



The Equivalence of ADPCM and CELP Coding

Peter Kabal

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

Version 1.2 March 2011

© 2011 Peter Kabal



You are free:



to Share – to copy, distribute and transmit this work



to Remix – to adapt this work

Subject to conditions outlined in the license.

This work is licensed under the *Creative Commons Attribution 3.0 Unported License*. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Revision History:

- 2011-03 v1.2: Creative Commons licence, noise feedback in non-predictive coders, corrections
- 2010-04: Initial release

1 Introduction

This document examines a coding schemes which differentially code signals while at the same time controlling the frequency characteristics of the coding (quantization) error. We show that a (vector quantized) version of an Adaptive Differential Pulse Code Modulation (ADPCM) system using noise feedback to shape the quantization noise can be converted to an equivalent system which is in the form of a Code Excited Linear Prediction (CELP) system. While this equivalence is known by, or at least not a surprise to, the signal processing cognoscenti, it is not widely appreciated by many others. We also try to add a historical perspective on the development of these systems.

In comparing CELP and ADPCM, we point out that the CELP error weighting scheme approach can be more general than that for the ADPCM system and so not all CELP systems have an equivalent ADPCM implementation. Nonetheless, most CELP systems in practical use do use weighting filters which can be converted back to an equivalent ADPCM noise feedback filter.

2 DPCM

Pulse Code Modulation (PCM) coding involves the direct quantization of an input signal sample-by-sample. Differential PCM (DPCM) is a variant in which the difference between the input signal and a predicted version of that signal is coded.¹ This scheme is shown in Fig. 1. The left side is the encoder; the right side is the decoder.

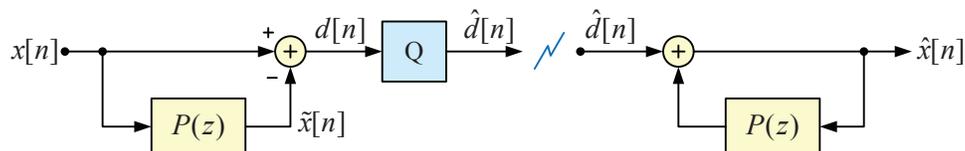


Fig. 1 DPCM System

2.1 Prediction

In the figure, we use z -transform notation for the filters. The predictor filter $P(z)$ creates an estimate of $x[n]$ based on past values. This implies that $P(z)$ does not have a delayless path. In a simple implementation, $P(z)$ could be equal to az^{-1} . Then if $a = 1$, $d[n]$ is the difference between $x[n]$ and $x[n - 1]$.

¹Our designations of DPCM and ADPCM systems differ a bit from what appears in the classic text on the digital coding of waveforms [11]. These systems are referred to in that text as D*PCM and DPCM, respectively.

In more complex systems, the filter $P(z)$ is a multi-coefficient filter whose coefficients are adapted to the local statistics of the input signal. The coefficients can be found, for instance, by solving for the linear predictor coefficients which minimize the mean-square error of the difference signal $d[n]$ (also known as the prediction residual). For low bit-rate coding of speech signals, a typical configuration uses a linear predictor with 10 coefficients which are updated every 10–30 ms.

The difference signal $d[n]$ is the output of the analysis filter (also known as the prediction error filter),

$$A(z) = 1 - P(z). \quad (1)$$

The received (quantized) signal $\hat{d}[n]$ is the *excitation* signal for the synthesis filter $1/A(z)$.

2.2 Quantizer

The quantizer is shown as a block which takes in the signal $d[n]$ and outputs the quantized signal $\hat{d}[n]$. The quantizer is a memoryless device, acting on the instantaneous value of its input signal. In more detail, the quantizer maps non-overlapping regions of input values to indices. In a communications scenario, it is the index which is transmitted. Reconstruction is by table look-up, mapping the index to an output value – here $\hat{d}[n]$. We elide the intermediate step of determining the index to show the quantizer as a block which directly creates the reconstructed value. Quantizers can also be adapted to scale the quantizer step size to the local statistics of the input signal – we will not explicitly show this scaling.

Since our main focus is on the coder side of a communications system, we do not consider the effect of errors introduced during the transmission of the quantizer values.

2.3 Coding Error

Now we are in a position to analyze the quantization error. In the time domain,

$$q[n] = d[n] - \hat{d}[n]. \quad (2)$$

We assume that the quantization error is uncorrelated with the input signal and independent from sample to sample. The quantization error is modelled as an additive white noise which is subtracted from the signal which appears at the input to the quantizer. Now we can express the output at the decoder as the sum of the input signal and a filtered quantization noise,

$$\begin{aligned} \hat{X}(z) &= \frac{1}{1 - P(z)} \hat{D}(z) \\ &= X(z) - \frac{1}{1 - P(z)} Q(z). \end{aligned} \quad (3)$$

From this relationship, we see that the frequency response of the quantization noise $Q(z)$ is shaped by the synthesis filter.

3 ADPCM

In DPCM the quantization noise is shaped by the synthesis filter. This may not be entirely desirable. In Adaptive DPCM (ADPCM), we adopt another structure. Here the adaptive part of the name implies that the prediction filter is adapted to the local characteristics of the input signal. It is generally understood that ADPCM refers to a structure with feedback around the quantizer as shown in Fig. 2. ITU-T Recommendation G.726 [7] describes a family of ADPCM coders. These ADPCM coders use pole-zero predictors; see Appendix A for a discussion of such predictors.

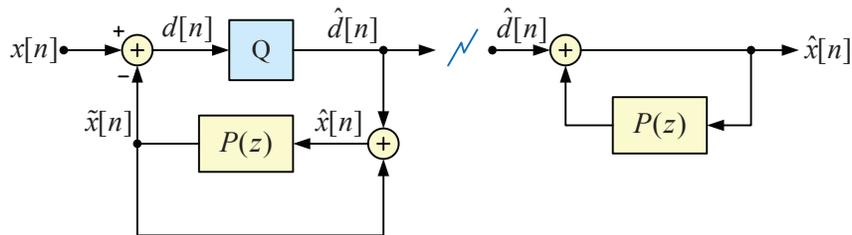


Fig. 2 ADPCM

We notice that the encoder includes a local decoder which produces the signal $\hat{x}[n]$ – the same signal which appears at the output of the decoder. The decoder for ADPCM is unchanged from that in the DPCM system. We note some relationships,

$$\hat{X}(z) = \hat{D}(z) + \tilde{X}(z) \quad (4)$$

$$D(z) = X(z) - \tilde{X}(z) \quad (5)$$

$$\hat{D}(z) = D(z) - Q(z). \quad (6)$$

Together these give the result that

$$\hat{X}(z) = X(z) - Q(z). \quad (7)$$

The effect of the feedback around the quantizer has been to make the output error the same as the error introduced by the quantizer alone. The result is that the noise on the reconstructed signal is white.

3.1 Noise Feedback Structure

With the result of Eq. (7), we see that the input to the filter $P(z)$ in the feedback loop around the quantizer is the sum of two components $-X(z)$ and $Q(z)$. The computations can be rearranged into a noise feedback version of the ADPCM coder as shown in Fig. 3. In this scheme, $X(z)$ and $Q(z)$ are processed separately through $P(z)$.

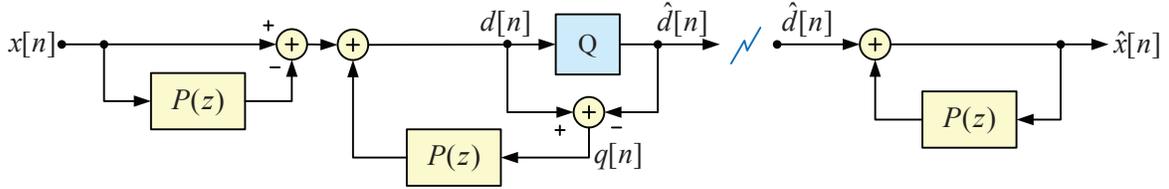


Fig. 3 ADPCM implemented using noise feedback

The noise feedback version of the ADPCM system has the prediction filter $P(z)$ in the noise feedback path. Consider a change to the noise feedback filter as shown in Fig. 4. The noise feedback filter is now a more general filter $N(z)$. With the more general noise shaping filter, the output of the system is (c.f. Eq. (7))

$$\hat{X}(z) = X(z) - \frac{1 - N(z)}{1 - P(z)} Q(z). \quad (8)$$

The filter $N(z)$ must not include a delayless path. One can see that if $N(z)$ is set equal to $P(z)$ (as in the basic form of ADPCM), the noise weighting becomes the identity filter. For speech and audio signals (processed by the human auditory system), it can be argued that the noise feedback filter should take advantage of the masking of noise by the frequency regions encompassing the more energetic signal components.

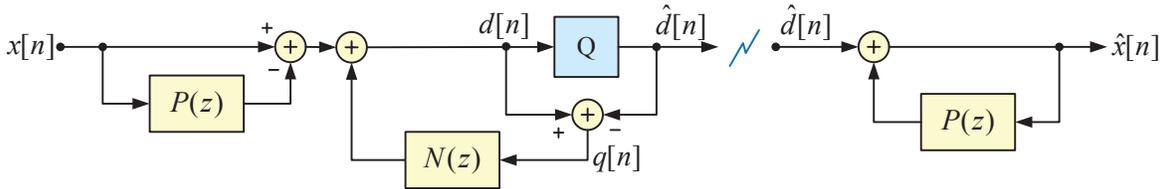


Fig. 4 ADPCM implemented using a general noise feedback filter

An early application of noise feedback applied to speech coding appeared as Adaptive Predictive Coding (APC) [1]. That system incorporated an additional prediction step to account for long term periodicities (pitch) in voiced speech. We will discuss long term prediction later in the context of CELP. In APC, quantization was carried out sample by sample, but the prediction filter was adapted frame-by-frame. A notable feature of APC is the use of the noise feedback to shape the quantization noise according to perceptual principles. This type of noise shaping carries over

to CELP coding as discussed later.

3.2 Noise Feedback in Non-Predictive Coding

Before venturing on towards vector quantization and CELP, we will take a digression to consider noise feedback in a non-predictive coder. If we remove the predictor for ADPCM in Fig. 4, we get the system shown in Fig. 5. The output of the system is

$$\hat{X}(z) = X(z) - (1 - N(z))Q(z). \quad (9)$$

This system implements noise shaping.

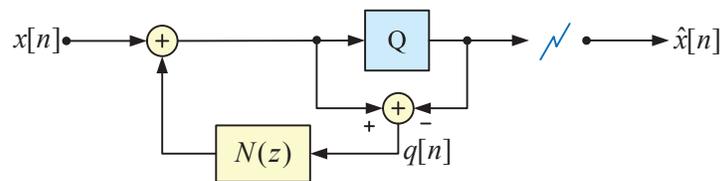


Fig. 5 Noise feedback in non-predictive coding

A restructured block diagram is shown in Fig. 6. One can note that the input signal is filtered by $1 - F(z)$ and the signal component of the output is fed back to cancel the effect of the $F(z)$ term. This leave only the noise filtered. The output of the system is

$$\hat{X}(z) = X(z) - \frac{1}{1 - F(z)}Q(z). \quad (10)$$

The systems shown in Fig. 5 and Fig. 6 are equivalent if we set

$$N(z) = -\frac{F(z)}{1 - F(z)}. \quad (11)$$

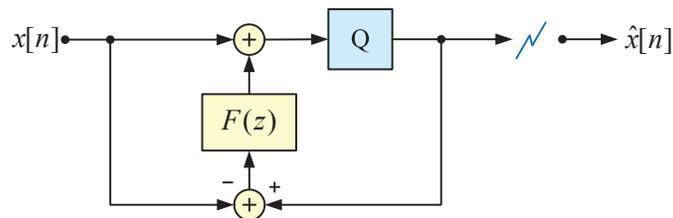


Fig. 6 Modified noise feedback in non-predictive coding

Noise shaping using the implementation in Fig. 6 is described in [13]. Noise shaping in a non-

predictive coder is part of the ITU-T G.711.1 standard [8]. The filter $F(z)$ in G.711.1 is based on a fourth order linear prediction (LP) analysis.

3.3 Vector Quantization in ADPCM

Vector quantization (VQ) involves the coding of a group of samples at one time, rather than coding sample by sample. Vector quantization was applied to ADPCM/APC systems in [3]. Two recent papers have revisited ways of looking at ADPCM in which the coder processes a vector at a time [4][16].

The use of VQ approach can be motivated by treating the quantizer as a codebook of values. For the vector quantization case, one looks at all possible sequences (vectors) of quantizer outputs for a time interval (a frame of data), and chooses the sequence of quantizer outputs which minimize the error in the frame. For instance, the error in the frame can be measured as the sum of squared errors over the frame.

To emphasize the change to vector quantization, Fig. 7 shows the quantizer output being taken from a codebook. The error calculation is shown as a block labelled MSE. Consider first the scalar version, i.e., the codebook consists of the possible quantizer outputs. In principle, each of the possible quantizer outputs $\hat{d}[n]$ is compared to $d[n]$, and that which gives the smallest quantization error is chosen. This is quantization to the nearest quantizer output level. In the case of a vector quantizer, the codebook emits a sequence of M samples. Each such vector of samples is then filtered with noise feedback and the codebook vector which minimizes the sum of M squared errors is chosen.

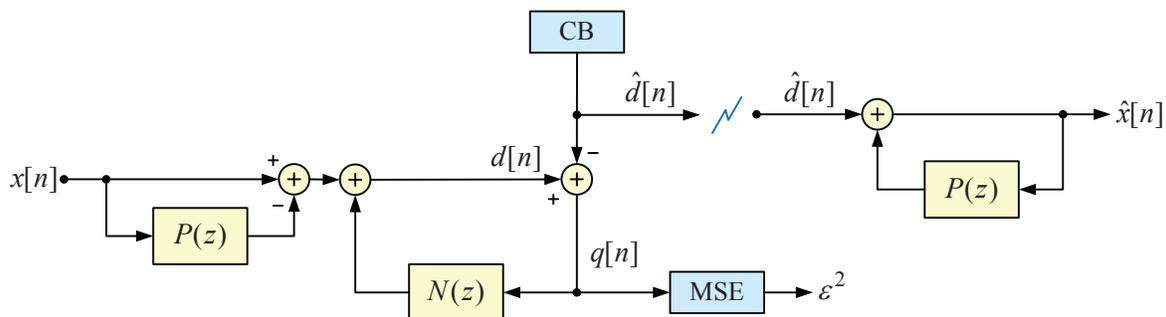


Fig. 7 Vector quantized ADPCM using noise feedback

3.4 Rearrangements of Noise Feedback ADPCM

We will manipulate the noise feedback ADPCM block diagram in several steps. In Fig. 8 we have encapsulated the filters into meta blocks.

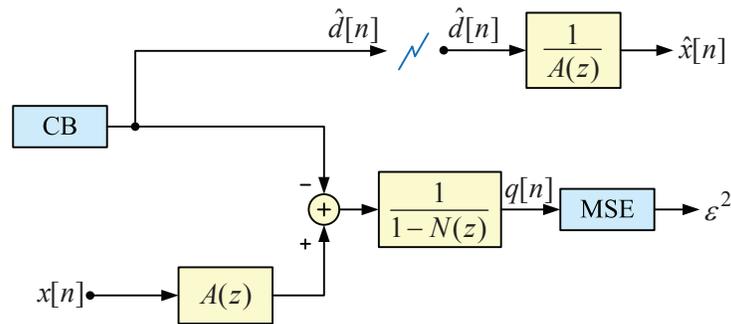


Fig. 8 Vector quantized ADPCM using noise feedback – rearrangement A

In the next step we merge two of the filters into a weighting filter $A(z)/(1-N(z))$ as shown in Fig. 9. To compensate for the inclusion of $A(z)$ into the weighting filter, the synthesis filter $1/A(z)$ is added to the signal from the codebook. In effect, we have created a locally decoded signal $\hat{x}[n]$ which is compared to the input signal and then the difference is weighted to form the error signal $q[n]$. In the case of time-varying filters – the adaptive case – care must be taken to use compatible filter structures to ensure that $1/A(z)$ in cascade with $A(z)$ gives an identity system. We also must be careful in the way the combined weighting filter $A(z)/(1-N(z))$ is implemented to ensure that with time varying coefficients, the system gives the same result as the ADPCM system with noise feedback.

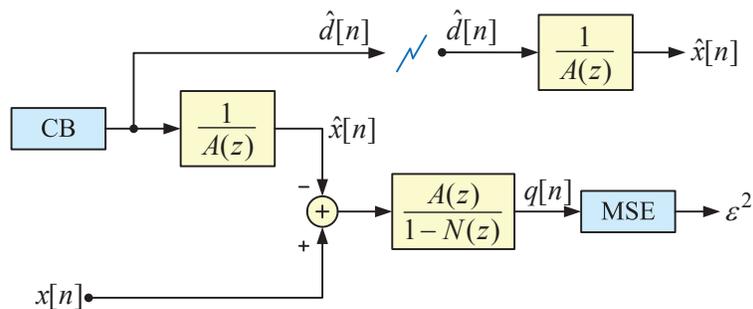


Fig. 9 Vector quantized ADPCM using noise feedback – rearrangement B

The first rearrangement of the ADPCM system, Fig. 8, forms the difference between the codebook output and the residual signal ($x[n]$ passed through $A(z)$). The second rearrangement of the ADPCM system, Fig. 9, forms the difference between reconstructed signal (codebook processed through $1/A(z)$) and the original input signal.

4 CELP

Code-Excited Linear Prediction (CELP) coding has become the paradigm of choice for speech coding in mobile communications and low bit-rate Voice-over-IP. CELP coding can be motivated by postulating a structure at the decoder. This design approach for CELP can be considered to extend parametric coders.

In a parametric coder, the structure of the decoder is specified and the task of the encoder is to determine the parameters for the structural elements. A classic example is Linear Predictive (LP) coding. The decoder uses an all-pole filter (modelling the human vocal tract), excited by either a periodic pulse train (voiced speech), or a noise-like signal (unvoiced speech). The coding of the excitation signal to the synthesis filter is simple: a gain value, a pitch period, and a voiced-unvoiced decision. The synthesis filter is the same as used in the ADPCM models. Well before the advent of CELP, LP coding had reached maturity, for example as described in the 1976 text by Markel and Gray [14]. The ideas of modelling the human vocal tract using linear prediction (the $P(z)$ filter) became implanted in the adaptive versions of ADPCM and APC.

A more sophisticated modelling of the excitation signals appeared as multipulse coding [2]. In this scheme, the dichotomy between a pure noise-like and a pure periodic signal is broken. An appropriate excitation in the form of pulses specified by amplitudes and positions is determined within an analysis-by-synthesis procedure. By analysis-by-synthesis, we mean that a search for the best pulse amplitudes and positions is found by synthesizing the output for trial pulses. The pulses which best match the original speech are chosen. The pulses can then model voiced speech (periodic pulse placement), unvoiced speech (aperiodic pulse placement), and mixed excitation. This procedure fits into the codebook view of ADPCM as shown in Fig. 9. Multipulse coding uses the weighting filter as shown in that figure. The original form of multipulse did not use a pitch synthesis filter.

In its simplest form, the CELP decoder is exactly the same as that for the DPCM and ADPCM systems that we have seen. One key to CELP is the use of vector quantization instead of the sample-by-sample scalar quantization typically used for ADPCM. CELP is often described as an Analysis-by-Synthesis system.

The block diagram of a CELP system (without pitch filtering) is the same as shown in Fig. 9. However, CELP coders generally also employ a synthesis filter which inserts pitch periodicity into the reconstructed signal. This is described in a subsequent subsection.

4.1 Calculation of the Error Signal

There are a number of practical considerations for frame based processing. These also apply to the vector quantized ADPCM systems.

- A non-zero internal state of the synthesis and weighting filters results in an output even without an input from the codebook. A simplification of the computations arises if we pre-calculate a target signal. This target signal is $x[n]$ with the zero-input response of the filters subtracted from it. In addition, pre-computing the response of the weighting filter to $x[n]$ is useful as this response does not depend on the codebook entry selected. These manipulations are described in detail as they apply to a practical speech coding standard in [12].
- The form of the weighting filter is typically derived from the analysis filter $A(z)$. For instance, a common form of the weighting filter is

$$W(z) = \frac{A(\gamma_1 z)}{A(\gamma_2 z)}, \quad (12)$$

For the coder describe in [12], $\gamma_1 = 0.9$ and $\gamma_2 = 0.5$. A plot of the response of this weighting filter for a frame of speech is shown in that report.

4.2 Pitch Modelling

CELP coders use an additional step in the synthesis process. In a basic implementation, this step is an all-pole filter with a single coefficient with a delay of L samples. This filter adds in a scaled replica of the past excitation delayed by L samples,

$$A_L(z) = \frac{1}{1 - g_p z^{-L}}. \quad (13)$$

The overall excitation signal $\hat{d}[n]$ is the sum of a fixed contribution $\hat{d}_i[n]$ (the i th vector) from a fixed codebook and an adaptive contribution from the past excitation signal,

$$\begin{aligned} \hat{d}[n] &= \hat{d}_i[n] + \hat{d}_L[n] \\ &= \hat{d}_i[n] + g_p \hat{d}[n - L]. \end{aligned} \quad (14)$$

The delay L is kept constant for an interval of time, then allowed to change, often at a subframe interval. The fixed codebook contribution is also scaled (not shown explicitly), so that the $\hat{d}[n]$ is a scaled mixture of two contributions. To model a periodic signal, g_p is near unity and L is the period.

The signal $\hat{d}_L[n]$ can be considered to be the output from an Adaptive Codebook (ACB). The elements of the codebook are vectors indexed by the lag L and scaled by the gain g_p . The new arrangement with two codebooks is shown in Fig. 10.

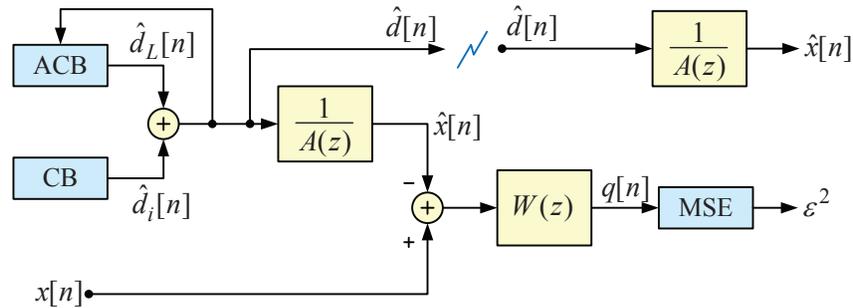


Fig. 10 CELP coder with a fixed codebook (CB) and an adaptive codebook (ACB)

4.3 Filter Updates

In CELP, the filter $1/A(z)$ is often referred to as the formant synthesis filter. This filter models the resonances of the human vocal tract – these resonances are known as the formants. The filter $A(z)$ is determined by analyzing the windowed input signal $x[n]$. The coefficients of this filter are quantized for transmission. There is one set of coded coefficients per frame (typically 10–30 ms long). However, for synthesis, the coefficients can be interpolated between quantized sets to form a formant synthesis filter which varies more smoothly over a frame. See [12] for a system which uses a 30 ms frame subdivided into 4 subframes. For that system, the formant filter is interpolated every subframe.

The weighting filter which appears in Fig. 10 can be more general than the filter which appears in ADPCM (the first coefficient of the impulse response need not be unity). It can also include a filter based on the pitch structure of voiced speech (see [12]). It is to be noted the weighting filter which is used only at the encoder, can be based on unquantized formant filters which can be updated as often as needed.

The ACB parameters are usually determined using an analysis-by-synthesis approach.² To reduce computational complexity, the search of the adaptive and fixed codebooks is typically done sequentially. First one assumes zero output from the fixed codebook CB. For the adaptive codebook ACB, one then searches over pitch lags L , finding the best g_p for each. The combination of L and g_p which results in the best match to the signal to be coded is chosen. The fixed codebook search then ensues while the output of the adaptive codebook is kept constant. See [15] for a discussion of this sequential optimization approach.

²The ACB parameters can be determined by analyzing the residual signal after LP analysis. However, the analysis-by-synthesis procedure which optimizes the pitch synthesis filter directly, taking into account the (noisy) excitation signal, provides a better match. Reference [15] has a comparison of these alternatives.

4.4 Adaptive Codebook

Problems present themselves when the lag L is smaller than the frame length. If L is larger than the frame length, past excitation values are available. In that case, the vectors in the adaptive codebook are the frame length chunks of the past excitation values, each vector being delayed by a different value of L . For L less than the frame length, one can extend the adaptive codebook vector entries by repeating samples with period L . The result can be described as using a pitch synthesis filter with time-varying lag – specifically, the lag starts out at L , and after generating L samples, jumps to $2L$, and so on.

Some CELP coders use more than a single filter coefficient for the pitch filter (the scheme described in [12] uses 5 coefficients). One can also interpolate between past excitation samples to effectively give fractional delays.

5 Tree Encoding

CELP coding operates frame-by-frame. We have seen that the state of the filters at the end of a frame affects the results in the next frame. The frame-based error criterion does not fully take these effects into account. An alternate is tree-based encoding. This scheme can be described as delayed-decision coding. One does not commit to the coding value until one has the opportunity to look at its consequences on future outputs. The decision delay implements a sliding window.

Gibson in [6] reviews the state of the art in speech coding just before the introduction of CELP. In a section on tree coding, he notes that “Any of the adaptive prediction, filtering, and noise spectral shaping algorithms discussed in the previous sections can be employed to adapt the code generator ...”. This motivates the use of adaptive noise shaping in a non-scalar quantizer context. A version of tree coding was described in [17]. This coder used a tree-search *within* a frame of data. It also used a pitch synthesis filter and applied perceptually-motivated adaptive weighting. An example of a true tree coder which employed adaptive noise shaping appears in [9, 10]. An example of tree encoding that replaces the block codes used in CELP by the sliding codes of tree coding appears in [5].

6 Summary

This report has shown that noise feedback ADPCM can be used with a vector quantizer. With vector quantization, the structure of the system can be recast to be identical with a CELP system. The noise feedback in ADPCM manifests itself as a weighting filter in CELP. CELP usually is implemented as having two codebooks – a fixed codebook for noise-like excitations and an adaptive codebook to insert periodicity into the excitation signal. These two codebooks can be mapped directly back into the vectorized form of ADPCM.

Appendix A Pole-Zero Predictors

In a block diagram such as Fig. 1, one might assume that the predictor $P(z)$ is an FIR filter. With this assumption the prediction error filter $A(z) = 1 - P(z)$ is an all-zero filter, and the synthesis filter $1/A(z)$ is an all-pole filter. But in fact the system is more general than that. We will consider the case of pole-zero predictors such as used in the ITU-T standard ADPCM coder [7][11].

With pole-zero predictor, the prediction occurs in a DPCM coder as the cascade of two predictors as shown in Fig. 11.

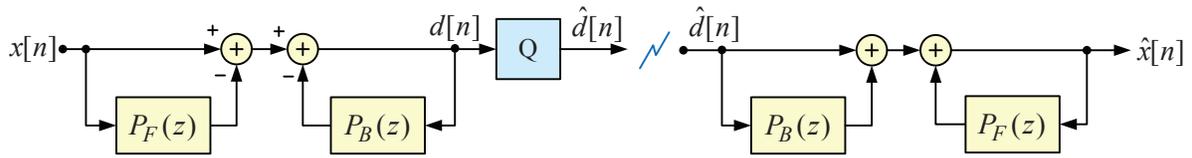


Fig. 11 DPCM with a pole-zero predictor

The response of the analysis filter consisting of the cascade of the two predictors can be written as

$$\begin{aligned} A(z) &= \frac{1 - P_F(z)}{1 + P_B(z)} \\ &= 1 - \frac{P_F(z) + P_B(z)}{1 + P_B(z)}. \end{aligned} \quad (15)$$

With this rearrangement, we find the equivalent predictor in the form shown in Fig. 1 is

$$P(z) = \frac{P_F(z) + P_B(z)}{1 + P_B(z)}. \quad (16)$$

For the ADPCM case, we show the two predictors separately, although they can again be combined into the predictor $P(z)$. The form with separate predictor filters is shown in Fig. 12.

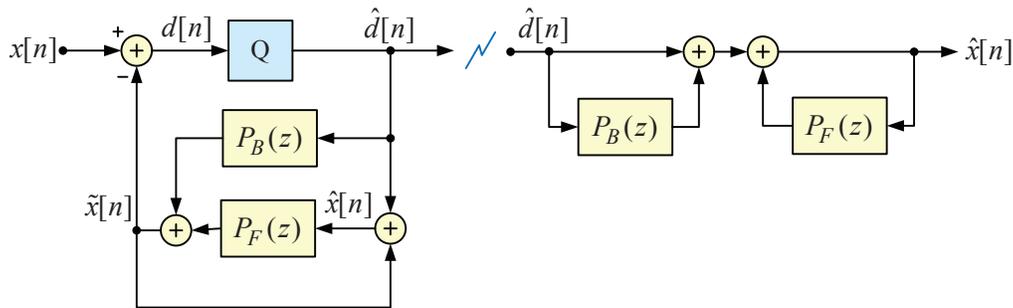


Fig. 12 ADPCM with a pole-zero predictor

References

- [1] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 27, no. 2, pp. 247–254, June 1979.
- [2] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rate", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Paris), pp. 614–617, May 1982.
- [3] J.-H. Chen and A. Gersho, "Vector Adaptive Predictive Coding of Speech at 9.6 kb/s", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Tokyo), pp. 1693–1696, May 1996.
- [4] J.-H. Chen, "Novel Codec Structures for Noise Feedback Coding of Speech" *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Toulouse, France), pp. I-681–I-684, May 2006.
- [5] C. C. Chu, *Tree Encoding of Speech Signals at Low Bit Rates*, M.Eng. Thesis, Dept. Electrical & Computer Engineering, McGill University, March 1986 (available on-line at www-mmsp.ece.mcgill.ca/Theses).
- [6] J. D. Gibson, "Adaptive Prediction in Speech Differential Encoding Systems", *Proc. IEEE*, vol. 68, pp. 488–525, April 1980.
- [7] ITU-T Recommendation G.726, 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM) Dec. 1990.
- [8] ITU-T Recommendation G.711.1, *Wideband embedded extension for G.711 pulse code modulation*, March 2008.
- [9] V. Iyengar, *A Low Delay 16 kbit/sec Coder for Speech Signals*, M.Eng. Thesis, Electrical & Computer Engineering, McGill University, Aug. 1987 (available on-line at www-mmsp.ece.mcgill.ca/MMSP/Theses).
- [10] V. Iyengar and P. Kabal "A Low Delay 16 kb/s Speech Coder", *IEEE Trans. Signal Processing*, vol. 39, no. 5, pp. 1049–1057, May 1991.
- [11] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984 (ISBN 978-0-13-211913-9).
- [12] P. Kabal, *ITU-T G.723.1 Speech Coder: A Matlab Implementation*, Technical Report, Dept. Electrical & Computer Engineering, McGill University, July 2009 (available on-line at www-mmsp.ece.mcgill.ca/Documents).
- [13] J. Lapierre, R. Lefebvre, B. Bessette, V. Malenovsky and R. Salami, "Noise Shaping in an ITU-T G.711-Interoperable Embedded Codec", *Proc. Eur. Signal Processing Conf.* (Lausanne, Switzerland), 5 pp., Aug. 2008.
- [14] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, 1976.

-
- [15] J.-L. Moncet and P. Kabal, *Codeword Selection for CELP Coders*, Technical Report, INRS-Telecommunications, University of Quebec, July 1987 (available on-line at www-mmsp.ece.mcgill.ca/Documents).
- [16] J. Thyssen and J.-H. Chen, "Efficient VQ Techniques and General Noise Shaping in Noise Feedback Coding", *Proc. Interspeech 2006 - ICSLP* (Pittsburgh, PA), pp. 221–224, Sept. 2006.
- [17] M. R. Schroeder and B. S. Atal, "Speech Coding Using Efficient Block Codes", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Paris), pp. 1668–1671, May 1982.