Real Time Implementation of Speech Codec G.729 using CS-ACELP on TM 1000 VLIW DSP processor

Vivek Kapur, M. M. Raghuvanshi, A. B. Maidamwar

Abstract— Conjugate structure algebraic CELP (G.729) is a voice codec that compresses speech signal based on model parameter of human voice. This paper deals with implementation of a speech-coding algorithm CS-ACELP using ITU-T's G.729 recommendation and optimize it for real-time implementation on a Very Long Instruction Word (VLIW) Digital Signal Processor (DSP) Central Processing Unit (CPU). Very long instruction word or VLIW refers to a CPU architecture designed to take advantage of instruction level parallelism (ILP). A processor that executes every instruction one after the other (i.e. a non-pipelined scalar architecture) may use processor resources inefficiently, potentially leading to poor performance

Keywords: G.729, CS-ACELP, DSP processor.

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 this 8 kbits/s algorithm is equivalent to that of a 32 kbits/s ADPCM under most operating conditions. Typical input rates include mu-law or A-law **64** kbits/s PCM or 128 kbit/s linear PCM providing a compression ratio of 16:l. These coders are all based on a model of the human vocal system. In that model, the throat and mouth are modelled as a linear filter, and voice is generated by a periodic vibration of air exciting this filter. In the frequency domain, this implies that speech looks somewhat like a smooth response (called the envelope), modulated by a set of discrete frequency components. CELP coders all vary in the manner in which the excitation is specified, and the way in which the coefficients of the filter are represented. All of them generally break speech up into units called frames, which can be anywhere from 1ms to 100ms in duration. For each frame of speech, a set of parameters for the model are generated and sent to the decoder. This implies that the frame time represents a lower bound on the system delay; the encoder must wait for at least a frames worth of speech before it can even begin the encode process. G.729 will be used for Voice over IP (VoIP), Videophones, Digital Satellite Systems, Integrated Services Digital Network (ISDN), Land-Digital Mobile Radio, Future Public Land Mobile Telecommunication Systems (FPLMTS), Digital Circuit Multiplication Equipment (DCME), Digital Simultaneous Voice and Data (DSVD), and other applications. This speech codec's relative low complexity makes it an attractive choice for Internet telephony.The analog voice signal, sampled 8000 times per second, is taken as input signal of the coder G.729. The coder operates on frames of 10 ms. For each frame, the speech signal is analyzed to extract the parameters of the CELP model (Linear Prediction (LP) filter coefficients, adaptive and fixed codebook index), which are encoded and transmitted. The input signal is high-pass filtered in the preprocessing block. A 10th order linear prediction analysis yields a set of LP filter coefficients, which are converted to Line Spectrum Pairs (LSP) and quantized using Vector Quantization (V Q). The excitation signal is chosen and an open-loop pitch delay is estimated with a perceptually weighted and low-pass filtered speech signal.

II. ENCODER

Figure 1 show the block diagram of encoder of G.729. First block of encoder is preprocessing in the stated in clause 2, the input to the speech encoder is assumed to be a 16-bit PCM signal. Two preprocessing functions are applied before the encoding process:

1) Signal scaling; and

2) high-pass filtering.

The scaling consists of dividing the input by a factor 2 to reduce the possibility of overflows in the fixed-point implementation. 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. Both the scaling and high-pass filtering are combined by dividing the coefficients at the numerator of this filter by 2. The resulting filter is given by:

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

The input signal filtered through $H_{h1}(z)$ is referred to as $s(n)$, and will be used in all subsequent coder operations.

The short-term analysis and synthesis filters are based on 10th order linear prediction (LP) filters.

The LP synthesis filter is defined as:

Manuscript Received October 09, 2011.

Vivek Kapur, NYSS college of engineering, Nagpur, Maharashtra 440002, India, (Email:vivek.kapur11@gmail.com)

Dr. M. M. Raghuvanshi, NYSS College of engineering, Nagpur, Maharashtra 440002, India.

A. B. Maidamwar, NYSS College of engineering, Nagpur, Maharashtra 440002, India.

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

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

The encoding principle of G.729 is shown in Figure 1. The input signal is high-pass filtered and scaled in the pre-processing block. The pre-processed signal serves as the input signal for all subsequent analysis. LP analysis is done once per 10 ms frame to compute the LP filter coefficients. These coefficients are converted to Line Spectrum Pairs (LSP) and quantized using predictive two-stage Vector Quantization (VQ) with 18 bits [3][4]. The excitation signal is chosen 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. This is done by filtering the error signal with a perceptual weighting filter, whose coefficients are derived from the unquantized LP filter. The amount of perceptual weighting is made adaptive to improve the performance for input signals with a flat frequency response. The excitation parameters (fixed and adaptive codebook parameters) are determined per sub-frame of 5 ms **(**40 samples) each. The quantized and un-quantized LP filter coefficients are used for the second sub-frame, while in the first sub-frame interpolated LP filter coefficients are used (both quantized and un-quantized). An open-loop pitch delay T_{OP} is estimated once per 10 ms frame using the perceptually weighted speech signal $S_w(n)$ [1][2].

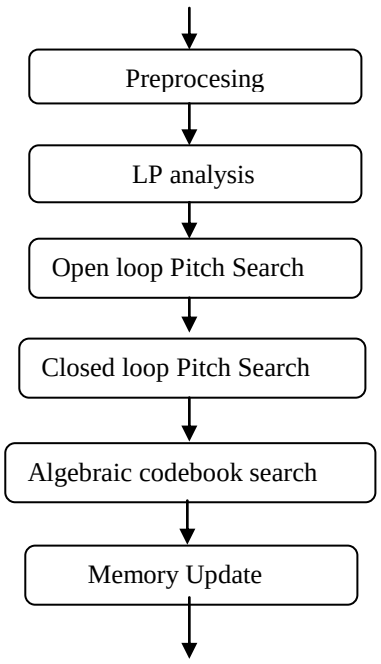


Figure 1. Block diagram of encoder

Either a closed or an open pitch loop is essential for good performance of the CELP algorithm at intermediate bit rates. The closed pitch loop can be interpreted as an adaptive codebook of overlapping candidate vectors. Either the endpoint correction method or the energy recursion method can be applied to the closed pitch loop, since 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), sing the target $x(n)$ and impulse response $h(n)$, by searching around the value of the open-loop pitch delay. A fractional pitch delay with 1/3 resolution is used. The pitch delay is encoded with 8 bits in the first subframe and differentially encoded with 5 bits in the second subframe.

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

Table 1:- Structure of fixed codebook search

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

An update of the states of the synthesis and weighting filters is needed to compute the target signal in the next subframe. After the two gains are quantized, the excitation signal, $u(n)$, in the present subframe is obtained using:

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

where $\hat{g}_p$ and $\hat{g}_c$ are the quantized adaptive and fixed-codebook gains, respectively, $v(n)$ is the adaptive-codebook vector (interpolated past excitation), and $c(n)$ is the fixed-codebook vector including harmonic enhancement. The states of the filters 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 filter operations. A simpler approach, which 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 output of the filter 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 states of the filter $A(z/\gamma_1)/A(z/\gamma_2)$ can be done by filtering the error signal $e(n)$ through this filter to find the perceptually weighted error $ew(n)$. However, the signal $ew(n)$ can be equivalently 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 available, the states of the weighting filter are updated by computing $ew(n)$ as in equation (76) for n = 30,...,39. This saves two filter operations.

III. DECODER

The decoder principle is shown in Figure 2 (b). First, the parameter's indices are extracted from the received bit-stream. These indices are decoded to obtain the coder parameters corresponding to a 10 ms speech frame. These parameters are the LSP coefficients, the two fractional pitch delays, the two fixed-codebook vectors, and the two sets of adaptive and fixed-codebook gains. The LSP coefficients are interpolated and converted to LP filter coefficients for each sub-frame.

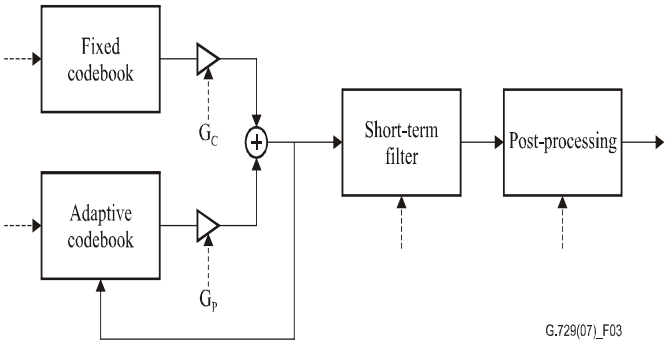


Figure 2 – Principle of the CS-ACELP decoder

The CS-ACELP decoder (figure 2) which is describes about the decoding and post-processing. Decoding process generates the LP filter coefficients from the transmitted information with the same procedures as used in the encoder. Post-processing consists of adaptive post filtering and high pass filtering. The main applications for this coder are 1) personal communication systems (PCS), 2) digital satellite systems, and 3) other applications such as packetized speech and circuit multiplexing equipment.

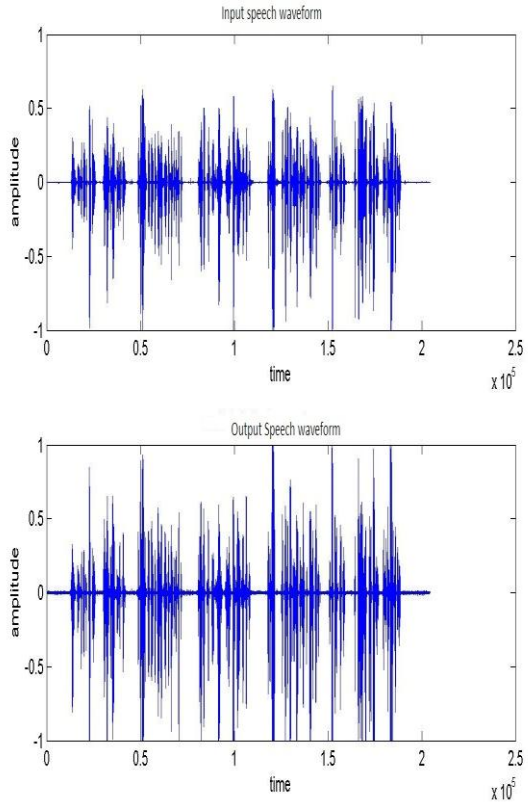
IV. REAL TIME IMPLEMENTATION WITH DSP

The implementation of G.729 involves translating the ITU-T C specification into highly efficient DSP code. This was a significant task requiring large amount of resources. The task was further complicated by the fact that each routine must be carefully examined in order to take advantage of the specific **DSP** chip capabilities. Also, to maintain bit-exact compliance, some basic operations, which were normally processed in one cycle by the TM-1000 chip, had to be coded using many lines of code. This was necessary to handle exceptions imposed by the 'ITU-T code. In addition, lots of testing was required to ensure bit-exact compliance.

One way to minimize the development time was to use the right balance of C and assembly language. The routines, which perform control operations rather than signal processing, were coded in C without many penalties. This allowed easy maintainability and debugging. In order to ensure maintainability and reusability, calling conventions were defined to standardize the interface to assembly routines. Two methods available for passing parameters are a) using the stack or b) TM-1000 registers. The stack method was used when interfacing with C. The register technique turned out to be faster for assembly routine to assembly routine calls. Bit exact compliance was a challenge when implementing G.729. An occasional error on the Least Significant Bit (LSB) of a value could have easily caused a failure. The implemented code was aimed to truncate data to the same precision as in the ITU-T code. When overflow happened in calculations, saturation was to be performed the same way. Fortunately, the TM-1000 chip processes overflow the same way as in the ITU-T's primitives.

V. CONCLUSION AND SIMULATION

This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering of the analogue input signal, then sampling it at 8000 Hz, followed by conversion to 16-bit linear PCM for the input to the encoder[1]. The output of the decoder should be converted back to an analogue signal by similar means. In this paper we have presented the real-time implementation of a full-duplex G.729 speech coder on a TM-1000 VLIW DSP processor. The implementation meets the ITU-T specification over all test files supplied by the ITU-T. The coder providing a simple, yet flexible high level interface, is fully re-entrant and is written to allow easy integration with other applications. Speech encoding and decoding tasks can be independently controlled.
Fig show the time domain representation of original speech and reconstruction speech, while implementing CS-ACELP on DSP processor it surely prove that the reconstructed speech is exactly similar of input speech



REFERENCES

1. Real-Time Implementation And Optimization Of ITU-T's G.729 Speech Codec Running At 8kbitsBec Using CS-ACELP On TM-1000 VLIW DSP CPU.
2. Salami et al: 'Design and Description of CS-ACELP: A toll quality 8kb/s speech coder', IEEE trans Speech Audio Process, 1996.
3. ITU-T G.729: 'Coding of speech at 8 kb/s using CS-ACELP', 1996.
4. Kataoka et al: 'An 8 kb/s speech coder based on conjugate structured CELP', IEEE int. conf. acoustic, speech, signal processing, 1993.
5. kataoka et al: 'LSP and gain quantization for proposed ITU-T 8 kb/s speech coding standard', IEEE workshop on speech coding, 1995.
6. Shaw Hwa Hwang: ' Computational improvement for G.729 standard', 2003.
7. A. B. Roach, "Session Initiation Protocol (SIP) -specific event notification," RFC 3265, June 2002.
8. 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.
9. R. Sparks, "The Session Initiation Protocol (SIP) refer method," RFC 3515, April 2003.
10. 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.
11. ITU-T Recommendation P.862 Amendment 1, "Source code for reference implementation and conformance tests," March 2003.