SELECTED METHODS FOR IMPROVING SYNTHESIS SPEECH QUALITY USING LINEAR PREDICTIVE CODING: SYSTEM DESCRIPTION, COEFFICIENT SMOOTHING AND STREAK

by

Steven Frank Boll
Ph.D.

November 1974 UTEC-CSc-74-151

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-73-C-0363 and Contract No. F30602-70-C-0300.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

This report develops two generalizations of the standard Linear Predictive Coding (LPC) implementation of a narrow band speech compression system. The purpose of each method is to improve the speech quality that is available from a standard LPC system. Attention is focused primarily upon the pitch excited system and therefore, the improvements considered focus upon the improved estimation of the reflection coefficients and the pitch period. Specifically, a parameter filtering algorithm is developed for dynamically smoothing the reflection coefficients to both increase naturalness in synthetic speech as well as eliminate the possibility of synthesis filter instabilities. Secondly, a new method for calculating the k-parameters of an LPC inverse filtering algorithm is developed, STREAK. New values for each k-parameter are calculated at each sample point directly using the lattice formulation of the inverse filter model. It is shown this technique can be used to improve a pitch detection scheme based upon the autocorrelation of inverse filter output sequence.

INTRODUCTION

This report is concerned with methods for improving narrow band synthesis speech quality generated by an LPC analysis synthesis system. As such it is assumed that the reader is familiar with the basic theory behind Linear Predictive Coding applied to speech. There are numerous references available which describe the techniques, advantages, and disadvantages of LPC, [1] [2] [3]. The intent of this report is to introduce and develop two new methods for improving speech quality by enlarging or replacing parts of so-called standard approaches. Therefore, only a brief description of LPC will be presented so as to provide a foundation to which these new techniques can be referenced.

Needless to say, there are many ways for improving speech quality. This report will primarily focus its attention on two major areas: one, improved estimation of reflection coefficients and two, improved estimation of pitch. Other methods for improvements such as parameter quantizations, and coding [4], fixed point implementation [5], and mode of transmission [6],although extremely important,will not be addressed.

Following a brief description of LPC are the reports in three major parts. Part one describes the various procedures which comprise the complete analysis-synthesis system. Part two describes a technique for the improved estimation of reflection coefficients using a minimum variance a priori least squares estimator. Part three describes a new method for calculating the reflection coefficients or k-parameters associated with the lattice form of the inverse filter and shows how this procedure for inverse filtering can be used for improved pitch tracking estimates.

Narrow Band Speech Compression

Digital speech transmission using conventional pulse code modulation requires channel bandwidths on the order of 60,000 bits per second. In order to reduce this rate to what might be called a narrow band speech compression system (typically 4000 bps or less) it is necessary to parameterize the speech waveform into a smaller (typically 13 to 20) set of slowly varying parameters. Estimates of these parameters are computed at some prescribed analysis rate, typically from 40 to 200 times per second. The parameters are then quantized, encoded and sent to the synthesizer across a transmission channel at a prescribed rate. Here they are decoded and supplied to a synthesizer algorithm which generates a synthesis speech waveform which hopefully sounds like the original in some acceptable manner. Thus, if the speech waveform were characterized by say 13 parameters, which could be coded to an average of 5 bits each and updated and sent every 200 ms, one would have a speech compression system requiring 3250 bps. The major components of such a system consist of an analyzer, a coder, a decoder, and a synthesizer. The standard parameters estimated in the analyzer consist of the signal energy, the voiced-unvoiced decision, the pitch period, and the set of vocal track descriptors. The vocal track is assumed to be accurately modeled by a digital filter defined as a ratio of polynomials. If the filter is assumed to have only poles, then linear predictive coding can be used to estimate the filter parameters, as well as determine energy, pitch and voicing.

Speech Analysis Using Linear Prediction

A linear prediction analysis of speech assumes that the n^{th} speech sample, s_n can be predicted approximately by a linear combination of the preceding p samples. Thus its approximation is given by

$$s_n' = \sum_{i=1}^{p} a_i \, s_{n-i}$$

where $\{a_i, i=1,2,..p\}$ is the set of real constants called predictor coefficients which are to be estimated. Values for these coefficients are found by minimizing the sum of the squares of the prediction error sequence, e_n where

$$e_n = s_n - s_n' = s_n - \sum_{i=1}^{p} a_i \, s_{n-i}$$

Thus values for the predictor coefficients are determined using a least squares estimator having as a loss function to be minimized

$$E = \sum_n e_n^2 = \sum (s_n - \sum_{i=1}^{p} a_i s_{n-i})^2$$

Note, as shown in Chapter II, this loss function E can be expanded to include a priori information leading to a smoother minimum variance estimate.

There are two basic approaches to linear prediction analysis. They vary according to the range of n used in defining the loss function E, and the definition of the signal s_n in that range. When using the covariance method the signal is defined over a finite range $-p \leq n \leq N-1$ and minimizing E leads to the set of normal equations [7]

$$\sum_{i=1}^{p} a_i \phi_{ij} = \phi_{j,o} \qquad j=1,2,\ldots,p$$

where

$$\phi_{ij} = \sum_{n=1}^{N} s_{n-i} s_{n-j}$$

The coefficient matrix $[\phi_{ij}]$ is positive definite covariance matrix and the system of equations can be efficiently solved using a triangularization method sometimes called the Cholesky decomposition [8] (See Chapter II). When using the autocorrelation method,the signal s_n is multiplied by a window of length N such that $s_n=0$ for $n<0$ and $n>N-1$. The range of n is assumed infinite and minimization of E leads to the normal equations [7]

$$\sum_{i=1}^{p} a_i r_{|i-j|} = r_j \qquad j=1,2,\ldots,p$$

where

$$r_i = \sum_{n=0}^{N-1} s_n s_{n+i}$$

The coefficient matrix $[R_{|i-j|}]$ is a positive definite Toeplitz matrix and the system of equations can be efficiently solved using Levinson's recursion [9].

Itakura [10] initially showed that a linear prediction analysis can be formulated in terms of another equivalent set of parameters $\{k_i i=1,2,..p\}$ called PARCOR, or reflection coefficients. It had been shown that these k-parameters are well suited as transmission parameters for a narrow band speech compression systems since they exhibit superior quantization properties [4], [10] and stability of the synthesis filter is guaranteed if $|k_i|<1$ [11].

Using the linear prediction model provides an effective method for detecting the pitch period. If the all-pole model defined by linear prediction accurately represents the vocal tract transfer function, and the radiation and glottal volume flow effects, then the output of the inverse filter, that is, the error signal e_n, should resemble an impulse like driving function having a period equal to the pitch for voiced speech. Absence of periodicity would imply unvoiced speech. Two approaches to pitch detection using inverse filtering are addressed in this report. The first concerns the standard block analysis approach, SIFT [12] and is described in Chapter I. The second uses a point by point analysis, STREAK and is described in Chapter III.

The energy needed to appropriately scale the synthesized output waveform can be obtained from the coefficient matrix (either $\phi_{0,0}$ or r_0), or by using the energy of the error signal itself, $\sum_{n=0}^{N-1} e_n^2$.

Methods for Improving Speech Quality

Considerable effort is currently being devoted to methods for improving the quality of synthesized speech generated from a narrow band compression system. Already noted are the studies in parameter quantization, fixed point implementation and improved modes of transmission. Other techniques exist such as more elaborate vocal tract models, and analysis [13], [14] as well as voice excited and error excited synthesizers [15]. However, implementing such techniques as these introduces the added drawbacks of increase complexity and computation and increased channel bandwidth. If the compression system is expected to operate in real time then complexity and computation must be minimized and if the bandwidth is to be constrained at a rate less than 4000 bps then the more elaborate excitation sequences must be simplified to a unit impulse driven sequence. In order to conform to these constraints of minimizing both computation rate and channel bandwidth, the techniques discussed in this report focus primarily upon the improved estimation of reflection coefficients and the pitch period using methods which are uncomplicated enough not to prohibit real time implementation or narrow band transmission.

I. DESCRIPTION OF THE ANALYSIS - SYNTHESIS SYSTEM

General Data Flow

A diagram showing the various stages in the entire system is shown in Figure I.1. The original analog speech waveform is first low pass filtered by an anti-aliasing filter having cutoff frequency f_c, sampled at a rate f_s, and quantized to q bits per sample. The digitized waveform is then stored on magnetic tape or disk.

The analysis portion of the system consists of three parts: (1) the data control routine for determining how the data is analyzed and transmitted to the synthesizer; (2) the actual analysis routines for estimating the vocal tract parameters: reflection coefficients, pitch, voicing and energy; and (3) the coding routines for optimally quantizing the analysis parameters for channel transmission.

The coded parameters are transmitted through the channel at a constant rate called the channel frame rate.

The synthesis portion of the system consists of two parts: (1) a routine for decoding the transmitted parameters; and (2) the synthesis routine for recursively generating synthetic speech. These samples are also stored on magnetic tape or disk.

A synthesized analog waveform is obtained from the D to A conversion of the processed samples which has been low pass filtered by an anti-imaging filter having cutoff frequency f_c.

Data Management (The Data Control Routine)

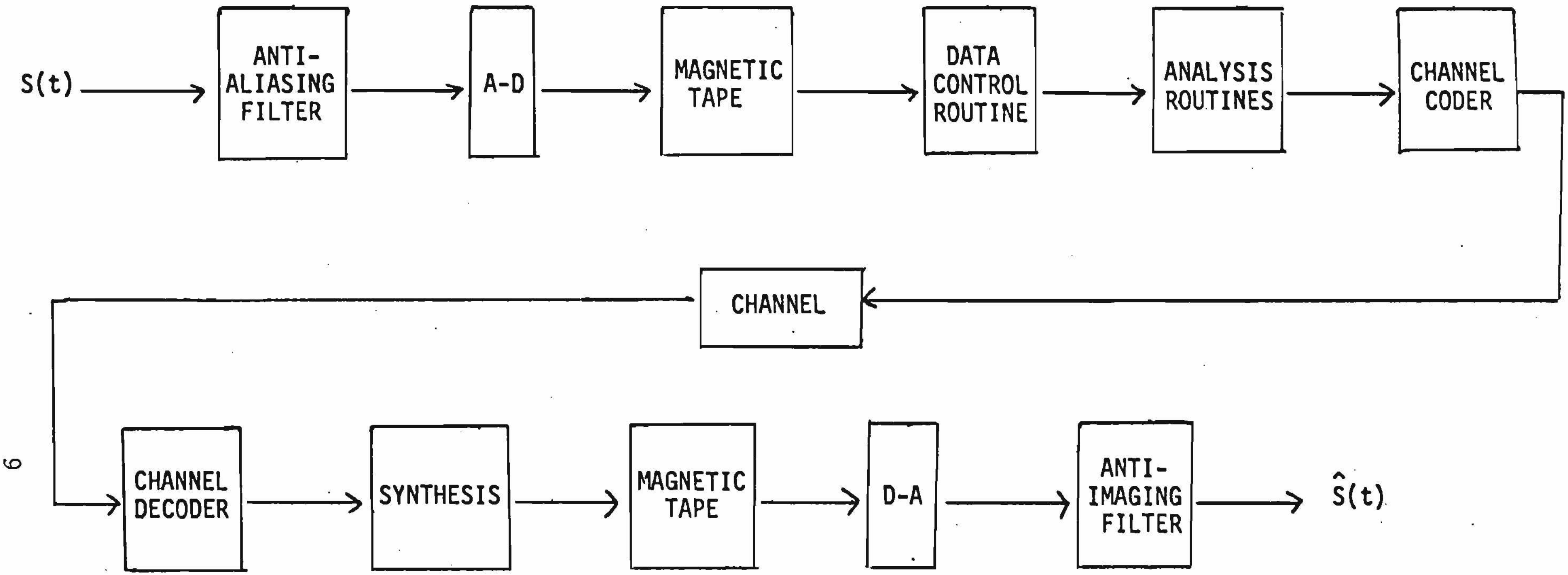
The approach taken for analyzing the incoming speech waveform is to separate it into possibly overlapping data sections called analysis frames, and extract a set of analysis parameters from each frame. At the completion of each analysis, the frame is shifted down the time line by loading new samples into the front end and dropping old samples off the back end. Thus one can view this approach as extracting the analysis parameters from that portion of the waveform which lies under a sliding analysis window.

Advancing the Analysis Frame

The approach used is to advance the analysis frame "pitch synchrounously". Specifically the analysis frame is shifted by an amount equal to the last estimated pitch period. This policy is followed except when the pitch period becomes less than a preset minimum jump distance for which the frame is then shifted a multiple of that pitch period. For example, if the minimum jump distance is 5 ms. and the pitch period is 4 ms., the frame is then advanced by 8 ms.

Size of the Analysis Frame

The size of the analysis frame is dictated by the amount of data needed to extract estimates of the pitch and reflection coefficients accurately. The analysis frame size for estimating pitch is set at 40 ms. whereas the frame size for estimating the reflection coefficients is set at 16 ms when using the covariance method and at 32 ms when using the autocorrelation method. Thus the overall analysis



ANALYSIS - SYNTHESIS SYSTEM

Figure I.1

frame size is set at 40 ms. with either the middle 16 ms or 32 ms used for reflection coefficient estimation.

Time Intervals Between Analyses

There are two different analysis frame rates used in the analysis. For coefficient estimation the frame rate is pitch synchronous as described above. As is described in Chapter II, this higher rate is necessary for smoothing the reflection coefficients.

However, the analysis portion of the system does no smoothing on the pitch estimates and therefore they need only be calculated as often as is dictated by the channel frame rate. That is, new estimates of pitch and voicing are needed only as often as they must be transmitted to the synthesizer. (Typically the channel frame rate is set at 50 frames/second or less.) Thus the pitch extraction analysis rate is set up to be multiple-pitch synchronous. A new pitch estimate is computed when the analysis frame has been shifted in time to a point required for a new pitch estimate to be transmitted to the synthesizer. For example, if the channel frame rate is set at 50 frames per second, then new pitch estimates are required every 20 ms. If the previous pitch period was found to be 5 ms. then the next pitch estimate will be computed after four shifts of the analysis frame.

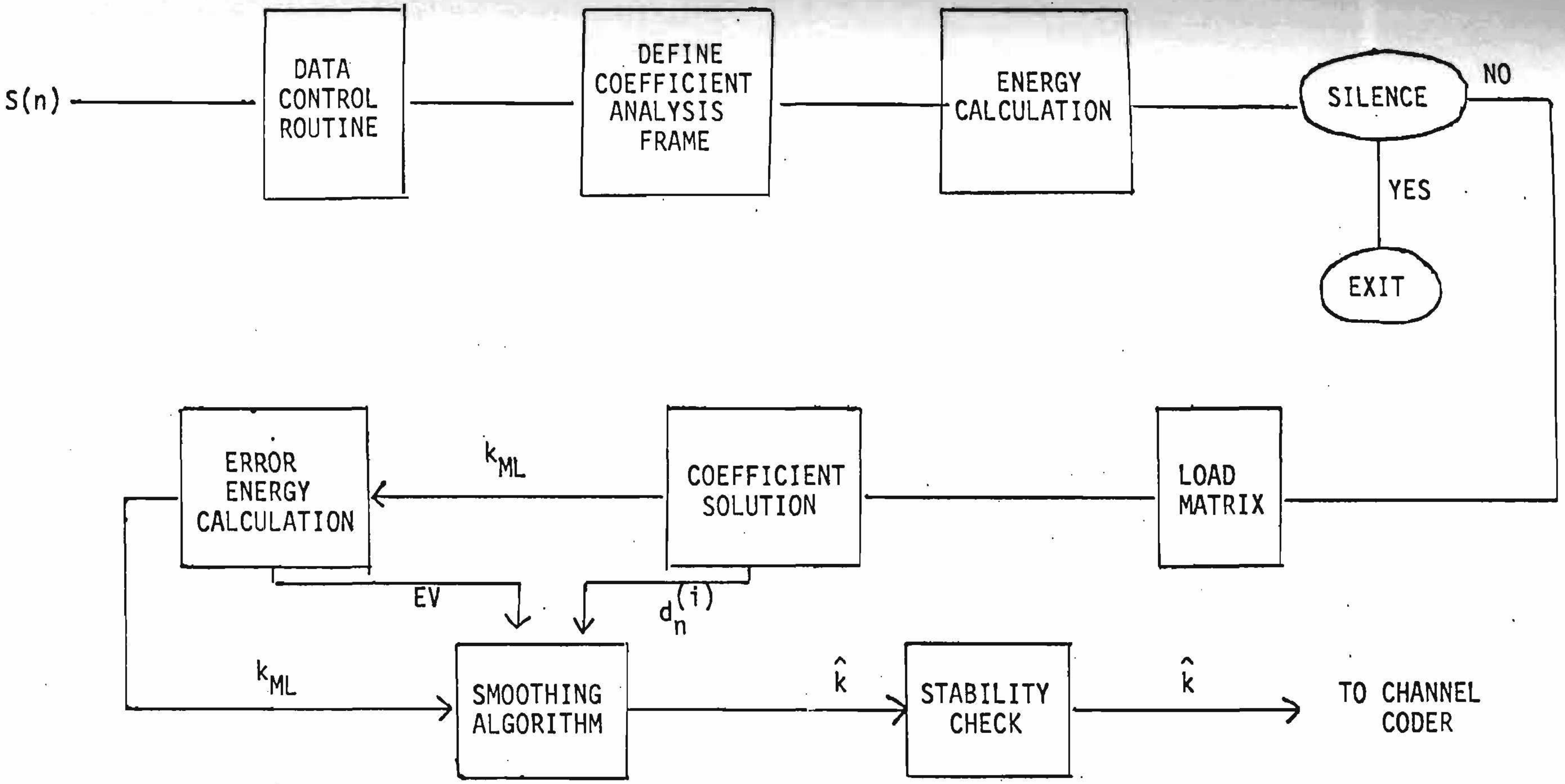
In summary, a data control routine specifies the length of the analysis frame and determines how it is shifted down the time line and which analysis parameters are to be computed at each shift. The analysis is pitch synchronous. New reflection coefficients are computed and smoothed at each shift. The reflection coefficients which are transmitted to the synthesizer consist of these values present at the points required for channel transmission. Thus if the channel frame rate is less than the pitch rate, the transmitted reflection coefficients represent a down-sampled version of the coefficients being estimated. Pitch and voicing are computed and transmitted at multiple shifts as dictated by the channel frame rate frequency. The length of each shift is set equal to the last pitch period estimated.

Reflection Coefficient Estimation and Smoothing

A diagram showing the various parts of the coefficient estimation routine is given in Figure I.2. Except for the smoothing algorithm which is appended at the end, the operations are similar if not identical to standard methods for estimating reflection coefficients. The routine can be used to estimate reflection coefficients using either the covariance [1], (Atal) method or the autocorrelation [2] (Markel, Itakura) method, which method is used depends, of course, upon the form of the linear system of equation which is to be solved.

Coefficient Analysis Frame Length

The overall analysis frame size specified by the data control routine is considerably larger (typically 40 ms.) than the coefficient analysis frame size to be used for estimating the reflection coefficients, (either 16 or 32 ms). Thus the first step in this routine is to extract a subset of length N from the center of the analysis frame. This subset, the coefficient analysis frame, is used to compute an energy term and reflection coefficients.



REFLECTION COEFFICIENT ESTIMATION

Figure I.2

Energy Calculation and Silence Detection

The zeroeth correlation term,

$$\phi_{0,0} = \sum_{n=1}^{N} s_n^2 \quad \text{or} \quad r_0 = \sum_{n=0}^{N-1} s_n^2 \qquad \text{I.1}$$

is initially computed and from it is computed the energy estimate,

$$R1 = \left(\frac{\phi_{0,0}}{N} \right)^{\frac{1}{2}} \quad \text{or} \quad R1 = \left(\frac{r_0}{N} \right)^{\frac{1}{2}} \qquad \text{I.2}$$

R1 is then compared against a threshold value, THRES, to determine if the waveform to be analyzed represents a silent region. For 12-bit samples, values of R1 less than a THRES equal to 12 imply the analysis frame represents silence and the routine is exited.

Matrix Loading (Covariance Method)

The linear system of equations to be used for estimating the reflection coefficients was developed in Reference [7].

The linear system is given by:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1,p} \\ \phi_{2,1} & \phi_{22} & \cdots & \phi_{2,p} \\ \cdot & & & \\ \cdot & & & \\ \phi_{p,1} & \phi_{p,2} & \cdots & \phi_{p,p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_{1,o} \\ \phi_{2,o} \\ \cdot \\ \cdot \\ \phi_{p,o} \end{bmatrix} \tag{I.3}$$

WHERE $\phi_{ij} = \sum_{n=1}^{N} s_{n-i} \; s_{n-j}$

and a_i, $i = 1, 2, \ldots, p$ = Maximum Likelihood or the Classical estimate of the predictor coefficients

Initially $\phi_{o,i,}$ $i = o, 1, \ldots, p$, are computed and from these values the remaining elements of the matrix are found using the standard method:

$$\phi_{i,i} = \phi_{i-1,\, i-1} + s^2_{-i+1} - s^2_{N+1-i} \qquad i = 2, \ldots, p$$

$$\phi_{i+1,i} = \phi_{i,i-1} + s_{-1}\, s_{-i+1} - s_{N-i}\, s_{N-i+1} \qquad i = 1,2,\ldots p-1$$

$$\phi_{i,i+1} = \phi_{i+1,i}$$

Matrix Loading (Autocorrelation Method)

The linear system of equations used to estimate the reflection coefficients for the autocorrelation method is given by [7]

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_{p-1} \\ r_1 & r_0 & \cdot & \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ r_{p-1} & & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_p \end{bmatrix} \qquad \text{I.4}$$

WHERE

$$r_n = \sum_{i=0}^{N-1-i} s_i' s_{i+n}'$$

a_i, i=1,2,...,p= Maximum likelihood or the classical estimate of the predictor coefficients.

s_n', n=0,1,...N-1= windowed speech sampled (usually using a Hamming type window). Makhoul [3] and Markel [2] have shown that preemphasizing the input speech improves the synthesis speech quality. With that implementation, the samples used to form r_n are given by

$$s_n' = (s_n - c \cdot s_{n-1}) \cdot W_n$$

WHERE s_n = speech samples

W_n = window samples

c = preemphasis coefficient, which is sampling frequency dependent. See Markel [2].

An efficient method for calculating the short term reflection coefficients r_n has been developed by both Blankenship [16] and Pfiefer [17].

Coefficient Solution from Linear Equations

The maximum likelihood, unweighted least squares estimate of the reflection coefficients is obtained from the linear system of equations using the Cholesky decomposition method [8] (Mitsui). In matrix notation let equation I.3 or I.4 be represented as:

$$H^T H\alpha = H^T y \qquad \text{I.5}$$

as can be shown

$$H^T H = LDU$$

$$L = U^T,\ D = \text{diagonal matrix} \qquad \text{I.6}$$

where L is a nonsingular triangular matrix obtained from the Cholesky decomposition. Substituting gives:

$$LDU\alpha = H^T y \qquad \text{I.7}$$

$$\text{or} \qquad LDK_{ML} = H^T y \qquad \text{I.8}$$

$$U\alpha \quad = k_{ML} \qquad \text{I.9}$$

WHERE $\hat{k}_{ML}$ = p × 1 vector of maximum likelihood reflection coefficients. The $\hat{k}_{ML}$ parameters represent those reflection coefficients obtained using the classical unweighted least squares solution. They can now be smoothed using some method such as the a priori least smoothing technique. If no smoothing is to be used, the analysis except for stability checks and the error energy calculation, has been completed.

Reflection Coefficient Smoothing

This portion of the algorithm represents a primary contribution to this report and is discussed in considerable detail in the next chapter.

Error Energy Calculation

A term which is used as part of a secondary criterion for the voiced-unvoiced decision is the energy of the prediction error sequence. Define the error sequence as

$$e_n = s_n - \sum_{i=1}^{p} a_i \, s_{n-1} \qquad \text{I.10}$$

Then it can be shown that

$$EV = \sum_{n=1}^{N} e_n^2 = \phi_{0,0} - \sum_{i=1}^{p} a_i \, \phi_{i,0} \qquad \text{(Covariance)} \qquad \text{I.11}$$

or

$$EV = \sum_{n=1}^{N} e_n^2 = r_0 - \sum_{i=1}^{p} a_i \, r_i \qquad \text{(Autocorrelation)}$$

From this energy term is computed a ratio term which is used as a secondary voiced-unvoiced decision criterion (Atal [1]). Define

$$RATIO = \frac{\phi_{0,0}^2}{EV} \qquad \text{I.12}$$

Then assuming 14-bit speech samples, for RATIO less than 0.7×10^8 the analysis frame is defined to be unvoiced.

Stability Check

The reflection coefficients are checked for stability before exiting the routine. Any reflection coefficient having a magnitude greater than or equal to 1 is redefined to have a magnitude of 0.97. Using the autocorrelation method guarantees stability assuming floating point implementation [11].

Pitch and Voicing Detection (Block Analysis Approach)

A diagram showing the various parts of the pitch and voicing detection routine is shown in Figure I.3. This routine is called at a rate defined by the channel frame rate frequency. Thus, if the channel frame rate is set at 50 frames per second, pitch and voicing values will be estimated 50 times per second. The time interval between each estimate, being defined as that multiple of the previous pitch period, which shifts the analysis frame into the next 20 ms. interval.

The estimate of the pitch period is found by autocorrelating on the prediction error signal. This method is similar to Markel's SIFT algorithm [12] and Itakura's modified autocorrelation method [18].

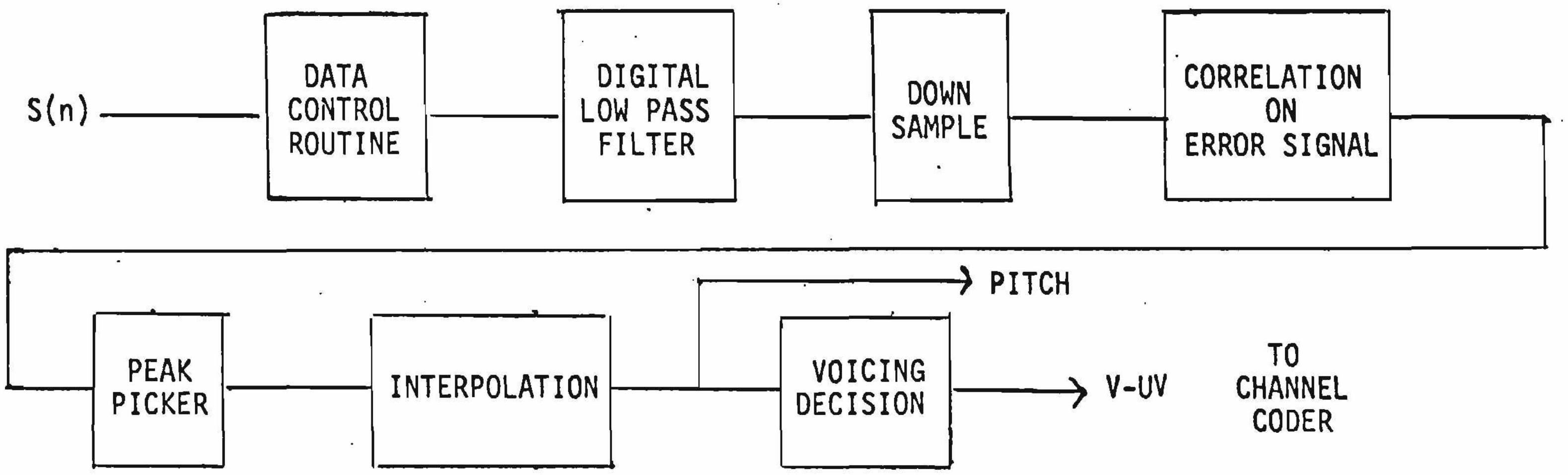
Digital Low-Pass Filtering and Down-Sampling

From the data control routine a 40 ms. analysis frame is extracted from the speech waveform. This analysis frame is then digitally low-pass filtered with a three pole Chebyshev 2-dB ripple filter having a 3-dB cutoff at 750 Hz for an 8 kHz sampling rate and at 600 Hz cutoff for a 6.4 kHz sample rate. This signal is then down-sampled four to one and passed on to the pitch detection routine.

Correlation on Error Signal and Pitch Estimate

The down-sampled signal is windowed with a Hamming window and the correlation W(n), of the error signal e(n), is formed using a third order predictor. Define

$$e_n = s_n - \sum_{i=1}^{3} a_i s_{n-i} \qquad n=1,\ldots, N \qquad \text{I.14}$$



PITCH AND VOICING DETECTION

Figure I.3

as the error signal, and

$$W(n) = \sum_{i=1}^{N} e_i \, e_{i+n} \qquad n = 0,1,\ldots, \text{maximum pitch period} \qquad \text{I.15}$$

as its short term autocorrelation. Where s_n equals the down-sampled windowed speech samples. Chapter III discusses a method for generating the error sequence, e_n using the STREAK algorithm.

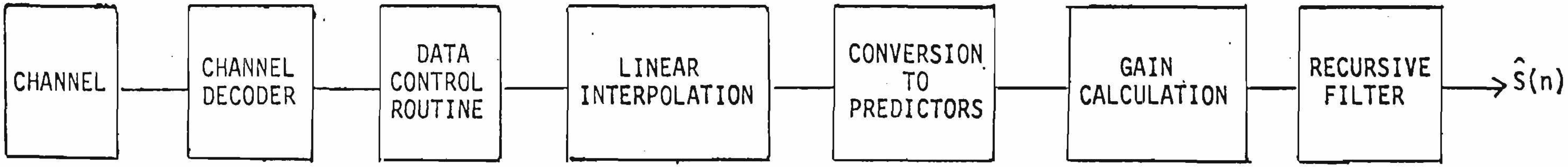
The W(n) sequence is searched between the limits of four and forty for its maximum value. The index M having the largest value after interpolation defines the pitch period estimate. The magnitude of W(M)/W(0) is used for voicing detection.

Voicing Decision

The correlation detector will have a magnitude between zero and one. If the correlation detector has a value greater than or equal to 0.3, the analysis frame is initially defined to be voiced speech. If the value of RATIO is defined in the previous section is also greater than the threshold value 0.7×10^8, then the analysis frame remains defined as voiced. If RATIO is less than 0.7×10^8 the voicing decision is switched to unvoiced.

Speech Synthesis

A diagram showing the various parts of the synthesizer is shown in Figure I.4. The decoded channel parameters consisting of p reflection coefficients, pitch, voicing and energy are received and used for updating the synthesizer at the channel frame rate.



SYNTHESIS SYSTEM

Figure I.4

Parameter Updating and Interpolation

There are two sets of channel parameters available to the synthesizer at any one time, a left set and a right set. Each set corresponds to successive left and right analyses separated by the multiple of the pitch period estimated in the left analysis. The synthesizer will generate the same number of pitch periods as the analysis was shifted down before performing the right analysis. To insure that the first set of parameters used by the synthesizer are always synchronized properly with the left analysis set, the left set pitch period is repeated for each synthesis until a new channel parameter set is received. Stated another way, the synthesis is always advanced by the same amount as the analysis is advanced.

Even though this approach prevents the pitch from being interpolated,the reflection coefficients and energy are interpolated prior to each new pitch period interval.

In summary, the pitch is not interpolated and the left set pitch period is repeated until a new channel parameter set arrives. The reflection coefficients and energy are linearly interpolated using the left and right channel sets as end points, with the first interpolated set equaling the left set. New interpolated values are used at the beginning of each pitch period. Since the analysis is advanced at the same rate, this method insures that the first set of parameters used for synthesis corresponds (except for quantization error) to the left analysis set. Finally, after that multiple of pitch periods have been advanced such that a new set of channel parameters can be used, the right set becomes the left set and the channel set becomes the right set.

Conversion to Predictor Coefficients

Synthesis is performed using the transversal filter configuration:

$$\hat{s}_n = \sum_{i=1}^{p} a_i \hat{s}_{n-i} + g.e_n \qquad n = 1,2,\ldots M \qquad \text{I.16}$$

WHERE $\hat{s}_n$ = synthesized speech

$$e_n = \begin{cases} 1 \text{ for } n=1,\ 0\ n\neq 1, \text{ voiced} \\ \text{random numbers, unvoiced} \end{cases}$$

g = gain term

M = pitch period

Therefore, since reflection coefficients were transmitted, a set of predictor coefficients must be obtained. Using the standard mapping from reflections to predictors Atal [1], a set of predictor coefficients are obtained for each interpolated set of reflection coefficients.

Gain Calculation and Speech Synthesis

A value for the gain term, g is estimated to insure that the energy of the synthesized speech signal equals the energy calculated from the original waveform. The method used is that proposed by Atal [1]. This method requires considerably more computation than simply using the square root of the energy of the error signal. However, it has been determined that using the latter method can cause amplitude modulation of the synthesized waveform, whereas, if gain is found by matching energies, this modulation does not occur.

Speech is synthesized using the recursion defined in equation I.16. The resulting samples are then stored on magnetic tape or disk for subsequent conversion to an audio waveform by D to A conversion. A detailed study of synthesis using fixed point arithmetic has been developed by Markel and Gray [4].

Channel Parameter Coding and Decoding

This study did not consider new methods for optimally quantizing the channel parameters. Procedures were written to quantize the channel parameters based upon studies of Makhoul and Viswanathan [4] and Markel [2]. The reflection coefficients, k_i were coded by linearly quantizing the log area functions g_i derived from the reflection coefficients, where

$$g_i = \log \frac{1+k_i}{1-k_i}$$

the pitch and energy were logarithmically quantized.

II. SMOOTHING REFLECTION COEFFICIENTS USING AN A PRIORI LEAST SQUARES ESTIMATOR

Introduction

This chapter discusses how the a priori least squares algorithm can be used to obtain a smoothed, minimum variance estimate of the reflection coefficients. As will be shown, smoothing results from the fact that each coefficient is filtered by a time-varying, first order, recursive low pass filter with filter coefficients defined by the least squares algorithm. Values for the coefficients are dynamically updated based upon the short-term characteristics of the speech waveform itself. As a result of this approach to smoothing, the filtering action is adaptive: heavy smoothing during stationary portions of the waveform, and light or negligible smoothing during nonstationary, transition portions. Secondly, the smoothing is efficient: since it is accomplished using a set of first order filters, the additional computation required does not become so excessive as to prevent real-time implementation.

The chapter has three main sections; the general development of the Minimum Variance A Priori least squares estimate, the simplification of the algorithm to scalar equations using the Gram-Schmidt orthoganalization, and the implementation and results of the algorithm.

The Minimum Variance A Priori Least Squares Estimate

The problem of estimating successive sets of reflection coefficients from successive sets of analysis frames can be viewed in general terms as the estimation of one random process, (the reflection coefficients) from observations of a different but related process, (the speech waveform itself). For each analysis frame one extracts a set of speech samples, s_n and constructs a data vector y_n and a measurement matrix H_n. Using this information an estimate $\hat{k}_n$ of the random process k_n is calculated. The dynamic model relating y_n to k_n is given by

$$y_n = H_n B_n k_n + \varepsilon_n \qquad \text{II.1}$$

where y_n is an N×1 vector of speech samples, and $B_n H_n$ is an N×p matrix constructed according to the linear predictive coding model, (see Ref. [7] for a discussion of H_n and next section for a discussion of B_n). The vector ε_n represents the modeling or prediction error. The vector k_n represents the set of p reflection coefficients to be estimated during the n^{th} analysis frame.

The minimum variance a priori least squares estimate $\hat{k}_n$ of k_n is found by minimizing the loss function, L_n of the n^{th} analysis frame

$$L_n = (y_n - H_n B_n k_n)^T R_n^{-1} (y_n - H_n B_n k_n) + (k_n - \bar{k}_n)^T M^{-1} (k_n - \bar{k}_n) \qquad \text{II.2}$$

where

$R_n = E\{\varepsilon_n \varepsilon_n^T\}$, the N×N positive definite covariance matrix of ε_n

$M_n = E\{(k_n - \bar{k}_n)(k_n - \bar{k}_n)^T\}$, the p×p positive definite matrix of the a priori covariance of $\bar{k}_n$.

The estimate, $\hat{k}_n$ is found by minimizing L_n with respect to k_n and is given by (see Ref. [7] for detailed development)

$$\hat{k}_n = (B_n^T H_n^T R_n^{-1} H_n B_n + M_n^{-1})^{-1} (B_n^T H_n^T R_n^{-1} y_n + M_n^{-1} \bar{k}_n) \tag{II.3}$$

The covariance, P_n of $\hat{k}_n$ is given by

$$P_n = \mathrm{Cov}\ (\hat{k}_n) = E\{(k_n - \hat{k}_n)(k_n - \hat{k}_n)^T\} = (B_n^T H_n^T R_n^{-1} H_n B_n + M_n^{-1})^{-1} \tag{II.4}$$

An equivalent expression for k_n is given by

$$\hat{k}_n = \bar{k}_n + P_n H_n^T R_n^{-1} (y_n - H_n B_n \bar{k}_n) \tag{II.5}$$

The vector $\bar{k}_n$ represents the a priori estimate of k_n. As such it represents the best estimate of k_n prior to knowledge of y_n. For this analysis, it is assumed that the reflection coefficients k_n obey the following relationship

$$k_{n+1} = k_n + w_n \tag{II.6}$$

where

$$E\{w_n\} = 0$$

$$E\{w_n w_j\} = Q_n \delta_{n,j} \tag{II.8}$$

Thus the expected value of k_n before knowledge of y_n denoted

$$\bar{k}_n = E\ \{k_n \mid y_1, y_2, \ldots, y_{n-1}\} \tag{II.9}$$

is given by

$$\bar{k}_n = \hat{k}_{n-1} \tag{II.10}$$

with covariance, M_n given by

$$M_n = E\{(k_n - \bar{k}_n)(k_n - \bar{k}_n)^T\}$$

$$= E\{(k_{n-1} + w_n - \hat{k}_{n-1})(k_{n-1} + w_n - \hat{k}_{n-1})^T\}$$

$$= P_{n-1} + Q_n \qquad \text{II.11}$$

Thus the expected variance of $\bar{k}_n$ equals the variance of $\hat{k}_{n-1}$ plus the variance of the difference between k_n and k_{n-1}.

Note that this definition of M_n allows the estimate of k_n to adjust dynamically to the changes in the speech waveform itself. For stationary sections of speech where $k_n \approx k_{n-1}$, $Q_n \approx 0$ and $M_n \approx P_{n-1}$ whereas for transition regions where k_n differs appreciably from k_{n-1}, the variance on $\bar{k}_n$ will equal the variance of $\hat{k}_{n-1}$ (the best estimate prior to measurement) plus the variance of the change from k_{n-1} to k_n. Thus the degree to which you believe in $\bar{k}_n$ as an estimate of k_n is directly affected by how things have changed from the previous frame, namely the covariance Q_n.

The expressions given in equations II.3 through II.11 are matrix equations and as such their implementation would be computationally prohibative without further reduction. The next section describes how these equations can be reduced to p sets of scalar equations.

Simplification of the Algorithm to Scalar Equations Using the Gram-Schmidt Orthoganalization

The a priori estimator equations, II.3 through II.5 can be reduced to p-sets of scalar equations by taking advantage of two facts. First, since as Mitsui [8] has pointed out, the reflection coefficients represent the weights of an orthoganal basis set, which add up to form the predicted speech wave, they are independent of each other. Thus the covariance matrices M_n and P_n are diagonal. Second, by using the Cholesky decomposition the Gram matrix, $H^T H$ can be diagonalized. Specifically, we have from the Cholesky decomposition that

$$H^T H = LDU \qquad \text{I.5}$$

and

$$B = U^{-1}$$

Thus the a priori estimate, equation II.3 can be diagonalized as follows: given

$$\hat{k}_n = (B_n^T H_n^T R_n^{-1} H_n B_n + M_n^{-1})^{-1} (B_n^T H_n^T R_n^{-1} y_n + M_n \bar{k}_n) \qquad \text{II.3}$$

let $\quad R_n = r_n I$

$$M_n = \text{diag } [m_n^{(i)}]$$

then $\quad B_n^T H_n^T R_n^{-1} H_n B_n = \frac{1}{r} L^{-1} LD_n UU^{-1} = \frac{1}{r} D_n \qquad$ II.12

where $\quad D_n = \text{diag } [d_n^{(i)}]$

thus $\quad \hat{k}_n = (\frac{1}{r_n} D_n + M_n^{-1})^{-1} (\frac{1}{r_n} B_n H_n^T y_n + M_n^{-1} \bar{k}_n) \qquad$ II.13

or $\quad \hat{k}_n = (\frac{1}{r_n} + D_n^{-1} M_n^{-1})^{-1} (\frac{1}{r_n} D_n^{-1} B_n H_n^T y_n + D_n^{-1} M_n^{-1} \bar{k}_n) \qquad$ II.14

but $$D_n^{-1} B_n H_n^T y = S^{-1} H_n^T y_n = k_{ML}, \text{ Least squares classical estimate} \qquad \text{II.15}$$

thus $$k_n = [\frac{1}{r_n} I + (M_n D_n)]^{-1} [\frac{1}{r_n} k_{ML} + (M_n D_n)^{-1} \bar{k}_n] \qquad \text{II.16}$$

The matrix product $M_n D_n$ will be diagonal, equaling

$$M_n D_n = \text{diag } [m_n^{(i)} d_n^{(i)}] \qquad \text{II.17}$$

Thus equation II.16 and therefore equation II.3 reduces down to p-sets of scalar equations given by

$$\hat{k}_n^{(i)} = \frac{\frac{m_n^{(i)} d_n^{(i)}}{r_n}}{1 + \frac{m_n^{(i)} d_n^{(i)}}{r_n}} k_{ML}^{(i)} + \frac{1}{1 + \frac{m_n^{(i)} d_n^{(i)}}{r_n}} \bar{k}_n^{(i)} \qquad \text{II.18}$$

$$i=1,2,\ldots,p$$

In addition $$\bar{k}^{(i)} = \hat{k}_{n-1}^{(i)} \qquad i=1,2,\ldots,p \qquad \text{II.10}$$

Thus the a priori estimator reduces to p-sets of first order recursive filters on the reflection coefficients.

$$\hat{k}_n = \frac{A_n^{(i)}}{1 + A_n^{(i)}} k_{ML}^{(i)} + \frac{1}{1 + A_n^{(i)}} \hat{k}_{n-1}^{(i)} \qquad \text{II.20}$$

where $$A_n^{(i)} = \frac{m_n^{(i)} d_n^{(i)}}{r_n} \qquad i=1,2,\ldots,p.$$

Starting with equation II.5 an equivalent expression for $\hat{k}_n$ can be derived as

$$\hat{k}_n^{(i)} = \hat{k}_{n-1}^{(i)} + \frac{A_n^{(i)}}{1 + A_n^{(i)}} (k_{ML}^{(i)} - \hat{k}_{n-1}^{(i)}) \quad i=1,2,\ldots,p \qquad \text{II.20}$$

The covariance, P_n of $\hat{k}_n$ can likewise be diagonalized as follows:

$$P_n = (B_n^T H_n^T R_n^{-1} H_n B_n + M_n^{-1})^{-1} = (\frac{1}{r} D_n + M_n^{-1})^{-1} \qquad \text{II.21}$$

or for each coefficient

$$p_n^{(i)} = E\{(k_n^{(i)} - \hat{k}_n^{(i)})^2\} = m_n^{(i)} / (1 + \frac{m_n^{(i)} d_n^{(i)}}{r_n}) \quad i=1,2,\ldots,p$$

$$\text{II.22}$$

Thus equation II.20 can be expressed in terms of $p_n^{(i)}$ as

$$\hat{k}_n^{(i)} = \hat{k}_{n-1}^{(i)} + \frac{p_n^{(i)} d_n^{(i)}}{r_n} (k_{ML}^{(i)} - \hat{k}_{n-1}^{(i)}) \quad i=1,2,\ldots,p \qquad \text{II.23}$$

Using equation II.11 each diagonal element of the covariance, M_n on $\bar{k}_n$ is given by

$$m_n^{(i)} = E\{(k_n^{(i)} - \bar{k}_n^{(i)})^2\} = E\{(k_{n-1}^{(i)} - \hat{k}_{n-1}^{(i)})^2\} + E\{(w_n^{(i)})^2\} \qquad \text{II.24}$$

or $$m_n^{(i)} = p_{n-1}^{(i)} + q_n^{(i)} \quad i=1,2,\ldots p \qquad \text{II.25}$$

Implementation and Results

The a priori least squares estimate is obtained from the following five step process.

Step

1 Compute the maximum likelihood estimate, $k_{ML}^{(i)}$ and diagonal elements, $d_n^{(i)}$ of D_n.

2 Estimate the elements of the covariance matrix,

$$m_n^{(i)} = p_{n-1}^{(i)} + q_n^{(i)}$$

3 Estimate the covariance on the modeling error, r_n.

4 Compute the covariance of $p_n^{(i)}$ of $k_n^{(i)}$, $p_n^{(i)} = \dfrac{m_n^{(i)}}{1 + \dfrac{m_n^{(i)} d_n^{(i)}}{r_n}}$

5 Compute the final estimate, $\hat{k}_n^{(i)}$

$$\hat{k}_n^{(i)} = \hat{k}_{n-1}^{(i)} + \frac{p_n^{(i)} d_n^{(i)}}{r_n} \left(k_{ML}^{(i)} - k_{n-1}^{(i)}\right)$$

The covariance r_n of the modeling error, ε_n is approximated by computing the residual

$$EV = (y_n - H_n B_n k_{ML})^T (y_n - H_n B_n k_{ML}) \qquad \text{II.26}$$

or $$EV = \phi_{0,0} - \sum_{i=1}^{p} \phi_{1,0}\, a_i \text{ or } r_0 - \sum_{i=1}^{p} r_i a_i \qquad \text{II.27}$$

and setting $$r_n = \frac{EV}{N} \qquad \text{II.28}$$

where N = Coefficient analysis window size

The covariance q_n defining expected variance on the difference in reflection coefficients is approximated as

$$q_n = E\,(k_n - k_{n-1})^2 = (k_{ML}|_n - k_{ML}|_{n-1})^2 \qquad \text{II.29}$$

The initial values for starting the algorithm are defined to be

$$\hat{k}_0^{(i)} = 0$$

$$m_0^{(i)} = (k_{ML}^{(i)})^2$$

Pitch Synchronous Analysis

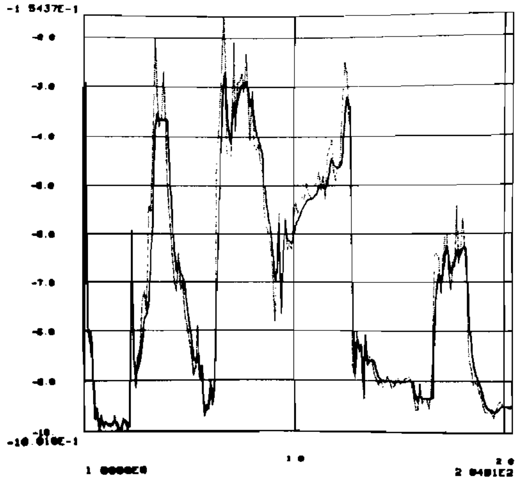
To insure that the a priori estimate $\bar{k}_n$ of the reflection coefficients represents a reasonable estimate of k_n prior to using the data vector y_n, the coefficient analysis is done pitch synchronously, rather than at the slower rate defined by the channel frame rate. By shortening the distance between successive updates, the variance M_n of $(\bar{k}_n - k_n)$ decreases during stationary speech segments thus increasing the amount of smoothing.

This analysis approach is similar to that used in providing the down-sampled speech to the pitch detection system. That is, the coefficients are estimated at the high, pitch synchronous rate, smoothed, then down-sampled for channel transmission. The primary difference between the two down-sampling methods is that the smoothing filter applied to the coefficients must be time-varying. During stationary segments of speech, the filter has a narrow pass band with its pole near the unit circle, while during transition regions, the filter essentially locks onto the input, mainly k_{ML}, with its pole near the origin.

Results

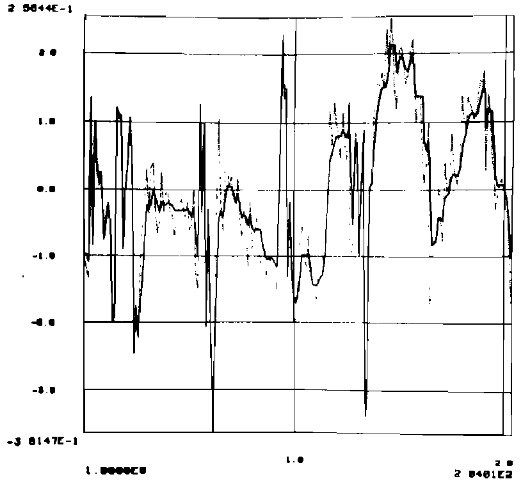
The results of using smoothing are best demonstrated by listening to processed speech using smoothed reflection coefficients. Informal listening tests indicate that smoothing improves speech quality in terms of (1) reducing the number of instabilities when using the covariance method and thereby reducing the number of annoying non-speech like pops resulting from hard limiting the coefficient values back to 0.97; (2) reducing the roughness on sustained vowel regions induced by slow update rate, (3) eliminating the warbling induced by step discontinuities in the spectrum, and (4) less degrading of speech quality as the analysis window size is narrowed.

Figures II.1 and II.2 show the result of the adaptive smoothing applied to the k_1 and k_{10} reflection coefficients. The successive phonemes /eI/, /i/, /aI/, /oU/, /u/, were digitized at 6.4 kHz and processed using a tenth order filter using the covariance method. The time histories of both coefficients with and without smoothing is displayed, with the darker curve corresponding to the smoothed estimate.



k_1 with and without adaptive smoothing

Figure II.1



k_{10} with and without adaptive smoothing

Figure II.2

SUMMARY AND CONCLUSIONS

As has been shown, when the reflection coefficients are estimated using the a priori algorithm, smoothing results from the fact that each coefficient is filtered by a time-varying, first order, recursive low-pass filter. Values for the coefficients are dynamically updated based upon the short-term characteristics of the speech waveform itself. There are two major advantages to smoothing in this manner: (1) the filtering is adaptive, heavy smoothing during stationary portions of the waveform, and light or negligible smoothing during non-stationary, transition portions; and (2) the smoothing algorithm is efficient, requiring only first order filters and therefore, can possibly be implemented in real time.

III. STREAK: A Simplified Technique for Recursively Estimating Autocorrelation k-Parameters

Introduction

A Simplified Technique for Recursively Estimating Autocorrelation k-parameters (hereafter called STREAK), defines a method for calculating the k-parameters associated with the lattice form of the inverse filter model used in the linear prediction analysis. This method differs from standard LPC approaches in two major respects: one, the k-parameters are estimated directly from the lattice model; and two, new estimates are calculated for each A-D sample. This chapter describes how these coefficients are calculated, how they may be used in an analysis-synthesis system, and how this technique could be used to improve the quality of a pitch extraction routine based upon the autocorrelation of the inverse filter output sequence.

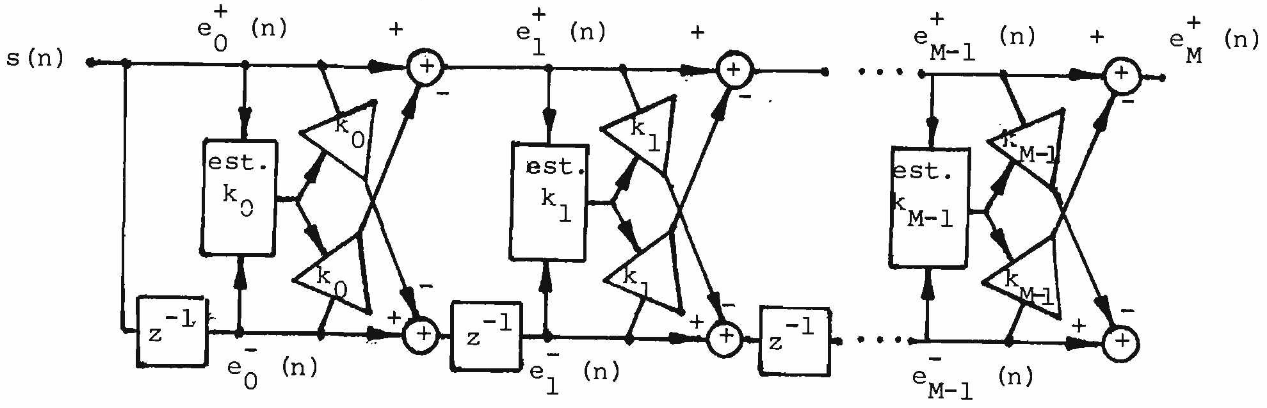
The standard approach in linear prediction is to estimate one set of M coefficients from a block of N data points, [1], [2], [3]. Values for these coefficients are calculated so as to minimize the sum of the squares of the prediction error sequence. As such, the least squares curve fit is applied uniformly over the entire block of N samples. This paper introduces a new concept in inverse filtering. Rather than estimating one set of parameters for a window of N samples, a new least squares estimate of each parameter is calculated at each point. The analysis is based upon the lattice form [2] of the inverse filter. Values for the k-parameters are obtained directly in terms of the forward and backward prediction error sequences.

The method is called a Simplified Technique for Recursively Estimating Autocorrelation k-parameters, or STREAK since: (1) only scalar equations are involved in the analysis, thus reducing the complexity of implementation; (2) successive k-parameter estimates are recursively estimated from preceding values; and (3) like the standard autocorrelation approach, these k-parameter estimates are bounded in magnitude by one.

The chapter is divided into two parts: a development of estimating k-parameters directly from the lattice form of the inverse filter, and a comparison of the inverse filter output using STREAK versus the output using the standard autocorrelation method.

The Lattice Formulation for Inverse Filtering

Itakura and Saito [10] developed a formulation for linear prediction analysis using a lattice form for the inverse filter. A block diagram of their PARCOR analyzer is shown in Figure III.1.



Lattice Form Inverse Filter

Figure III.1

Both forward e_m^+ (n) and backward e_m^- prediction error sequences for this filter are defined as

$$e_m^+ (n) = \sum_{i=0}^{m} a_{m,i} \; s(n-i) \qquad \text{III.1}$$

$$e_m^- (n) = \sum_{i=1}^{m+1} b_{m,i} \; s(n-i) \qquad \text{III.2}$$

The z-transforms for these prediction error sequences are defined as

$$e_m^+ (n) \leftrightarrow U_m^+ (z) = A_m (z) \; S (z) \qquad \text{III.3}$$

$$e_m^- (n) \leftrightarrow U_m^- (z) = B_m (z) \; S (z) \qquad \text{III.4}$$

where

$$A_m (z) = \sum_{i=0}^{m} a_{i,m} \; z^{-i} \qquad \text{III.5}$$

$$B_m (z) = \sum_{i=1}^{m+1} b_{i,m} \; z^{-i} \qquad \text{III.6}$$

and S(z) is the z transform of the input signal, s(n).

It can be shown [19] that these filter polynomials, A_m (z) and B_m (z) satisfy the following recursive relation

$$A_{m+1} (z) = A_m (z) - k_m \, B_m (z) \qquad \text{III.7}$$

$$z \, B_{m+1} (z) = B_m (z) - k_m \, A_m (z) \qquad \text{III.8}$$

where k_m is the m^{th} k parameter.

Using equations III.7 and III.8 the forward and backward prediction error sequences satisfy

$$e^{+}_{m+1}(n) = e^{+}_{m}(n) - k_m e^{-}_{m}(n), \quad e^{+}_{0}(n) = s(n) \qquad \text{III.9}$$

$$e^{-}_{m+1}(n+1) = e^{-}_{m}(n) - k_m e^{+}_{m}(n), \quad e^{-}_{0}(n) = s(n-1) \qquad \text{III.10}$$

The analysis procedure consists of estimating k_m based on $e^{+}_{m}(n)$ and $e^{-}_{m}(n)$ then advancing to the next stage of the filter using equations III.9 and III.10.

From the analysis, estimates of the M reflection coefficients k_m m=0, 1, ..., M-1 and the final prediction errors $e^{+}_{M}(n)$ and $e^{-}_{M}(n)$ are obtained.

Estimating k_m (Block Analysis Method)

The standard procedure for estimating k_m was to assume that the input waveform s(n) was stationary over an interval of N samples and estimate k_m based on the short-term autocorrelation of $e^{+}_{m}(n)$ with $e^{-}_{m}(n)$ [2], [19]. Using this approach each k_m was calculated as

$$k_m = \frac{\sum_{n=1}^{N} e^{+}_{m}(n)\, e^{-}_{m}(n)}{\left[\sum_{n=1}^{N} e^{+}_{m}(n)^2 \quad \sum_{n=1}^{N} e^{-}_{m}(n)^2\right]^{1/2}}$$

For comparison purposes this approach will be defined as the block analysis method since one set of k-parameters are estimated for a block of N sample points.

The next section develops a method for estimating new values for the k-parameters at each sample, n.

Estimating k_m Single Sample Analysis Method, STREAK

An estimate of each k-parameter at each sample point can be obtained by calculating that value for estimating k_m such that the sum of the squares of e^+_{m+1} (n) and e^-_{m+1} (n+1) is minimized for each n. That is, referring to Figure III.1, a logical criterion for estimation is that the energies at the next stage of the filter should be minimized. Thus, a value for k_m is calculated to minimize $[e^+_{m+1}(n)]^2 + [e^-_{m+1}(n+1)]^2$.

Therefore, define the loss function, L_m at the mth stage of the filter as

$$L_m = [e^+_{m+1}(n)]^2 + [e^-_{m+1}(n+1)]^2 \qquad \text{III.12}$$

L_m is to be minimized with respect to k_m. Substituting the expressions for e^+_{m+1} (n) and e^-_{m+1} (n+1) from equations III.9 and III.10 gives

$$L_m = [e^+_m(n) - k_m e^-_m(n)]^2 + [e_m(n) - k_m e^+_m(n)]^2 \qquad \text{III.13}$$

L_m is minimized by equating to zero the derivative of L_m with respect to k_m and solving for k_m. Thus

$$\frac{d\,L_m}{d\,k_m} = 0 = -2[e^+_m(n) - k_m e^-_m(n)]e^-_m(n) - 2[e^-_m(n) - k_m e^+_m(n)]e^+_m(n)$$

III.14

or

$$k_m(n) = \frac{2\, e^+_m(n)\, e^-_m(n)}{[e^+_m(n)]^2 + [e^-_m(n)]^2} \qquad m=0,\ 1,\ \ldots,\ M-1 \qquad \text{III.15}$$

Equation III.15 along with the updating equations III.9 and III.10 define the complete analysis procedure.

As each new sample, s(n) enters the inverse filter, new values for each k-parameter are estimated. Thus, from e_0^+ (n) and e_0^- (n), k_0 (n) is formed using equation III.15. Next e_1^+ (n) and e_1^- (n+1) are computed using equations III.9 and III.10. The analysis then advances to the next section of the filter and everything repeats.

Examining the analysis equations shows that the total number of arithmetic operations for each sample consists of five multiples, three adds, and one divide per section, times M sections. M+1 storage locations are required for the e_m^+ (n) and e_m^- (n) arrays and M locations for the k_m (n) array.

The Relation Between the k-parameters and the Forward and Backward Prediction Error Sequences

From equations III.9, III.10 and III.15 a recursion relating the m^{th} k-parameter to the $m+1^{st}$ forward and backward prediction error can be obtained which is identical to that using the block analysis method. Thus substituting the expression for k_m (n) given in equation II.15 into the forward and backward prediction error sequences, equations III.9 and III.10 gives

$$e_{m+1}^+ (n) = e_m^+ (n) - \frac{2\, e_m^+ (n) (e_m (n))^2}{(e_m^+ (n))^2 + (e_m^- (n))^2} \qquad \text{III.16}$$

$$e_{m+1}^- (n+1) = e_m^-(n) - \frac{2 (e_m^+(n))^2\, e_m^-(n)}{(e_m^+(n))^2 + (e_m^-(n))^2} \qquad \text{III.17}$$

Simplifying gives

$$e^{+}_{m+1}\ (n)\ =\ e^{+}_{m}\ (n)\ \frac{(e^{+}_{m}\ (n))^{2}\ -\ (e^{-}_{m}\ (n))^{2}}{(e^{+}_{m}\ (n))^{2}\ +\ (e^{-}_{m}\ (n))^{2}} \qquad \text{III.18}$$

$$e^{-}_{m+1}\ (n+1)\ =\ -e^{-}_{m}\ (n)\ \frac{(e^{+}_{m}\ (n))^{2}\ -\ (e^{-}_{m}\ (n))^{2}}{(e^{+}_{m}\ (n))^{2}\ +\ (e^{-}_{m}\ (n))^{2}} \qquad \text{III.19}$$

but

$$1-k^{2}_{m}\ (n)\ =\ 1\ -\left[\frac{2e^{+}_{m}\ (n)\ e^{-}_{m}\ (n)}{(e^{+}_{m}\ (n))^{2}\ +\ e^{-}_{m}\ (n))^{2}}\right]^{2} \qquad \text{III.20}$$

$$=\ \left[\frac{(e^{+}_{m}\ (n))^{2}\ -\ e^{-}_{m}\ (n))^{2}}{(e^{+}_{m}\ (n))^{2}\ +\ (e^{-}_{m}\ (n))^{2}}\right]^{2} \qquad \text{III.21}$$

thus

$$e^{+}_{m+1}\ (n)\ =\ e^{+}_{m}\ (n)\ (1\ -\ k^{2}_{m}\ (n))^{\frac{1}{2}} \qquad \text{III.22}$$

$$e^{-}_{m+1}\ (n+1)\ =\ -e^{-}_{m}\ (n)\ (1\ -\ k^{2}_{m}\ (n))^{\frac{1}{2}} \qquad \text{III.23}$$

Equation III.22 shows that the energy of the forward prediction error, $(e^{+}_{m}\ (n))^{2}$ will be a monotonically decreasing function of m.

Note also that if

$$|e^{+}_{m}(n)|\ =\ |e^{-}_{m}\ (n)|$$

then

$$|k_{m}\ (n)|\ =\ 1$$

and

$$e^{+}_{m+1}\ (n)\ =\ e^{-}_{m+1}\ (n+1)\ =\ 0$$

For this situation, $e_m^+ (n)$ and $e_m^- (n)$ are predicted exactly from $e_m^- (n)$ and $e_m^+ (n)$ respectively and thus the prediction errors $e_{m+1}^+ (n)$ and $e_{m+1}^- (n+1)$ must be zero.

Likewise if

$$k_m (n) = 0$$

then

$$e_{m+1}^+ (n) = e_m^+ (n)$$

$$e_{m+1}^- (n+1) = e_m^- (n)$$

Speech Synthesis Using STREAK

The original waveform $s_n = e_0^+ (n)$ can be reconstructed from the k-parameters, $k_m (n)$, m=0, 1, ... M-1, and the final forward prediction error, $e_M^+(n)$. The synthesis equations are determined by expressing $e_m^+ (n)$ equation III.9 in terms of $e_{m+1}^+ (n)$, $e_m^- (n)$ and $k_m (n)$. The resulting synthesis filter (commonly called the two multiply lattice filter [20]) is defined as

$$e_m^+ (n) = e_{m+1}^+ (n) + k_m (n)\, e_m^-(n) \qquad \text{III.24}$$

$$e_{m+1}^- (n+1) = e_m^- (n) - k_m (n)\, e_m^+ (n) \qquad \text{III.25}$$

$$m = M-1, \ldots 0.$$

with $e_M^+ (n)$ given and $e_0^+ (n) = s_n$.

Note that since STREAK calculates new values for k-parameters at each sample point, n, the synthesis filter can also be updated for each new sample. Thus the filter characteristics can vary at the sampling rate. This approach represents a somewhat radical departure from the standard procedure of supplying the synthesizer with just one set per pitch period or per anlaysis frame. To do this of course demands that new k-parameters be transmitted at the sampling rate thus requiring an enormous channel bandwidth. However, preliminary experiments show that a pitch excited synthesizer updated continuously with k-parameters estimated at the sampling rate, generates synthetic speech having a more natural quality than that obtained using standard block analysis-synthesis methods such as described in Section I. Currently methods are being investigated for reducing the bandwidth but retaining this quality.

A Geometrical Interpretation of k_m

From the fact that

$$a^2 + b^2 \leq 2ab \qquad \text{III.26}$$

it can be seen that

$$|k_m| \leq 1 \qquad \text{III.27}$$

Geometrically by defining two lines ℓ_1 and ℓ_2 extending from the origin having coordinates $[e_m^+ (n), e_m^- (n)]$ and $[e_m^- (n), e_m^+ (n)]$ respectively, it is shown that k_m equals the cosine of the angle between ℓ_1 and ℓ_2 (see Figure III.2). Thus, by definition

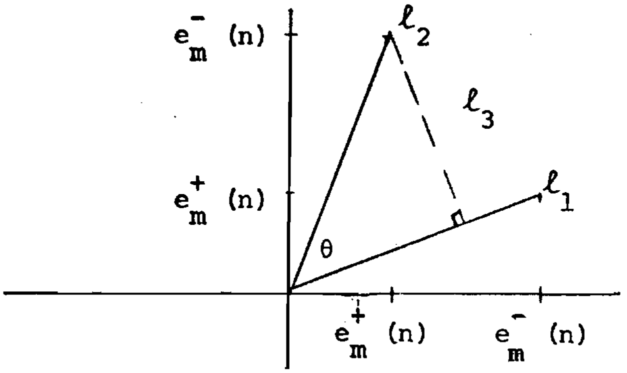
$$\cos \theta = \frac{\ell_1 \cdot \ell_2}{|\ell_1| \; |\ell_2|} \qquad \text{III.28}$$

but

$$\ell_1 \cdot \ell_2 = 2 \, e_m^+ (n) \, e_m^- (n) \qquad \text{III.29}$$

and

$$|\ell_1| = |\ell_2| = \{[e_m^+ (n)]^2 + [e^- (n)]^2\}^{\frac{1}{2}} \qquad \text{III.30}$$



Geometrical Interpretation of k_m

Figure III.2

Thus, $k_m = \cos\theta$

It should also be noted that ℓ_2 minus the projection of ℓ_2 onto ℓ_1 defines a vector ℓ_3 having coordinates $[e_{m+1}^+(n), e_{m+1}^-(n+1)]$. It is, of course, the length of this vector which is minimized, by defining k_m as $\cos\theta$.

Using STREAK for Pitch Detection

A well-established technique for pitch detection consists of performing an autocorrelation on the error signal obtained from the inverse filter output [7], [12], [18]. The idea behind this method is that if the all-pole model accurately represents the vocal tract transfer function and the radiation and glottal volume flow effects, then the output of the inverse filter should resemble an impulse-like driving function having a period equal to the pitch for voiced speech. The autocorrelation of this error sequence should therefore exhibit a large spike located at a distance from the origin equal to the pitch period. This method, however, will occasionally fail, generating a dominant peak at a distance other than the true pitch period. In these cases, the inverse filter has not done an adequate job of removing everything. Or, stated another way, the curve fit was insufficient.

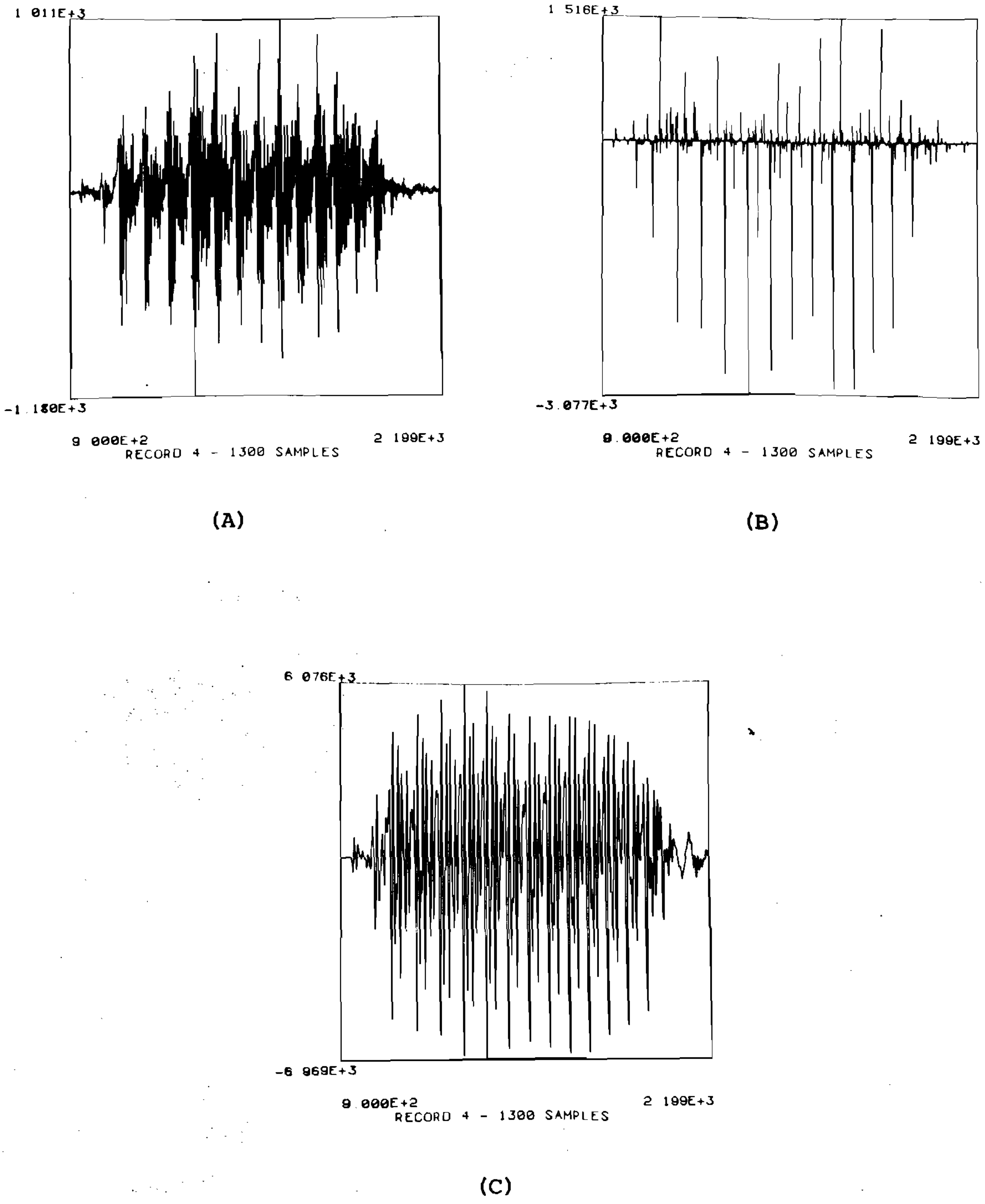
A solution to this problem is to generate the error sequence using STREAK rather than from conventional block analysis methods. This approach results in a superior least squares curve fit since the fit is applied on a sample by sample basis rather than over an entire block of N samples.

To illustrate the improvement in inverse filtering using STREAK over the block analysis method, the next section compares the forward prediction error sequences for various phonemes using the two methods.

Comparison of Inverse Filter Outputs

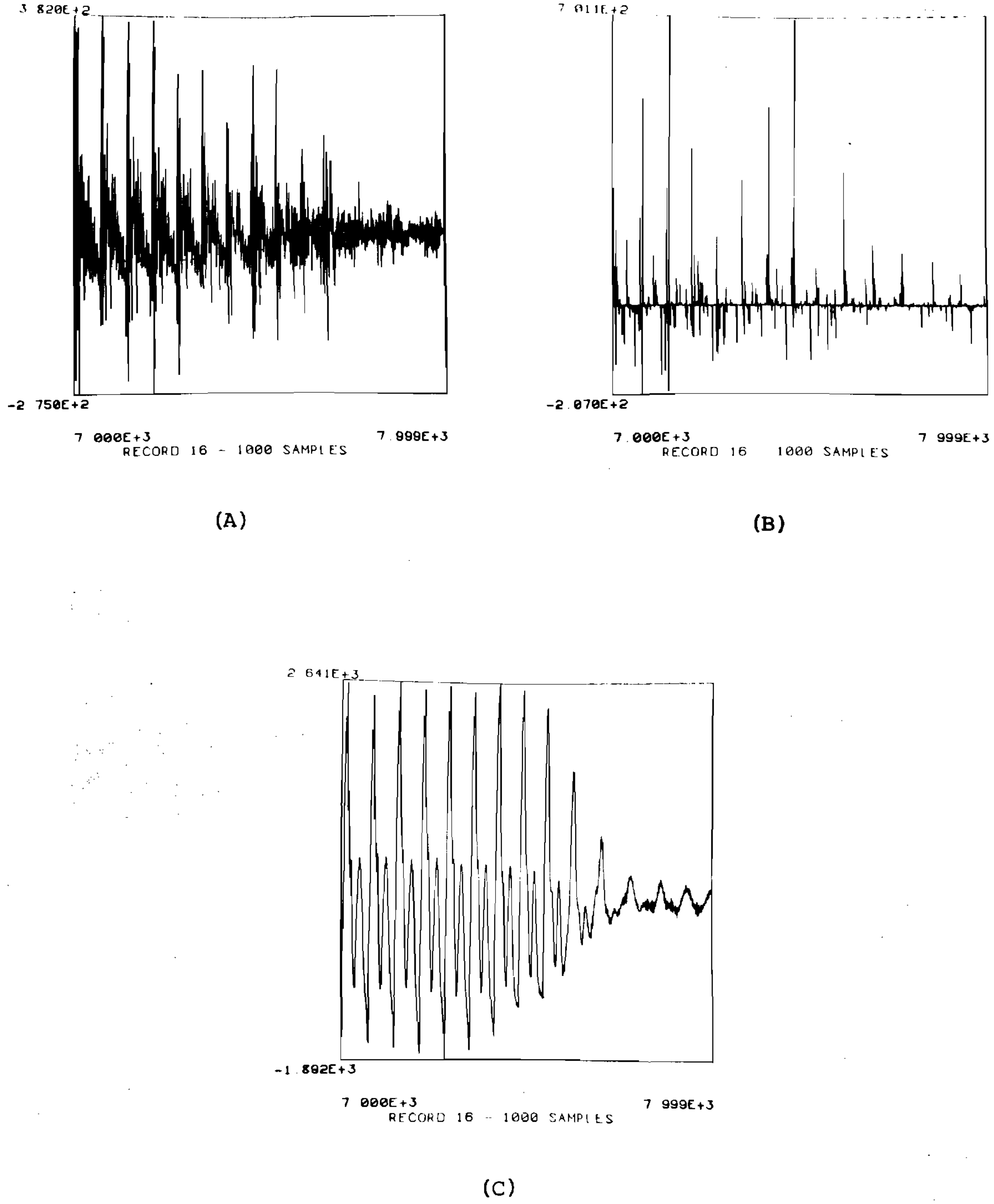
The forward prediction error sequence using the block analysis method was generated using twelve k-parameters estimated from a 20 ms Hamming window. The window was advanced in 20 ms steps. The sampling rate was 8k Hz. The data was not preemphasized. Twelve k-parameters were also used in generating the error sequence using the STREAK algorithm.

For each comparison three figures are presented: the original waveform, the error sequence using the block analysis, and the error sequence using STREAK. In Figure III.3 the entire work "oak" as spoken by a low-pitched male in the context "Oak is strong . . ." is displayed in Figure III.3 (c), along with the prediction error



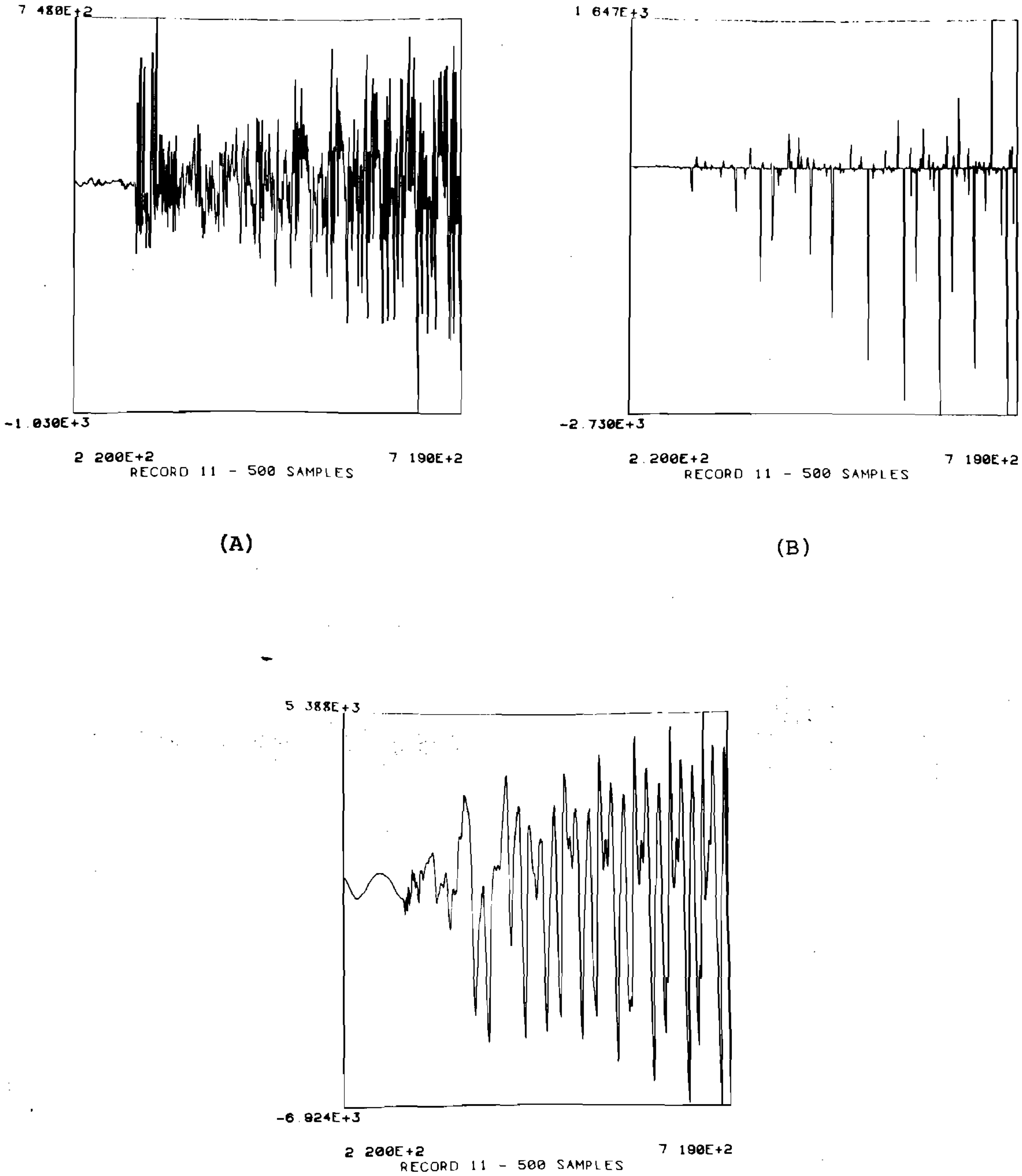
Comparison of Inverse Filter Outputs for the Word "Oak". (A) Block Analysis Error Sequence, (B) STREAK Error Sequence, (C) Original

Fiqure III.3



Comparison of Inverse Filter Outputs for the Nasal /n/ in "Friends". (A) Block Analysis Error Sequence, (B) STREAK Error Sequence, (C) Original

Figure III.4



(A)

(B)

(C)

Comparison of Inverse Filter Outputs for the Voiced Stop /b/ Followed by the Semivowel /r/ in "Break". (A) Block Analysis Error Sequence, (B) STREAK error sequence, (C) Original

Figure III.5

using the block analysis method in Figure III.3 (A) and using STREAK in Figure III.3 (B). Note that the error sequence using STREAK exhibits a considerably flatter spectral character than the block analysis error sequence.

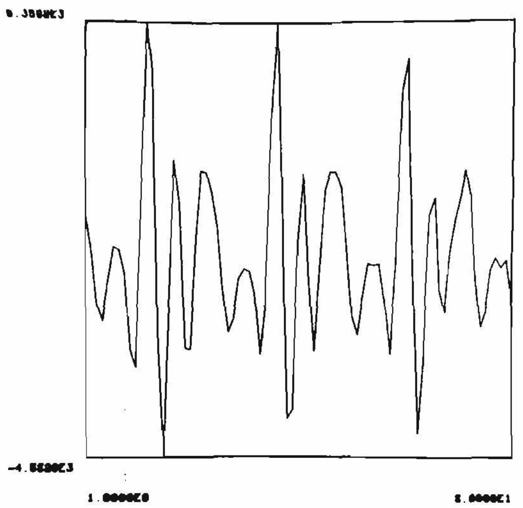
In Figure III.4 (C) the nasal /n/ from the word "friends" spoken in the context "Thieves who rob friends deserve jail" is displayed. Figure III.4 (A) displays the block analysis error sequence and Figure III.4 (B) displays the error sequence using STREAK. Note the absence of periodicity at the trailing end of the nasal in Figure III.4 (A), whereas with STREAK, the pitch period is clearly evident.

Finally, Figure III.5 (C) displays the voiced stop /b/ followed by the semivowel /r/ in the work "break" spoken in the context "Don't break the glass". Again, Figure III.5 (A) displays the prediction error using block analysis and Figure III.5 (B) using STREAK. Examining the error waveform during both the /b/ and /r/ sections shows again that STREAK produces a spectrally flatter error sequence, with the fundamental more clearly evident.

These three examples were taken from a large group of sentences spoken by both male and female speakers. For every sentence analyzed, the two error sequences exhibited the same general characteristics with the STREAK algorithm always producing the superior curve fit.

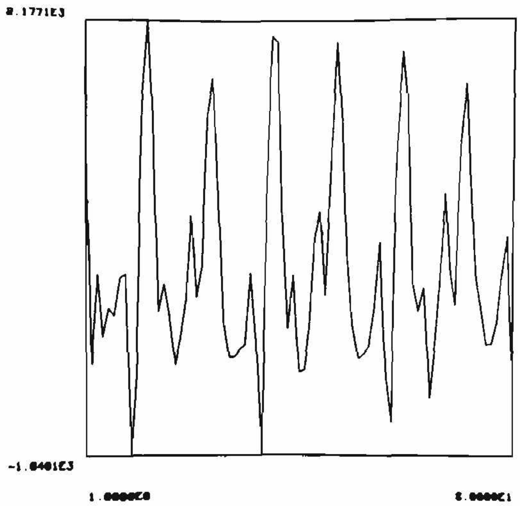
Implementing STREAK into a Pitch Tracking Algorithm

In Chapter I a pitch detection method was described in which a new sequence was formed by prefiltering and down-sampling the original speech waveform. (See Figure I.3). The pitch period was estimated by autocorrelating the error sequence generated from the smoothed down-sampled waveform, and locating the distance of the largest positive peak. STREAK can be incorporated into this algorithm by simply replacing that procedure for estimating of the inverse filter and error sequence generation using the block analysis method, with the error sequence generator using STREAK. All other procedures are left unchanged. An example comparing the results of the two methods is shown in Figure III.6 through III.10. Figure III.6 shows the smoothed, down-sampled waveform from which the pitch estimate is to be determined. This speech sample is from the phoneme /o/ in "four". It was obtained by low pass filtering an 8 kHz sampled waveform at 750 Hz and down-sampling by four. Thus, the eighty samples represent a 40 ms window. The error sequence using the block analysis approach is shown in Figure III.7. This sequence was generated by a four pole inverse filter using predictor coefficients estimated from the eighty sample window. Figure III.8 shows the error sequence generated by STREAK from the same data using a fourth order filter. Figures III.9 and III.10 show the auto-correlations of each of these error sequences. Note that a pitch doubling error results when the block analysis is used but that the correct pitch period of 25 will be chosen when the STREAK analysis is used.



Smoothed Down-Sampled Waveform

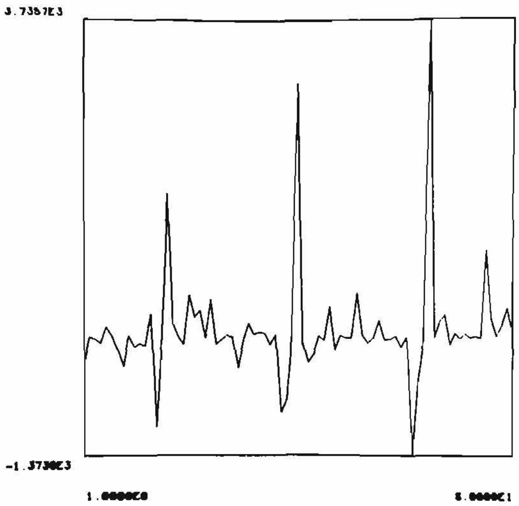
Figure III.6



Error sequence using

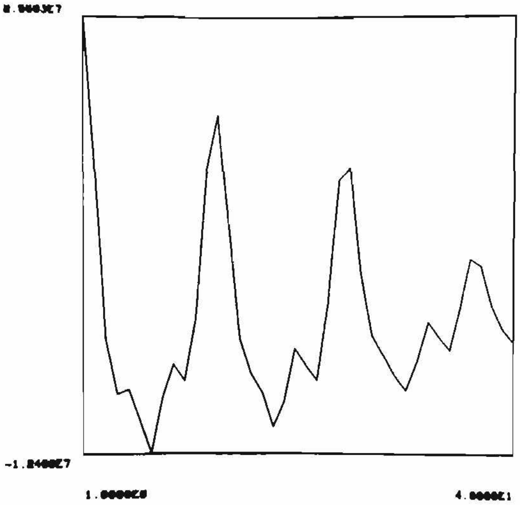
Block Analysis

Figure III.7



Error Sequence using STREAK

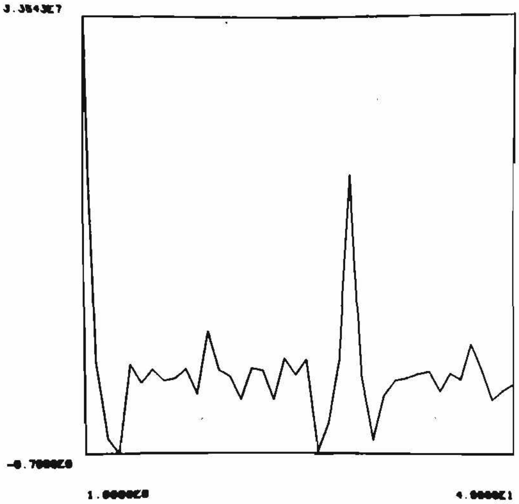
Figure III.8



Autocorrelation of Block Analysis

Error Sequence

Figure III.9



Autocorrelation of STREAK Analysis

Error Sequence

Figure III.10

SUMMARY AND CONCLUSIONS

A technique for recursively estimating the k-parameters of the Linear Predictive Coding inverse filter has been developed. The k-parameters are estimated directly from the lattice form of the inverse filter. The criterion for estimation was that a value for k_m be calculated so as to minimize the sum of the squares of the $m + 1^{st}$ forward and backward prediction error sequences. New estimates of each k-parameter of calculated at each sample point.

It was shown that the least squares curve fit using this method was superior to that using the block analysis method and therefore that this method may improve pitch detection schemes based upon the autocorrelation of the inverse filter output.

REFERENCES

1. B. S. Atal and S. L. Hanauer, "Speech Analysis by Linear Prediction of the Speech Wave", The Journal of the Acoustical Society of America, Volume 50, pp. 637-655, 1971.

2. J. D. Markel, A. H. Gray, Jr., and H. Wakida, Linear Prediction of Speech-Theory and Practice, Speech Comm. Res. Lab., Santa Barbara, California SCRL Monograph 10, September 1973.

3. J. I. Makhoul and J. J. Wolf, "Linear Prediction and the Spectral Analysis of Speech", Bolt, Beranek, and Newman, Inc., Cambridge, Mass., BBN Report 2304, August 1972.

4. John Makhoul and R. Viswanathan, "Quantization Properties of Transmission Parameters in Linear Predictive Systems", Bolt, Beranek, and Newman, Inc., Submitted for publication in the IEEE Trans. Acoust., Speech and Signal Processing.

5. J. D. Markel and A. H. Gray, Jr., "Fixed-Point Truncation Arithmetic Implementation of a Linear Prediction Autocorrelation Vocoder", IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-22, No. 4, August 1974.

6. D. T. Magill, "Adaptive Speech Compression for Packet Communication Systems", Network Speech Compression Note 9, Stanford Research Institute, December 3, 1974.

7. S. F. Boll, "A Priori Digital Speech Analysis", UTEC-CSc.-73-123. Computer Science, University of Utah, March 1974.

8. E. Mitsui, T. Nakajima, T. Suzuki, and H. Omura, "An Adaptive Method for Speech Analysis Based on Kalman Filtering Theory", Bull. Electro. Tech. Lab., Vol. 36, No. 3 Tokyo, Japan 1972.

9. N. Levinson, "The Wiener RMS (root mean square) Error Criterion in Filter Design and Prediction", J. Math Phys., Vol. 25, pp. 261-278, 1947; also N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series, Cambridge: MIT Press 1966, Appendix B.

10. F. Itakura and S. Saito, "On the Optimum Quantization of Feature Parameters in the PARCOR Speech Synthesizer", Conference Record of the IEEE 1972 Conference on Speech Communication and Processing, paper L4, New York, 1972.

11. J. D. Markel and A. H. Gray, Jr., "On Autocorrelation Equations with Application to Speech Analysis", IEEE Trans. on Audio Electro Acoust., Vol. AU-21, No. 2, pp. 69-79, April, 1973.

12. J. D. Markel, "The SIFT Algorithm for Fundamental Frequency Estimation", IEEE Trans. on Audio Electro Acoust., Vol. AU-20, No. 5, pp. 367-377.

13. A. V. Oppenheim, "Speech Analysis and Synthesis Based on Homomorphic Filtering", J. Acoust. Soc. Am., Vol. 45, pp. 458-465, February 1969.

14. N. J. Miller, "Removal of Noise from a Voice Signal by Synthesis", UTEC-CSc-74-013, Computer Science, University of Utah, May 1973.

15. C. K. Un and D. T. Magill, "Residual Excited Linear Prediction Vocoder", Stanford Research Institute, March 20, 1974.

16. W. A. Blankinship, "Note on Computing Autocorrelaions", IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-22, No. 1, pp. 76-77, February 1974.

17. L. L. Pfeifer, "Multiplication Reduction in Short-Term Autocorrelation", IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No. 6, pp. 556-557, December 1973.

18. F. Itakura and S. Saito, "Analysis Synthesis Telephony Based Upon the Maximum Likelihood Method", Reprints of the 6th International Congress of Acoustics, Y. Zonasi, Ed.; Tokyo, Japan, Rep. C-5-5, August 21-28, 1968.

19. F. Itakura, et al., "An Audio Response Unit Based on Partial Autocorrelation", IEEE Trans. on Communications, Vol. COM-20, No. 4, pp. 792-797, August 1972.

20. A. H. Gray Jr. and J. D. Markel, "Digital Lattice Filter Synthesis", IEEE Trans. on Audio and Electroacoustics, Vol. Au-21, No. 6, pp. 491-500, December 1974.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
UTEC-CSc-74-151

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
SELECTED METHODS FOR IMPROVING SYNTHESIS SPEECH QUALITY USING LINEAR PREDICTIVE CODING: SYSTEM DESCRIPTION, COEFFICIENT SMOOTHING AND STREAK

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Steven Frank Boll

8. CONTRACT OR GRANT NUMBER(s)
DAHC15-73-C-0363

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer Science Department
University of Utah
Salt Lake City, Utah 84112

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ARPA Order #2477

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, Virginia 22209

12. REPORT DATE
November 1974

13. NUMBER OF PAGES
64

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

linear predictive coding, speech compression, vocoder, reflection coefficient smoothing, a priori least squares, lattice inverse filter, prediction error, pitch detection, autocorrelation, speech synthesis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report develops two generalizations of the standard Linear Predictive Coding (LPC) implementation of a narrow band speech compression system. The purpose of each method is to improve the speech quality that is available from a standard LPC system. Attention is focused primarily upon the pitch excited system and therefore, the improvements considered focus upon the improved estimation of the reflection coefficients and the pitch period. Specifically, a parameter filtering algorithm is developed for dynamically

20. Abstract (Continued)

smoothing the reflection coefficients to both increase naturalness in synthetic speech as well as eliminate the possibility of synthesis filter instabilities. Secondly, a new method for calculating the k-parameters of an LPC inverse filtering algorithm is developed, STREAK. New values for each k-parameter are calculated at each sample point directly using the lattice formulation of the inverse filter model. It is shown this technique can be used to improve a pitch detection scheme based upon the autocorrelation of inverse filter output sequence.