Preprints IFAC/IFORS Symp. Identification, System Parameter Estimation (Budapest), pp. 203-207, July 1991

Adaptive Linear Prediction in Speech Coding

Peter Kabal

Department of Electrical Engineering, McGill University
Montreal, Quebec Canada

INRS-Télécommunications, Université du Québec
Verdun, Quebec Canada

Abstract Adaptive linear prediction is commonly used as a key step in digital coding of speech. This paper discusses some of the techniques that have been developed for adapting and coding the predictor coefficients in speech coders. The linear predictors in high quality speech coding often consist of two stages, a short-time span (formant) filter and a long-time span (pitch) filter. The use of such filters in analysis-by-synthesis coders is examined. In addition, backward adaptive strategies can be used to achieve high quality, low delay coding. The filters in these coders can be high-order (50 or more time lags) filters. Computational complexity and numerical stability of the algorithms is of prime concern for these filters. A number of new directions in the application of adaptive prediction in speech coding are also discussed.
Keywords adaptive systems, prediction, speech analysis

1. Introduction

Adaptive linear prediction is used in speech coding to remove redundancies from the speech signal. The prediction residual (error) signal can then be coded for transmission. If the predictor is working well, the residual signal is smaller in amplitude and easier to code than the original speech signal. The receiver uses the decoded residual signal to excite a synthesis filter.

This paper reviews the application of predictor filters in digital coding of speech. The practical considerations of speech coding means that conventional methodologies (see for instance [1]) for linear prediction must modified for this application. We refer to means to make the predictor adaptive, alternative filter structures, parameter coding considerations, and backward updates.

Fig. 1 shows a linear predictive speech coder, perhaps more appropriately called an adaptive predictive coder (APC). The conventional predictor filter is $F(z)$, given as

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

The predictor output is the linear combination of past input speech samples. The residual signal $r(n)$ is formed as the difference between the current sample and the predictor output. The residual is quantized with the block labelled Q. The coder includes an quantization error feedback filter $N(z)$.

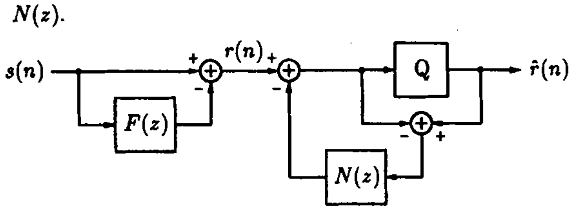

Fig. 1 A speech coder

The corresponding decoder is shown in Fig. 2. The receiver synthesis filter is

$$H(z) = \frac{1}{F(z)} . \qquad (2)$$

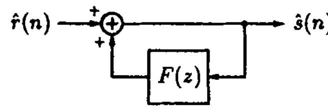

Fig. 2 Decoder

In the absence of quantization, the connection of the coder to the decoder results in distortionless reproduction of the input signal. The quantization is implied to be sample-by-sample, but in low bit rate coders, additional gains can be had by using vector, or delayed decision (tree) coding. In any case, the quantizer output is represented by an index which is transmitted to the receiver. The receiver decodes the index to form the quantized residual $\hat{r}(n)$.

The effect of the predictor can be described by the signal-to-noise ratio of the output. The overall SNR depends on the choice of noise feedback filter [2]. We model the quantizer as generating additive noise uncorrelated with the input. If the filter is absent ($N(z) = 0$), the output signal is the input signal plus quantization noise filtered through $H(z)$. If $N(z) = F(z)$, a simple analysis shows that the output signal has a flat quantization noise spectrum. In that case, the SNR of the output can be expressed as the product

$$\mathrm{SNR}_O = \mathrm{SNR}_Q \mathrm{P}_G . \qquad (3)$$

In this expression, SNR_Q is the SNR for the quantizer alone, and P_G is the predictor gain, defined to be the ratio of the input signal energy to the residual signal energy.

In practice, $N(z)$ is related to $F(z)$, often as a bandwidth expanded version of $F(z)$. Such a filter will (generally) result in a lower SNR, but is subjectively preferable.

2. Adaptive prediction

A speech signal can be thought of as a signal which is quasi-stationary, changing between different modes. One can argue from a speech production point of view that the vocal tract is a time-varying filter which instills a spectral shaping. A compromise predictor, either trained for a single speaker or even an ensemble of speakers can achieve only modest predictor gains.

Benefits accrue if the predictor coefficients are updated to reflect the local properties of the speech signal. The dynamics of the filter can be tuned to the dynamics of the vocal tract. This type of argument would point to an adaptive predictor as a time-varying linear filter.

However, when the adaptation process is taken into account, the name linear predictive coder turns out to be a misnomer. In most implementations of adaptive filters, the adaptation is linked to minimizing the mean-square error over an appropriate window of time. Thus there is a feedback from the error signal to adjust the filter parameters. This makes the overall adaptive filter, non-linear.

The "degree" of non-linearity depends on the adaptation speed. A slowly adapting filter or a filter which has constant coefficients for a block of time, would appear to be linear in the appropriate time interval.

2.1 Forward adaptive predictors

The conventional approach for low-bit rate coding, is to adapt the predictor based on analysis of the input speech signal. The coder does this analysis, and transmits the predictor coefficients as side information — the receiver needs to be able to generate the inverse filter. This is termed *forward* adaptation.

To keep the amount of side information under control, the number of parameters and the rate of update must be held in check. This suggests a block update strategy.

Transversal implementation

A model for calculating the predictor coefficients for a transversal implementation is shown in Fig. 3. 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 ,

$$e(n) = x_w(n) - \sum_{k=1}^{L} c_k\, x_w(n - M_k)\,. \tag{4}$$

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 a error window $w_e(n)$ to obtain a windowed error signal $e_w(n)$ where $e_w(n) = w_e(n)e(n)$. The squared error is defined by,

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

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

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

where

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

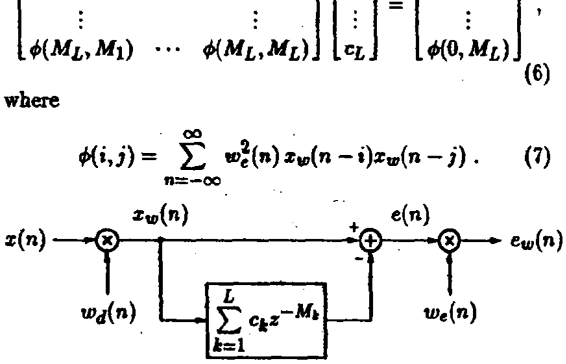

Fig. 3 Predictor analysis model

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). The window has the effect of deemphasizing the high order lag products. An important consideration is the minimum phase property of the prediction error filter $A(z) = 1 - F(z)$. If $A(z)$ is minimum phase, the corresponding synthesis filter $H(z)$ used at the decoder is stable. In the case of general delays, M_i, 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 is the case most widely studied.

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

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 $0 \le n \le N-1$. More general error windows in a covariance approach have been suggested by Singhal and Atal [3]. The covariance method does not guarantee that $A(z)$ is minimum phase but does minimize the error energy for each frame.

Lattice implementation:

Lattice analysis methods have been employed in linear prediction and are useful in implementing a lattice structured predictor [4]†. 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. 4. 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 [4].

For those stages for which a reflection coefficient is calculated, the aim, in terms of maximizing the prediction 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 property of guaranteeing that the mean-square value of the forward residual is non-increasing across each stage of the lattice.

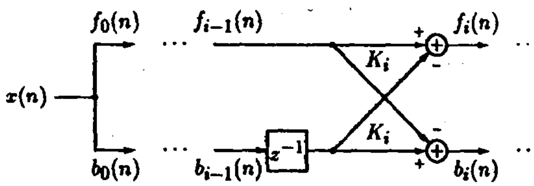

Fig. 4 Analysis model for lattice predictors

† One can convert between transversal and lattice implementations, achieving identical impulse responses. However in a time-varying environment, they two structures are not entirely equivalent due to their different initial conditions at frame boundaries.

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

$$K_i = \frac{2C_{i-1}}{F_{i-1} + B_{i-1}} , \qquad (8)$$

where

$$C_i = < f_i(n) b_i(n-1) >, \ F_i = < f_i^2(n) >, \ B_i = < b_i^2(n-1) >, \qquad (9)$$

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 [4], calculates the reflection coefficients using Eq. (8) 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.

Note also that the lattice coefficients can be transformed to direct form (impulse response) coefficients, allowing for an alternate implementation of the filter in transversal form.

2.2 Formant predictors

In applying a predictor in a speech coder, we identify two windows. The analysis window was described earlier. The filter window is the interval over which the filter coefficients are kept constant. The analysis window generally overlaps the filter window for the best prediction gain, although delay considerations may suggest only partial overlaps. The length of the filter window is determined by transmission rate considerations, and by considerations of intervals over which a fixed filter is appropriate. The number of prediction coefficients is limited by the rate dedicated for their transmission. Such a filter is often termed a formant filter. Formants in speech are the resonances in the speech spectral envelope. The formant filter models these resonances. Speech tends to have 4 or 5 formants, so that the orders of 8–10 are appropriate. Additional coefficients help fit other spectral details.

2.3 Pitch predictors

Speech is quasi-periodic in voiced regions. That means that samples separated by the pitch period tend to be similar. This pitch period is in the order of 40–120 samples at a 8 kHz sampling rate. To capture this redundancy, the prediction filter has to have the corresponding number of delays. As an alternative, however, we can use only a small number of non-zero weights. The pitch filter then consists of a bulk delay corresponding to the pitch lag M_p and then a small number of weights (typically 1–3).,

$$P(z) = \sum_{i=1}^{N_p} b_i z^{-M_p-i-1} . \qquad (10)$$

In the case of the pitch filter, the pitch lag as well as the weight values must be made known to the receiver.

Given a pitch lag, the general set of equations given earlier can be used to find a minimum mean-square error pitch predictor. However due to the relative large value of the pitch lag (filter order) with respect to the window size, data windowing is not appropriate, i.e., the autocorrelation method does not give high prediction gains [5]. The covariance method must be used. The pitch lag can be estimated from correlation calculations [5].

There are a number of ways that the pitch (long-term) predictor can be combined with the formant (short-term) predictor. The two predictors can be placed in cascade in either order, or can be placed in parallel. Experiments have shown that a F-P configuration, with the formant filter preceding the pitch filter outperforms the other configurations. Furthermore, the F-P combination can be jointly optimized to give an overall better prediction gain [6].

Recent studies have shown that multi-tap pitch filters act somewhat as interpolators, compensating for non-integer pitch delay values. An alternative is to use fractionally-spaced pitch filters [7]. These are implemented as a fixed interpolator in cascade with the pitch predictor. The number of coefficients for the pitch filter can be reduced, but at the expense of an increased resolution in the lag values that must be transmitted.

2.4 Analysis-by-synthesis

The conventional view of the use of adaptive prediction in speech coding is that the coder finds a predictor which minimizes the residual energy. Let us step back and look at the problem from the viewpoint of the decoder. Let the decoder structure use an all-pole synthesis filter (cascade of a formant filter and a pitch filter). The coding problem is to jointly choose the best excitation waveform and synthesis filter parameters. For each block of data, the excitation waveform is chosen from a finite repertoire of waveforms The error criterion is a frequency weighted mean-square error.

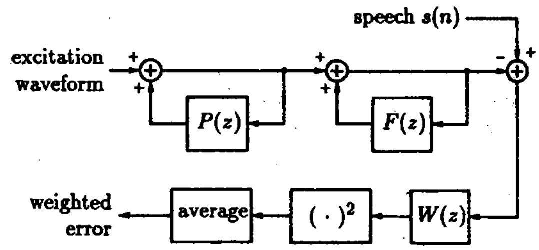

Fig. 5 Analysis-by-synthesis

The coder can form a local copy of the decoder and search over the parameter space for the combination that gives the best reconstructed waveform. Consider the subproblem of choosing the best synthesis filter parameters for a given excitation waveform. An exhaustive search over the parameter space (typically 20-50 bits of data for the filter parameters) is not practical. Setting up to solve for the minimum mean-square error formant filter leads to highly non-linear equations (due to the recursive nature of the filter). Now we are back to the starting point, the formant synthesis filter has to be obtained by analyzing the input speech, i.e., solving for the minimum mean-square error predictor.

However, in the case of the pitch filter, if the pitch lag is larger than the analysis block length, the feedback loop is effectively open. Solving for the minimum mean-square error filter parameters results in a set of linear equations [8].

Schemes that use this approach, for example Code-Excited Linear Prediction (CELP), have been dubbed analysis-by-synthesis schemes. Consider the three major elements of a CELP decoder: the excitation waveform, the pitch filter and the formant filter. The formant filter is preselected by analyzing the input speech (linear prediction). Given this filter, for each excitation waveform, we can solve for the best pitch filter (lag and coefficients). The benefits in choosing a pitch filter jointly with the excitation waveform are great. This pitch filter will take into account the imperfections in the excitation waveform. The number of excitation waveforms in a CELP coder can be relatively small (typically 128–1024 for a 40 sample block — a fraction of a bit per sample). For CELP, the transmission rate allocated for the excitation (the "main" information stream) is much smaller than that for the "side" information (filter parameters).

The remaining open problem for this type of configuration is to discover better strategies to choose the formant synthesis filter parameters — reoptimizing them to account for the coding noise.

3. Stability

The filters in the speech coding system are updated frame by frame. Conventional notions of stability are in essence asymptotic properties of systems. In speech coding, an "unstable" filter may persist for a few frames (often corresponding to an interval with increasing energy, but eventually periods of stable filters are encountered. In practice, the output does not continue to increase in amplitude with time.

Consider the case of an all zero prediction error filter in cascade with a quantizer, followed by an all pole synthesis filter. The quantizer can be modelled as adding noise, possibly correlated with the signal, to the residual signal. As long as the synthesis filter is the inverse to the prediction error filter, and the filter coefficients are updated in step, the signal component emerges unaltered. For the signal component, stability is not a problem because of pole/zero cancellation. However, the quantization noise passes through only the synthesis filter. An "unstable" synthesis filter can cause the output noise to build up during the period of instability and can lead to locally degraded speech quality.

3.1 Noise enhancement

The effect of filtering on the quantization noise may be measured in a number of different ways. If the quantization noise is modelled as white noise, the output noise power can be expressed as the input noise power times the power gain of the filter. The power gain is the sum of the squares of the filter coefficients.

One approach to the problem of noise buildup is to constrain the power gain of the synthesis filter. Formally this can be approached as a calculus of variations problem. Consider augmenting the residual correlation matrix with a term of the form $\lambda \mathbf{c}^T \mathbf{c}$. The solution is found by augmenting the each term on the diagonal of the correlation matrix in the standard covariance formulation by the term λ. The augmenting term in that case was the correlation matrix for high frequency noise. In practice, the Lagrange multiplier λ must be determined iteratively.

In practice the system of equations to be solved can be ill-conditioned. The eigenvalue spread can be large if the input speech spectrum has nulls such as due to lowpass filtering. There may be a continuum of solutions all very close to the optimum, but which may have very different power gains. In this case, constraining the power gain need not reduce the prediction gain by a large amount.

Conventionally the problem of noise enhancement is tackled by ensuring a stable synthesis filter. For formant synthesis filters, the autocorrelation method gives stable filters. With conventional windows, the autocorrelation and covariance methods differ mainly by the way they treat block edges. For sufficiently long (with respect to the order of the filter) windows, the covariance method also gives stable filters, at least most of the time.

The situation is far different for pitch filters, the covariance method often gives unstable filters. These filters correspond to physical situations at the onset of speech where the waveform is growing with each pitch period. If the pitch filter is determined from the clean input speech, severe degradation due to the noise enhancement effects of the "unstable" synthesis filter are observed. The filters can be stabilized by moving the singularities inward [9]. The analysis-by-synthesis coder strategies do not need to explicitly consider stability. If the pitch filter is optimized for a given excitation waveform, the noise enhancement effect is automatically included.

4. Filter parameter coding

Vector quantization has been much studied for the coding of linear predictor filter parameters. The fidelity criterion is often Euclidean distance in the autocorrelation domain. However, the number of bits required for good coding (20–40 bits for 10 formant filter coefficients), precludes an exhaustive search vector quantization strategy.

Further gains can be had if inter-frame correlations are also considered. In practice, this is rarely used; first differential coders can cause error propagation in the present of channel errors and second, inter-frame coding involves added coding delay.

The direct form coefficients representation of the filter parameters is not conducive to efficient quantization. Instead, non-linear functions of the reflection coefficients (e.g. log-area ratio) are often used as transmission parameters.

Recently, there has been a growing interest in the use of line spectral frequencies (LSF) to code the filter parameters. LSF's are an alternative to the direct form predictor coefficients or the lattice form reflection coefficients for representing the filter response. The line spectral frequencies are an ordered set of frequencies obtained from the filter description [10]. There is an intimate relationship between the LSF's and the formant frequencies. Accordingly, LSF's can be quantized taking into account spectral features known to be important in perceiving speech sounds.

Other possibilities are combined vector-scalar quantizers in which the fidelity criterion is based on an LSF distance. This type of quantizer finds a middle ground between performance and computational complexity [11].

5. Backward adaptive predictors

Forward adaptive predictors have analysis windows and filter windows which overlap. We now consider the case in which the analysis window entirely precedes the filter window. Furthermore, consider the case in which the adaptation is based on the reconstructed signal rather than the original speech signal. Since the reconstructed signal is available to both the coder and decoder, both can adapt the filters and no explicit transmission of filter parameters is required. This is a backward adapted strategy.

Backward adaptation has the advantage that since no side information is transmitted, there is no inherent limit to the number of coefficients that can be used or the update rate — there is only a causality requirement that the predictor update must utilize only past reconstructed outputs.

Backward adaptation is however, susceptible to mistracking of the coder and decoder filters. Such systems have to be carefully designed so that the effect of channel errors dies off quickly and the coder and decoder get back into synchronism. In addition, the coding noise present in the reconstructed signal prevents a full realization of the predictor gain available from the clean input signal.

Backward adaptation has been used for some time in relatively high rate coders with simple predictor structures. More recently, very high quality medium rate coders have been designed using backward adaptation. For instance at 16 kb/s (8 kHz sampling rate), a low delay speech coder using 2 bits/sample (no side information) can achieve very high qualities [12]. The fact that adaptation feeds back from the reconstructed signal places a heavy burden on each component of the coder. Each component must perform well or the whole loop collapses. The coder cited uses a delayed decision quantizer (sliding block or tree coder), sample-by-sample updated formant predictor and a 3-tap pitch predictor.

The pitch filter with its long delay and backward adaptive estimation of the pitch lag, renders the coder susceptible to mistracking if the channel error rate becomes significant. Newer alternative are the use of a very high order filter (50–100). As long as the number of taps encompasses the pitch range, this filter can act as pitch filter without explicit estimation of the pitch lag. This configuration is more robust to channel errors.

5.1 Issues in high-order predictors

The high-order filter (combining both formant and pitch functions) can be determined using the methods described earlier. With such high orders, numerical considerations for the solution of the equations becomes an important consideration. The autocorrelation method and the Levinson recursion offer good numerical properties. However, the predictor gain for the autocorrelation method for the pitch periodicities becomes small (the analysis window deemphasizes the large lag correlation components). The covariance method gives much superior predictor gains, but conventional solutions are plagued by numerical difficulties.

Recently, a solution method due to Cumani [13] has shown itself to be very useful. This method is a stage-by-stage optimization in which numerical errors in previous stages can be compensated for in later stages. This method allows one to realize the increased predictor gains of the covariance formulation [14].

We can refer to the samples of the past signal as being stale. We reduce the staleness by keeping the analysis window as close to the filter window as possible. However, reductions in computations can often be had by increasing the filter window length and sliding the the analysis window by a similar amount. For instance if the filter coefficients are kept constant and used over 5 samples, the analysis window can be moved in steps of 5 samples. The derivation of the coefficients corresponding to the samples in the analysis window then need be done only every 5 samples. The length of the analysis window is determined by that which gives the best prediction gain. Too short a window and longer term trends are not taken care of. In fact random variations in the signal will cause the prediction gain to fall. Too long a window and the window may span non-stationary segments of the signal. The number of coefficients is not a direct factor, since they are not transmitted.

6. New directions

Speech coding is a rapidly evolving area, spurred on as it is by a demand for digital transmission and the availability of low cost signal processing hardware for implementing sophisticated algorithms. In the application of new prediction techniques, we have been limited by the analytic tools. Recent work [15] has indicated that non-linear predictors may have a larger role to play. In this work, non-linear dynamical system concepts are used to show that after linear predictive analysis, a significant predictable component remains. This component lies on an attractor of dimension less than 3. Our own work indicates that a rapidly updated, backward adaptive adaptive filter (of a more conventional kind, but is are non-linear nonetheless) can also improve the prediction gain when used in the same context. This is preliminary work, but shows great promise — the backward adaptive nature means that no additional side-information need be transmitted.

Another area of interest is in tracking the dynamics of speech. The conventional approach has been to treat speech as quasi-stationary. We assume that within a frame the signal is stationary. Additional gains can be made by allowing the predictor to change within the frame to better track the changing signal. The same viewpoint will allow for improvements in backward adaptive predictors. The problem there is that the analysis window is displaced from the filter window. If we can track the signal changes, the prediction can be improved.

A related problem is the use of fixed frames for analysis. If the frame straddles two different types of speech segments, the prediction gain will be small. A segmentation of speech into variable length frames can improve the situation. We have preliminary results based on a Hidden Markov filter to automatically segment a speech signal. We fix the number of frames in an interval of speech and then use dynamic programming to optimize the frame boundaries. This scheme dramatically improves the local prediction gain. The drawback is the additional processing delay introduced.

References

1. P. Strobach, *Linear Prediction Theory*, Springer-Verlag, 1990.
2. J. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.
3. S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC codes at low bit rates", *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, San Diego, CA, March 1984, pp. 1.3.1–1.3.4.
4. J. Makhoul, "Stable and efficient lattice methods for linear prediction", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-25, pp. 423–428, October 1977.
5. R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-37, pp. 467–478, April 1989.
6. P. Kabal and R. P. Ramachandran, "Joint optimization of linear predictors in speech coders", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-37, pp. 642–650, May 1989.
7. P. Kroon and B. S. Atal, "Pitch predictors with high temporal resolution", *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Albuquerque, NM, April 1990, pp. 661–664.
8. P. Kabal, J.-L. Moncet and C. C. Chu, "Synthesis filter optimization and coding: Applications to CELP", *Proc. of IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, New York, NY, pp. 147–150, April 1988.
9. R. P. Ramachandran and P. Kabal, "Stability and performance analysis of pitch filters in speech coding", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-35, pp. 937–946, July 1987.
10. P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1419–1426, December 1986.
11. J. Grass and P. Kabal, "Methods of improving vector-scalar quantization of LPC coefficients", to appear in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Toronto, Ont., May 1991.
12. V. Iyengar and P. Kabal "A low delay 16 kbit/sec speech coder", to appear in *IEEE Trans. Signal Processing*, May 1991.
13. A. Cumani, "On a covariance lattice algorithm for linear prediction", *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Paris, May 1982, pp. 651–654.
14. M. Foodeei and P. Kabal, "Backward adaptive prediction: high-order predictors and formant-pitch configurations" to appear in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Toronto, Ont., May 1991.
15. B. Townshend, "Non-linear prediction of speech", to appear in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Toronto, Ont., May 1991.