論 文

Robust, Low Delay Multi-tree Speech Coding at 9.6Kbits/sec

Hong Chae Woo* Byung Hyun Moon* Chae Wook Lee** *Regular Members*

견실, 저지연 멀티트리 9.6Kbits/s 음성부호기에 관한 연구

正會員 禹 洪 棣* 正會員 文 炳 顯* 正會員 李 採 昱**

ABSTRACT

In this research, a multi-tree coder at 9.6Kbits/sec using a novel scheme for adaptation of the short-term coefficients is developed. The overall delay of the tree coder is maintained at 2.5 msec (16 samples at the 6.4KHz sampling frequency). This coder produces good quality speech over ideal channels, and it is very robust to channel errors up to a bit error rate (BER) of 10^{-3}. This robustness is achieved by using a parallel adaptation scheme in combination with the use of a smoothed version of the received excitation sequence for adaptation of the short-term prediction coefficients. For the multi-tree coder, reconstructed output speech is evaluated using signal-to-quantization noise ratios (SNR), segmental SNRs, and informal listening tests.

要 約

본 논문에서는 음성의 short-term 계수 추출에 대한 새로운 방식을 제안하였으며, 데이타량 9.6Kbits/sec의 멀티 트리 부호기를 실현하였다. 이 트리 부호기는 총 지연 시간 2.5msec을 (6.4KHz 샘플링 주파수에서 16 샘플) 가지며, 좋은 출력 음질을 가지며, bit 오율 (BER) 10^{-3} 에서도 견실한 상태를 유지한다. 이 견실성은 short-term 계수 추출을 위해 수신된 여기 신호를 smoothing 하여, 병렬 구성과 함께 사용하므로 가능 하였다. 이 부호기의 출력 음성은 SNR, SNRSEG, 그리고 듣기 시험으로 평가 되었다.

I. Introduction

A tree coding scheme which is low delay and robust to errors is investigated. The requirement for low delay is important in telephone networks since delay can be accumulated as signals are transmitted through certain transmission media and switches. According to the International Telegraph and Telephone Consultative Com-

*大邱大學校 電子工學科
**大邱大學校 情報通信工學科
論文番號：93-37

mittee (CCITT) specifications, delay is required to be less than 5 msec and less than 2 msec delay is desirable [1]. Based on these specifications, an encoding delay of 2.5 msec (16 samples at the 6.4KHz sampling frequency) is maintained throughout this research. The delay requirement only allows the speech coder to estimate the spectral envelope of the signal in a backward adaptive fashion.

Another challenge in the backward mode is maintenance of robustness to channel errors at low data rates. At a low data rate coder, one bit error results in more serious coder performance degradation than it does in a high bit rate coder. Thus, numerous different adaptation structures for short-term prediction and long-term prediction have been investigated. We have found that in order to achieve good adaptation, it is critical to use the correct signal with parallel adaptation structure suggested by Cuperman and Gersho [2]. As the correct adaptation signal for short-term prediction, the smoothed excitation signal, which is obtained by filtering the excitation signal through the all-zero filter, is proposed. With this adaptation scheme in the 9.6Kbits/sec multi-tree coder, the coder produces good quality speech and is quite robust to errors.

Ⅱ. Multi-tree Coder

Fig. 1 shows the overall block diagram of a tree coder. Tree coders search for the optimal path by minimizing the distortion D. A path is a sequence of branches which is specified by the path map symbols. The released path map symbols are transmitted to the receiver. The received path map symbols are used in the code generator which reconstructs the output signal.

At low bit rates (below 16Kbits/sec), the mean squared error (MSE) is not a very meaningul performance measure for speech coding. The auditory masking effect in the human hearing system is thus included as the following weighting filter [3]

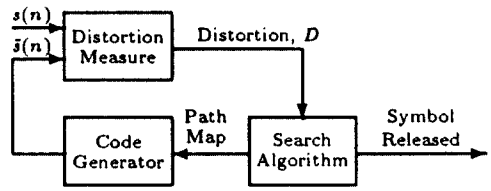

Figure. 1 Elements of a tree coder

$$W(Z)=\frac{A(Z)}{A(\gamma^{-1}Z)} \tag{1}$$

where

$$A(\gamma^{-1}Z)=1+\sum_{k=1}^{p} a_k\gamma^k Z^{-k} \tag{2}$$

and $A(Z)$ is the short-term predictor and γ is the degree of deemphasis of the error spectrum. Based on computer simulation, $\gamma=0.86$ was selected after informal listening tests.

As the searching algorithm, the (M,L) algorithm which keeps only the M best paths at each level of the code tree is implemented. In alow delay 16Kbits/sec tree coder, Iyengar and Kabal report that the prediction gain increases as delay (L) increases up to 8 samples and then saturates, but they say that the value of L is not critical [4]. They also report that the system performance (segmental SNR) rapidly improves for M up to 8 but it saturates when M is about 16. In our work, the (M,L) algorithm is used with $M=8$ and $L=16$ samples to have moderate encoding complexity and to meet the low delay requirement of our tree coders.

Gibson and Chang developed a multi-tree coder which consists of different rate trees interleaved with each other [5]. A multi-tree coder is known to produce rich high frequency components in the reconstructed speech signal. The multi-tree coder in our research consists of a tree with a rate of 2 bits and a tree with a rate of 1bit, which are interleaved with each other. The overall data rate of this coder is 9.6kbits/sec at a 6.4Kbits/sec sampling frequency.

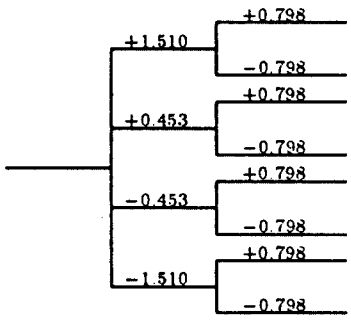

Figure. 2 An example of multi-tree code

In Fig. 2, an example of this multi-tree is shown at depth 2. The output symbols on the branches of the multi-tree are assigned as the output levels of MMSE Gaussian quantizers. The 1 bit Gaussian quantizer has output levels of ±0.798 while the 2 bit Gaussian quantizer has output levels of ±0.453 and ±1.510.

Ⅲ. Predictor Adaptation

1 Short-term Predictor Adaptation

In our research, the least squares (LS) method for extracting the coefficients of the short-term linear predictor is mainly concerned because of its good performance. The residual-driven lattice (RDL) recursive algorithm (which is the gradient adaptive lattice (GAL) type) and the least squares lattice (LSL) recursive algorithm are chosen in our research. These two algorithms are listed here.

• The RDL Algorithm [6]

Time Update

At each time instant, set the forward prediction error (FPE) $f_0(n)$ and the backward prediction error (BPE) $b_0(n)$

$$f_0(n)=b_0(n)=u(n). \tag{3}$$

where $u(n)$ is the quantized residual signal.

Order Update

For $k=1,2,\ldots,p$, and a leakage factor $\lambda=0.96875$, compute

$$f_k(n)=f_{k-1}(n)-\Gamma_k(n)b_{k-1}(n-1), \tag{4}$$

$$b_k(n)=b_{k-1}(n-1)-\Gamma_k(n)f_{k-1}(n), \tag{5}$$

$$c_k(n)=\lambda c_k(n-1)+2f_{k-1}(n)b_{k-1}(n-1), \tag{6}$$

$$d_k(n)=\lambda d_k(n-1)+f_{k-1}^2(n)+b_{k-1}^2(n-1), \tag{7}$$

$$\Gamma_k(n+1)=\frac{c_k(n)}{d_k(n)}, \tag{8}$$

where $c_k(n)$ and $d_k(n)$ are the prediction error variances, and the reflection coefficient Γ_k is expressed as the ratio between $c_k(n)$ and $d_k(n)$.

• The LSL Algorithm [7]

Time Update

At each time instant, set

$$f_0(n)=b_0(n)=\tilde{s}(n), \tag{9}$$

$$\xi_0(n)=0 \tag{10}$$

$$E_1^f(n)=E_1^b(n)=\lambda E_1^f(n-1)+f_0^2(n) \tag{11}$$

where $\tilde{s}(n)$ is the reconstructed output. The leakage factor λ is chosen so that $\lambda=0.99$

Order Update

For $k=1,2,\ldots,p$, compute

$$\Gamma_k(n)=\lambda\Gamma_k(n-1)+\frac{f_{k-1}(n)b_{k-1}(n-1)}{1-\xi_{k-1}(n)} \tag{12}$$

$$\Gamma_k^f(n)=\frac{\Gamma_k(n)}{E_k^f(n)}, \tag{13}$$

$$\Gamma_k^b(n)=\frac{\Gamma_k(n)}{E_k^b(n-1)}, \tag{14}$$

$$E_{k+1}^f(n)=\frac{E_k^f(n)-\Gamma_k^b(n)\Gamma_k(n)}{0.98^2}, \tag{15}$$

$$E_{k+1}^b(n)=\frac{E_k^b(n-1)-\Gamma_k^f(n)\Gamma_k(n)}{0.98^2}, \tag{16}$$

$$\xi_k=\xi_{k-1}(n)=\frac{b_{k-1}^2(n-1)}{E_k^b(n-1)}, \tag{17}$$

$$f_k(n) = f_{k-1}(n) - \Gamma_k^b(n) b_{k-1}(n-1), \quad (18)$$

$$b_k(n) = b_{k-1}(n-1) - \Gamma_k^f(n) f_{k-1}(n), \quad (19)$$

where $\xi k(n)$ is the gain parameter, and $E_{k+1}^f(n)$ and $E_{k+1}^b(n)$ are the variances of the FPE and the BPE.

2. Long-term Predictor Adaptation

In addition to short-term correlation, there is long-term correlation from pitch period to pitch period in a speech signal. The general form of a three-tap long-term (pitch) predictor can be represented in Z transform notation by

$$P(Z) = \sum_{i=-1}^{1} \beta_i Z^{-(M1+i)}, \quad (20)$$

where M_1 is the pitch lag and $(\beta_{-1}, \beta_0, \beta_1)$ is the set of long-term coefficients. The pitch coefficients of the backward prediction method can be calculated in the block method using the previous frame of data.

Another approach to backward long-term prediction is the backward recursive method developed by Pettigrew and Cuperman [8]. In the backward recursive method, the coefficients are updated every sample so that the computational complexity is evenly distributed. This backward method, which consists of pitch period tracking and coefficient adaptation, uses a least mean squares (LMS) type algorithm

In hybrid backward adaptive pitch adaptation, backward block adaptation and backward recursive adaptation are combined. In our tree coders, hybrid adaptation is always implemented. The hybrid adaptation performs better than either the block adaptation or the recursive adaptation alone [8].

3. Gain Adaptation Algorithm

The gain adaptation is given by the following equation :

$$g(n) = M(n-1) g^{\lambda}(n-1), \quad (21)$$

where $g(n)$ is the current step sizer (or the gain term in the vector case), $M(n-1)$ is the step size multiplier, which is a function of the index of the quantizer output level, and λ is a "forgetting" factor to combat channel errors. The value of the multiplier controls the expansion and contraction of the quantizer step size.

In our multi-tree coder, the robust Jayant adaptive algorithm in (21) is used to adapt the step size for the 2 bit quantizer. For the 1 bit quantizer, the step adaptation algorithm of adaptive delta modulation (ADM) is used.

Ⅳ. Short-term Predictor Adaptation Scheme

The output signal adaptation scheme, in which the output of each predictor is used as the adaptation input, produces the best error free performance, but it is very sensitive to channel errors because of long memory in the output signal. The complementary scheme to the output signal case is the use of the error input signal, which has short memory, as the adaptation input. The error signal is actually the quantized and received residual signal. Cuperman and Gersho have used this architecture in low delay vector excitation coding (LD-VXC) [2]. In LD-VXC, the 2 pole 6 zero short-term predictor (which is the LMS type) and the pitch predictor are adapted in parallel rather than in cascade Parallel adaptation means that the error $e(n)$ drives both the pitch predictor and the shortterm predictor. Additional work in this class is the RDL algorithm developed by Yatrou and Mermelstein [6]. Their research was based on an ADPCM system with the GAL type algorithm. The works in the LD-VXC and the RDL algorithm have significantly improved noisy channel performance at the ex-

pense of degradation in error free performance.

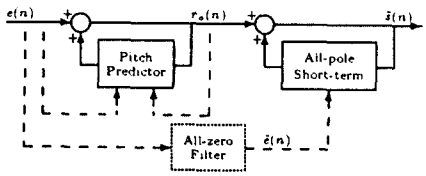

Figure. 3 Smoothed error adaptation scheme

Figure 3 is the new scheme we have proposed for predictor adaptation. The dashed line in the figure indicate which signals are used as in puts to the backward adaptation algorithm. The short-term lattic predictor is now driven with the smoothed error signal $\tilde{e}(n)$ which contains significantly more signal informaition than the error signal $e(n)$. The sensitivity to channel errors in the proposed scheme is reduced by using the error signal as the adaptation input, after the error signal has been shaped by an all-zero filter. This shaping provides the signal memory which is necessary for the coefficients of the all-zero filter are obtained as a truncated FIR approximaition to the all-pole short-term synthesizer. If the short-term predictor is denoted as $A(Z)=\sum_{k=1}^{p} a_k Z^{-k}$, $p=8$ and the all-zero filter is $1+C(Z)$ where $C(Z)=\sum_{k=1}^{q} a_k Z^{-k}$, $C(Z)$ is chosen to satisfy

$$\frac{1}{1-A(Z)} \cong 1+C(Z). \tag{22}$$

We then have

$$c_1 = a_1, \tag{23}$$

$$c_k = a_k + \sum_{j=1}^{k-1} c_j a_{k-j},\ 2 \le k < q. \tag{24}$$

The smoothed error signal $\tilde{e}(n)$ used for adaptation is given by

$$\tilde{e}(n) = e(n) + \sum_{k=1}^{q} c_k e(n-k). \tag{25}$$

In our tree coders, $C(Z)$ is usually chosen to be the eighth order filter ($q=8$). Note that the pitch predictor and the short-term predictor in the proposed scheme are updated in parallel which is very robust to transmission errors.

V. Multi-tree Coder Performance

Table 1: Ideal channel performance: LSL

sentence	adaptation	SNR(dB)	SNRSEG(dB)
1	output	16.90	16.17
	error	14.66	12.86
	smoothed	16.66	15.58
2	output	17.29	15.94
	error	14.42	12.68
	smoothed	16.63	15.26
3	output	14.40	13.11
	error	11.69	10.22
	smoothed	14.24	12.87
4	output	12.34	13.86
	error	10.75	11.43
	smoothed	12.27	13.33
5	output	13.97	14.56
	error	12.95	11.94
	smoothed	13.77	14.33

Table 1 shows the SNRs and the SNRSEGs for the output adaptation, the error adaptation, and the smoothed error adaptation signals in the LSL algorithm. It is clear from the Table 1 that the smoothed error adaptation scheme yields performance virtually equivalent to that of the output adaptation scheme for ideal channels. The LSL with the smoothed error signal adaptation improves 0.8-2.5dB in SNR and 2.0-2.7dB in SNRSEG over the LSL with the error signal adaptation.

Table 2: Ideal channel performance: RDL

sentence	adaptation	SNR(dB)	SNRSEG(dB)
1	error	13.92	12.34
	smoothed	15.56	15.24
2	error	13.74	12.34
	smoothed	15.43	14.94
3	error	10.93	9.77
	smoothed	12.56	12.06
4	error	10.27	10.80
	smoothed	11.13	12.95
5	error	11.54	11.23
	smoothed	12.89	14.07

This improvement is also evident in the case of the RDL predictor as shown in Table 2. Informal subjective listening tests substantiate the objective results in Tables 1 and 2. The output speech produced by the error signal adaptation scheme is very noisy while output speech produced by the output signal adaptation scheme and the smoothed error signal adaptation scheme have high quality with low granular noise. Comparing the two lattic predictors, the LSL type is superior to the RDL type.

Table 3: Error performance: LSL, sentence 1

target BER	adaptation	SNR(dB)	SNRSEG(dB)
0.0	output	16.90	16.17
	error	14.66	12.86
	smoothed	16.66	15.58
10^{-4}	output	8.84	10.23
	error	12.70	11.43
	smoothed	13.87	13.42
10^{-3}	output	1.59	3.10
	error	9.38	8.08
	smoothed	9.56	8.53
10^{-2}	output	-2.96	-2.04
	error	4.37	3.42
	smoothed	1.80	1.48

The simulation of error performance for each test sentence was repeated 20 times with different seeds in the generation of a random number. The values of SNR and SNRSEG in Tables 3 and 4 are the average values of simulation results.

Table 4: Error performance: RDL, sentence 1

target BER	adaptation	SNR(dB)	SNRSEG(dB)
0.0	error	13.92	12.34
	smoothed	15.56	15.24
10^{-4}	error	11.54	10.23
	smoothed	12.10	12.48
10^{-3}	error	8.80	7.36
	smoothed	6.48	6.19
10^{-2}	error	4.00	2.90
	smoothed	0.57	0.69

The performance of the LSL with the output singal adaptation rapidly drops as the error rate increases while the performance of the LSL with the error signal adaptation very slowly drops according to the error rates. The error performance of the LSL with the smoothed error adaptation is located between the two extrems, ie, the output adaptation case and the error adaptation case. However, the LSL predictor with the smoothed erroradaptation almost performs as well as the LSL with the e rror adaptation even at the error rate of 10^{-3}. The performance of the RDL case in the Table 4 has the similar trend as that of the LSL.

Ⅵ. Conclusions

In the 9.6Kbits/sec multi-tree coder, the over-all delay is maintained at 2.5 msec. Using the smoothed error signal adaptation LSL algorithm, the performance of the coder is improved 0-2.7dB in the SNRSEG term over that of this coder using the error signal adaptation. This coder produces quite good output speech quality. Moreover, the multi-tree coder is robust to errors p to 10^{-3} bit error rate. Consequently, the multi-tree coder using the smoothed error signal adaptation is, we can say, an excellent tradeoff between the error free performance and the error performance.

References

1. N. S. Jayant, "High-quality coding of telephone speech and wide band audio," *EIII Commun. Mag.*, vol. 28, pp. 10-20, Jan. 1990.
2. V. Cuperman, A. Gersho, R. Pettigrew, J. J. Shynk, and J. H. Yao, "Backward adaptation for low delay vector excitation coding of speech at 16Kbit/s," in *Proc. GLOBECOM*, pp. 34.2.1-34.2.5., 1989.
3. B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 614-617, 1982.
4. V. Lyengar and P. Kabal, "A low delay 16 kb.s speech coder," *IEEE Trans. Signal Processing*, vol. 39, pp. 1049-1057, May 1991.
5. J. D. Bibson and W. W. Chang, "Factional rate

multi-tree speech coding," in *Proc. GLOBECOM*, pp. 1906-1910, 1989.
6. P Yatrou and P. Mermelstein, "Ensuring predictor tracking in ADPCM speech coders under noisy transmission conditions," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 249-261, Feb. 1988.
7. R. C. Reininger and J. D. Gibson, "Backward adaptive lattice and transversal predictors in ADPCM," *IEEE Trans. Commun.*, vol. 33, pp. 74-82, Jan. 1985.
8. R. Pettigrew and V. Cuperman, "Backward pitch prediction for low-delay speech coding," in *Proc. GLOBECOM*, pp.34.3.1-34.3.6, 1989.

禹 洪 棣(Hong Chae Woo) 정회원
1957년 1월 20일생
1980년 2월 : 경북대학교 전자공학과(공학사)
1979년 12월~1985년 12월 : 국방과학연구소 연구원
1988년 12월 : Texas A&M 대학교 전기공학과(공학석사)
1991년 12월 : Texas A&M 대학교 전기공학과(공학박사)
1992년 3월~현재 : 대구대학교 공과대학 전자공학과 조교수

李 琡 昱(Chae Wook Lee) 正會員
1957年 12月 24日生
1980年 2月 : 韓國航空大學 通信工學科(工學士)
1987年 3月 : 東京工業大學 大學院 電氣電子工學科(工學碩士)
1990年 3月 : 東京工業大學 大學院 電氣電子工學科(工學博士)
1990年 3月~現在 : 大邱大學校 工科大學 情報通信工學科 助敎授
※主關心分野 : 디지탈 信號處理, 光通信시스템, 符號理論

文 炳 顯(Byung Hyun Moon) 正會員
1960년 10월 16일생
1985년 6월 : Southern Illinois University 전자공학과(공학학사)
1987년 5월 : University of Illinois 전자공학과(공학석사)
1990년 12월 : Southern Methodist University 전자공학과(공학박사)
1991년 9월~현재 : 대구대학교 전자공학과 전임강사