Joint optimization of adaptive predictive coding.

Seema A. Ranka
JOINT OPTIMIZATION OF ADAPTIVE PREDICTIVE CODING

by

Seema A. Ranke

A Thesis

Presented to the Graduate Committee

of Lehigh University

in candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1983
This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

September 14, 1983
(date)

Professor in Charge

Chairman of the Department
ACKNOWLEDGEMENTS

I wish to express my deep gratitude to my thesis adviser, Professor Bruce D. Fritchman, for the guidance and the encouragement he gave me to undertake this project.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>v-vi</th>
</tr>
</thead>
</table>

## CHAPTER 1
INTRODUCTION

## CHAPTER 2
INFORMATION THEORETIC CONCEPTS OF COMPRESSION AND ADAPTIVE PREDICTIVE CODING.

- 2.1 Introduction
- 2.2 Information Theoretic Concepts of compression
- 2.3 Adaptive Predictive Coding

## CHAPTER 3
JOINT OPTIMIZATION OF APC

- 3.1 Introduction
- 3.2 Meaning and Necessity of Joint Optimization
- 3.3 Algorithms Development in Optimizing APC
  - 3.3.1 First Algorithm
  - 3.3.2 Second Algorithm
  - 3.3.3 Third Algorithm
  - 3.3.4 Fourth Algorithm

## CHAPTER 4
RESULTS AND CONCLUSIONS

- 4.1 Introduction
- 4.2 First Algorithm
- 4.3 Second Algorithm
- 4.4 Third Algorithm
- 4.5 Fourth Algorithm

## REFERENCES

- 43-44

## VITA

- 45
LIST OF FIGURES

Figure 3.1  General differential scheme  
(a) Coder (b) Decoder.

Figure 4.1  First Algorithm - Plot of the residual distribution  
entropy Versus number of taps in the filter.

Figure 4.2  First Algorithm - Plot of the first predictor  
coefficient entropy multiplied by number of taps  
in the filter Versus number of taps in the filter.

Figure 4.3  First Algorithm - Plot of number of bits required  
to represent each sample of the speech Versus  
number of taps in the filter.

Figure 4.4  Second Algorithm - Plot of the residual distribu-  
tion entropy Versus number of iterations.

Figure 4.5  Second Algorithm - Plot of the first predictor  
coefficient entropy multiplied by number of  
taps in the filter Versus number of iterations.

Figure 4.6  Second Algorithm - Plot of number of bits required  
to represent each sample of the speech Versus  
number of iterations.

Figure 4.7  Long - time - average autocorrelation function of  
speech signal.

Figure 4.8  Second Algorithm - Long - time - average autocorre-  
lation function of difference signal.

Figure 4.9  Second Algorithm - Long - time - average autocorre-  
lation function of difference signal.

Figure 4.10  Third Algorithm - Plots of the residual distribu-  
tion entropy Versus threshold difference signal.

Figure 4.11  Third Algorithm - Plots of the first predictor  
coefficient entropy multiplied by number of taps  
Versus threshold difference signal

Figure 4.12  Third Algorithm - Plots of number of times differ-  
ence signal was set to zero Versus threshold differ-  
signal.
Figure 4.13  Plots of number of bits required to represent each sample of the speech Versus threshold difference signal.

Figure 4.14  Fourth Algorithm - Plot of the residual distribution entropy Versus threshold difference signal.

Figure 4.15  Fourth Algorithm - Plot of number of bits required to represent each sample Versus threshold difference signal.
ABSTRACT

This work investigates the behavior of the residual signal in adaptive predictive coding (APC), which is one of the waveform encoding techniques. It also studies the variation of the predictor coefficients as a function of the structure of the predictor.

Using continuous estimation of the predicted sample values of the speech signal, it is possible to force the residual signal to zero. Using the entropy of the continuously estimated prediction coefficients and of the residual, the structure of the adaptive algorithm is investigated. Results obtained using computer simulation show that the continuous estimation of predictor coefficients does result in a very low entropy for the residual distribution but predictor coefficients change dramatically from sample to sample, resulting in a very high entropy for the predictor coefficient distribution. Hence various different algorithms have been investigated to jointly optimize the APC. Prediction coefficients are changed only when the residual signal exceeds a particular value. Two different algorithms, which use this criterion to estimate the prediction coefficients are investigated. The results show that this represents a better approach towards reducing the number of bits/sample for the representation of the speech signal. The prediction coefficients do not change too much from sample to sample. It also keeps the residual distribution entropy low.
CHAPTER 1

INTRODUCTION

This thesis develops some algorithms in an attempt to optimize adaptive predictive coding (APC). Schemes devised to extract only significant information from the output of a source and to eliminate redundant or irrelevant information are called "data compression algorithms". Many different methods and compression algorithms have been developed with varying degrees of success and complexity over the past decade. But APC is especially attractive since it is most easily implemented and has the potential for considerable improvement.

In predictive coding, redundancy is reduced by subtracting from the signal that part which can be predicted from its past. For many signals and especially speech, the first order entropy of the difference signal is much smaller than the first order entropy of the original signal. Therefore the difference signal is better suited to memoryless encoding. It is difficult to assess exactly how much compression is theoretically possible for a given level of distortion (1), but Information Theory and Rate-Distortion Theory (2) are helpful in providing some knowledge of expected performance. Measured compression limits exceed the theoretical limits (3) because the definite theoretical bounds can not be
obtained unless the statistics of the source are completely characterized. Unfortunately that has not been possible to date for speech signal because of its time varying characteristics.

In spite of this successful efforts have been made towards reducing the number of bits required to represent speech signals. In a recent review (1), Haskell has made a good attempt to summarize as to where we are today in audio (and video) coding.

Methods developed to represent speech signals can be divided into two general categories (4).

1. Waveform Encoding.
2. Source encoding.

Waveform coding systems attempt to digitally communicate a good reproduction of the actual waveform, whereas source coding systems attempt to estimate and communicate a linear model of speech production process rather than the specific waveform. i.e. the primary emphasis is on preserving intelligibility rather than on preserving the naturalness of reproduced sounds.

The best waveform encoders are currently capable of producing good quality speech at about 1 bit/sample. For example adaptive transform coding (5), sub-band coding (6-10) and tree coding (11-16) along with APC (17-22) are capable of this level of
performance. Also the best source encoders are capable of producing speech at about 1/8 bit/sample with very high intelligibility. But such signals are very highly compressed and hence are very unnatural sounding. It is quite unlikely that natural sounding speech can be obtained at 1/8 bit/sample, but there is considerable room for improvement between 1 bit/sample and 1/8 bit/sample.

Chapter 2 discusses some Information theoretic concepts of compression. It also gives a basic background about APC and discussion of work done by other researchers. Chapter 3 is the discussion of work carried out in this study. Discussion of the new algorithms developed and ideas behind them are included. Chapter 4 summarizes the results and conclusions drawn from this study.
CHAPTER 2

INFORMATION THEORETIC CONCEPTS OF COMPRESSION

AND

ADAPTIVE PREDICTIVE CODING

2.1 Introduction:

In this chapter the basic concepts concerning the information content of the speech signal are reviewed. This is followed by the discussion of APC. It is realized that the entropy of the distribution is a more appropriate cost function to compress the speech, rather than the minimum variance calculations.

2.2 Information Theoretic Concepts of Compression:

Digital encoding of speech has been a topic of long standing interest for purposes of digital communications and digital storage. The efficiency of such encoding techniques strongly depends on the degree to which the speech can be compressed without impairing the quality of the decoded signal. Typically signals such as speech have a high degree of redundancy that can be exploited to reduce the bit rate without introducing perceptible degradations in the quality of the decoded signal.

In order to take advantages of these properties, a considerable
amount of signal processing is necessary. The natural occurring speech signal is time continuous and bandlimited, and hence can be sampled at a rate equal to or greater than the Nyquist rate, without loss of information. However the process of quantizing these samples introduces some distortion. Ideally, to represent the signal digitally without distortion would require an infinite data set. A finite data representation is possible if some distortion is permitted. Thus the amount of data required depends on the amount of distortion permitted. There are some other factors which influence the amount of data required. For example, by appropriately coding the quantization levels, it is generally possible to significantly reduce the amount of data required for a given level of distortion.

A given set of sample values along with a given level of distortion can be viewed as having a specific information content. What is not information is the redundancy and as said earlier, this redundancy is to be removed to compress the speech.

Redundancy can be thought of as arising in two ways.

1. If each allowable sample value is assigned a symbol and symbols are not equilikely to occur. And/or

2. If there is statistical dependence between symbols from sample to sample.

The entropy function can be used to precisely express how much
information is present. Predictive coding is a well known method of reducing redundancy. Also the appropriate coding can be used to reduce the size of the data set required to represent a sequence of speech samples. A well known method of coding, called Huffman coding, optimally removes the redundancy and preserves the information. But unfortunately, unless it is possible to reduce the number of quantization levels and/or the extent of statistical dependence, practical implementation of Huffman encoding becomes impractical. Huffman coding might still be practical if the redundancy arising from statistical dependence were removed by some method (for example APC), or if the number of required quantization levels could be partially reduced along with some statistical dependence. Considering all these things, APC is one of the most attractive methods. Also as said earlier it is quite easily implemented and has the potential for considerable improvement. The next section gives the basic background for APC.

2.3 Adaptive Predictive Coding:

APC is one of the most efficient methods of compressing speech (17). In this method both the transmitter and the receiver estimate the signal's current value by linear prediction on the previously transmitted signal. The difference between the estimate and the true value of the signal is quantized, and then transmitted to the receiver. At the receiver the difference signal is added to
the predicted signal to reproduce the original speech signal. If the prediction is good then the difference sequence should in some sense be smaller than the original sequence. Consequently by quantizing the residual samples rather than the original samples, the same level of distortion can be achieved with fewer quantization levels and hence less data.

The performance of this procedure depends on the ability to predict accurately. Because of the nonstationary nature of the speech signal, prediction can be good and accurate only if the predictor adapts to changes in the signal. This requires the adaption of predictor coefficients with time. Hence data will be required to represent both the predictor coefficients and the difference samples.

Different methods which are used currently to change the predictor coefficients with time, and then quantize the difference samples and predictor coefficients do result in considerable reduction of the bit rate compared to Pulse Code Modulation (PCM). But these methods are based on some dubious assumptions. For example in the case of APC, the signal is assumed to be stationary over a 5 to 10 msec interval. This assumption is based on the physical models of the vocal tract (23). Hence predictor coefficients are assumed to remain constant over this interval.
Also the criterion commonly used to optimize the APC is minimization of the variance of the difference sequence. But from the discussion in the previous section it is clear that it is the entropy of the distribution which determines the amount of data required to represent the difference sequence, and minimizing the variance of this difference is not the best approach.

Hence it is clear that for a significant improvement in APC, an important step is to give up minimum-variance calculation of the prediction coefficients over a fixed data block, and try to optimize the problem considering entropy of the distribution a more appropriate cost function. Also it is realized that some method of continuously estimating the predictor coefficients is required which does not limit the structure of the predictor. But now if predictor coefficients are updated with each new sample, then though it is possible to force the residual signal to zero, predictor coefficients might change dramatically from sample to sample, thereby greatly increasing the data required to represent them. Hence there may be a net increases in the data required to represent the signal. On the other hand if predictor coefficients are set to zero then the residual signal will be same as the original sequence and no reduction in the data representation will be possible. Hence the optimum solution falls between these two extremes.
The next chapter describes the research done in an attempt to find a good solution, falling between the extremes of too rapidly changing predictor coefficients produced by forcing the difference signal to a very small value, and no prediction at all.
CHAPTER 3

JOINT OPTIMIZATION OF APC

3.1 Introduction:

This chapter discusses the joint optimization of APC. This is followed by a discussion of different algorithms developed in this research to optimize APC, and the ideas behind them. The significant difference between this research and the research done by other investigators is the way the prediction coefficients are adapted.

3.2 Meaning And Necessity Of Joint Optimization

As discussed in chapter 2, an important step in optimizing APC is to give up the minimum-variance calculation of the predictor coefficients over a fixed data block. Instead, if the predictor coefficients are updated with each new sample, then it is possible to force the residual signal to zero. This will result in reducing the data required to represent the residual sequence. But now it is possible that the predictor coefficients will change dramatically from sample to sample. Hence we might require greatly increased data to represent them. And therefore the net result may be an increase in the data required to represent the voice signal.

On the other hand if the predictor coefficients are set to zero,
then the data required to represent them will be zero. But now the residual will be the same as the original sequence and hence there will be no reduction in the data. Therefore the optimum solution falls between these two extremes. It is therefore realized that data representation of both, the predictor coefficients and residual signal should be investigated jointly.

3.3 Algorithms Development In Optimizing APC:

Four different algorithms have been investigated, in an attempt to optimize APC. The First and Second algorithms, which estimate the prediction coefficients continuously, result in quite a high number of bits/sample of the speech signal. Hence a different approach is taken in the Third and Fourth algorithms. Here prediction coefficients are changed only when the difference signal exceeds a particular value.

3.3.1 First Algorithm:

The first algorithm that was developed centered around the idea of continuously estimating the predictor coefficients. This is important in studying the joint variation of the predictor coefficients and the residual signal.

The predictor coefficients were changed with each sample, using the gradient method of computing predictor coefficients (24). An all pole filter was used allowing the number of coefficients to
Figure (3.1) (a) and (b) show a general differential scheme. The variance of the prediction error in figure 3.1 (a) is,

$$\sigma_d^2 = E[d^2(n)] = E[\{x(n) - \hat{x}(n)\}^2] \quad (3.1)$$

If P is a linear predictor then x(n) is a linear combination of past quantized values.

$$\hat{x}(n) = \sum_{k=1}^{P} a_k x(n-k) \quad (3.2)$$

Where $a_k$'s are the predictor coefficients. From equations (3.1) and (3.2),

$$\sigma_d^2 = E[\{x(n) - \sum_{k=1}^{P} a_k x(n-k)\}^2] \quad (3.3)$$

In order to choose a set of predictor coefficients $[a_k]$, 1 ≤ K ≤ P, that minimizes $\sigma_d^2$, we must differentiate $\sigma_d^2$, with respect to each parameter and set the derivative equal to zero.

$$\frac{\delta \sigma_d^2}{\delta a_k} = E[2 \{x(n) - \hat{x}(n)\} \{x(n-k)\}] = 0$$

or, $\frac{\delta \sigma_d^2}{\delta a_k} = -2 E[d(n)x(n-k)] = 0, \ 1 \leq K \leq P \quad (3.4)$

The conventional approach is to solve this set of equations to obtain the predictor coefficients. The alternative approach (24-25) is the LMS - Gradient method. It solves equations (3.4) indirectly by operating directly on the data.

It is clear from equation (3.4), that if the predictor
Fig. 3.1 - General differential scheme. (a) Coder (b) Decoder.
coefficients are to be such as to minimize \( \sigma_d^2 \) then the difference signal should be uncorrelated with the past values of the predictor input \( x(n-k), 1 < k < P \). This can be achieved by a subtraction of some auxiliary input from the input signal, after the auxiliary input is adjusted in amplitude to maximize its correlation with the main input. Thus this LMS - Gradient method does not require any intermediate computations. This continuous LMS prediction algorithm can be written as,

\[
a_k(n+1) = a_k(n) + C d(n) x(n-k)
\]  

where \( a_k \) is the kth predictor coefficient, \( x(n) \) is the input signal, and \( C \) is the gradient coefficient \(<\ 1\). \( C \) should be chosen to assure the stability of the gradient convergence process.

Using this algorithm, the joint variation of the residual signal and the predictor coefficients was studied as a function of the number of taps in the predictor. Gradient coefficient \( C \) was chosen as 0.205.

3.3.2 Second Algorithm:

The second algorithm that was developed was an extension of the first one. In this, the estimation of prediction coefficients and calculation of predicted sample value for a particular sample was
done repeatedly until a good predicted value was obtained. In other words in-place iterations were increased from 1 (in algorithm 1) to a higher value. The gradient method was used to estimate the predictor coefficients continuously.

Again using this algorithm, the joint variation of the residual signal and predictor coefficients was studied as a function of number of iterations, gradient coefficients and number of taps in the predictor. Also the long-time-average autocorrelation function was estimated for the voice signal $R_X$, and for the residual signal $R_D$ using equations (3.6) and (3.7) respectively.

$$R_X = \frac{1}{L} \left[ \sum_{n=0}^{L-1-m} x(n) x(n+m) \right], \quad 0 \leq |M| \leq L-1$$  \hspace{1cm} (3.6)

$$R_D = \frac{1}{L} \left[ \sum_{n=0}^{L-1-m} d(n) d(n+m) \right], \quad 0 \leq |M| \leq L-1$$ \hspace{1cm} (3.7)

Where $L$ is the block length.

3.3.3 Third Algorithm:

Results of the first and second algorithms (discussed in details in chapter 4) indicated that continuous estimation of predictor coefficients does result in a very low entropy of residual distribution. And hence very little data is required to represent the residual sequence. But it also results in a very high entropy of the predictor coefficients distribution. Therefore we require a large amount of data to represent them. And the net result is an
increase in the data required to represent the speech signal. Hence it was realized that there is need for some other method of estimating these coefficients, which would result in lower entropy of the predictor coefficient distribution, keeping the residual distribution entropy low.

In this algorithm, prediction coefficients were not changed for each sample. The absolute value of the difference signal was kept within some threshold value. Whenever the difference signal exceeded the threshold value, it was brought down to zero in one step. Thus predictor coefficients changed only when the difference signal exceeded the threshold.

The algorithm development dealt primarily with how to bring the difference signal value to zero in one step. It is described as follows:

\[ d_0(n) = x(n) - \hat{x}(n) \]  
\[ d_0(n) = x(n) - \sum_{i=1}^{\infty} a_0(i) x(n-i) \]  

Where subscript \( '0' \) indicates 0th iteration, \( d(n) \) is the nth difference sample, \( a(i) \) is the ith predictor coefficient, and \( x(n) \) is the nth input sample.

After one iteration the ith coefficient, \( a_0(i) \) changes by say
\[ \Delta a_0(i). \] When the gradient method of computing predictor coefficients is used,

\[ \Delta a_0(i) = c_i d_0(n) x(n-i) \] (3.10)

Hence after one iteration ith predictor coefficient,

\[ a_1(i) = a_0(i) + \Delta a_0(i) \] (3.11)

From equation (3.10) and (3.11),

\[ a_1(i) = a_0(i) + c_i d_0(n) x(n-i) \] (3.12)

Also,

\[ d_1(n) = x(n) - \sum_{i=1}^{P} a_1(i) x(n-i) \] (3.13)

From equation (3.12) and (3.13),

\[ d_1(n) = x(n) - \sum_{i=1}^{P} a_0(i) x(n-i) - \sum_{i=1}^{P} c_i d_0(n) x^2(n-i) \] (3.14)

From equation (3.9) and (3.14),

\[ d_1(n) = d_0(n) - \sum_{i=1}^{P} c_i d_0(n) x^2(n-i) \] (3.15)

From equation (3.15), it is clear that after one iteration the difference signal has been reduced by the amount \( c_i d_0(n) x^2(n-i) \).

\[ d_0(n) - d_1(n) = \sum_{i=1}^{P} c_i d_0(n) x^2(n-i) \] (3.16)

From equation (3.16),

\[ \frac{[d_0(n) - d_1(n)]}{d_0(n)} = \sum_{i=1}^{P} c_i x^2(n-i) \] (3.17)

If \( c_i \gg 0 \) then the right hand side of equation (3.17) \( \gg 0 \). Also, the
right hand side of equation (3.17) should be $\leq 1$. The reasons for this can be explained as follows:

(a) If $\sum_{i=1}^{P} C_i x^{2(n-i)} > 1$,

Then from equation (3.17),

$$d_0(n) - d_1(n) > d_0(n)$$

Therefore the signs of $d_0(n)$ and $d_1(n)$ will be different. Hence choosing,

$$C_i x^{2(n-i)} < 1.$$  

(b) If $\sum_{i=1}^{P} C_i x^{2(n-i)} > 1$,

Then from equation (3.17),

$$d_0(n) - d_1(n) > d_0(n)$$

or, $$1 - [d_1(n)/d_0(n)] > 1.$$  

or, $$|d_1(n)/d_0(n)| > 0$$

or, $$d_1(n) > d_0(n)$$

Which is not desirable. Hence choosing again,

$$\sum_{i=1}^{P} C_i x^{2(n-i)} \leq 1$$

From equation (3.16) after kth iteration,

$$d_{k+1}(n) - d_k(n) = - \sum_{i=1}^{P} C_i d_k(n) x^{2(n-i)}$$  \hspace{1cm} (3.18)

But for each in-place iteration,
\[ \sum_{i=1}^{P} C_i x^{2(n-i)} = M, \quad M < 1 \]  

(3.19)

where \( M \) is a constant. Therefore from equation (3.18) and (3.19),

\[
d_{k+1}(n) - d_k(n) = - M d_k(n) \quad (3.20)
\]

or, \( d_{k+1}(n) - (1 - M) d_k(n) = 0 \)  

(3.21)

let, \( 1 - M = \Delta M, \quad 0 \leq \Delta M < 1 \)  

(3.22)

Therefore from equation (3.21) and (3.22),

\[
d_k(n) = (\Delta M)^k d_Q(n) \quad (3.23)
\]

From equation (3.23) we can conclude that a sufficient number of iterations will drive the iterated error to zero.

\[
\lim_{k \to \infty} d_k(n) = 0 \quad (3.24)
\]

The larger the value of \( M \), the faster is the convergence. Moreover it makes no difference whether the \( C_i \)'s are different or the same. If for all \( i, \quad C_i = C \) then,

\[ \sum_{i=1}^{P} C_i x^{2(n-i)} = C \sum_{i=1}^{P} x^{2(n-i)} = M \]  

(3.25)

If \( M = 1 \), then from equation (3.22), \( \Delta M = 0 \). And from equation (3.25),

\[ C = 1 / [ \sum x^{2(n-i)} ] \]  

(3.26)

Choosing the value of \( C \) given by equation (3.26) will cause the convergence in one step.
From equation (3.12) and (3.26) the value of ith predictor coefficient \( a_i(i) \), which will force the difference signal to zero in one step is,

\[
a_i(i) = a_0(i) + \left[ d_0(n) x(n-i) / \sum_{i=1}^{P} x^2(n-i) \right]
\]

From equation (3.9) and (3.27),

\[
a_i(i) = a_0(i) + [x(n) - \sum_{k=1}^{P} a_0(k) x(n-k)] x(n-i) / \sum_{j=1}^{P} x^2(n-j)
\]

Now let \( a_0(i) = 0 \) for all \( i \neq 0 \), then

\[
a_i(i) = \left[ x(n) x(n-i) \right] / \sum_{j=1}^{P} x^2(n-j)
\]

where \( i = 1, 2, \ldots, p \). Values of predictor coefficients given by equation (3.29) force the difference signal to zero in one step.

Using these ideas, again the residual distribution entropy and predictor distribution entropy were estimated as functions of the threshold difference signal. And the process was used to reduce, bits per sample being the criterion for the optimization. The long-time average autocorrelation function was estimated for voice signal \( R_X \) and for difference signal \( R_D \), using equations (3.6) and (3.7) respectively. Also the number of taps in the predictor were changed, and all the calculations repeated.

3.3.4 Fourth Algorithm:

This algorithm differs very little from the Third algorithm. The
idea behind it is the same, i.e. to keep the predictor coefficients from changing too much from sample to sample (as was observed in the first and Second algorithms), but at the same time keeping the residual distribution entropy low.

Here again as in the third algorithm, the absolute value of the difference signal is kept within some threshold value. But whenever it exceeds the threshold value, it was not brought down to zero in one step (as in the Third algorithm), instead the number of in-place iterations was increased from one to a higher value. It was iterated until the difference value becomes less than or equal to the threshold value. Here again the gradient method of estimating predictor coefficients was used. The residual distribution entropy and predictor distribution entropy were estimated as functions of threshold difference signal. Long-time average autocorrelation functions were estimated for voice signal $R_X$ and for the difference signal $R_D$ using equations (3.6) and (3.7) respectively. Also the number of taps in the predictor were changed and all the calculations repeated.

Results and conclusions drawn from all this work are discussed in the next chapter.
CHAPTER 4

RESULTS AND CONCLUSIONS

4.1 Introduction:

In this chapter, the results of the various computer simulations, regarding the four different algorithms discussed in chapter 3 are discussed. Various plots are drawn for each algorithm and conclusions drawn from them.

4.2 First Algorithm:

Fig. 4.1 is the plot of residual distribution entropy versus number of taps in the filter. The gradient coefficient was chosen as 0.205 and residual distribution entropy was calculated with number of taps L as 2, 4, 6 and 8. Residual distribution entropy as low as 1.76 was obtained.

But as can be seen from fig. 4.2, predictor coefficients varied too much from sample to sample, and resulted in high predictor coefficient distribution entropies. As a result, data required to represent the predictor coefficients became very high compared to the data required to represent the speech samples.

Fig. 4.3 is a plot of bits/sample versus number of taps in the filter. In case of minimum number of taps (L = 2), the fewest bits/sample were obtained, even though the residual distribution
Fig. 4.1- First Algorithm - Plot of the residual distribution entropy versus number of taps in the filter.
the fitted
entropy versus number of taps in
Fig. 4.1 - Fitted algorithm - plot of the residual distribution

Fig. 4.0 - No. of taps versus entropy

NO. OF TAPS

0 2 4 6 8 10

RESIDUAL DISTRIBUTION ENTROPY

2.0 2.3 2.5 2.7 1.9 1.7

C = 0.205

NO. OF QUANTIZATION BITS = 6
Fig. 4.2 - First Algorithm - Plot of the first predictor coefficient distribution entropy multiplied by number of taps in the filter versus number of taps in the filter.
Fig. 4.2 - First Algorithm - plot of the first predictor coefficient distribution entropy multiplied by number of taps in the filter versus number of taps.
Fig. 4.3 - First Algorithm - Plot of number of bits required to represent each sample of the speech versus number of taps in the filter.

C=0.205

□ NO. OF QUANTIZATION BITS=6
Fig. 4.3 - First Algorithm - Plot of number of bits required to represent each sample of the speech versus number of taps in the filter.

The approximate bits/sample in Eq. 2.6 is obtained by multiplying the number of taps by the square of the number of quantization bits and then dividing by 2 to the power of the number of bits/sample.

\[ \text{Bits/sample} = \frac{\text{No. of taps} \times \text{No. of quantization bits}^2}{2^\text{No. of bits/sample}} \]

For example, if there are 64 taps and 6 quantization bits, the number of bits/sample is approximately 5.9.
entropy was the highest in this case. This is due to the fact that encoding of predictor coefficients constitutes the major part of the total data required. Note the similarity between the shapes of plots in the figures 4.2 and 4.3. Again this is due to the fact that the data required to represent the predictor coefficients constitutes a very large part of the total data required.

The fewest number of bits/sample obtained using this algorithm were 5.25, which is quite high.

From all this it can be concluded that changing predictor coefficients for each sample is not the best method for minimizing the residual distribution entropy in order to minimize the bits/sample of the speech signal. It does result in a very low data for the representation of the difference signal, but predictor coefficients change dramatically from sample to sample resulting in a very high data requirement for the representation of the coefficients, resulting in a high number of bits/sample for the representation of the speech signal.

4.3 Second Algorithm:

Fig. 4.4 is the plot of residual distribution entropy versus number of iterations for the case of number of taps L = 4 and gradient coefficient c = 0.205. Residual distribution entropy as low as 1.0 was obtained when number of inplace-iterations were
Fig. 4.4 - Second Algorithm - Plot of the residual distribution entropy versus number of iterations.
increased to 20. But again as can be seen from fig. 4.5 predictor coefficients varied more and more from sample to sample as number of iterations were increased. As a result, representation of the predictor coefficients again constituted the major part of the total data required to represent the speech signal. Fig. 4.6 is the plot of bits/sample versus number of iterations. Again note the similarity between the shapes of figures 4.5 and 4.6. The fewest number of bits/sample were obtained for the case of number of iterations = 1 (Fig. 4.6). This special case is the same as First Algorithm. Therefore it is concluded from this that increasing the number of iterations did not result in any gain over the First Algorithm, in terms of data representation.

Fig. 4.7 is the long time average autocorrelation function of the speech signal. The correlation is high between adjacent samples and it decreases rapidly for greater spacings.

Fig. 4.8 and 4.9 are the long time average autocorrelation functions of the difference signal for the different cases as explained in the plots. Again as can be seen from these plots, the correlation is higher between the adjacent samples and decreases for greater spacings. Also as expected the correlation in case of difference signal is quite low compared to the correlation in case of speech signal. As number of iterations were increased, the
Fig. 4.5 - Second Algorithm - Plot of the first predictor coefficient distribution entropy multiplied by number of taps in the filter versus number of iterations.
Fig. 4.5 - Second algorithm - plot of the first predictor
Fig. 4.6 - Second Algorithm - Plot of number of bits required to represent each sample of the speech versus number of iterations.

L=4, C=0.205
\( \square \) NO. OF QUANTIZATION BITS=6

-30-
Fig. 4.6 - Second Algorithm - Plot of number of bits required to represent each sample of the speech versus number of iterations.
Fig. 4.7 - Long-time average autocorrelation function of speech signal.
Speech segment function

Fig. 4.7 - Long time average autocorrelation function of

\[ M \]
Fig. 4.8 - Second Algorithm - Long - time - average autocorrelation functions of difference signal.
Fig. 4.9 - Second Algorithm - Long-time average autocorrelation functions of difference signal.
Fig. 4.9 - Second Algorithm - Long-time average autocorrelation functions of difference signal.

- $L=4$, $C=0.205$
- $\text{NI}=1$
- $\text{NI}=20$
difference signal became less and less correlated.

4.4 Third Algorithm:

Fig. 4.10 shows the plots of residual distribution entropy versus the threshold difference signal, for number of taps in the filter $L = 2, 4$ and $8$. The threshold difference signal, $DT$ is defined as follows:

$$DT = \frac{\text{Threshold Difference Signal}}{\text{Maximum value of Speech Signal}}$$

As expected, as $DT$ is increased, the residual distribution entropy increased in general.

Fig. 4.11 shows the plots of predictor coefficient distribution entropies versus the threshold difference signal, for $L = 2, 4$ and $8$. All the plots show a minimum and then again the values increase.

Fig. 4.12 is the plot of number of times the difference signal was set to zero (i.e. number of times predictor coefficients were changed) versus the threshold difference signal. The total number of bits/sample for the speech signal were then calculated from equation 4.1.

$$\text{Bits/Sample} = \left[\text{R.D.E.} \times \text{B.L.}\right] + \left[\text{F.P.C.D.E.} \times N \times L\right]/\text{B.L.}$$

Where R.D.E. is Residual distribution entropy,

B.L. is block length,

F.P.C.D.E. is First predictor coefficient distribution entropy,

$N$ is Number of times prediction coefficients changed, and
Fig. 4.10 - Third Algorithm - Plots of the residual distribution entropy versus threshold difference signal.
Fig. 4.10 - Plots of the residual distribution entropy versus threshold.
Fig. 4.11 - Third Algorithm - Plots of the first predictor coefficient distribution entropy multiplied by number of taps versus threshold difference signal.
Fig. 4.11 - Third Algorithm - Plots of the first predictor coefficient distribution entropy multiplied by number of taps versus threshold difference signal.
Fig. 4.12 - Third Algorithm - Plot of number of times difference signal was set to zero versus threshold difference signal.
Fig. 4.12 - Third algorithm - plot of number of times different
L is Number of taps in the filter.

The fewest number of bits/sample, obtained using this algorithm were 2.49 for \( L = 4 \), as can be seen from the plots in fig. 4.13. This is quite an improvement over First and Second Algorithms.

This again supports the conclusion drawn earlier that adapting predictor coefficients for each sample is not the best approach for reducing the bits/sample.

4.5 Fourth Algorithm:

Fig. 4.14 shows the plot of residual distribution entropy versus the threshold difference signal, for number of taps in the filter \( L = 4 \), using the Fourth Algorithm.

Fig. 4.15 is the plot of number of bits required to represent each speech sample versus the threshold difference signal for \( L = 4 \). The fewest number of bits/sample using this algorithm were 2.17 for \( L = 4 \) as can be seen from the plot in fig. 4.15. This is again calculated from equation 4.1. This is a slight improvement over the Third Algorithm.

From the results of the First, Second, Third and Fourth Algorithms, it can be concluded that adapting predictor coefficients for each speech sample is not the best approach.
Fig. 4.13 - Third Algorithm - Plots of number of bits required to represent each sample of the speech versus threshold difference signal.
Approximate bits/sample

Fig. 4.13 - Third algorithm - plots of number of bits required to represent each sample of the speech signal versus threshold difference.
Fig. 4.14 - Fourth Algorithm - Plot of the residual distribution entropy versus threshold difference signal.

NO. OF QUANTIZATION BITS=6

\[ L=4 \]
Fig. 4.14 - Fourth Algorithm - plot of the residual distribution entropy versus threshold difference signal.
Fig. 4.15 - Fourth Algorithm - Plot of number of bits required to represent speech sample versus threshold difference signal.
signal

versus threshold difference

to represent speech sample

Fig. 4.15 - Fourth Algorithm - plot of number of bits required

APPROXIMATE BITS/SAMPLE

NO. OF QUANTIZATION BITS = 6

DL = 4
towards reducing bits/sample. But some other criteria for example those used in the Third and Fourth Algorithms represent a better approach, where predictor coefficients are kept from changing too much from sample to sample, but keeping the residual distribution entropy low.
REFERENCES


VITA

Seema Ranka, daughter of Shanti and Hukumchand Rathi, and wife of Ajay Ranke, was born in Nagpur, India on August 25, 1957. She obtained her high school education from the local institutions. She received her Bachelor of Science in Electrical Engineering from Nagpur University, Nagpur, India in 1979. She came to the United States in 1981 and was a teaching assistant at Lehigh University, Bethlehem, Pennsylvania from 1981 to 1983, where she completed requirements for her Master of Science degree in Electrical Engineering.