-Word Memory," <u>Bell Syst. Tech. J.</u>, Vol. 49, pp. 321-342, March 1970.

6. N. S. Jayant, "Adaptive Quantization With a One-Word Memory," <u>Bell Syst. Tech. J.</u>, Vol. 52, pp. 1119-1143, Sept. 1973.

7. P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech," <u>Bell Syst. Tech. J.</u>, Vol. 52, pp. 1105-1117, Sept. 1973.

8. B. S. Atal and M. R. Schroeder, "Adaptive Prediction Coding of Speech Signals," <u>Bell Syst. Tech. J.</u>, Vol. 49, pp. 1973-1986, Oct. 1970.

9. L. J. Greenstein, "Slope Overload Noise in Linear Delta Modulators with Gaussian Inputs," <u>Bell Syst. Tech. J.</u>, Vol. 52, pp. 387-421, March 1973.

10. R. A. McDonald, "Signal-to-Noise Performance and Idle Channel Performance of Differential Pulse Code Modulation Systems with Particular Applications to Voice Signals," <u>Bell Syst. Tech. J.</u>, Vol. 45, pp. 1123-1151, Sept. 1966.

11. L. R. Rabiner, L. B. Jackson, R. W. Schafer, and C. H. Coker, "A Hardware Realization of a Digital Formant Speech Synthesizer," <u>IEEE Trans. on Comm. Tech.</u>, Vol. COM-19, pp. 1016-1020, Dec. 1971, Pt. I.

12. J. B. Anderson and J. B. Bodie, "Encoding of Speech by Tree Coding with a Fidelity Criterion," presented at IEEE Symposium on Information Theory, Notre Dame, Indiana, Oct. 1974 and to appear in IEEE Trans. on Information Theory, 1975.

13. J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Information Theory, Vol. IT-6, pp.7-12, March 1960.

14. T. J. Goblick and J. L. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice," IEEE Trans. on Information Theory, Vol. IT-13, pp. 323-326, May 1967.

15. N. S. Jayant and A. E. Rosenberg, "The Preference of Slope Overload to Granularity in the Delta Modulation of Speech," Bell Syst. Tech. J., Vol. 50, pp. 3117-3125, Dec. 1971.

16. J. E. Abate, "Linear and Adaptive Delta Modulation," Proc. IEEE, Vol. 55, pp. 298-308, March 1967.

17. J. A. Greefkes and F. de Jager, "Continuous Delta Modulation," Philips Res. Rep., Vol. 23, pp. 233-246, April 1968.

18. N. S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers," Proc. IEEE, Vol. 62 pp. 611-632, May 1974.

19. T. Berger, Rate Distortion Theory, A Mathematical Basis for Data Compression, Prentice Hall, New York, 1971.

20. A. J. Viterbi and J. K. Omura, "Trellis Encoding of Memoryless Discrete-Time Sources with a Fidelity Criterion," IEEE Trans. on Information Theory, Vol. IT-20, pp. 325-332, May 1974.

21. R. J. Dick, T. Berger, and F. Jelinek, "Tree Encoding of Gaussian Sources," IEEE Trans. on Information Theory, Vol. IT-20, pp. 332-336, May 1974.

22. J. P. Odenwalder, A. J. Viterbi, I. M. Jacobs, and J. A. Heller, "Study of Information Transfer Optimization for Communication Satellites," Final Report on Contract NAS2-6810, LINKABIT Corporation, San Diego, California, January 1973.

23. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Information Theory, Vol. IT-13, pp. 260-269, April 1967.

24. J. Makhoul, "Linear Prediction: A Tutorial Review,", *Proceedings of the IEEE*, April 1975.

25. D. L. Chaffee, "Final Report for Phases I, II and III of the Longbrake Program," 20 November 1974.