ISSN: 2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Simulation of Conjugate Structure Algebraic Code Excited Linear Prediction Speech Coder

Ritisha Virulkar[1], A.P.Khandait[1], Gautam Bacher[2], Abhijit.B.Maidamwar[3]
[1]PCOE, Nagpur
[2]BITS, Goa
[3]RGCER, Nagpur

Abstract : The CS-ACELP is a speech coder that is based on the linear prediction coding technique. It gives us the bit rate reduced to up to 8kbps and at the same time reduces the computational complexity of speech search described in ITU recommendation G.729. This codec is used for compression of speech signal. The idea behind this algorithm is to predict the next coming signals by the means of linear prediction. For his it uses fixed codebook and adaptive codebook. The quality of speech delivered by this coder is equivalent to 32 kbps ADPCM. The processes responsible for achieving reduction in bit rate are: sending less number of bits for no voice detection and carrying out conditional search in fixed codebook.

Keywords: 8 kbps algorithm, codebook search, CS-ACELP

I INTRODUCTION

The ITU-T standardized 8 kbits/s speech codec to operate with a discrete-time speech signal. G.729 provides coding of speech signals used in multimedia applications at 8 kbits/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP) [1][2]. The quality of speech produced by our coder is equivalent to a 32 kbits/s ADPCM for most operating conditions. These conditions include clean and a noise containign speech, multiple levels of encoding, variations in level and non-speech inputs.The typical input rates are mu-law or A-law **64** kbits/s PCM or 128 kbit/s linear PCM providing a compression ratio of 16:l. The coder designed is robust against channel errors. This means that the coder should be able to withstand these errors without introducing any major effects. Also if radio channels suffer from long distance fades and complete frames are lost then with minimum loss in the quality of speech the decoder should be able to retain those missing frames.

The coder generally breaks up the speech into small units called frames. For each speech frame a set of parameters are generated and are sent to the decoder. This signifies that the frame time represents a lower bound on the system delay and the encoder must wait for at least a frame worth of speech before it can even begin the encode process. Then the input signal is passed through a preprocessing block which consists of a high pass filter. A 10th order linear prediction analysis gives a set of coefficients called the LP filter coefficients.These are further converted to Line Spectrum Pair (LSP) coefficients and are quantized using Vector Quantization (VQ).

The excitation signal is chosen and an open-loop pitch delay is estimated with a speech signal that is perceptually weighted and low-pass filtered.This speech codec’s relative low complexity makes it an attractive choice for Internet telephony.

The algorithm can be divided into two sections. Section I will describe the CS-ACELP encoder and Section II will describe the CS-ACELP decoder. The encoder can be subdivided into various parts:

a. Preprocessing
b. Linear Prediction Analysis
c. Open loop pitch search
d. Closed loop pitch search
e. Fixed codebook search
f. Memory update

A. Preprocessing

A 16 bit pulse code modulated signal is assumed to be the input to the encoder. But before encoding the signal is needed to pass through two preprocessing blocks. They are:

1) Signal scaling

2) high-pass filtering

The scaling process consists of dividing the input signal by a factor 2 so that the possibility of overflows in the fixed-point implementation is reduced. The high-pass filter is used as a precaution against the undesired components that are of low frequency. A second order filter of pole/zero type with a cut-off frequency of 140 Hz is used. Both the processes of scaling and high-pass filtering are combined together by dividing the coefficients at the numerator of this filter by 2. And we get the resulting filter which is is given by:

$$H_{h1}(z) = \frac{0.46363718 - 0.92724705z^{-1} + 0.46363718z^{-2}}{1 - 1.9059465z^{-1} + 0.9114024z^{-2}} \qquad (1)$$

This input signal that is filtered through $H_{h1}(z)$ is referred to as $s(n)$, and is used further in all the subsequent coder operations.

B. Linear Prediction Analysis

In the LP analysis the redundancy in the speech signal is exploited. The primary objective of LP analysis is to compute the LP coefficients which minimized the prediction error. The popular method for computing the LP coefficients is autocorrelation method. This achieved by minimizing the total prediction error. The short-term analysis and synthesis filters are based on 10th order linear prediction (LP) filters. The LP synthesis filter is defined as:

$$\frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{10} \hat{a}_i z^{-i}} \qquad (2)$$

where $\hat{a}_i$, i = 1,...,10, are the (quantized) linear prediction (LP) coefficients. The short-term prediction, or linear prediction analysis is performed once per speech frame using the autocorrelation method with a 30 ms asymmetric window. After every 80 samples (10 ms), the autocorrelation coefficients of windowed speech are computed and are converted to the LP coefficients making use of the Levinson-Durbin algorithm. Then these LP coefficients are transformed to the LSP domain for quantization and interpolation purposes. The quantized interpolated and unquantized filters are converted back to the LP filter coefficients (to construct the synthesis and weighting filters for each subframe).

$$\frac{\mathbf{1}}{\hat{A}(z)} = \frac{\mathbf{1}}{\mathbf{1} + \sum_{i=1}^{10} \hat{a}_i z^{-i}}$$

C. Open loop pitch search

The input signal is passed through high-pass filter and is scaled in the pre-processing block. This pre-processed signal act as an input signal for all the further analysis. LP analysis is performed once for per 10 ms frame for purpose of the computation of the LP filter coefficients. These LP coefficients are then converted to Line Spectrum Pair (LSP) coefficients and are quantized using predictive two-stage Vector Quantization (VQ) with 18 bits [3][4]. By using an analysis-by-synthesis search procedure in which the error between the original and reconstructed speech is minimized according to a perceptually weighted distortion measure, the excitation signal is chosen. To do this the error signal is filtered with a perceptual weighting filter, the coefficients of which can be derived from the unquantized LP filter. The perceptual weighting is made adaptive so that the performance for input signals with a flat frequency response is improved. The excitation parameters (fixed and adaptive codebook parameters) are determined per sub-frame of 5 ms (40 samples) each. The LP filter coefficients (both quantized and un-quantized) are used for the second sub-frame, whereas in the first sub-frame interpolated LP filter coefficients (both quantized and un-quantized) are used . An open-loop pitch delay denoted by T_{OP} is estimated once per 10 ms frame by using the perceptually weighted speech signal $S_w(n)$ [1][2].

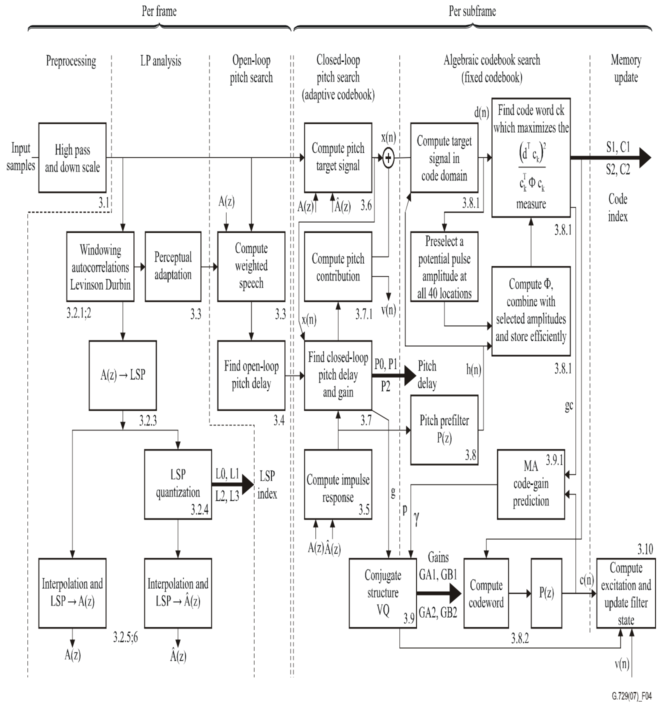

Per frame
Per subframe
Preprocessing
LP analysis
Open-loop pitch search
Closed-loop pitch search (adaptive codebook)
Algebraic codebook search (fixed codebook)
Memory update
Input samples
High pass and down scale
3.1
Compute pitch target signal
x(n)
A(z)
Â(z)
3.6
Compute target signal in code domain
3.8.1
d(n)
Find code word ck which maximizes the
$\frac{(d^T c_k)^2}{c_k^T \Phi c_k}$
measure
3.8.1
S1, C1
S2, C2
Code index
Windowing autocorrelations Levinson Durbin
3.2.1;2
Perceptual adaptation
3.3
A(z)
Compute weighted speech
3.3
Compute pitch contribution
3.7.1
v(n)
x(n)
Preselect a potential pulse amplitude at all 40 locations
Compute Φ, combine with selected amplitudes and store efficiently
3.8.1
A(z) → LSP
3.2.3
Find open-loop pitch delay
3.4
Find closed-loop pitch delay and gain
3.7
P0, P1
P2
Pitch delay
h(n)
gc
Pitch prefilter P(z)
3.8
LSP quantization
3.2.4
L0, L1
L2, L3
LSP index
Compute impulse response
3.5
A(z) Â(z)
g
p
γ
MA code-gain prediction
3.9.1
Interpolation and LSP → A(z)
Interpolation and LSP → Â(z)
3.2.5;6
A(z)
Â(z)
Conjugate structure VQ
3.9
Gains GA1, GB1
GA2, GB2
Compute codeword
P(z)
c(n)
3.10
Compute excitation and update filter state
3.8.2
v(n)
G.729(07)_F04

Figure 1:- Block diagram of CS-ACELP Encoder

The weighted speech signal $S_w(n)$ is used for the open loop pitch lag estimation.

$$R(k) = \sum_{n=0}^{79} s_w(n) s_w(n-k)$$

The three maxima of the correlation are found and they are in following three ranges; (20:39), (40:79), (80:143). The open loop pitch is obtained by taking the maxima of the three ranges by using the normalized autocorrelation function.

$$R'(t_i) = \frac{R(t_i)}{\sqrt{\sum_n s_w^2(n-t_i)}}, \quad i = 1, \cdots, 3.$$

For one frame, the total operations required are 10160 multiplications, 10033 additions, 123 comparisons, 3 radical and 3 division operations and estimate the open loop pitch.

The computation of the pitch is dependent on the voiced and the unvoiced signal. The pitch contour lies in the voiced signal only. The weighted delta-LSP function (Wd) is used to differentiate between voice and unvoiced signal. The function Wd is given by:

$Wd = \sum_{k=1}^{10} w_k * [LSP_i(k) -$

$LSPi-1k2$

If the value of Wd is greater than some pre-defined threshold, then the open loop pitch lag is estimated otherwise the pitch value is taken as same as that of previous frame. The $LSP_i(k)$ is the LSP coefficient of the kth order at the ith frame and w$_k$ is the weighted coefficient [5]. Hence the calculations that are required in this are automatically reduced.

D. Closedloop pitch search

For good performance of the CELP algorithm at an intermediate bit rate either a closed or an open pitch loop is essential. The closed pitch loop can be called as an adaptive codebook of overlapping candidate vectors. Either a method called the endpoint correction or the energy recursion method can be applied to the closed pitch loop, as both these procedures take advantage of the overlapping nature of the codebook and are not affected by its dynamic character. Closed-loop pitch analysis is then done (to find the adaptive-codebook delay and gain), using the target signal $x(n)$ and impulse response $h(n)$, by searching around and estimating the value of the open-loop pitch delay. A fractional pitch delay having a resolution of 1/3 is used. The pitch delay is encoded with 8 bits in the first subframe and is differentially encoded with 5 bits in the second subframe

E. Fixed codebook search

The fixed codebook usually occupies 17 bits. The case where it takes 11 bits can be considered as mentioned in [4]. The pulse positions of the first two pulses are each encoded with the help of three bits, whereas the third pulse position is encoded with the help of four bits. The global sign for the three pulses is encoded with one bit. The first two pulses in the sequence have fixed amplitudes of +1, and the last pulse has fixed amplitude of -1.

Pulse	Sign	Positions
i_0	s_0: ±1	m_0:0, 5, 10, 15, 20, 25, 30, 35
i_1	s_1: ±1	m_1:1, 6, 11, 16, 21, 26, 31, 36
i_2	s_2: ±1	m_2:2, 7, 12, 17, 22, 27, 32, 37
i_3	s_3: ±1	m_3:3, 8, 13, 18, 23, 28, 33, 38 4, 9, 14, 19, 24, 29, 34, 39

Table 1:- Fixed codebook search structure

F. Memory Update

The states of the synthesis and weighting filters are needed to be updated to compute the target signal in the next subframe. After quantizing the two gains, the excitation signal denoted by $u(n)$, in the present subframe is obtained using the equation:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \quad n = 0,...,39$$

where $\hat{g}_p$ are the quantized adaptive-codebook gains and $\hat{g}_c$ are fixed-codebook gains, $v(n)$ is the vector of adaptive-codebook (past interpolated excitation), and $c(n)$ is the vector of fixed-codebook including harmonic enhancement. The filter states can be updated by filtering the signal $r(n) - u(n)$ (difference between residual and excitation) through the filters $1/\hat{A}(z)$ and $A(z/\gamma_1)/A(z/\gamma_2)$ for the 40 sample subframe and saving the states of the filters. This would require three operations of the filter. A simpler approach, that requires only one filter operation, is as follows. The locally reconstructed speech $\hat{s}(n)$ is computed by filtering the excitation signal through $1/\hat{A}(z)$. The filter output due to the input $r(n) - u(n)$ is equivalent to $e(n) = s(n) - \hat{s}(n)$. So the states of the synthesis filter $1/\hat{A}(z)$ are given by $e(n)$, $n = 30,...,39$. Updating the filter states $A(z/\gamma_1)/A(z/\gamma_2)$ can be done by filtering the error signal $e(n)$ through this filter to find the error $ew(n)$ which is perceptually weighted. However, the signal $ew(n)$ can also be found by:

$$ew(n) = x(n) - \hat{g}_p y(n) - \hat{g}_c z(n)$$

Since the signals $x(n)$, $y(n)$ and $z(n)$ are now available, the weighting filter states are updated by computing $ew(n)$ as in equation (76) for $n = 30,...,39$. This saves two filter operations.

II BIT ALLOCATION OF THE 8 KBIT/S CS-ACELP ALGORITHM

The CS-ACELP coder is based on the code-excited linear prediction (CELP) coding model. This coder operates on 10 ms speech frames that corresponds to 80 samples at a sampling rate of 8000 samples per second. For each frame of 10 ms, the speech signal is analyzed to extract the parameters of the CELP model (linear prediction filter coefficients, the indices and gains of adaptive and fixed-codebook). These parameters are then encoded and further transmitted. The bit allocation of the coder parameters is shown in Table 1. At the decoder, these filter parameters are used to retrieve the excitation and synthesis filter parameters. The speech signal is reconstructed by filtering this excitation through a filter called the short-term synthesis filter, as shown in Figure 1. The short-term synthesis filter is based on a 10th order linear prediction (LP) filter. The long-term, or pitch synthesis filter is implemented using the approach of adaptive-codebook. After the computation of the reconstructed speech, it is passed through a postfilter to further enhanced its properties.

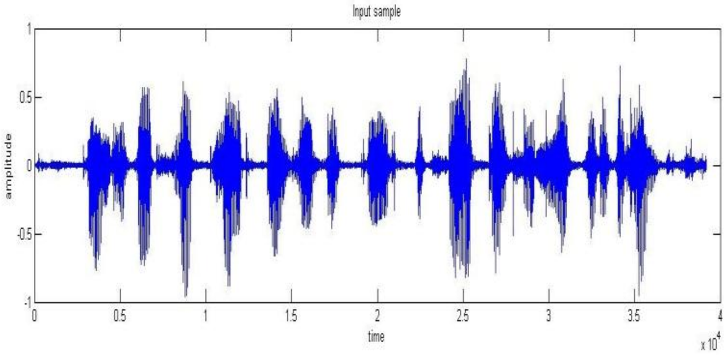
Parameter	**Code-word**	**Sub-frame 1**	**Sub-frame 2**	**Total per frame**
Line spectrum pairs	*L*0, *L*1, *L*2, *L*3			18
Adaptive-codebook delay	*P*1, *P*2	8	5	13
Pitch-delay parity	*P*0	1		1
Fixed-codebook index	*C*1, *C*2	13	13	26
Fixed-codebook sign	*S*1, *S*2	4	4	8
Codebook gains (stage 1)	*GA*1, *GA*2	3	3	6
Codebook gains (stage 2)	*GB*1, *GB*2	4	4	8
Total				80

Table2:- Bit allocation of CS-ACELP algorithm for 8 kbit/s

III CONCLUSION AND SIMULATION RESULT

This coder is designed to operate with a digital signal which is obtained by first performing telephone bandwidth filtering of the analogue input signal, then sampling it at 8000 Hz, and is followed by conversion to 16-bit linear PCM for the input to the encoder. The output of the decoder is to be converted back to an analogue signal by similar method. Another input/output characteristics of the signal, like those specified by for 64 kbit/s PCM data, is needed to be converted to 16-bit linear PCM before encoding, or from 16-bit linear PCM to the appropriate format after decoding.

For simulation we used a matlab Software. The graph shows the original speech and the same type of graph is expected at the decoder output.



Graph1:- Original Speech

IV REFERENCES

[1] Salami et al: 'Design and Description of CS-ACELP: A toll quality 8kb/s speech coder', IEEE trans Speech Audio Process, 1996.

[2] ITU-T G.729: 'Coding of speech at 8 kb/s using CS-ACELP', 1996.

[3] Kataoka et al: 'An 8 kb/s speech coder based on conjugate structured CELP', IEEE int. conf. acoustic, speech, signal processing, 1993.

[4] kataoka et al: 'LSP and gain quantization for proposed ITU-T 8 kb/s speech coding standard', IEEE workshop on speech coding, 1995.

[5] Shaw Hwa Hwang: 'Computational improvement for G.729 standard', 2003.

[6] A. B. Roach, "Session Initiation Protocol (SIP) -specific event notification," RFC 3265, June 2002.

[7] A. Johnston, S. Donovan, R. Sparks, C. Cunningham, and K. Summers, "Session Initiation Protocol (SIP) Public Switched Telephone Network (PSTN) call flows," RFC 3666, December 2003.

[8] R. Sparks, "The Session Initiation Protocol (SIP) refer method," RFC 3515, April 2003.

[9] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-

end speech quality assessment of narrow-band telephone networks and speech codecs," Feb. 2001.

[10] ITU-T Recommendation P.862 Amendment 1, "Source code for reference implementation and conformance tests," March 2003.

[11] A. E. Conway, "Output-based method of applying PESQ to measure the perceptual quality of framed speech signals", in IEEE Wireless Communications and Networking Conference, Vol. 4, pp. 2521-2526, March 2004.

[12] Prof M Noor,Israr K., "Real-Time Implementation And Optimization Of ITU-T‘s G.729Speech Codec Running At8kbits/Sec Using CS-ACELP On TM-1000VLIW DSP CPU", Communications Magazine,IEEE, 1997, 35 (9) :82-91.

[13] Duttweiler D L., "Proportionate normalized least mean squares adaptation in echo cancellers", IEEE Transactions on Speech and Audio Processing, 2000, 8 (5) :508-518.

[14] Texas Instruments Incorporated, Codec Engine Application Developer User's Guide, www.ti.com, 2007.