REAL-TIME IMPLEMENTATION OF HIGH-QUALITY 32 KBPS WIDEBAND LD-CELP CODER

Oded Gottesman
Yair Shoham

Speech Coding Research Department
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

The Wideband-Audio Low-Delay CELP (LD-CELP) coder produces speech with quality as high as the CCITT 64 kb/s standard (G.722) at half the bitrate. The computational load of the encoder is almost 900% processor time of the 12.5 MIPS DSP32c. This makes a real-time implementation impractical. We investigated the Gain-Shape Vector-Quantization (GSVQ) in order to reduce the computational load of the encoder. This paper describes a real-time implementation of the LD-CELP encoder based on the AT&T SURFboard using two DSP32c operating in parallel. A computational load of 180% processor time has been achieved. The respective decoder requires 42% processor time. The implementation of a full-duplexed coding system requires three 12.5 MIPS Digital-Signal-Processors (DSPs) and has one-way coding delay of less than 1ms. The coder also performs well for non-speech wideband audio signals such as music.

Keywords: Wideband, LD-CELP.

1. INTRODUCTION

The growing pool of ISDN applications intensifies the interest in new and more advanced coding algorithms for wideband speech [6, 7]. The major requirements expected from such coders are: a) the coded speech quality should be comparable to that of the G.722; b) the bitrate should be at least halved; and c) the one-way coding delay should be minimal. The 32 kb/s wideband-speech LD-CELP coder was shown to potentially satisfy these requirements [10, 11]. However, the high computational load of the encoder, which is approximately 900% of the processor-time of a 12.5 MIPS DSP [5], makes the implementability of this algorithm questionable. With the present DSP technology, the use of several DSPs operating in parallel seems to be unavoidable even if the algorithm is greatly simplified. Therefore, we were challenged to implement the encoder using only two DSPs.

In this paper we present a real-time DSP implementation of this coder and describe its performance. Section 2 gives a brief overview of the initial 32 kb/s wideband-speech LD-CELP algorithm and analyzes its computational load. Section 3 shows how we dealt with the problem of the computational load reduction. Section 4 describes the development of the parallel-processing operated DSP software, the processor time and memory usage. Section 5 discusses the subjective performance test results.

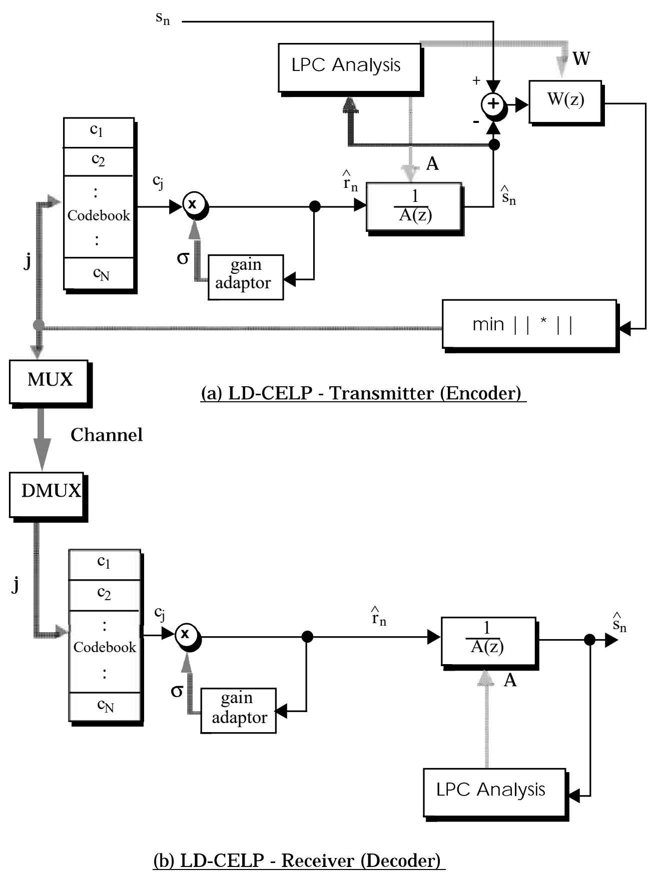

Figure 1. Fully backward adaptive LD-CELP coder

2. OVERVIEW OF THE 32 KB/S WIDEBAND-SPEECH LD-CELP ALGORITHM

The LD-CELP is basically a backward-adaptive version of the conventional CELP coder [1, 8, 12]. The basic structure of the LD-CELP [3-5, 10, 11] is illustrated in Fig. 1. The LD-CELP encoder implements a closed-loop (analysis-by-synthesis) search procedure for finding the best excitation c_j drawn from an excitation codebook. Each possible excitation vector is passed through the adaptive gain σ [2, 5] and the LPC filter 1/A(z) and results with a synthetic output vector. The encoder selects the excitation whose synthetic output vector $\hat{s}_n$ is the best match to the input speech s_n, usually in a Weighted-Mean-Squared Error

(WMSE) sense. The WMSE matching is accomplished via the use of a noise-weighting filter W(z). The parameter j that describes the selected excitation vector is then transmitted to the decoder where the synthesis process is duplicated.

The parameters of the filters 1/A(z) and W(z) are determined via the LPC analysis applied to the recent past output speech in a backward-adaptive mode. The filter W(z) is important for achieving a high perceptual quality in CELP systems. The conventional form of noise-weighting filter $W_c(z)$ is given by [1,3-5, 8, 10, 11]:

$$W_c(z) = \frac{1-\sum_{k=1}^{P} a_k \gamma_z^{\,k} z^{-k}}{1-\sum_{k=1}^{P} a_k \gamma_p^{\,k} z^{-k}} \quad ; \quad 0<\gamma_p<\gamma_z<1 \tag{1}$$

where A(z) is the LPC polynomial. Such a filter has an inherent limitation in modeling concurrently the formant structure and the spectral tilt. Since at high frequencies the data is highly unstructured and the initial unweighted SNR tends to be highly negative [11], noise-weighting filter is more critical in wideband speech coding. Therefore, an enhanced form of noise-weighting filter [5, 10, 11] is used for wideband speech LD-CELP coder, given by:

$$W(z) = W_c(z)\, T(z) \tag{2}$$

where T(z) is a tilt controlling second order section given by:

$$T(z) = \frac{1}{1-\sum_{k=1}^{2} \tau_k \gamma_t^{\,k} z^{-k}} \tag{3}$$

where the coefficients $\{\tau_k\}_{k=1}^{2}$ are computed by applying the standard LPC procedure to the first three correlation coefficients of the current frame LPC coefficients $\{a_k\}_{k=0}^{p}$ [5, 10, 11]. The parameter γ_t is used to adjust the spectral tilt of T(z). Table 1 shows the configuration of the wideband LD-CELP coder investigated and implemented [5].

3. COMPUTATIONAL LOAD REDUCTION

The computational complexity of our initial LD-CELP coder is depicted in Table 2 [5]. It is measured as a percentage of 12.5 MIPS processor time. The overall complexity of the encoder is approximately 900% real-time. The most intensive task (429.36%) is the convolution of the synthesis filters with the entire set of excitation vectors and the computation of the energy of the resulted vectors [5]. The second intensive task is VQ search (341.56%). We selected to reduce the complexity of these two tasks by using Gain-Shape VQ [3-5]. Additional reduction of the algorithm complexity may be obtained by performing the LPC analysis once in every given number of vectors rather than every vector.

Element	**Encoder Real-Time (%)**
Sampling rate	16 kHz
Coded data	7 kHz audio
Bitrate	32 kb/s (2 bit/sample)
Vector length	5 samples (0.3125 ms)
LPC analysis	backward mode
LPC Synthesis filter order	32
Noise-Weighting filter order	16
Spectral-Tilt filter order	2
Noise-Weighting filter	γ_z=0.95, γ_p=0.8, γ_t=0.7
Quantization	Excitation signal, 10 bit
Pitch-Synthesis filter	Not used
Adaptive predictive gain	Backward-mode

Table 1. The Wideband LD-CELP configuration

Process	**Encoder Real-Time (%)**	**Decoder Real-Time (%)**
Convolution plus energy	429.36	0
VQ search	341.56	0
LPC of order 32	61.67	61.67
Recursive autocorrelation	19.62	19.62
Impulse response	10.62	0
Zero input response	10.62	0
Update filters states	10.62	5.31
Pre-filter the input	5.57	0
Autocorrelation of order 3	2.37	0
LPC of order 2	2.25	0
Weight filters	2.05	2.05
Predictive gain update	1.66	1.66
Compute the VQ's target	0.32	0
Total Real-Time %	**898.30**	**90.32**
MIPS	**112.29**	**11.29**

* All real-time % computations are with respect to DSP32C having 12.5 MIPS

Table 2. The computational complexity of the initial LD-CELP coder [5].

Fig. 2 illustrates the complexity of the two investigated LD-CELP encoders [5] as a function of the LPC update period. The initial system performs an exhaustive search in a 10-bit codebook for the best matched shape-vector $c_j(n)$, hence is denoted by Shape-VQ (SVQ). The quantized excitation vector $\hat{r}(n)$ for the SVQ system is given by:

$$\hat{r}(n) = \sigma\, c_j(n) \quad ; \quad 0 \le n \le N-1 \tag{4}$$

where N is the vector length and σ is the adaptive predictive gain [2, 5] illustrated in Fig. 1. The second system performs a Gain-Shape VQ (GSVQ) [3-5], where 7 bits are allocated to represent a shape vector $q_j(n)$ and 3 bits are used for gain factor g_k. The quantized excitation vector $\hat{r}(n)$ for the GSVQ system is given by:

$$\hat{r}(n) = \sigma\, g_k\, q_j(n) \quad ; \quad 0 \le n \le N-1 \tag{5}$$

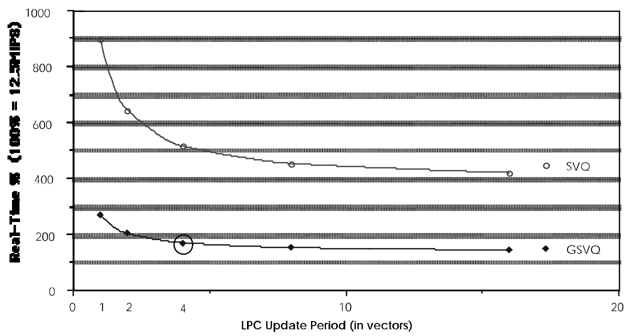

Figure 2. The computational complexity vs LPC update rate for the two LD-CELP encoders [5].

Fig. 3 illustrates the output SNR obtained [5] for the respective systems. The GSVQ encoder having an update rate of every 4 vectors requires 180% real-time was selected for real-time implementation on a two-DSP hardware. The computational load of the respective decoder is 42% processor time. Therefore it is implemented on a third DSP [5].

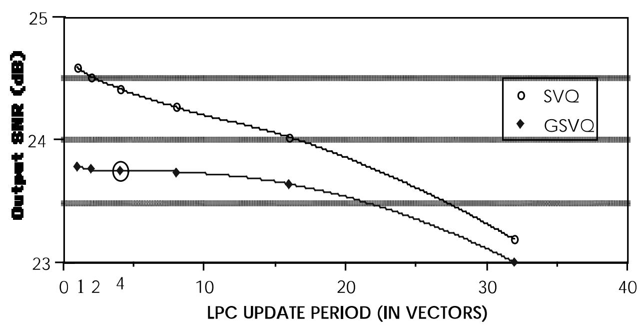

Figure 3. Output SNR vs LPC update rate for the two LD-CELP coders [5].

4. SURFBOARD IMPLEMENTATION

The original LD-CELP algorithm was written in C language. First we compiled and simulated the algorithm in general-purpose computer. Later we used the AT&T DSP32 C Language Compiler to compile the entire C code to DSP32 assembly code [13]. We ran this code on the AT&T DSP32 SURFboard. The encoding algorithm was then divided into two parts, to distribute its processing over two DSPs [5]. The first part includes the processes that are directly related to the VQ search and are performed every vector. We denoted this class of processes by *VQ processes*. The second part includes the processes that are directly related to the LPC analysis and are performed once in every given number of vectors. We denoted this class of processes by *LPC processes*. We ran these two parts of the algorithm on two DSPs where each one of them ran a different part of the algorithm in a master-slave manner. During this phase, the interface between the two DSPs was developed. We were greatly helped at this phase by a locally developed program called "*dspx*" which handled the downloading and the I/O between the Unix environment and the SURFboard. At this point we completed the allocation and scheduling of the tasks and interfaces performed by each one of the DSPs, but we still processed data files rather then real-time sampled data. The next step was to take a conservative approach in converting C subroutines step-by-step into DSP32 assembly code [13]. After all the C subroutines were converted to hand optimized DSP32 subroutines, we wrote a DSP32 assembly code to handle the DMA for the real-time processing. Fig. 5 illustrates the processing stream, performed by each DSP in the real-time implementation of the selected GS LD-CELP encoder. The master DSP handles the DMA with the analog interface. As soon as a new vector of samples is filled, the first (master) DSP starts processing the VQ processes. In the background the LPC processes are handled by the second (slave) DSP. The slave DSP is synchronized to the master DSP such that they share the VQ search. The arrows denote parameter transfer between the two DSPs. The illustrated process is repeated in a 4 vector period (the LPC update period). The 4 vectors in this period are denoted by vector #1 to vector #4. Table 3 summarizes the complexity of our real-time implemented GSVQ LD-CELP encoder [5]. Table 4 summarizes the memory usage of the implementation.

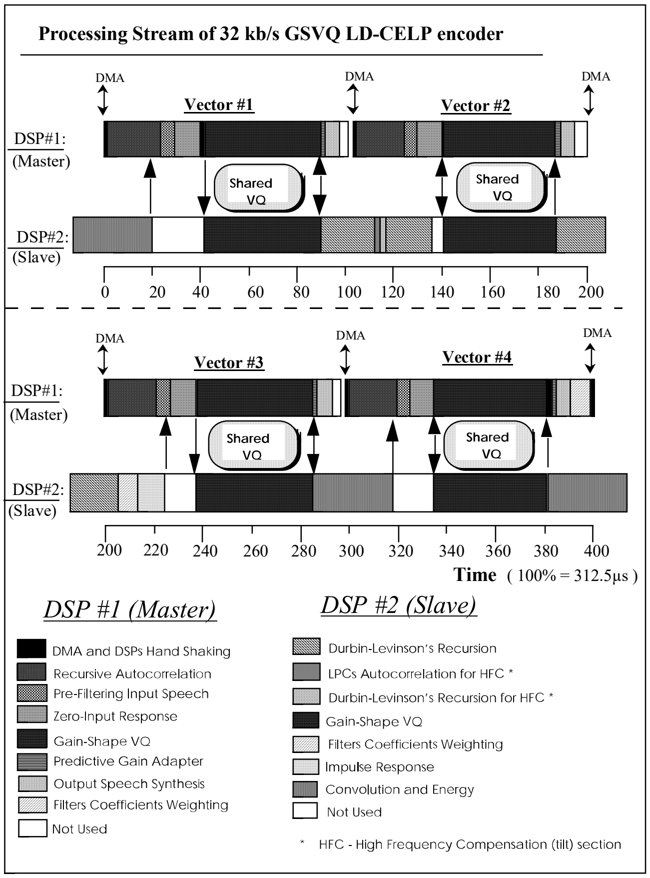

Figure 4. Processing Stream of 32 kb/s Wideband GSVQ LD-CELP encoder [5].

5. RESULTS

The performance of the 32 kb/s wideband LD-CELP was evaluated by comparing it to the 64 kb/s G.722 CCITT standard wideband coder [9]. The test material included four male and four female utterances. Each utterance was coded by the G.722 and by the real-time LD-CELP to form a pair of utterances. Twenty-four listeners took part in the test. Twelve of the listeners work in speech processing and are well acquainted with this kind of test, and therefore were denoted "*trained*" listeners. The other twelve listeners, who are not experienced with this kind of test, were denoted "*naive*" listeners. The listener was asked to vote for the better sounding utterance in his judgment, or, to split his vote equally, if no preference could be made. The final scores were defined as the percentage of the number of votes for each system.

Process	Encoder Real-Time (%)	Decoder Real-Time (%)
Convolution plus energy	13.42	0
VQ search	88.70	0
LPC of order 32	15.42	15.42
Recursive autocorrelation	18.83	18.83
Impulse response	2.66	0
Zero input response	10.62	0
Update filters states	10.62	5.31
Pre-filter the input	5.57	0
Autocorrelation of order 3	0.59	0
LPC of order 2	0.56	0
Weight filters	1.02	0.51
Predictive gain update	1.66	1.66
Compute the VQ's target	0.32	0
DSP interface	10.03	0
Total Real-Time %	**180.03**	**41.73**
MIPS	**22.50**	**5.22**

* All real-time % computations are with respect to DSP32C having 12.5 MIPS

Table 3. The computational complexity of the initial LD-CELP coder [5].

\ System Block \	Encoder DSP#1	Encoder DSP#2	Decoder
Program	4476	3460	2402
Data	7556	7172	6940
Total	**12032**	**10632**	**9342**

Table 4. Memory usage of the Wideband LD-CELP (in bytes) [5]

Type of input	64 kb/s ADPCM (G.722) (%)	32 kb/s GSVQ LD-CELP (%)
Total Score	54.43	45.57
Trained Listeners Score	58.85	41.15
Naive Listeners Score	50.00	50.00
Male's utterances only	53.39	46.61
Female's utterances only	55.47	44.53

Table 5. A/B-test results for 32 kb/s GS LD-CELP vs 64 kb/s ADPCM (G.722) [5].

The experimental results of the subjective test are summarized in Table 5 [5]. The total results indicate that, on the average, our real-time 32 kb/s coder and the 64 kb/s ADPCM standard, which operates at twice the bit rate, provide comparable speech quality. Among naive-listeners, the two systems performed alike, on the average. We may, therefore, be able to halve the bitrate while preserving the high quality of the reproduced speech. Another observation is that LD-CELP does better on males than on females.

6 CONCLUSIONS

The main conclusion of this work is that high-quality coding of wideband audio at 32 kb/s is feasible while keeping the computational complexity at a reasonable level. The results of the subjective A/B comparison tests indicate that the reproduced-speech *quality of the 32 kb/s GS LD-CELP is comparable to the 64 kb/s ADPCM standard*. The two major advantages of our real-time implemented GS LD-CELP coder over the ADPCM standard are: a) it operates at half the bitrate of the ADPCM standard; and b) it has an extremely low one-way-delay of less than 0.94 ms compared to about 1.5 ms for the ADPCM standard. This work presents a real-time implemented coder which can be an excellent candidate for wideband audio coding in high-quality communication networks.

REFERENCES

[1] B. S. Atal, M. R. Schroeder, "Code Excited Linear Predictive (CELP): High Quality Speech at Very Low Bit Rates", *Proc. IEEE Int. Conf. ASSP*, 1985, pp. 937-940.

[2] J. H. Chen and Allen Gersho, "Gain-Adaptive Vector Quantization with Application to Speech Coding", *IEEE Trans. Comm.*, Vol. 35 No. 9, September 1987, pp. 918-930.

[3] J. H. Chen, " A Robust Low-Delay CELP Speech Coder at 16 kb/s", *Proc. GLOBECOM-89*, Vol. 2, November 1989, pp. 1237-1240.

[4] J. H. Chen et. al., "Low-delay CELP coder for the CCITT 16 kb/s speech coding standard", *IEEE SAC*, vol. 10 no. 5, June 1992, pp. 830-849.

[5] O. Gottesman, "Algorithm Development and Real-Time Implementation of High-Quality 32kbps Wideband Speech LD-CELP Coder", *MS Thesis, ECE Dept., Drexel University*, January 1993.

[6] N. S. Jayant et al., "Coding of Wideband Speech", *Proc. 2nd Europ. Conf. Speech. Commun. Technol.*, Sept 1991.

[7] N. S. Jayant , "Signal Compression: Technology Targets and Research Directions", *IEEE SAC*, vol. 10 no. 5, June 1992, pp. 796-818.

[8] Peter Kroon and Ed. F. Deprettere, "A Class of Analysis-by-Synthesis Predictive Coder for High Quality Speech Coding at Rates Between 4.8 and 16 kbits/s", *IEEE J. SAC*, vol. 6 No. 2, February 1988, pp. 353-363.

[9] P. Mermelstein, "G.722, A New CCITT Coding Standard for Digital Transmission of Wideband Audio Signals", *IEEE Communications Magazine*, January 1988, pp. 8-15

[10] E. Ordentlich, "Low Delay Code Excited Linear Predictive (LD-CELP) Coding of Wide Band Speech at 32Kbit/sec", *MS Thesis, EE Dept., MIT*, March 1990.

[11] E. Ordentlich, Y. Shoham, "Low-delay code-excited linear-predictive coding of wideband speech at 32 kbps", *Proc. ICASSP-91*, pp. 9-12.

[12] Y. Shoham, "Constrained-Stochastic-Excitation Coding of Speech at 4.8 Kb/s", In B. S. Atal et al., editor, *Advances in Speech Coding, Kluwer Academic Publishers*, 1990, pp. 339-348.

[13] "WE® DSP32C Digital Signal Processor - Information Manual", *AT&T,* January 1990.