N88-25750

A 4.8 KBPS CODE EXCITED LINEAR PREDICTIVE CODER

THOMAS E. TREMAIN, JOSEPH P. CAMPBELL, JR., VANOY C. WELCH

U.S. Department of Defense, R5
Fort Meade, Maryland, U.S.A. 20755-6000

Abstract

In 1984, the Department of Defense established a program under the direction of the National Security Agency (NSA) to develop a secure voice system (STU-III) capable of providing end-to-end secure voice communications to all segments of the Federal Government, Federal Government contractors, and citizens of the United States. Based on the work performed by the Digital Voice Processor Consortium, NSA determined that the terminal for the new system should be built around the DoD Standard LPC-10 voice processor algorithm.

While the performance of the present STU-III as a processor of speech is considered to be good, its response to nonspeech sounds, such as whistles, coughs and impulse-like noises, may not be completely acceptable. Speech in noisy environments also causes problems with the LPC-10 voice algorithm. In addition, there is always a demand for something better. We hope to complement LPC-10's 2.4 kbps voice performance with a very high quality speech coder operating at a higher data rate. This new coder is one of a number of candidate algorithms being considered for an upgraded version of the STU-III in late 1989. In this paper, we address the problems of designing a code excited linear predictive (CELP) coder to provide very high quality speech at a 4.8 kbps data rate that can be implemented on today's hardware.

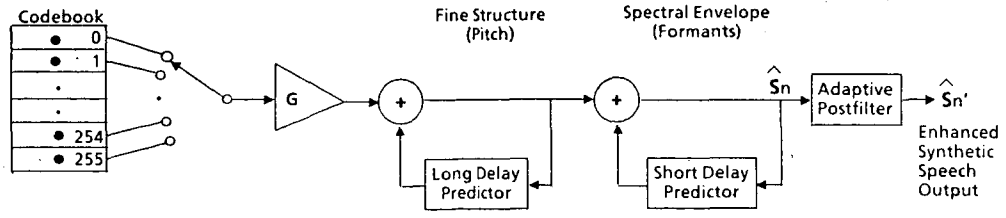
1 INTRODUCTION

Speech coding algorithms that achieve high quality speech at high data rates are well known (i.e., 64 kbps PCM, 32 kbps ADPCM and 16 kbps APC). However, high quality speech coding at lower data rates has been achieved only very recently. Codebook excited linear prediction (CELP) was introduced by B.S. Atal and M.A. Schroeder at the 1984 International Communications Conference [1]. The introduction of CELP sparked one of the speech coding community's greatest research efforts to achieve high quality speech at 4.8 kbps within reasonable computational complexity.

In this paper, we present a detailed description of our CELP coder, various tradeoffs to obtain a 4.8 kbps data rate, and computational complexity reduction methods to allow practical implementation of our coder using a pair of new generation digital signal processor (DSP) chips.

2 CELP DESCRIPTION

Our CELP decoder is shown in Figure 1. The parameters required for this model are the codebook index and gain, the pitch index and gain, and the short-term predictor parameters. To further



Codebook Index	Codebook Gain	Pitch Index & Gain	Spectrum (LSPs)	Transmit
4	4	4	1	Rate/Frame
ℓ	ℓ	$\ell_p = \ell$	$\ell\ell$	Frame Size

Figure 1: CELP decoder.

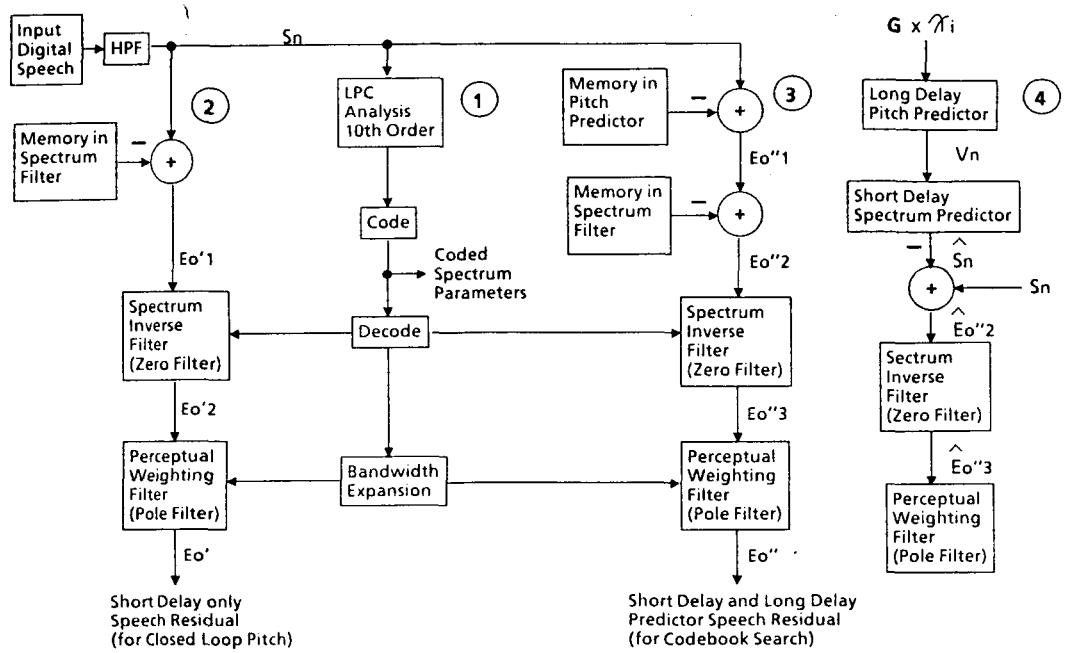


Figure 2: CELP encoder.

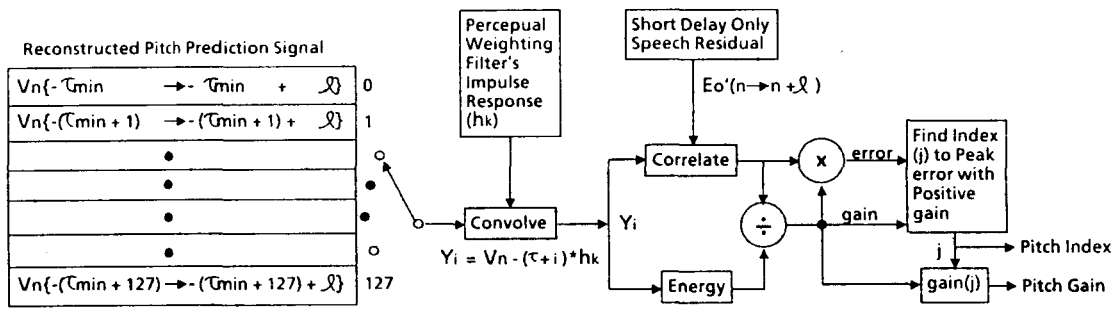


Figure 3: Closed loop pitch calculation.

reduce the quantizing noise at the expense of negligible computation, we added a postfilter based on the pole-zero filter described in reference [2] with adaptive spectral tilt compensation.

As labeled in Figure 2, the CELP encoder consists of four parts: 1)The short-term predictor (spectrum) parameters are calculated in an open loop format; 2) The pitch index and gain are calculated on a spectrum-only residual signal in a closed loop format on a subframe basis relative to the spectrum parameters; 3) The codebook index and gain are calculated on a spectrum and pitch residual signal in a closed loop format on a subframe basis relative to the spectrum parameters; 4) The decoder is run in the transmitter to update all the filter states, which are then used to calculate the next frame of speech in the closed loop format.

Figure 3 shows the closed loop pitch calculation using the filtering approach. The reconstructed pitch prediction signal, V_n, is convolved with the perceptual weighting filter's impulse response. The convolution is calculated for each of the 128 pitch lags, which vary from a minimum (τ_{min}) of 16 to a maximum of 143. For this convolution, there is only one impulse response in the convolution interval. Each pitch lag's convolution is then correlated with the short-delay (spectrum only) predictor's speech residual. The optimum pitch lag for a 1-tap predictor maximizes the error function and has positive gain. Since the pitch predictor is calculated over 128 lags, a brute force calculation requires 33 million instructions (multiplies and adds) per second (33 MIPS). However, as shown in Section 4, the number of instructions can be significantly reduced.

The codebook search using the filtering approach is given in Figure 4. The codebook is convolved

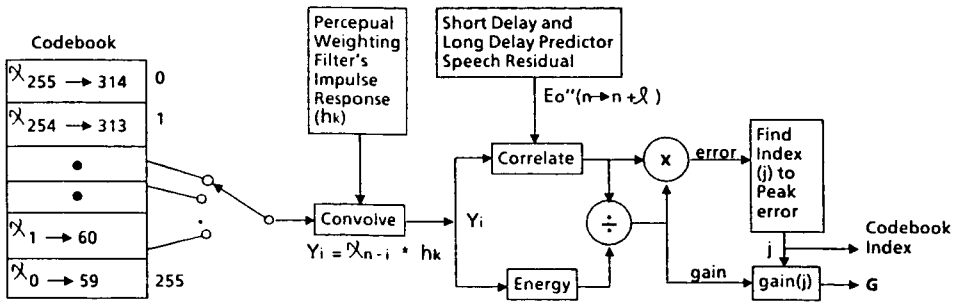


Figure 4: Codebook search.

Table 1: Options studied for a 4.8 kbps CELP coder.

Option	*1*	*2*	*3*	*4*	*5*	*6*
Spectrum Frame	20 msec	20 msec	22.5 msec	25 msec	25 msec	30 msec
Samples @8 kHz	160	160	180	200	200	240
Bits @4.8 kbps	96	96	108	120	120	144
Spectrum (LSPs)	36	36	36	36	36	36
Pitch Index	$2*7=14$	$2*7=14$	$2*7=14$	$4*7=28$	$2*7=14$	$4*7=28$
Pitch Gain	$2*3=6$	$2*3=6$	$2*4=8$	$4*4=16$	$2*4=8$	$4*5=20$
Codebook Index	$2*14=28$	$4*5=20$	$4*6=24$	$4*5=20$	$4*8=32$	$4*8=32$
Codebook Gain	$2*6=12$	$4*5=20$	$4*6=24$	$4*5=20$	$4*6=24$	$4*6=24$
Bits Used	96	96	106	120	114	140

with the perceptual weighting filter's impulse response. The convolution is calculated for each of the 256 codewords. For this convolution, there can be several impulse responses in the convolution interval as determined by the pitch index. Each codeword convolution is then correlated with the speech residual from the short-delay (spectrum) predictor and long-delay (pitch) predictor. The optimum codeword maximizes the error function for the codebook. For a 256 codeword codebook, a brute force calculation requires 66 MIPS. However, as shown in Section 4, the number of instructions can be significantly reduced.

3 CODING AND COMPLEXITY ESTIMATES

Option 1 in Table 1 is a coding scheme presented by Bell Labs in reference [3]. This approach was not studied because it is too computationally complex. As shown in Table 2, it requires more than 5 billion multiplies and adds per second (5 GIPS) using brute force calculations. Options 2 and 3 are more practical from the implementation point of view; however, the output speech is very rough because of the small codebook size and the pitch is not updated at the same rate as the codebook. Option 4 provides better speech quality than 2 or 3 because the pitch is updated at the same rate as the codebook. However, the speech is still rough because of the small codebook. Option 5 has an acceptable size codebook, but the speech is rough because the pitch and codebook are updated at different rates.

We selected Option 6 because it provides the best speech quality of the options listed in Table 1 and it is practical to implement. Tables 2 and 3 show that the codebook and pitch calculations with brute force calculations require about 100 MIPS. Table 4 shows most of CELP's remaining calculations, which add up to only 1.27 MIPS.

Table 2: Computational Complexity Comparisons for Codebook Calculations.

Option	*1*	*2*	*3*	*4*	*5*	*6*
Convolution	3240	820	1035	1275	1275	1830
Size	*16,384	*32	*64	*32	*256	*256
Instructions	53,084,160	26,240	66,240	40,800	326,400	468,480
Correlation	32,768	64	128	64	512	512
Frame	*80	*40	*45	*50	*50	*60
Instructions	2,621,440	2560	5760	3200	25,600	30,720
Total Inst.	55,705,600	28,800	72,000	44,000	352,000	499,200
Rate	*100	*200	*177,777	*160	*160	*133.33
Total IPS	5,570,560,000	5,760,000	12,800,000	7,040,000	56,320,000	66,560,000

Table 3: Computational Complexity Comparisons for Closed Loop Pitch Calculations.

Option	*1*	*2*	*3*	*4*	*5*	*6*
Convolution	3240	3240	4095	1275	5050	1830
Pitch Lags	*128	*128	*128	*128	*128	*128
Instructions	414,720	414,720	524,160	163,200	646,400	234,240
Correlation	256	256	256	256	256	256
Pitch Frame	*80	*80	*90	*50	*100	*60
Instructions	20,480	20,480	23,040	12,800	25,600	15,360
Total Inst.	435,200	435,200	547,200	176,000	672,000	249,600
Pitch Rate	*100	*100	*88.88	*160	*80	*133.33
Total IPS	43,520,000	43,520,000	48,640,000	28,160,000	53,760,000	33,280,000

Table 4: Remaining calculations in CELP coder

Operation	*IPS/call*	*Times Called*	*Total IPS*
Zero Filter	88,000	3	264,000
Pole Filter	80,000	4	320,000
Pole Weighting Filter	81,333	5	406,667
Pitch Filter	8,000	3	24,000
Matrix Load & Invert	92,667	1	92,667
Adaptive Postfilter	162,667	1	162,667
Total IPS			1,270,000

Table 5: End-Correction Computational Complexity Reduction.

Operation	*Closed Loop Pitch*			*Codebook*		
	Brute Force	End-Correct	End-Correct 30 Impulse	Brute Force	End-Correct	End-Correct w/75% zeros
Convolve	1830	1830	1830	1830	1830	1830
	*128	+127*60	+127*30	*256	+255*60	+255*15
Instructions	234,240	9,450	5,640	468,480	17,130	5,655
Correlate	256	256	256	512	512	512
	*60	*60	*60	*60	*60	*60
Instructions	15,360	15,360	15,360	30,720	30,720	30,720
Total Inst.	249,600	24,810	21,000	499,200	47,850	36,375
Rate	*133.33	*133.33	*133.33	*133.33	*133.33	*133.33
Total IPS	33,280,000	3,308,000	2,800,000	66,560,000	6,380,000	4,850,000

4 COMPUTATION REDUCTION STUDIES

End-correction, shown in equations 1 - 3, can be applied to the closed loop pitch convolution calculation. The results are identical to the brute force convolution. As shown in Table 5, the computation is reduced by an order of magnitude from 33 to 3 MIPS. Also, a slight further reduction can be obtained by reducing the weighted impulse response from 60 to 30 samples.

$$y_{0,0} = 0 \tag{1}$$

$$y_{i,0} = \sum_{j=0}^{i-1} x_{offset+j} \bullet h_{i-j} \qquad where\ 1 \leq i \leq \ell \tag{2}$$

$$y_{i,k} = y_{i-1,k-1} + x_{offset-k} \bullet h_i \qquad where\ 1 \leq i \leq \ell\ and\ 1 \leq k \leq lags \tag{3}$$

In Atal and Schroeder's original design [1], their codebook was generated from a zero-mean unit-variance white Gaussian sequence where each codeword consisted of an independent segment of this sequence. End-correction can also be applied to the codebook convolution calculation if a special form of codebook with overlapping codewords is used. In our case, each codeword contains one new sample and all but one sample of the previous codeword. When using overlapped versus independent codebooks, the difference in synthesized speech is virtually unnoticeable and the reduction in segmental signal-to-noise ratio is less than a fraction of a decibel for a 256 codeword codebook. As shown in Table 5, end-correction reduces the computation by an order of magnitude from 66 to 6 MIPS for a 256 codeword codebook. Also, a slight further reduction can be obtained by taking advantage of any zero samples in the codebook (which we center clip 75% of the samples to reduce high frequency noise).

5 IMPLEMENTATION

Implementation of our CELP coder on a pair of new generation DSP chips is very practical. The end-correction techniques we have described, when applied to the pitch and codebook calculations, yield a 10 MIPS CELP algorithm. When implementing speech coders of this type on DSP chips, typically, only one-third to one-half of the DSP chip's peak computational power can be used because of breaks in the multiply-accumulate pipeline, random logic, and control overhead. By analyzing our CELP coder's computationally intensive algorithm segments, we determined that it can be implemented on various new generation DSP chips or other processor chips (i.e., AT&T's Graph Search Machine). For example, our CELP algorithm can be implemented on a pair of AT&T's 12.5 MIPS DSP32C DSP chips or on a pair of TI's 16.6 MIPS TMS320C30 DSP chips. If slightly optimistic processor

utilization efficiency can be attained, our CELP algorithm may even fit on a pair of TI's 10 MIPS TMS320C25 DSP chips!

6 CONCLUSIONS

In this paper, we described the practical implementation of a CELP coder, we examined different bit allocations that can be used to implement a 4.8 kbps coder, and we determined the computational complexity of the coder for each bit allocation. The CELP coder we describe produces high quality speech and is practical to implement on a pair of new generation DSP chips.

The end-correction techniques applied to the codebook search and closed loop pitch calculations have each been shown to reduce the computational requirements by an order of magnitude without causing a loss in the quality of the output speech signal. We also found that the best speech quality was obtained with the parameters referred to as Option 6 in Section 3.

Our CELP coder has no problems with background noise (in fact, it is faithfully reproduced) and even works well with multiple speakers. *Informal* listening tests indicate that our CELP's speech intelligibility and quality are comparable with 16 kbps APC and 32 kbps CVSD!

We formally measure speech intelligibility and quality using Dynastat's diagnostic rhyme test (DRT) and diagnostic acceptability measure (DAM), respectively. We will report our CELP's DRT and DAM scores in the future. (As reference points, 32 kbps CVSD has a DRT score of 93.2 and a DAM score of 63 while 2.4 kbps LPC-10 has a DRT score of 90 and a DAM score of 53.)

Our future work will focus on error protection techniques and determining a 4.8 kbps standard to allow interoperability between current and future CELP coders. We expect future CELP coders to offer even better performance by exploiting the next generation's more powerful hardware.

7 ACKNOWLEDGMENTS

We are grateful to the following for their contributions: Tom Barnwell, Shawn and Elizabeth Campbell, Allen Gersho, George Kang, Dan Lin and Bell Communications Research. We thank our in-house support, especially Dave Kemp, and our in-house tools development team. We wish to give special recognition to the AT&T Bell Laboratories Acoustics Research Department, especially Bishnu Atal and Peter Kroon, for their outstanding contributions.

References

[1] Atal, B.S. and M.R. Schroeder, "Stochastic Coding of Speech at Very Low Bit Rates," Proc. of International Communications Conference, 1984, pp. 1610-1613.

[2] Chen, J.H. and A. Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," Proc. of Int. Conf. Acoust., Speech, and Signal Process., 1987, pp. 2185-2188.

[3] Kroon, P. and B.S. Atal, "Quantization Procedures for the Excitation in CELP Coders," Proc. of Int. Conf. Acoust., Speech, and Signal Process., 1987, pp. 1649-1652.