Pitch Prediction Filters in Speech Coding

RAVI P. RAMACHANDRAN AND PETER KABAL

Abstract—**Prediction error filters which combine short-time prediction (formant prediction) with long-time prediction (pitch prediction) in a cascade connection are examined. A number of different solution methods (autocorrelation, covariance, Burg) and implementations (transversal and lattice) are considered. It is found that the F-P cascade (formant filter before the pitch filter) outperforms the P-F cascade for both transversal- and lattice-structured predictors. The performances of the transversal and lattice forms are similar. The solution method that yields a transversal structure requires a stability test and, if necessary, a consequent stabilization. The lattice form allows for a solution method which ensures a stable synthesis filter. Simplified solution methods are shown to be applicable for the pitch filter (multitap case) in an F-P cascade. Furthermore, new methods to estimate the appropriate pitch lag for a pitch filter are proposed for both transversal and lattice structures. These methods perform essentially as well as an exhaustive search in an F-P cascade. Finally, the two cascade forms are implemented as part of an APC coder to evaluate their relative subjective performance.**

I. Introduction

In this paper, speech coder configurations which use two nonrecursive prediction error filters to process the incoming speech signal are examined. Conventionally, the prediction is carried out as a cascade of two separate filtering operations. The first filter, referred to here as the formant filter, removes near-sample redundancies. The second is termed the pitch filter and acts on distant-sample waveform similarities. The resulting residual signal is quantized and coded for transmission. In an adaptive predictive coder (APC), these predictors are placed in a feedback loop around the quantizer. An additional quantization noise shaping filter can be employed to reduce the perceived distortion in the decoded speech [1], [2]. An alternative description of an APC coder uses an open-loop predictor configuration and a noise feedback filter [3]. A block diagram of such a configuration is shown in Fig. 1. This type of open-loop arrangement is also used in code-excited linear prediction (CELP) [4]. In CELP, the coding is accomplished by selecting a waveform from a given repertoire of waveforms. The selection process uses an analysis-by-synthesis strategy. Conceptually, each candidate waveform is passed through the synthesis filters to find that one which produces the best quality speech. Noise shaping is accomplished by including a frequency weighting in the error criterion which is used to choose the best waveform.

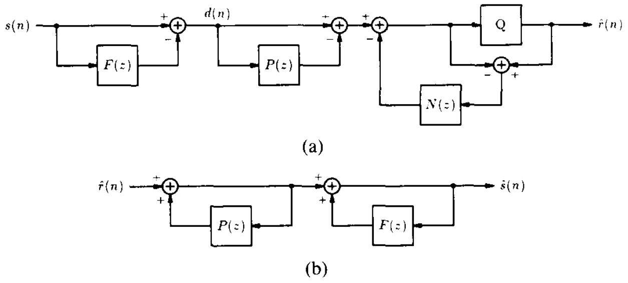


Fig. 1. Block diagram of an APC coder with noise feedback. (a) Analysis phase. (b) Synthesis phase.

In both APC and CELP, the residual signal or the selected codeword (after scaling by the gain factor) is passed through a pitch synthesis and a formant synthesis filter to reproduce the decoded speech. The filtering in the synthesis phase can be viewed in the frequency domain as first inserting the fine pitch structure and then shaping the spectral envelope (formant structure).

The analysis to determine the predictor coefficients is carried out frame by frame. The filter coefficients are then coded for transmission. The quantization of these coefficients is outside the scope of the present study. These parameters, along with the quantized excitation information, are used by the decoder to reconstruct the speech. The frame update rate is chosen to be slow enough to keep the transmission rate required small, yet fast enough to allow the speech segment under analysis to be adequately described by a set of constant parameters. Depending on the application, the effective frame size usually corresponds to time intervals between 5 and 20 ms.

The aim of this paper is to study predictors which incorporate both short-time and long-time prediction. The effect of the ordering of the prediction filters in the cascade connection is considered. The filters will be implemented in both lattice and transversal forms. In addition, methods to determine the lag used for the pitch filter will be derived. The two predictor configurations incorporating the transversal and lattice solutions are tested as part of an APC coder that is equivalent to the one shown in Fig. 1. This allows us to access the relative perceptual quality of the decoded speech that results from the use of different configurations and solutions.

The next section will introduce the different configurations for formant and pitch filters. This is followed by an

Manuscript received June 10, 1987; revised August 30, 1988. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

R. P. Ramachandran is with the Department of Electrical Engineering, McGill University, Montreal, P.Q., Canada H3A 2A7.

P. Kabal is with the Department of Electrical Engineering, McGill University, Montreal, P.Q., Canada H3A 2A7 and INRS-Telecommunications, Université du Québec, Verdun, P.Q., Canada H3E 1H6.

IEEE Log Number 8826113.

0096-3518/89/0400-0467$01.00 © 1989 IEEE

analysis of a prediction error filter which uses general delays. This general structure subsumes both formant and pitch filters and allows for both autocorrelation and covariance analyses. The following section makes the analysis specific to pitch filters. A comparison of the techniques is given in Section V. Then, the stability properties of the synthesis filters are examined for different configurations. Section VII examines means to determine an appropriate lag for the pitch filter. Finally, Section VIII discusses the relative performance of the different options when implemented as part of a speech coder.

II. Formant and Pitch Predictors

The conventional formant predictor has a transfer function

$$F(z) = \sum_{k=1}^{N_f} a_k z^{-k}. \qquad (1)$$

The order N_f is typically between 8 and 16. The system function of the noise feedback filter is usually related to that of the formant predictor. One choice is to let $N(z) = F(z/\alpha)$ where $0 < \alpha < 1$. This reduces the perceptual distortion of the output speech by improving the signal-to-noise ratio (SNR) in regions where the spectral level is low. However, this improvement comes at the expense of decreased SNR in the formant regions [1]. At the receiver, the formant synthesis filter has a transfer function $H_F(z) = 1/(1 - F(z))$.

The pitch predictor has a small number of taps N_p. The delays associated with these taps are bunched around the pitch lag value. The system function for a transversal form pitch predictor is

$$P(z) = \begin{cases} \beta_1 z^{-M} & \text{1 tap} \\ \beta_1 z^{-M} + \beta_2 z^{-(M+1)} & \text{2 taps} \\ \beta_1 z^{-M} + \beta_2 z^{-(M+1)} + \beta_3 z^{-(M+2)} & \text{3 taps.} \end{cases} \qquad (2)$$

The pitch lag M is usually updated along with the coefficients. The pitch synthesis filter has a system function $H_P(z) = 1/(1 - P(z))$.

The conventional predictor configuration uses a cascade of a formant predictor and a pitch predictor, referred to here as an F–P cascade. This structure can be motivated from a standard speech production model, which decouples the quasi-periodic source (the vocal folds) from the vocal tract filter.

In the context of speech coding, pitch predictors are most useful during voiced speech since voiced speech is characterized as a quasi-periodic signal with considerable correlation between samples separated by a pitch period. In the F–P cascade, the formant predictor removes the near-sample correlations to a large extent. The resulting formant predicted signal is a low-density quasi-periodic signal consisting mainly of pitch spikes. The pitch predictor acts on this residual signal. If the pitch period is an integral number of samples, a one-tap pitch predictor can remove pitch period correlations. For nonintegral pitch periods, a multitap pitch predictor serves somewhat as an interpolating filter for the removal of these distant-sample correlations.

The formulation used for the pitch filter is such that it removes long-term correlations, whether due to actual pitch excitation or not. The use of the term ''pitch filter'' is somewhat misleading in describing the action of this filter for unvoiced speech and even to some extent for voiced speech. However, for ease of reference and to keep with past nomenclature, in this paper the long delay filter will be referred to as the pitch filter, and the corresponding lag value will be referred to as the pitch lag.

The cascade connection of the predictors can also have the pitch predictor precede the formant predictor (referred to as a P–F cascade). In the P–F cascade, the pitch filter is chosen to reduce long-term correlations such as those due to quasi-periodic input signals. The remaining near-sample correlations are handled by the formant predictor. The filter coefficients for the two filters in the cascade are determined in a sequential fashion. The coefficients of the first filter are found, and then the coefficients of the second filter are determined. The sequential solution process gives different results for the F–P cascade and the P–F cascade connections. In addition, the initial conditions at the time of coefficient update are different for the F–P and P–F connections. This would account for differences even when the two forms use the same coefficients for their constituent parts.

The individual filters can be implemented in either transversal or lattice form. As shown later, the various implementations and solution methods give rise to systems with differing performance and differing stability properties.

III. Predictors with General Delays

In this section, general formulations for determining the predictor coefficients for both formant and pitch predictors in transversal or lattice form are developed. Later, these formulations are made more specific for the case of pitch filters.

A. Transversal Implementation

A model for calculating the predictor coefficients for a transversal implementation is shown in Fig. 2. The input signal $x(n)$ is multiplied by a data window $w_d(n)$ to give $x_w(n)$. The signal $x_w(n)$ is predicted from a set of its previous samples to form an error signal

$$e(n) = x_w(n) - \sum_{k=1}^{L} c_k x_w(n - M_k). \qquad (3)$$

The values M_k are arbitrary but distinct integers corresponding to delays of the signal $x_w(n)$. The final step is to multiply the error signal by an error window $w_e(n)$ to

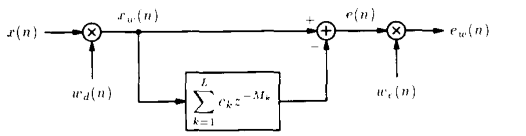


Fig. 2. Analysis model for transversal predictors.

obtain a windowed error signal $e_w(n)$ where

$$\begin{aligned} e_w(n) &= w_e(n)\, e(n) \\ &= w_e(n)\, x_w(n) - w_e(n) \sum_{k=1}^{L} c_k x_w(n - M_k). \end{aligned} \tag{4}$$

The squared error is defined by

$$\epsilon^2 = \sum_{n=-\infty}^{\infty} e_w^2(n). \tag{5}$$

The coefficients c_k are computed by minimizing ϵ^2. This leads to a linear system of equations that can be written in matrix form ($\boldsymbol{\Phi c} = \boldsymbol{\alpha}$):

$$\begin{bmatrix} \phi(M_1, M_1) & \phi(M_1, M_2) & \cdots & \phi(M_1, M_L) \\ \phi(M_2, M_1) & \phi(M_2, M_2) & \cdots & \phi(M_2, M_L) \\ \vdots & \vdots & & \vdots \\ \phi(M_L, M_1) & \phi(M_L, M_2) & \cdots & \phi(M_L, M_L) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_L \end{bmatrix} = \begin{bmatrix} \phi(0, M_1) \\ \phi(0, M_2) \\ \vdots \\ \phi(0, M_L) \end{bmatrix} \tag{6}$$

where

$$\phi(i, j) = \sum_{n=-\infty}^{\infty} w_e^2(n)\, x_w(n - i)\, x_w(n - j). \tag{7}$$

For both formant and pitch predictors, the delays M_k are grouped. A formant predictor has a set of delays $M_k = k$ for $k = 1$ to N_f. A pitch predictor has a small number of delays $M_k = M + k$ for $k = 0$ to $N_p - 1$. Grouping the pitch taps reduces the amount of side information (which is sent to the decoder) needed to specify the delay values.

1) Autocorrelation Method: The autocorrelation method results if $w_e(n) = 1$ for all n. The data window $w_d(n)$ is typically time-limited (rectangular, Hamming, or other). An important consideration is the minimum-phase property of the prediction error filter $A(z) = 1 - \Sigma_{k=1}^{L} c_k z^{-M_k}$. If $A(z)$ is minimum phase, the corresponding synthesis filter $1/A(z)$ used at the decoder is stable. The autocorrelation method can be shown to give a minimum-phase formant filter [5]. In the case of pitch filters, the minimum-phase property does not hold in general. An exception occurs if the delays corresponding to the coefficients are uniformly spaced, i.e., $M_k = kM_1$. This point is discussed further in Appendix A.

The matrix $\boldsymbol{\Phi}$ is always symmetric and positive definite. It is also Toeplitz if the intercoefficient delays are equal. Depending on whether $\boldsymbol{\Phi}$ is Toeplitz or not, either the Levinson recursion[1] or the Cholesky decomposition can be used to solve the autocorrelation equations.

2) Covariance Method: The covariance method results if $w_d(n) = 1$ for all n and the error window is rectangular, $w_e(n) = 1$ for $0 \leq n \leq N - 1$. More general error windows in a covariance approach have been suggested by Singhal and Atal [6]. The covariance method does not guarantee that $A(z)$ is minimum phase, but it does minimize the error energy for each frame. The resulting system of equations (6) has the entries in $\boldsymbol{\Phi}$ and $\boldsymbol{\alpha}$ defined with no window applied to the input signal,

$$\phi(i, j) = \sum_{n=0}^{N-1} x(n - i)\, x(n - j) \tag{8}$$

where N is the frame length.

An alternative method is the modified covariance technique, which does guarantee a minimum-phase filter [1]. This technique works well for formant predictors and is used in many of the experiments described later. A discussion of the modified covariance approach and its relevance to the pitch prediction problem appears in Appendix B.

B. Lattice Implementation

Lattice methods have been employed in linear prediction and are useful in implementing a lattice-structured formant preditor [7].[2] Here, we consider more general lattice forms with only a subset of the stages actually performing a filtering operation. A lattice-structured predictor consisting of a total of P stages is an all-zero filter, as depicted in Fig. 3. The input signal is $x(n)$, and the final error signal is $e(n) = f_P(n)$. Stage i has a reflection coefficient K_i and forms both the forward residual $f_i(n)$ and backward residual $b_i(n)$. Reflection coefficients will be calculated for stages corresponding to one of the delay values M_k. Other stages will have zero-valued reflection coefficients. For these null stages, the forward error term propagates unaltered, and the backward error term is merely delayed. A lattice form filter will be minimum phase if all of the reflection coefficients have magnitudes which are smaller than one [7].

For those stages for which a reflection coefficient is calculated, the aim, in terms of maximizing the prediction

[1] A distinction is made between the general form of the Levinson recursion, which allows an arbitrary right-hand-side vector, and the Levinson–Durbin recursion, which applies if the elements of $\boldsymbol{\alpha}$ appear in the first row of $\boldsymbol{\Phi}$.

[2] For formant predictors, one can convert between transversal and lattice implementations with identical impulse responses. However, in a time-varying environment, they are not equivalent due to their different initial conditions at frame boundaries.

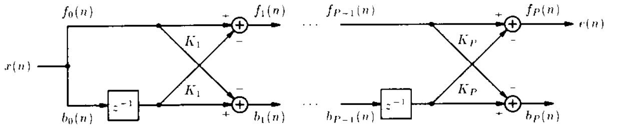


Fig. 3. Analysis model for lattice predictors.

gain alone, is to minimize the mean-square value of the forward residual. However, this criterion does not ensure that the magnitude of the resulting reflection coefficients is bounded by one and therefore does not ensure the stability of the corresponding synthesis filter. The Burg algorithm minimizes the sum of the mean-square values of the forward and backward residuals and ensures the stability of the synthesis filter. It also has the desirable property of guaranteeing that the mean-square value of the forward residual is nonincreasing across each stage of the lattice.

For the Burg method, the reflection coefficient K_i is calculated as

$$K_i = \frac{2C_{i-1}}{F_{i-1} + B_{i-1}} \tag{9}$$

where

$$C_i = \sum_{n=0}^{N-1} f_i(n)\, b_i(n-1), \quad F_i = \sum_{n=0}^{N-1} f_i^2(n),$$
$$B_i = \sum_{n=0}^{N-1} b_i^2(n-1), \tag{10}$$

and N is the frame length. The mean-square value of the forward residual is reduced by the factor $(1 - K_i^2)$ across stage i. A computationally efficient procedure, termed the covariance-lattice method [7], calculates the reflection coefficients using (9), but expresses them in terms of the covariance of the input signal. With this rearrangement, the computational complexity becomes comparable to the conventional covariance method.

When applying the Burg formula, the lattice-structured prediction error filter (as in Fig. 3) is minimum phase even if some of the reflection coefficients are constrained to be zero. Note also that the lattice coefficients can be transformed to direct form (impulse response) coefficients, allowing for an alternative implementation of the filter in transversal form.

IV. Pitch Filter Analysis Method

This section discusses the analysis methods that are used to implement both transversal- and lattice-structured pitch prediction error filters. Assume for the moment that the value of the pitch filter lag M is chosen so as to maximize the prediction gain. A discussion of the problem of estimating M is postponed until a later section. The input to the pitch filter is $d(n)$ for an F-P cascade and $s(n)$ for a P-F cascade.

A. Transversal Pitch Filters

The autocorrelation and covariance methods can be used to determine the coefficients for a transversal-structured prediction error filter $1 - P(z)$. For the autocorrelation method, the input signal must be windowed. Conventionally, the window is of finite duration, and all the samples outside the range of the window are set to zero. The method is effective for formant predictors since the largest filter delay is usually small compared to the length of the window used. This ensures that the frame edge effects due to the zero-valued analysis samples preceding and following the window are small. In contrast to the formant predictor, the delays used for a pitch predictor are comparable to, or even larger than, the frame and window lengths. For a pitch filter, frame edge effects are no longer negligible. The problem is not solved by using windows that are longer than the largest delay of the pitch predictor since too much time-averaging greatly reduces the performance and, in addition, changes in the pitch lag are not adequately tracked. Experiments involving various window shapes (including windows dynamically adapted to the pitch lag) confirm that the performance of the resulting pitch predictors is poor. Furthermore, there is no guarantee that the synthesis filter $H_P(z)$ derived using the autocorrelation method is stable if the filter has more than a single tap.[3]

The covariance method yields high prediction gains, but may give unstable pitch synthesis filters. Specifically, for three-tap pitch predictors in an F-P cascade, the system of equations is

$$\begin{bmatrix} \phi(M, M) & \phi(M, M+1) & \phi(M, M+2) \\ \phi(M+1, M) & \phi(M+1, M+1) & \phi(M+1, M+2) \\ \phi(M+2, M) & \phi(M+2, M+1) & \phi(M+2, M+2) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \phi(0, M) \\ \phi(0, M+1) \\ \phi(0, M+2) \end{bmatrix} \tag{11}$$

where

$$\phi(i, j) = \sum_{n=0}^{N-1} d(n-i)\, d(n-j) \tag{12}$$

and $d(n)$ is the input signal to the pitch predictor. The matrix Φ is not Toeplitz in general, and the Cholesky decomposition can be used to solve the system of equations. For reasonable frame sizes and the small number of taps used for pitch filters, Φ can be modified to become Toeplitz with little loss in prediction gain. Then, the general

[3] In our limited experiments with pitch filters derived using an autocorrelation formulation, no instability was observed.

form of the Levinson recursion can be used to determine the predictor coefficients. Note that the Toeplitz nature of Φ does not guarantee that $H_P(z)$ is stable, but does allow for a more efficient solution of the system of equations. Stabilization schemes can be employed whenever $H_P(z)$ is found to be unstable. Stabilization of the pitch filter is simple to implement and is derived from a computationally efficient stability test [8]. The degradation in average pitch prediction gain due to stabilization has been found to be small for an F-P cascade [8].

B. Lattice Pitch Filters

The Burg method works well for a formant predictor. Here, the technique is used to develop a lattice implementation for a pitch predictor that is used in cascade connections. The basic motivation for using the Burg approach is that the corresponding synthesis filter is stable. Hence, no stability test or fix-up is necessary. Even though a lattice filter involves more computations per sample than its transversal counterpart, computational convenience is provided in the above context. The computation required for a stability test and the consequent stabilization is saved.

Given the value of M, the reflection coefficients K_i for $i = 1$ to $M - 1$ are set to zero. The first nonzero coefficient in the pitch filter is K_M. In the case of formant filters and one-coefficient pitch filters, the impulse response of a lattice prediction error filter has the same form as that for a transversal filter. However, two- and three-coefficient lattice pitch prediction error filters do not have the same transfer functions as the transversal structured filters. The transfer functions of lattice prediction error filters are given by

$$A(z) = \begin{cases} 1 - K_M z^{-M} & N_p = 1 \\ 1 + K_M K_{M+1} z^{-1} - K_M z^{-M} - K_{M+1} z^{-(M+1)} & N_p = 2 \\ 1 + (K_M K_{M+1} + K_{M+1} K_{M+2}) z^{-1} + K_M K_{M+1} z^{-2} - K_M z^{-M} & \\ \quad - (K_{M+1} + K_M K_{M+1} K_{M+2}) z^{-(M+1)} - K_{M+2} z^{-(M+2)} & N_p = 3. \end{cases} \tag{13}$$

The two- and three-coefficient lattice filters have terms in z^{-1} and z^{-2} which are absent in the corresponding transversal filters. Note, however, that in the case of the three-coefficient lattice filter, the reflection coefficients control the five nontrivial impulse response values, hence giving a configuration with only three degrees of freedom.

In a pitch filter, the pitch lag changes from frame to frame. This variation of the position of the nonzero lattice coefficients can be detrimental to the performance. Consider the case when the pitch lag increases from one frame to another. In the new frame, the backward residual in the lattice will have been filtered by both the old coefficients and the new coefficients. A remedy for this problem is to reset the backward residual to the delayed filter input signal at each frame boundary. In addition, it is beneficial to back up the filter at each frame boundary by $N_p - 1$ samples to allow a "warm-up" period before generating actual output samples.[4] This resets the memory in the filter to be the same as if the filter had been used for the infinite past. This strategy will be used in the implementations.

V. Comparison of Techniques

The various predictor configurations were tested using the analysis phase of a general speech coder, such as shown in Fig. 1(a), as a test bed. In comparing different configurations and algorithms involving a pitch filter, prediction gain will be used as the performance measure. The prediction gain is used to avoid tying down the results to a specific type of coder. The aim is to assess the extent to which the predictors remove redundancies by measuring the energy of the resulting residual. For a general predictor, the prediction gain is the ratio of the average energy at the input to that predictor to the average energy of the prediction residual. For the system shown in Fig. 1(a), the formant gain is

$$G_F = \frac{\sum_n s^2(n)}{\sum_n d^2(n)}. \tag{14}$$

A similar formula applies to the pitch gain. The overall prediction gain is the prediction gain for the cascade of the filters.

For the present results, the value of pitch lag M is chosen to be the one that gives the highest prediction gain. Although an exhaustive search for the best M is not computationally practical, this approach will provide some insight into the relative performance of the various configurations. Also, for the present, the pitch filter is not stabilized. The issue of stability is deferred to Section VI.

The conditions common to all experiments involve the use of a formant predictor with ten coefficients and a pitch predictor with one, two, or three coefficients. The input speech samples comprise six utterances, three spoken by a male and three spoken by a female. The speech database consists of high-quality recordings (low-pass filtered to just below 4 kHz) at a sampling frequency of 8 kHz. The relevant average prediction gains for each sentence were computed and converted to decibels. Since the relative ordering of the methods is more or less preserved for each sentence, the tables present averages across the sentences of these decibel values.

[4] This strategy is equivalent to converting the reflection coefficients to direct form coefficients [see (13)] and then implementing the predictor in transversal form.

A. Cascade Configurations

In comparing the F–P and P–F configurations, analysis is carried out for 80-sample frames (corresponding to 10 ms intervals). This somewhat rapid update allows for a higher prediction gain and tends to illustrate the differences between schemes more clearly. The range of pitch lags is set to cover the range for both male and female speakers. Minimum and maximum values for M of 20 and 120 are used.

Table I shows a comparison of several techniques for one-, two-, and three-tap pitch filters. The formant predictor is implemented in transversal form with the coefficients determined by using the modified convariance method. The pitch filter is implemented in transversal form (covariance method) or lattice form (Burg method).

In the case of the F-P cascade, the pitch lag is that which maximizes the pitch prediction gain and hence also the overall prediction gain. Only a single figure appears for the formant prediction gain since the formant filter is unaffected by the choice of pitch coefficients and pitch lag.

For the P–F cascade, there is more of an interaction between the pitch predictor and the formant predictor. Values are given for the case in which the pitch lag is chosen to maximize the prediction gain for the pitch filter alone and the case in which the pitch lag is chosen to maximize the overall prediction gain. The myopic view of choosing the pitch lag to maximize only the pitch prediction gain gives the situation in which the pitch predictor has a higher prediction gain than the formant predictor. This phenomenon has also been observed in [9]. The situation reverses when the pitch lag is chosen to maximize the overall prediction gain. It should be noted here that the search for the best lag to maximize the overall prediction gain is impractically complex. Note also that as the number of taps in the pitch predictor is increased, there is an increase in the pitch prediction gain at the expense of a decrease in the formant prediction gain.

Note that the F–P cascade consistently outperforms the P–F cascade in average overall prediction gain, and even more so for the myopic choice of pitch lag.[5]

There is only a small difference between the lattice implementation of a pitch predictor and a transversal implementation, in spite of these forms having different impulse responses. In fact, the lattice form for two and three-tap filters in the F–P cascade slightly outperforms the transversal form for the utterances spoken by females. Examination of the final residual formed after formant and pitch prediction verifies that the pitch pulses are effectively removed by a lattice pitch filter.

TABLE I
PREDICTION GAINS FOR FORMANT/PITCH PREDICTORS (80-SAMPLE FRAMES). THREE NUMBERS IN AN ENTRY REFER TO ONE-,TWO-, AND THREE-TAP PITCH FILTERS

method		formant gain dB	pitch gain dB	overall gain dB
F-P	transversal P	16.1	4.2 5.3 5.8	20.3 21.4 21.9
	lattice P		4.0 5.2 5.7	20.1 21.3 21.8
P-F	transversal $P^{(1)}$	8.4 8.2 7.6	10.1 11.4 11.9	18.5 19.6 19.5
	transversal $P^{(2)}$	13.7 11.5 11.2	6.1 8.9 9.4	19.8 20.4 20.6
	lattice $P^{(1)}$	9.1 7.3 6.5	9.5 11.8 13.5	18.6 19.1 20.0
	lattice $P^{(2)}$	13.6 12.0 11.0	6.2 7.9 9.4	19.8 19.9 20.4

Notes: $P^{(1)}$ pitch lag chosen to maximize pitch prediction gain
$P^{(2)}$ pitch lag chosen to maximize overall prediction gain

For the previous experiments, a modified covariance approach guarantees a minimum-phase formant prediction error filter. Ancillary experiments were conducted to compare different options for the formant filter in an F–P cascade. A lattice implementation of the formant filter using the reflection coefficients determined by a modified covariance method results in essentially no change in prediction gain. Using a covariance formulation (implemented in transversal form) for the formant filter improves the prediction gain by only 0.1 dB, but introduces nonminimum-phase formant filters in about 4 percent of the frames.

B. Formant-Pitch Interaction: Effects of Frame Size

The success of the pitch predictor depends on the formant residual having adjacent pitch pulses which are similar in shape. Yet if the formant predictor varies significantly from frame to frame, the pitch pulses may differ in detail. For short frames, the formant predictor coefficients may change significantly from frame to frame just due to the asynchronism between the frames and the positions of the pitch pulses. An investigation was made of the variation in prediction gain with changes in the sizes of the analysis frames for the formant and pitch filters. We return to the F–P cascade with the modified covariance method used to determine the coefficients of the transversal formant predictor. Also, the covariance approach determines the coefficients of a transversal pitch predictor. The results for different combinations of frame sizes are shown in Table II.

Consider the performance of the pitch filter with a 40-sample analysis frame. The pitch gain increases as the length of the analysis frame for the formant predictor increases from 40 to 80 and then levels off for a 160-sample formant analysis frame. At the same time, since the formant prediction gain does not change significantly with the frame size, the overall prediction initially rises and then levels off.[6] For the 80-sample pitch analysis frame, the performance is again essentially constant with a change of the formant frame size from 80 to 160. Since the prediction gain remains high at the slow formant update rates, the slow formant update rates are to be preferred since they involve less computation and require a smaller bandwidth for transmission. The number of frames with unstable pitch synthesis filters also depends on the frame size combination chosen. But as shown in the next

[5] This ordering is also true for each utterance, except for one utterance in which the P-F cascade (transversal pitch filter) with pitch lag, chosen to maximize overall gain, slightly outperforms the F–P cascade (transversal pitch filter).

[6] One would expect the formant prediction gain to increase with decreasing frame size. This is so for a covariance approach, but not necessarily so for the modified covariance approach, especially for short frames.

TABLE II
PREDICTION GAINS FOR AN F-P CASCADE FOR VARIOUS COMBINATIONS OF FRAME SIZES. THREE NUMBERS IN AN ENTRY REFER TO ONE-,TWO-, AND THREE-TAP PITCH FILTERS

formant		pitch		overall gain dB
frame size	gain dB	frame size	gain dB	
40	15.8	40	3.8 4.8 5.4	19.6 20.6 21.2
80	16.1	40	5.2 6.8 7.7	21.3 22.9 23.8
		80	4.2 5.3 5.8	20.3 21.4 21.9
160	16.0	40	5.2 6.8 7.7	21.2 22.8 23.7
		80	4.2 5.4 5.8	20.2 21.4 21.8
		160	3.0 3.9 4.1	19.0 19.9 20.1

section, the loss in prediction gain due to stabilization of the pitch filter is small anyway.

C. *Simplified Pitch Filter Computation*

The computation of the pitch predictor coefficients can be simplified in two ways for the multitap case. First, the matrix of covariance terms can be forced to be Toeplitz [see (11)] and hence allow for the use of a generalized Levinson recursion. For the three-tap case, the predictor gain decreases by an average of only 0.02 dB with this change (80-sample frames).

The second, more radical, change involves assuming that the off-diagonal terms in the covariance matrix are negligible. These terms are correlation terms with small lags. This approximation is justified by the observation that in an F-P cascade the formant filter tends to remove the short-time correlations. With this change, the matrix becomes diagonal. For a three-tap pitch filter, the coefficients are given by

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \phi(0, M)/\phi(M, M) \\ \phi(0, M+1)/\phi(M+1, M+1) \\ \phi(0, M+2)/\phi(M+2, M+2) \end{bmatrix} \quad (15)$$

where $\phi(i, j)$ is the correlation function for the intermediate residual signal $d(n)$ as in (12). The average prediction gain of this simplified solution in an F-P cascade is only about 0.2 dB less than the full complexity solution (80-sample frames).

VI. STABILITY OF PITCH SYNTHESIS FILTERS

For the configurations considered until now, the pitch filter derived by the transversal formulation can give rise to a nonminimum-phase prediction error filter with a corresponding unstable pitch synthesis filter. In a one-tap transversal filter in an F-P cascade, the coefficient is obtained by minimizing the energy of the residual over a frame of N samples and is given by

$$\beta_1 = \frac{\sum_{n=0}^{N-1} d(n)\, d(n-M)}{\sum_{n=0}^{N-1} d^2(n-M)}. \quad (16)$$

An unstable pitch synthesis filter arises when the absolute value of the numerator is greater than the denominator ($|\beta| > 1$). This usually arises when a transition from an unvoiced to a voiced segment takes place. Such a transition is marked by an increase in the signal energy. When processing a voiced frame that occurs just after an unvoiced frame, the denominator quantity $\Sigma\, d^2(n - M)$ involves the sum of the squares of amplitudes in the unvoiced segment and does not reach a very large value. On the other hand, the numerator quantity $\Sigma\, d(n)\, d(n - M)$ involves the sum of the products of the higher amplitudes from the voiced frame and the lower amplitudes from the unvoiced frame. Under these circumstances, the numerator can often be larger in magnitude than the denominator, giving $|\beta_1| > 1$. From these considerations, one can conclude that unstable pitch synthesis filters can arise when the signal energy shows a sudden increase. This concept carries over to two- and three-tap pitch filters. Highly unstable pitch synthesis filters can introduce pops and clicks into the reconstructed speech [8]. These are a result of the amplification of coding noise for frames with unstable synthesis filters.

A. *Stability of the F-P and P-F Cascade Configurations*

For an F-P cascade, the degree of instability of $H_P(z)$ depends on the frame size. Figures are shown for different combinations of frame sizes in Table III. The stability of the pitch filter is checked using the test given in [8]. If found to be unstable, the coefficients are scaled downward in magnitude to the point at which they satisfy the stability test (for further details, see [8]).

For a fixed formant frame size, the number of frames with unstable pitch filters increases with decreasing pitch frame size. For fixed frame sizes, the number of unstable frames also generally increases as the number of pitch taps is increased. The loss in pitch gain due to stabilization is small enough that the relative ordering of the systems with different frame size combinations is not altered. One can also note that the loss in prediction gain due to stabilization of the pitch filter is highly correlated with the percentage of unstable frames for a given number of pitch coefficients.

The synthesis filter $H_P(z)$ is more susceptible to instability in a P-F cascade than in an F-P arrangement. This is because the original speech waveform (input to the pitch predictor in a P-F cascade) has a higher input energy that shows greater variations from frame to frame than the residual after formant prediction (input to the pitch predictor in an F-P cascade). Experiments show that when using an F-P cascade (80-sample frames), an average of 23 percent of the frames have unstable three-tap pitch synthesis filters. In a P-F cascade, the number increases by more than a factor of 2, to almost half of the total number of frames processed. Due to this phenomenon, stabilization sacrifices more prediction gain in a P-F cascade than in an F-P cascade. Even though some of the loss in gain can be retrieved by calculating the formant coefficients based on the output from the stabilized pitch filter, an examination of the residual shows that the pitch pulses are not effectively removed by the stabilized pitch filter.

TABLE III
PITCH FILTER STABILITY FOR VARIOUS COMBINATIONS OF FRAME SIZES (F-P CASCADE). THREE NUMBERS IN AN ENTRY REFER TO ONE-,TWO-, AND THREE-TAP PITCH PREDICTORS. THE PREDICTOR GAIN IS THE GAIN BEFORE STABILIZATION. THE LOSS FIGURE IS THE LOSS IN PREDICTION GAIN DUE TO STABILIZATION OF THE PITCH FILTER

formant	pitch			
frame size	frame size	gain dB	% unstable	loss dB
40	40	3.8 4.8 5.4	9 28 32	0.1 0.3 0.5
80	40	5.2 6.8 7.7	9 29 35	0.1 0.6 0.7
	80	4.2 5.3 5.8	6 26 23	0.0 0.3 0.2
160	40	5.2 6.8 7.7	8 32 39	0.1 0.6 0.7
	80	4.2 5.4 5.8	4 25 27	0.0 0.3 0.2
	160	3.0 3.9 4.1	2 15 16	0.0 0.1 0.1

VII. PITCH LAG ESTIMATION

In the preceding comparisons, the pitch lag M was chosen as that value which maximizes the prediction gain. In this section, practical methods to choose M are discussed.

A. Transversal Case

For this section, we consider the covariance method used to find the coefficients of the pitch filter in one of the cascade configurations. In the general transversal case, the system $\mathbf{\Phi c} = \boldsymbol{\alpha}$ is solved, with the matrix and vectors appropriate to the cascade connection. Then, the optimum coefficients are given by $\mathbf{c} = \mathbf{\Phi}^{-1}\boldsymbol{\alpha}$, and the resulting mean-square error is $\epsilon^2 = \phi(0, 0) - \mathbf{c}^T\boldsymbol{\alpha}$. Note that $\mathbf{\Phi}$, $\mathbf{c}$, and $\boldsymbol{\alpha}$ all depend on M. The value of M should be chosen so as to maximize $\mathbf{c}^T\boldsymbol{\alpha}$. In general, an exhaustive search for the optimal M is not practical for multitap pitch filters.

A one-tap pitch filter leads to a simpler case. Then, $\beta_1 = \phi(0, M)/\phi(M, M)$, and the resulting mean-square error ϵ^2 is

$$\epsilon^2 = \phi(0, 0) - \frac{\phi^2(0, M)}{\phi(M, M)}. \tag{17}$$

The pitch lag M should be chosen so as to maximize $\phi^2(0, M)/\phi(M, M)$.

Atal and Schroeder [1] describe a method of estimating the pitch lag. They apply this method to a three-tap pitch filter. A normalized correlation array $\tau(m)$ is first calculated, where

$$\tau(m) = \frac{\phi(0, m)}{\sqrt{\phi(0, 0)\, \phi(m, m)}}. \tag{18}$$

The correlation array is searched for the maximum value, which is in fact the optimal lag for a one-tap pitch filter. For a three-tap filter, the lag corresponding to the middle tap is set to this value.

1) A New Method to Determine the Pitch Lag for an F-P Cascade: When formant prediction is performed first (F-P cascade), the near-sample-based redundancies have been removed to a large extent before pitch analysis. Therefore, the off-diagonal terms in the matrix $\mathbf{\Phi}$ [as in (11)] are small. Neglecting these terms leads to the simplified formulation for the pitch predictor as in (15). With the same approximation, $\mathbf{c}^T\boldsymbol{\alpha}$ becomes

$$\mathbf{c}^T\boldsymbol{\alpha} \approx \sum_{m=M}^{M+N_p-1} \frac{\phi^2(0, m)}{\phi(m, m)}. \tag{19}$$

The value of M that maximizes this quantity is chosen. This is equivalent to maximizing $\Sigma_{m=M}^{M+N_p-1} \tau^2(m)$. This new method applies a sliding rectangular window to the normalized correlation array. This need involve little additional computation over that for a single-tap filter. The denominator term can be calculated efficiently if use is made of the recursion

$$\phi(i + 1, i + 1) = \phi(i, i) + d^2(-i - 1) - d^2(N - 1 - i). \tag{20}$$

For a one-tap predictor, the method involves no approximation. For two- and three-tap predictors, the difference in pitch prediction gain compared to an exhaustive search is small (0.02 and 0.07 dB for two and three taps, respectively). For the three-tap case, the new scheme gives 0.2 dB more prediction gain than using the lag found by maximizing $\tau^2(m)$ alone.

2) Pitch Lag Selection for a P-F Cascade: For the P-F cascade, the pitch lag will be chosen to maximize the pitch prediction gain alone. Estimating the pitch lag for a multitap pitch filter in a P-F cascade cannot be done in the same way as for an F-P cascade since the assumption that near-sample correlations are small no longer holds. For two- and three-tap predictors, a search for the best M is computationally burdensome. As an alternative, the value of M chosen in the one-tap case can be used as an estimate of the pitch lag. For a three-tap predictor, the delay associated with the middle coefficient corresponds to the value of M for a one-tap filter. Experiments reveal that for three-tap pitch predictors, the average difference in overall prediction gain between the proposed method and an exhaustive search for the pitch lag which maximizes the pitch gain alone is extremely small. There is a loss in pitch prediction gain of 0.5 dB, which is made up by an increase in formant prediction gain of almost the same amount.

B. Lattice Case

The lattice coefficients for a pitch filter are chosen to minimize the sum of the mean-square values of the forward and backward residuals at each stage. This strategy ensures a minimum-phase filter. However, the real objective is to minimize the mean-square value of the forward residual alone in order to maximize the prediction gain.

For an N_p coefficient lattice form pitch predictor, the resulting mean-square error is

$$\epsilon^2 = \prod_{i=M}^{M+N_p-1} (1 - K_i^2) F_{M-1}. \tag{21}$$

The best value of M is that which minimizes the product term. The calculation of this quantity as a function of M is impractical except in special cases. For a one-coeffi-

TABLE IV
METHODS OF CHOOSING M IN TWO-COEFFICIENT LATTICE PITCH PREDICTORS

Method	Choice of M	Non-zero Coefficients
1	$\max(\mu^2(m))$	K_m, K_{m+1} or K_{m-1}, K_m
2	$\max(\mu^2(M) + \mu^2(M+1))$	K_M and K_{M+1}

TABLE V
METHODS OF CHOOSING M IN THREE-COEFFICIENT LATTICE PITCH PREDICTORS

Method	Choice of M	Non-zero Coefficients
1	$\max(\mu^2(M))$	K_{M-1}, K_M and K_{M+1}
2	$\max(\mu^2(M) + \mu^2(M+1))$	K_M, K_{M+1} and K_{M+2}
3	$\max(\mu^2(M) + \mu^2(M+1) + \mu^2(M+2))$	K_M, K_{M+1} and K_{M+2}

cient pitch predictor, the best value of M is that value of m which maximizes the value of $\mu^2(m)$ where

$$\mu(m) = \frac{2\phi(0, m)}{\phi(0, 0) + \phi(m, m)}. \tag{22}$$

The normalized correlation value $\mu(m)$ is in fact the reflection coefficient for a one-coefficient pitch predictor determined from the Burg formulation.

1) Pitch Lag Selection for an F-P Cascade: For a one-coefficient lattice filter, using the value of M that maximizes $\mu^2(M)$ is the best choice. The search for the best value of M for the two- and three-coefficient cases involves a great deal of computation. Even for an F-P cascade in which the near-sample redundancies have been removed by the formant predictor, the simplifying assumptions used in the transversal case are not useful for the lattice form.

When determining the value of M for two-coefficient lattice filters, it is desirable that the overall prediction gain be consistently higher than in the one-coefficient case. A number of different methods for choosing M were considered. A subset consisting of the best of these is shown in Table IV.

Method 1 uses the one-coefficient criterion to choose the best lag for a two-coefficient lattice. Given the value of m chosen by method 1, the reflection coefficients K_M and K_{M+1} are used if $\mu^2(M+1) \geq \mu^2(M)$; otherwise, K_{M-1} and K_M are used. Methods 1 and 2 achieve gains which are 0.2 and 0.1 dB (respectively) less than that for an exhaustive search for the pitch lag.

A selection of methods considered for three-coefficient lattice predictors is shown in Table V. For each of the entries, the lattice coefficients can be placed at different lags. The entries represent the best of these placements. Method 2 adds a lattice stage to its two-coefficient counterpart and hence must have a higher prediction gain than the two-coefficient predictor. Methods 1 and 3 give essentially the same performance (0.4 dB less than for an optimal choice). Method 2 exhibits the highest prediction gain, which is 0.3 dB less than for the exhaustive search. Note that in any of these methods, if $\mu(m)$ is replaced by $\tau(m)$, the results are virtually identical.

2) Pitch Lag Selection for a P-F Cascade: For a P-F cascade with lattice form pitch filters, the value of M is determined differently than for its F-P counterpart. A reliable algorithm is devised by extending the approach that is optimal for maximizing the pitch prediction gain for a one-tap pitch filter. Choosing M in this fashion and adding additional lattice stages for two- or three-tap predictors guarantees a higher pitch gain and is comparable to the covariance formulation. The average loss in overall prediction gain over that corresponding to the choice of M that maximizes pitch gain alone is 0.45 dB for a three-tap predictor.

VIII. PERFORMANCE IN AN APC SPEECH CODER

To help assess the relative performance of the various cascade predictors, an APC speech coder was used as a test bed. The APC test bed is modeled after the APC coder described in [1] by Atal and Schroeder. For the F-P cascade, the coder is as in [1], with the formant predictor placed in an open-loop arrangement and the pitch predictor in a feedback loop around the quantizer. The noise shaping filter is $N(z)$ where $N(z) = F(z/\alpha)$ (see Fig. 1). Accomplishing the *same* noise shaping in a P-F cascade results in a more complex system than in the F-P case. The P-F cascade version requires two feedback loops around the quantizer.

Some of the experimental conditions reported in [1] are changed to maintain consistency with the parameters used throughout this paper. The speech is sampled at 8 kHz. The predictors are updated every 80 samples. The modified covariance method (with the addition of high-pass filtered white noise as in [1]) is used to calculate the coefficients of a tenth-order formant predictor. The pitch predictor has three coefficients. Transversal-structured pitch predictors are stabilized if necessary. The stabilization was found to be necessary to prevent undesirable pops and clicks in the output speech (particularly prevalent for the P-F cascade). The pitch lag is computed using the practical approaches outlined in Section VII. The quantization of the residual is performed by an adaptive three-level uniform quantizer. The noise feedback factor α is set to 0.75. The noise feedback samples are passed through a peak limiter which clips at twice the rms value of the final prediction residual.

The APC coder was run using the same speech database used in Section V to calculate the prediction gains. The F-P arrangement employing a transversal solution is the best of the methods. The F-P lattice and the P-F transversal solutions generate decoded speech of essentially the same quality and which is only slightly inferior to that for the F-P transversal method. However, the distortions present in the F-P lattice and P-F transversal approaches are different. The F-P lattice generates output speech that suffers from occasional warbles. The output speech resulting from a P-F transversal solution has audible background noise and occasional clicks. The fourth case, namely, the P-F lattice, generates decoded speech which is significantly worse than that of the other methods for most of the utterances.

IX. Summary and Conclusions

In terms of prediction gain, the F–P cascade stands out as being superior to the P–F cascade for both transversal and lattice implementations of the pitch filter. In addition, the minimum-phase condition can be enforced on a transversal implementation with only a small drop in prediction gain. To achieve minimum-phase prediction error filters, the formant coefficients are computed using the modified covariance method (or autocorrelation method), and the pitch coefficients are computed using a covariance approach and then modified if the pitch filter is not minimum phase. In the lattice case, the minimum-phase condition is guaranteed by the Burg formulation. An additional advantage of the F–P cascade is that it allows for a simplified solution for multitap pitch filters with little loss in performance. This solution involves approximating the matrix of correlations with a diagonal form. Whether a transversal or lattice structure is used to implement the formant and pitch filters affects the performance to only a small degree. The choice of structure then is dictated by ancillary considerations such as computational complexity and parameter coding.

For the P–F cascade, the pitch lag can be chosen to maximize the pitch prediction gain or the overall prediction gain. The latter option is preferable, but impractically complex and still falls short of the performance of an F–P cascade. For a transversal pitch filter in the P–F cascade, the incidence of instability is high. The stabilization process sacrifices prediction gain, and in addition, the stabilized pitch filter does not adequately remove the pitch pulses.

Experiments for the F–P configuration with different frame sizes reveal that a combination of a short analysis frame for the pitch predictor and a long analysis frame for the formant predictor produces high overall prediction gains. The degree of instability of the pitch filter increases with decreasing frame size, although the consequent loss in prediction gain due to stabilization is not very large.

Algorithms to determine a suitable value for the pitch predictor lag were developed. For the F–P cascade in transversal form, a simplified method to determine the pitch lag causes little loss in prediction gain compared to that from the use of the optimum lag. The computational complexity is a fraction of that for an exhaustive search.

An APC coder was used to evaluate speech quality. Of the predictor methods considered, the P–F lattice performs the worst. The F–P cascade employing a transversal solution is the best. The F–P lattice and the P–F transversal solutions also perform well and are only slightly inferior to the F–P transversal approach. Note, however, that noise shaping in a P–F configuration is more complex to implement than in an F–P arrangement.

The above results show that the F–P transversal approach is the method of choice from a number of different points of view. It has advantages in terms of prediction gain, subjective quality, and complexity of implementation in a noise feedback APC coder.

Appendix A
Autocorrelation Method: Minimum-Phase Properties

In the case of formant filters, the autocorrelation method is well known to lead to minimum-phase prediction error filters [5]. A formant filter is distinguished by the fact that the prediction is based on data values immediately preceding the current sample. A more general set of delay values leads to a formulation which lacks the structure imposed by this special case. A simple example is used to show that use of the autocorrelation method by itself does not guarantee a minimum-phase predictor.

Consider a two-tap pitch filter with coefficients corresponding to lags 2 and 3 ($M = 2$). Then, the equations to be solved are

$$\begin{bmatrix} \phi(0) & \phi(1) \\ \phi(1) & \phi(0) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \phi(2) \\ \phi(3) \end{bmatrix}. \tag{A1}$$

Let the autocorrelation values be $\{\phi(0), \phi(1), \phi(2), \phi(3)\} = \{1, 0, a, a\}$, with the parameter a being positive. This sequence of correlations is positive definite for $a < 0.618$. It can easily be shown that the sum of the optimal filter coefficients is $2a$. For positive pitch filter coefficients, a necessary and sufficient condition for a minimum-phase filter is that the sum of the filter coefficients be less than unity [8]. Then, for $0.5 \le a \le 0.618$, the pitch filter is not minimum phase. Another example of such a case has been given in [10].

Although the autocorrelation method does not give minimum-phase predictors for arbitrary spacing of the delays, one special case that does lead to a minimum-phase solution warrants discussion. Consider a predictor based on uniformly spaced delays of the form $M_k = kM_1$. This case can be considered to be an extension of the formant filter applied to a subsampled signal. The autocorrelation sequence for a subsampled signal is the subsampled version of the autocorrelation sequence for the original signal. The pitch filter with lags as given above can be considered to be a filter operating on interleaved subsampled signals. It can then be argued that the optimal prediction error filter with these lags derived using the autocorrelation formulation is minimum phase. This uniformly spaced case subsumes a one-tap pitch filter and a formant filter.

Appendix B
Modified Covariance Method

The modified covariance method [1], [11] is based on residual energy ratios and guarantees a minimum-phase prediction error filter. This method has been successfully applied to formant prediction. Unfortunately, this method cannot be derived as minimizing an error criterion.

The following description of the modified covariance method is intimately tied in to the Cholesky decomposition for solving a set of positive definite equations, ($\boldsymbol{\Phi c} = \boldsymbol{\alpha}$). The Cholesky decomposition forms a lower/upper triangular decomposition of the correlation matrix

$$\boldsymbol{\Phi} = \boldsymbol{LU} \tag{B1}$$

where $\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{U}$ is the upper triangular matrix formed by transposing $\boldsymbol{L}$ ($\boldsymbol{L}^T = \boldsymbol{U}$). Define the auxiliary vector

$$\boldsymbol{g} = \boldsymbol{Uc}. \tag{B2}$$

The solution of the equations proceeds by first solving a set of triangular equations by back substitution,

$$\boldsymbol{Lg} = \boldsymbol{\alpha}. \tag{B3}$$

The solution vector $\boldsymbol{g}$ is substituted into the triangular set of equations (B2), which is solved for the coefficients $\boldsymbol{c}$. The error resulting from using this optimal set of coefficients can be expressed as

$$\epsilon^2 = \phi(0, 0) - \boldsymbol{g}^T\boldsymbol{g}. \tag{B4}$$

One property of the LU decomposition of a positive definite matrix is a nesting of the solution. Denote $\boldsymbol{\Phi}^{(i)}$ as the $i \times i$ submatrix obtained by keeping only the first i rows and first i columns of $\boldsymbol{\Phi}$. The lower triangular matrix in the LU decomposition of $\boldsymbol{\Phi}^{(i)}$ is itself a submatrix of $\boldsymbol{L}$. This nesting implies that the vector $\boldsymbol{g}$ also nests. The mean-square error for the ith-order solution can be written as

$$\epsilon_i^2 = \phi(0, 0) - \boldsymbol{g}^{(i)T}\boldsymbol{g}^{(i)}. \tag{B5}$$

The coefficients found solving the equations using the Cholesky decomposition may result in a synthesis filter which is not stable. Using an analogy with conventional autocorrelation analysis, a stabilization strategy can be formulated based on a reflection coefficient parameterization for the filters. The normalized mean-square error for an autocorrelation approach can be expressed as

$$\epsilon_i^2 = \phi(0, 0) \prod_{n=1}^{i} (1 - K_n^2) \tag{B6}$$

where K_n is a reflection coefficient. The ith reflection coefficient in the autocorrelation method can be expressed in terms of the mean-square error for an ith-order filter and that for an $(i - 1)$th-order filter:

$$K_i^2 = 1 - \frac{\epsilon_i^2}{\epsilon_{i-1}^2}. \tag{B7}$$

The modified covariance strategy uses the ratios of the error energies to define reflection coefficients. Using (B5) and (B7), the reflection coefficients are obtained from

$$K_i^2 = \frac{g_i^2}{\epsilon_{i-1}^2}. \tag{B8}$$

With the autocorrelation method, the sign of the reflection coefficient is opposite to that of the predictor coefficient. In the modified covariance strategy, g_i assumes the role of the reflection coefficient in determining the sign. The reflection coefficients, which are bounded in magnitude by unity (and hence correspond to stable synthesis filters), can be transformed to direct form coefficients [12].

A. Application to Pitch Filters

For the general form of the Levinson recursion used to solve autocorrelation equations with an arbitrary right-hand-side vector, the reflection coefficients that correspond to the filter coefficients do not appear naturally in the formulation. That means that for the case of an arbitrary right-hand-side vector, while the mechanics of the modified covariance approach may be used, the method is no longer rooted as the analog to the autocorrelation approach.

As part of the study of pitch predictors, attempts were made to extend the modified covariance method to pitch filters using both transversal and lattice implementations. However, the resulting prediction gains were poor. In fact, the prediction gain does not necessarily increase monotonically as more reflection coefficients are added.

The stable solution produced by a modified covariance approach leads to lower pitch prediction gains than a stabilized covariance solution. Consider a one-tap pitch predictor. The modified covariance method leads to

$$\beta_1 = \frac{\phi(0, M)}{\sqrt{\phi(0, 0)\,\phi(M, M)}}. \tag{B9}$$

In this case, β_1 is a normalized correlation coefficient. The corresponding coefficient for the covariance approach is

$$\beta_1 = \frac{\phi(0, M)}{\phi(M, M)}. \tag{B10}$$

Analysis shows that the prediction gain is higher if β_1 is calculated by the covariance approach and the resulting pitch synthesis filter is stabilized by setting $|\beta_1| = 1 - \epsilon$ whenever $|\beta_1| > 1$ (see [8]) than if β_1 is determined by the modified covariance method.

References

[1] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 247-254, June 1979.

[2] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 63-73, Feb. 1979.

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[4] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 25.1.1-25.1.4.

[5] S. W. Lang and J. H. McClellan, "A simple proof of stability for all-pole linear prediction models," *Proc. IEEE*, vol. 67, pp. 860-861, May 1979.

[6] S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC codes at low bit rates," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, Mar. 1984, pp. 1.3.1-1.3.4.

[7] J. Makhoul, "Stable and efficient lattice methods for linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 423-428, Oct. 1977.

[8] R. P. Ramachandran and P. Kabal, "Stability and performance anal-

ysis of pitch filters in speech coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 937-946, July 1987.

[9] J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet, "Speech coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 710-736, Apr. 1979.

[10] T. L. Marzetta, "Two-dimensional linear prediction: Autocorrelation arrays, minimum-phase prediction error filters, and reflection coefficient arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 725-733, Dec. 1980.

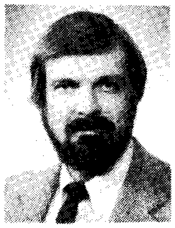
[11] B. W. Dickinson, "Autoregressive estimation using residual energy ratios," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 503-506, July 1978.

[12] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

Ravi P. Ramachandran was born in Bangalore, India, on July 12, 1963. He received the B.Eng. degree (with great distinction) from Concordia University, Montreal, P.Q., Canada, in 1984 and the M.Eng. degree (on the Dean's Honour List) from McGill University, Montreal, P.Q., Canada, in 1986. During the period of his undergraduate studies, he held a Concordia University Undergraduate Fellowship. When studying for the M.Eng. degree, he held a Natural Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Fellowship. He is currently a doctoral student at McGill University and holds another NSERC Postgraduate Fellowship for the Ph.D. degree.

From January to June 1988 he was a Visiting Postgraduate Researcher at the University of California, Santa Barbara. His main research interests are in speech coding, data communications, and digital signal processing.

In 1984 Mr. Ramachandran received the Order of Engineers of Quebec Student Award, the John H. Chapman Award (presented by Spar Aerospace for scholarship in communications), the electrical engineering medal as the most outstanding student in electrical engineering, and the Morris Chait medal as the highest-ranking student in the B.Eng. program.



Peter Kabal received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, Ont., Canada.

He is an Associate Professor in the Department of Electrical Engineering at McGill University, Montreal, P.Q., Canada, and a Visiting Professor at INRS-Telecommunications (a research institute affiliated with the Université du Québec), Verdun, P.Q. From September 1987 to June 1988 he spent a sabbatical year as a Visiting Professor at the University of California, Santa Barbara. His current research interests focus on digital signal processing as applied to speech coding, adaptive filtering, and data transmission.