Simulation of G.729 Speech Coding Algorithm for Wireless Application

Monika S.Yadav[1] Prof.Vinay Keswani[2]
[1]M.Tech. Student [2]Assistant Professor
[1,2]Department of Electronics Communication Engineering
[1,2]Vidarbha Institute Of Technology, Nagpur,India

Abstract— In modern communication systems, the number of users are increased rapidly to access the wired networks. Consequently the use of channel capacity has to be increased.G.729 is one of the widely used standard in ITU-T for Speech compression. Speech coding is the application of data compression of digital audio signal containing speech. Speech compression aims to compress the speech signal to attain maximum channel capacity with lower bit rate and highest quality. G.729 is one of the widely used standard in ITU-T for speech compression. The algorithm used for this is 8 kb/s speech coding algorithm. It is based on CS-ACELP(Conjugate Structure Algebric Code Excited Linear Prediction) coding technique. This algorithm predicts the next incoming signals by means of linear prediction. The CS-ACELP speech codec delivers quality of Speech equivalent to that of 32kb/s ADPCM for most operating conditions.

Key words: ITU-T G.729, Conjugate Structure Algebric Code Excited Linear Predictive Coding (CS-ACELP)

I. INTRODUCTION

Nowadays, Communication via internet(VOIP) is the most widely used technology for transmission of speech signal over packet switched IP networks.The G.729 is a low complexity continuous data transmission scheme for VOIP applications.It is an audio data compression algorithm for voice that compresses digital voice in packets of 10 milliseconds duration.G.729 operates at a bit rate of 8kb/s.It has low bandwidth requirements therefore it is mostly used in VOIP applications (such as skype), Videoconferencing, electronic toys, archiving, Digital simultaneous voice and data (DSVD), numerous computer based gaming and Multimedia applications, Most of the speech applications require minimum coding delay in order to avoid hindering the flow of the speech conversation because of long coding delays, A speech coder is one which converts a digitized speech signal into the coded representation and transmits it in a form of frames, At the receiving end, the speech decoder receives the coded frames and synthesizes reconstructed speech signal.

II. METHODOLOGY

The proposed method provides a better way of providing secured speech transmission to a reasonable extent. The data security can be accomplished during transmission by scrambling the speech samples before supplying to the speech encoder block, scrambling the output speech bit streams before transmission, and by changing the encoding scheme for codebook index only during speech sample processing. It uses the fixed codebook and adaptive codebook for this purpose.

A. *Cs-Acelp Architecture*

The proposed system architecture is shown in fig1.

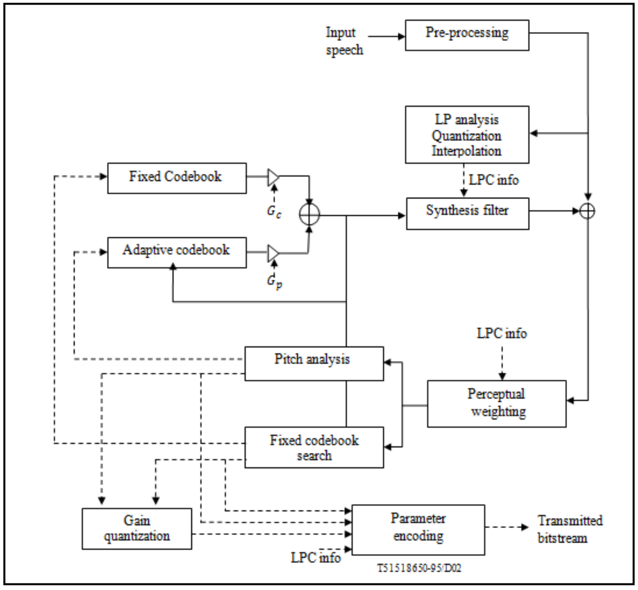

Fig. 1: CS-ACELP Encoder

The complete coder is devided into CS-ACELP encoder and CS-ACELP decoder. The encoder block consist of

1) Preprocessing

The input to the speech encoder is assumed to be a 16-bit PCM signal. Two preprocessing functions are applied before the encoding process:

a) Signal scaling; and
b) High pass filtering.

The high-pass filter serves as a precaution against undesired low-frequency components. A second order pole/zero filter with a cut-off frequency of 140 Hz is used.

2) Linear Prediction Analysis

Linear prediction analysis is performed once per speech frame using the autocorrelation method with a 30 ms asymmetric window. For every 80 samples (10 ms), the autocorrelation coefficients of windowed speech are computed and converted to the LP coefficients. Then the LP coefficients are transformed to the LSP domain for quantization and interpolation purposes. The input frame is divided into two subframes of 5ms each. The use of subframes allows better tracking pitch and gain parameters and reduces complexity of codebook searches. Once the LSF coefficients are quantized and interpolated they are converted back to LP coefficients. These LP coefficients are used to construct synthesis and weighting filters for each subframe. The perceptual weighting filter is based on unquantized LP filter coefficients. Its procedure often results in improvement in the speech coder performance.

3) Pitch Analysis

The weighted speech signal is used to find an estimation of the pitch delay in the speech frame.For each subframe the excitation is represented by contribution of both adaptive codebook and fixed codebook.The adaptive codebook approach is used to represent the periodic component in the excitation signal.It is searched by using a two step procedure.

a) Open loop pitch analysis: The open loop pitch analysis is done once per 10ms frame using weighted speech signal.This procedure is used to find the position of the pitch in the adaptive codebook so that the complexity of search is reduced.
b) Closed loop pitch analysis:The close loop pitch search is carried out to find index and gain of the signal.For each subframe,the signal to be matched referred as the target signal and the impulse response of the weighted synthesis filter are computed.
c) Algebric (Fixed) codebook search: Algebric codebook os used for the fixed codebook. These codebooks are deterministic codebooks in which the codebook vectors are determined from the transmitted index using simple algebra instead of lookup tables. This structure provides advantages such as less storage space, reduced search complexity and robustness. Each vector contains four nonzero pulses. These pulses can assume the specific amplitudes and positions as shown in table below.

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: Structure of fixed codebook

4) *Quantization of Gain*

The adaptive and fixed codebook gains are vector quantized. Vector quantization is a technique which allows modeling of probability density functions by distributions of prototype vectors. The gain codebook search is done by minimizing the mean squared weighted error between original and reconstructed speech.

5) *Memory Update*

The adaptive gain is vector quantized using a two stage conjugate structured codebook. Such a structure reduces both computational and memory requirements. The states of synthesis and weighting filters are updated to compute the target signal in the next subframe.

B. *CS-ACELP Decoder*

The decider consists of decoding and post processing. Decoding process generates the LP filter coefficients from the transmitted information with the same procedure as used in the encoder. In decoder the parameter indices are extracted from the received bit stream. These indices are decoded to obtain the coded parameter corresponding to a 10ms speech frame. The LSP coefficients are interpolated and converted to LP filter coefficient for each sub frame.

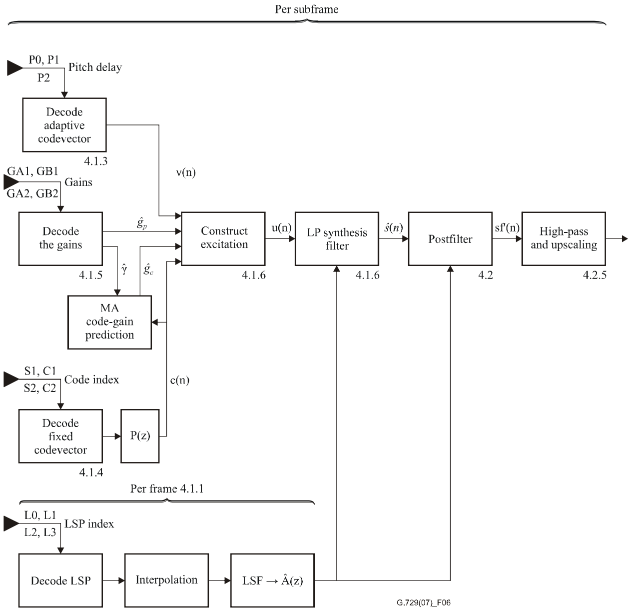

Fig. 2: Signal flow at the CS-ACELP decoder

The fig 2.shows the signal flow at the CS-ACELP decoder.The parameters LP coefficients,adaptive codebook vector,fixed codebook vector and gains are decoded.The decoded parameters are used to compute the reconstructed speech signal.This reconstructed signal is enhanced by a post processing operation consisting of a post filter,high pass filter and up scaling.For each subframe decoding of the adaptive and fixed codebook vector,decoding of the adaptive and fixed codebook gains,and the computation of the reconstructed speech takes place.

1) *Decoding Of the Adaptive Codebook Vector*

If no parity error has occurred,the received adaptive index P1 is used to find the integer and fractional parts of the pitch delay.

2) *Decoding Of the Fixed Codebook Vector*

The received fixed codebook index is used to extract the positions of the excitation pulses.The pulse signs are obtained from S.Once the pulse positions and signs are decoded,the fixed codebook vector c(n) is constructed.

3) *Computing the Reconstructed Speech*

The excitation $u(n)$] is input to the LP synthesis filter. The reconstructed speech for the subframe is given by:

$$\hat{s}(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i) \qquad n = 0,...,39$$

where $\hat{a}_i$ are the interpolated LP filter coefficients for the current subframe. The reconstructed speech $\hat{s}(n)$ is then processed by the post-processor.

a) Post-processing

Post-processing consists of three functions: adaptive postfiltering, high-pass filtering and signal upscaling. The adaptive postfilter is the cascade of three filters: a long-term postfilter $H_p(z)$, a short-term postfilter $H_f(z)$ and a tilt compensation filter $H_t(z)$, followed by an adaptive gain control procedure.

1) Long-term postfilter

The long-term postfilter is given by:

$$H_p(z) = \frac{1}{1+\gamma_p g_l}\left(1+\gamma_p g_l z^{-T}\right)$$

where T is the pitch delay, and g_l is the gain coefficient. Note that g_l is bounded by 1, and it is set to zero if the long-term prediction gain is less than 3 dB. The factor γ_p controls the amount of long-term postfiltering and has the value of $\gamma_p = 0.5$. The long-term delay and gain are computed from the residual signal $\hat{r}(n)$ obtained by filtering the speech $\hat{s}(n)$ through $\hat{A}(z/\gamma_n)$, which is the numerator of the short-term postfilter.

$$\hat{r}(n) = \hat{s}(n) + \sum_{i=1}^{10} \gamma_n^i \hat{a}_i \hat{s}(n-i)$$

The long-term delay is computed using a two-pass procedure. The first pass selects the best integer T_0 in the range [$int(T_1)$ – 1, $int(T_1)$ +1], where $int(T_1)$ is the integer part of the (transmitted) pitch delay T_1 in the first subframe. The best integer delay is the one that maximizes the correlation.

$$R(k) = \sum_{n=0}^{39} \hat{r}(n)\hat{r}(n-k)$$

The second pass chooses the best fractional delay T with resolution 1/8 around T_0. This is done by finding the delay with the highest pseudo-normalized correlation.

$$R'(k) = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sqrt{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)}}$$

where $\hat{r}_k(n)$ is the residual signal at delay k. Once the optimal delay T is found, the corresponding correlation $R'(T)$ is normalized with the square-root of the energy of $\hat{r}(n)$. The squared value of this normalized correlation is used to determine if the long-term postfilter should be disabled. This is done by setting $g_l = 0$ if: Otherwise the value of g_l is computed from:

$$g_l = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)} \quad \text{bounded by } 0 \le gl \le 1.0$$

The non-integer delayed signal $\hat{r}_k(n)$ is first computed using an interpolation filter of length 33. After the selection of T, $\hat{r}_k(n)$ is recomputed with a longer interpolation filter of length 129. The new signal replaces the previous one only if the longer filter increases the value of $R'(T)$.

2) Short-term postfilter

The short-term postfilter is given by:

$$H_f(z) = \frac{1}{g_f}\frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)} = \frac{1}{g_f}\frac{1+\sum_{i=1}^{10}\gamma_n^i \hat{a}_i z^{-i}}{1+\sum_{i=1}^{10}\gamma_d^i \hat{a}_i z^{-i}}$$

where $\hat{A}(z)$ is the received quantized LP inverse filter (LP analysis is not done at the decoder) and the factors γ_n and γ_d control the amount of short-term post filtering, and are set to $\gamma_n = 0.55$, and $\gamma_d = 0.7$. The gain term g_f is calculated on the truncated impulse response $h_f(n)$ of the filter $\hat{A}(z/\gamma_n)/\hat{A}(z/\gamma_d)$ and is given by:

$$g_f = \sum_{n=0}^{19} |h_f(n)|$$

3) Tilt compensation

The filter $H_t(z)$ compensates for the tilt in the short-term postfilter $H_f(z)$ and is given by:

$$H_t(z) = \frac{1}{g_t}\left(1 + \gamma_t k_1' z^{-1}\right)$$

where $\gamma_t k_1'$ is a tilt factor k_1' being the first reflection coefficient calculated from $h_f(n)$ with:

$$k_1' = -\frac{r_h(1)}{r_h(0)} \qquad r_h(i) = \sum_{j=0}^{19-i} h_f(j)h_f(j+i)$$

The gain term $g_t = 1 - |\gamma_t k_1'|$ compensates for the decreasing effect of g_f in $H_f(z)$. Furthermore, it has been shown that the product filter $H_f(z)H_t(z)$ has generally no gain. Two values for γ_t are used depending on the sign of k_1'. If k_1' is negative, $\gamma_t = 0.9$, and if k_1' is positive, $\gamma_t = 0.2$.

4) Adaptive gain control

Adaptive gain control is used to compensate for gain differences between the reconstructed speech signal $\hat{s}(n)$ and the postfiltered signal $sf(n)$. The gain scaling factor G for the present subframe is computed by:

$$G = \frac{\sum_{n=0}^{39} |\hat{s}(n)|}{\sum_{n=0}^{39} |sf(n)|}$$

The gain-scaled postfiltered signal $sf'(n)$ is given by:

$$sf'(n) = g^{(n)} sf(n) \qquad n = 0,\ldots,39$$

where $g^{(n)}$ is updated on a sample-by-sample basis and given by:

$$g^{(n)} = 0.85\, g^{(n-1)} + 0.15\, G \qquad \text{n} = 0,\ldots,39$$

The initial value of $g^{(-1)} = 1.0$ is used. Then for each new sub frame, $g^{(-1)}$ is set equal to $g^{(39)}$ of the previous sub frame.

5) High-pass filtering and upscaling

A high-pass filter with a cut-off frequency of 100 Hz is applied to the reconstructed post filtered speech $sf'(n)$. The filter is given by:

$$H_{h2}(z) = \frac{0.93980581 - 1.8795834\, z^{-1} + 0.93980581\, z^{-2}}{1 - 1.9330735\, z^{-1} + 0.93589199\, z^{-2}} \qquad (91)$$

The filtered signal is multiplied by a factor 2 to restore the input signal level.

III. RESULTS

Figure 3.1 shows input speech waveform of original signal. The original signal is of 4 second file of wav format. At encoder 4 seconds wave file are applied and through decoder we reconstruct the original wave file. This simulation has been done in Matlab software.

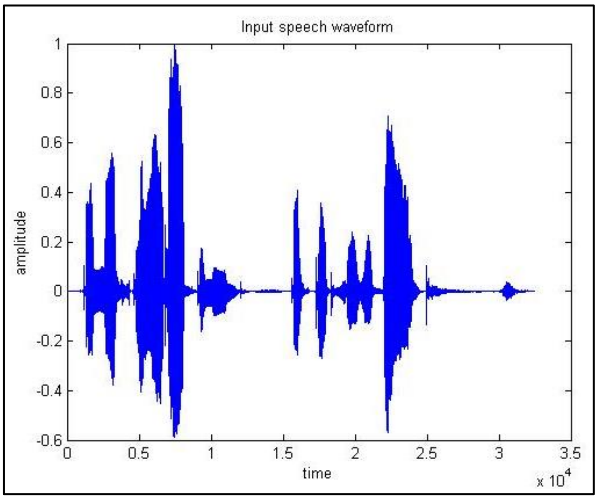

Fig. 3: Input speech waveform

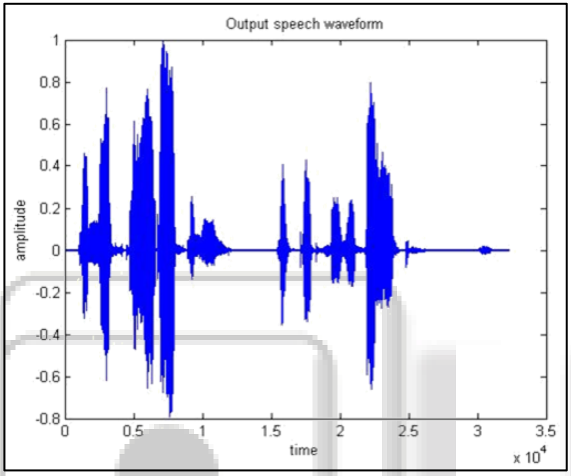

Fig. 4: Output speech waveform

While comparing the Input speech waveform and Output speech waveform, we recognized that there is no such change in both signal. The transferring rate of signal is 8 kbps from encoder to decoder. This means that our simulation method is correct. The speech quality produced by this coder should be equivalent to that of 32 kb/s ADPCM for most operating conditions.

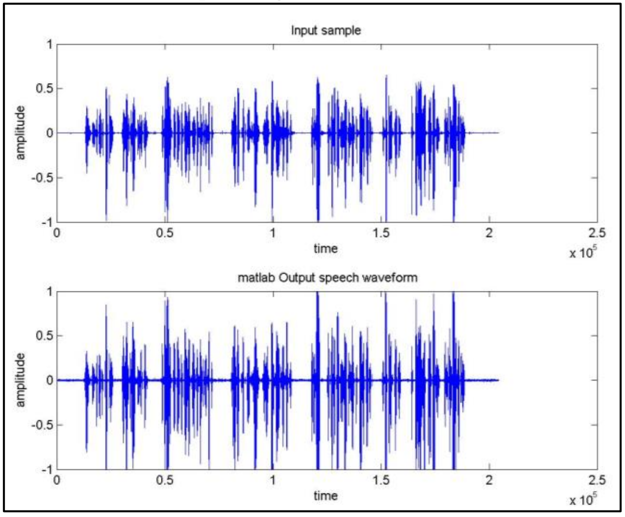

Fig. 5: Input & Output speech waveform

Figure 3.3 shows the input & output waveform of another signal in wav format. When we are applying input signal to the encoder, it is transfer the signal with the speed of 8 kbps. And we reconstruct our original signal from decoder. From figure 3.3 we can observe that there is no change in Input waveform and output waveform.

IV. CONCLUSION

The designed and developed 8 kbps was implemented through simulation and tested for different speakers with various conditions.The implementation and analysis of an efficient algorithm for providing secured speech transmission for various application with different speech input. From the experimental results, it is evident that the algorithm yields good compression and obtain very good perceptual quality. The overall performance targets low processing delay and low computational complexity with nearly same quality of speech synthesis and thus more suited for application in the real time implementation of the spectrally efficient speech coder. In matlab software we have provided the simulation.

REFERENCES

[1] Nimisha Susan Jacob,Ancy S. Anselam, " Performance Analysis Of CS-ACELP Speech Coder"(IJEAT) ISSN:2249-8958 ISSUE-5,June-2015.
[2] An Efficient Algebric Codebook Search for G.729 speech codec" IEEE,19 June 2014.
[3] Koji Seto and Tokunbo Ogunfunni," Scalable Wideband Speech Coding for IP Networks,"Dec.2012.
[4] T. Ogunfunmi and M. 1. Narasimha, "Speech over VoIP Networks:Advanced Signal Processing and System Implementation," IEEE Circuits and Systems Magazine, Vol. 12, no. 2, pp. 35 - 55, 2012.
[5] V. Kapur,M.M.Raghuvanshi,A.B.Maidamwar,"Real Time Implementation Of Speech Codec G.729 Using CS-ACELP TM 1000 VLIW Dsp Processor" IJSCE ISSN:2231-2307,Volume-1,Nov 2011.
[6] G. Madre , "Design of a variable rate algorithm for CS-ACELP coder", IEEE, 2003.
[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 Amendment 1, "Source code for reference implementation and conformance tests," March 2003.
[10] A. B. Roach, "Session Initiation Protocol (SIP) -specific event notification," RFC 3265, June 2002.
[11] 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. 20
[12] A. Kataoka , "A 6.4 kbit/s extension to the G.729 speech coder", IEEE, 1997.
[13] Salami , 'Design and Description of CS-ACELP: A toll quality 8kb/s speech coder', IEEE trans Speech Audio Process, 1996.
[14] ITU-T G.729: 'Coding of speech at 8 kb/s using CS-ACELP', 1996.