IMPROVED CS-CELP SPEECH CODING IN A NOISY ENVIRONMENT USING A TRAINED SPARSE CONJUGATE CODEBOOK

Akitoshi Kataoka, Sachiko Hosaka, Jotaro Ikedo, Takehiro Moriya and Shinji Hayashi*

NTT Human Interface Laboratories
3-9-11, Midori-Cho
Musashino-Shi, Tokyo, 180 Japan

*NTT Wireless Systems Laboratories
1-2356 Take
Yokosuka-Shi, Kanagawa, 238-03 Japan

ABSTRACT

A high-quality 8-kbit/s speech coder based on Conjugate Structure CELP (CS-CELP) is proposed that uses a trained sparse conjugate codebook. The trained sparse conjugate codebook improves speech quality for noisy speech. This codebook consists of two sub-codebooks and each sub-codebook consists of a random component and a trained component. Each component has excitation vectors consisting of a few pulses. In the random component, pulse position and amplitude are determined randomly. The trained component is determined by training. Subjective tests (Differential Mean Opinion Score, DMOS and Mean Opinion Score, MOS) indicated that this codebook improves speech quality compared with the conventional trained codebook for noisy speech. The MOS showed that the quality of improved CS-CELP is equivalent to that of the 32-kbit/s ADPCM for clean speech.

1. INTRODUCTION

The ITU-T is currently standardizing an 8-kbit/s speech coding algorithm. During the selection phase, there were two candidates: Algebraic Code Excited Linear Predictive (ACELP) from France Telecom and the University of Sherbrook[1] and Conjugate Structure Code Excited Linear Predictive (CS-CELP) from NTT[2][3]. The speech assessment group found that both codecs had quality equivalent to that of the 32-kbit/s Adaptive Differential Pulse Code Modulation (ADPCM) in error-free conditions and that both codecs met the requirements for random bit error and tandem conditions. Unfortunately, neither codec met the requirements for input level variation, frame erasure, or environmental noise conditions. In the Tokyo ad-hoc meeting of the ITU-T in March 1994, France Telecom and NTT agreed to form a task group to arrange a compromise.

The ITU-T 8-kbit/s codec will be used for FPLMTS (Future Public Land Mobile Telecommunication Systems), which is the next generation of communication systems. Since these systems use compact sets, people can communicate with anyone in stations, cars, airports, etc. But these are noisy environments. Therefore, good speech quality in the environmental noise conditions is important.

This paper focuses on environmental noise conditions by using CS-CELP coding. The ITU-T requires that the quality of the 8-kbit/s coder be no worse than that of the 32-kbit/s ADPCM under environmental noise conditions. Speech quality is evaluated by using the DMOS: this compares the coded speech with the original speech plus noise. Therefore, the coder must efficiently encode the noise as well as the speech.

To improve the quality of coded speech in a noisy environment, we tried several techniques and found that two factors were important: the Line Spectrum Pair (LSP) parameters and the fixed-shape codebook. The LSP parameters have to be trained using a mixed database consisting of clean speech and noisy speech. To minimize spectrum distortion in the clean speech, we rearranged the bit allocations of the coder and assigned one more bit for the LSP quantization.

This paper presents a trained sparse conjugate codebook that improves speech quality in a noisy environment. This codebook consists of a random component and a trained component. Each excitation vector consists of a few pulses. In addition to improving the quality of coded speech in a noisy enviroment, this codebook reduces the computational complexity of a fixed-shape codebook search.

2. CONJUGATE STRUCTURE

The conjugate structure is so named because the relationship between the two codebooks used in this scheme is conjugate[4]. In conjugate VQ, an output codevector is generated by summing two vectors, each stored in a different codebook. Both the shape vector and the gain vector are summations of two vectors from two sub-codebooks. The shape-excitation vector $\mathbf{C}_i$ is the sum of the two excitation vectors

$$\mathbf{C}_i = \theta_1 \cdot \mathbf{C}_{sub1j} + \theta_2 \cdot \mathbf{C}_{sub2k} , \qquad (1)$$

where θ_1 and θ_2 are signs and $\mathbf{C}_{sub1j}$ and $\mathbf{C}_{sub2k}$ are excitation vectors in the sub-codebooks.

We previously proposed pre-selection of the codebook search to reduce complexity[2][3]. The pre-selection process selects M (out of N) candidates by using the cross-correlation $(\mathbf{X}^T\mathbf{H})\mathbf{C}_i$, where N is the sub-codebook size, $\mathbf{X}$ is the weighted input speech, $\mathbf{H}$ is the impulse response matrix, and $\mathbf{C}_i$ are the fixed-shape excitation vectors. In closed-loop analysis, when the best excitation vector is determined, only the pre-selected candidates are filtered by the impulse response matrix. Since the fixed-shape codebook consists of two sub-codebooks, the pre-selection process is performed for each sub-codebook.

3. TRAINED SPARSE CONJUGATE CODEBOOK

The fixed-shape codebook consists of Gaussian random vectors or trained vectors. Since each vector has a set of full

0-7803-2431-5/95 $4.00

pulses, the search of the fixed-shape codebook involves a lot of computational complexity. To reduce the complexity, a sparse codebook has been proposed. Since a sparse codebook has only a few non-zero pulses, it can reduce the complexity and also memory requirement. Pulse position and amplitude were determined randomly, that is, each excitation vector was generated by clipping the center of a random Gaussian signal. But this did not provide high speech quality.

We found that each trained excitation vector is pulsive and that each vector can be replaced by a few pulses. We propose the trained sparse conjugate codebook, in which each excitation vector consists of a few pulses: pulse position and amplitude are determined by training. This sparse codebook reduces the complexity of the cross-correlation calculations during pre-selection and the complexity of the filtering operation during closed-loop analysis.

If the length of the excitation vector is L (=40 in CS-CELP), a full-pulse codebook has L pulses. In cross-correlation calculations, the full-pulse codebook needs L multiplication and addition operations for each excitation vector. On the other hand, if the sparse codebook has K(=2 or 5) pulses, these operations are needed only K times. In the filtering operation, the full-pulse codebook needs $L(L-1)/2$ multiplication and addition operations. Although the complexity of the filtering operation in the sparse codebook depends on pulse positions, the sparse codebook needs only $K(K-1)/2 + K(L-K)$ multiplication and addition operations in the maximum case.

Figure 1 shows the segmental SNR versus the number of pulses in the sub-excitation vector. Since the excitation vector in conjugate codebook is the summation of two sub-vectors, the excitation vectors have twice as many pulses. The dotted lines show that the sub-excitation vector is a full-pulse (=40) vector. Even if the sub-codebook consists of only two pulses, the difference compared with the full-pulse codebook is small.

4. SPARSE CONJUGATE CODEBOOK FOR NOISY ENVIRONMENT

Although the trained codebook provides high-quality for clean speech, it provides unsatisfactory quality for noisy speech, because the trained codebook is not always able to handle the noise. An untrained codebook may actually provide better quality than a trained one for noisy speech, but it cannot provide high quality for clean speech.

We therefore adapted the trained sparse conjugate codebook to handle a noisy environment. This modified sparse codebook consists of a random component and a trained component, as shown in Figure 2. Each component has excitation vectors consisting of a few pulses. In the random component, pulse position and amplitude are determined randomly; that is, each excitation vector is generated by center-clipping a random Gaussian signal. Since the random component does not depend on the speech characteristics, it handles noise better than the trained one. To maintain high quality for clean speech, the trained component is determined by training. However, the training must be done considering the influence of the random component, and the training database is a mixed one consisting of clean speech and noisy speech.

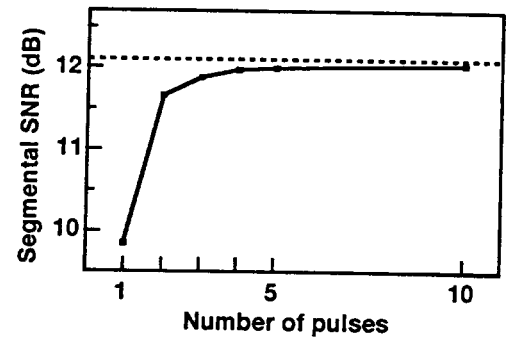

Fig.1. Segmental SNR versus number of pulses in sub-excitation vector.

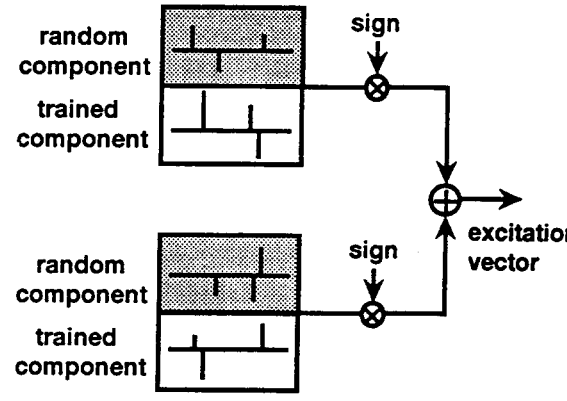

Fig.2. Trained sparse conjugate codebook for noisy environment.

Since each excitation vector is the summation of two sub-vectors, there are three possible combinations of sub-vectors: 1) a sub-vector from each of the random components, 2) one sub-vector from the random component and one from the trained component, 3) a sub-vector from each of the trained components. The coder selects the most appropriate combination by closed-loop analysis for type of speech, noisy or clean. We expected combination 1 to be used for noisy speech and combination 3 for clean speech. However, the proportion of each combination was as follows: for clean speech, combination 1: 14.0%, combination 2: 46.2% and combination 3: 39.8% and for noisy speech, combination 1: 14.2%, combination 2: 46.6% and combination 3: 39.2%. Combination 2 is selected the most often. The random component is useful for both clean speech and noisy speech. The new sparse codebook can handle various speech conditions by selecting the sub-vector from each component.

5. TRAINING THE SPARSE CONJUGATE CODEBOOK

The sub-codebooks are trained by the generalized Lloyd algorithm. Each sub-codebook is trained alternately. While one is being trained, the other is fixed. This section describes the procedures for training one sub-codebook.

The training algorithm alternately iterates two processes. One process first determines the code of the fixed-

shape codebook for a target vector, then for all frames accumulates the target vectors, the reconstructed speech, the gains, and the impulse response matrices. The target vector is generated by subtracting three vectors from the input speech; the three vectors are the zero input response from the previous frame, the synthesized output from the adaptive codebook, and the synthesized output from the other sub-codebook. The other process determines the trained excitation vector from these accumulated vectors. The first process is identical to the fixed-shape codebook search in an encoder. The second process generates the new excitation vector that minimizes the distortion between the input speech and the reconstructed speech for all frames. The trained full-pulse excitation vector $\mathbf{C}_i$ is given by

$$\mathbf{C}_i = \psi \sum_j (g_j \mathbf{H}_j)^T \mathbf{X}_j, where\ \psi = (\sum_j (g_j \mathbf{H}_j)^T (g_j \mathbf{H}_j))^{-1}, \quad (2)$$

where $\mathbf{X}$ is the target vector, g is the gain, and $\mathbf{H}$ is the impulse response matrix.

In the sparse codebook, the second process generates the excitation vector that consists of a few pulses. The position and amplitude of each pulse are determined sequentially; that is, first position and amplitude of one pulse are determined by minimizing the distortion between the input speech and the reconstructed speech, and then the second pulse position is determined while the first pulse position is fixed. The amplitudes of the two pulses are redetermined after the second pulse position has been determined. In this manner, M pulse positions and amplitudes are determined by minimizing the distortion between the input speech and the reconstructed speech. The trained sparse excitation vector $\mathbf{C}_i$ is given by

$$\mathbf{C}_i = \psi \sum_j (g_j \mathbf{H}_{(sub)j})^T \mathbf{X}_{(sub)j}, \quad (3)$$

where $\mathbf{H}_{sub}$ is the subset of the impulse response matrix and $\mathbf{X}_{sub}$ is the subset of the target vector. These procedures are similar to those in multi-pulse coding. If the code of a random component is selected in the first process, the accumulated vectors for the selected code are used in training the trained component of the other sub-codebook, because the excitation vector of a random component is not trained.

6. PERFORMANCE OF TRAINED SPARSE CONJUGATE CODEBOOK

The trained sparse conjugate codebook is evaluated by using an objective measure (segmental SNR). When the sub-excitation vectors of each sub-codebook have N pulses, the excitation vectors have 2N pulses. Figure 3 shows segmental SNR versus the ratio of the trained component to the random component for clean speech, when the sub-excitation vector of each component has the same number of pulses. When the sub-codebook consists only of the random component, it achieves a low score. As the ratio of the trained component is increased, the segmental SNR improves. When the sub-codebook consists only of the trained component, it achieves a high score. However, the fully trained codebook provides unsatisfactory quality

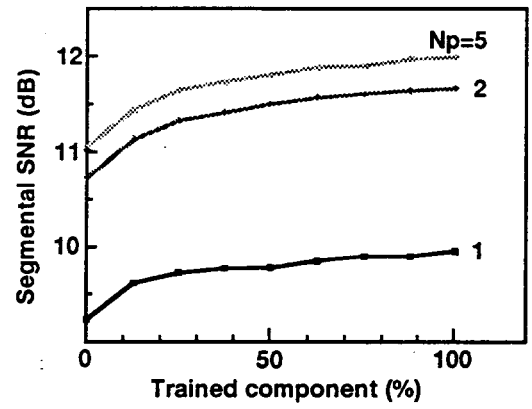

Fig.3. Segmental SNR versus ratio of the trained component to the random component in sub-codebook. Np is the number of pulses in the sub-codebook.

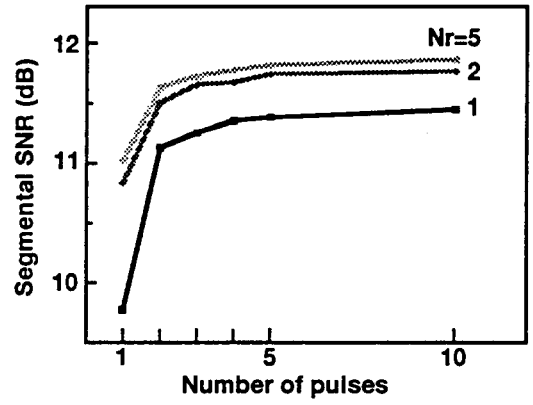

Fig.4. Segmental SNR versus the number of pulses of sub-excitation vectors in the trained component. Nr is the number of pulses in the random component.

for noisy speech, as is discussed in Section 4. Even if the ratio of the trained component is 25%, the degradation is small.

Figure 4 shows segmental SNR versus the number of pulses of the sub-excitation vector in the trained component for clean speech, when the number of pulses in the random component is a constant and each component has the same number of sub-excitation vectors (64 untrained excitation vectors and 64 trained ones). Even if the number of pulses in the trained component is 2 for various numbers of pulses in the random component, the degradation is small. When the number of pulses in the random component is 2 ($N_r = 2$), segmental SNR is almost as good as for five pulses. The two-pulse sub-codebook achieves almost the same performance as ones with more pulses.

7. EVALUATION OF SPEECH QUALITY

We evaluated the quality of the proposed coder by using both the DMOS and the absolute (ordinary) MOS. In the trained sparse conjugate codebook, the ratio of the random component to the trained component was 50% and each component had the same number of pulses (2 or 5).

Table 1 **Differential MOS test results**

(Each sub-codebook size is 128. The conventional trained codebook consists of 128 trained excitation vectors. The trained sparse codebook consists of 64 untrained excitation vectors and 64 trained ones.)

Additional noise condition S/N ratio (dB)	Original	ADPCM	CS-CELP	Improved CS-CELP	
			conventional trained full-pulse codebook	trained two-pulse codebook	trained five-pulse codebook
Car noise (10 dB)	4.67	4.01	2.46	3.20	3.21
Car noise (20 dB)	4.79	4.17	3.15	3.83	3.88
Babble noise (30 dB)	4.90	4.30	4.11	4.29	4.49
Motor noise (10 dB)	4.69	4.01	2.53	3.02	3.29

Table 2 **Absolute MOS test results**

Condition	Original	ADPCM	CS-CELP	Improved CS-CELP	
			conventional trained full-pulse codebook	trained two-pulse codebook	trained five-pulse codebook
Clean speech	4.54	4.01	4.10	4.18	4.20
Car noise (10 dB)	2.11	1.92	1.58	1.84	1.90
Car noise (20 dB)	2.69	2.63	2.30	2.52	2.53
Babble noise (30 dB)	3.60	3.22	3.15	3.47	3.40
Motor noise (10 dB)	2.11	1.97	1.67	1.85	1.92

The source material consisted of 30 sentence-pairs. The number of listeners was 24. The additional noise was car noise, babble (office) noise, and motor noise; three S/N ratios were used. The DMOS was rated in comparison with the original signal on a 5-point degradation category scale: degradation #5 was inaudible, #4 was audible but not annoying, #3 was slightly annoying, #2 was annoying, and #1 was very annoying. The absolute MOS of speech quality was rated as excellent (5), good (4), fair (3), poor (2), or unsatisfactory (1).

The subjective results are shown in Tables 1 and 2. The absolute MOS shows that the quality of improved CS-CELP is equivalent to that of the 32-kbit/s ADPCM for clean speech. Under environmental noise conditions (S/N=10, 20, 30 dB), the sparse conjugate codebook improves speech quality compared with the conventional trained codebook, in which all excitation vectors are trained. The absolute MOS shows that the subjective quality of improved CS-CELP with a sparse codebook is equivalent to that of the 32-kbit/s ADPCM; however, in the DMOS, the quality of improved CS-CELP is slightly worse than that of the ADPCM.

In the DMOS tests, listeners tend to concentrate only on the difference between the original speech with noise and the coded speech. Therefore, the effects of noise reduction by the postfilter of the coder, for example, are ignored. However, in practice, telephone users do not compare the original speech with the coded speech, but evaluate the quality absolutely. Consequently, the MOS results reflect the actual situation better than DMOS, while the DMOS results are useful to identify the difference.

The absolute MOS shows that the listeners give low scores even for the original speech with noise. The DMOS results also show that the difference between the original speech and coded speech becomes small, when the S/N ratio increases. The quality of improved CS-CELP is better than that of the ADPCM for a high S/N ratio (30 dB). We think that it will be uncommon to use a telephone in such a very noisy environment (10 dB). We concluded that the improved CS-CELP is useful for noisy environments.

8. CONCLUSIONS

We proposed an improved 8-kbit/s CS-CELP speech coder that uses a trained sparse conjugate codebook. This codebook consists of two sub-codebooks and each sub-codebook consists of a random component and a trained component. Each excitation vector consists of a few pulses. The trained sparse conjugate codebook can handle various speech conditions by selecting the sub-vector from each component. Subjective tests (DMOS and MOS) indicated that this codebook improves speech quality compared with the conventional trained codebook for noisy speech. The MOS showed that the quality of this new coder is equivalent to that of the 32-kbit/s ADPCM for clean speech.

Acknowledgements

We wish to thank Dr. Nobuhiko Kitawaki and Takao Kaneko for guiding our research. We are also grateful to Dr. Kazunori Mano for his helpful advice and discussion.

REFERENCES

[1] R. Salami, C. Laflamme, and J-P. Adoul: "ACELP Speech Coder at 8kbit/s with a 10ms frame: A Candidate for CCITT Standardization", *Proc.* IEEE Workshop on Speech Coding, pp.23-24, 1993

[2] A. Kataoka, T. Moriya and S. Hayashi: "An 8-kbit/s Speech Coder Based on Conjugate Structure CELP", *Proc. ICASSP'93*, pp.592-595, 1993

[3] A. Kataoka, T. Moriya and S. Hayashi: "Implementation and Performance of an 8-kbit/s Conjugate Structure CELP Speech Coder", *Proc. ICASSP'94*, II-93-96, 1994

[4] T. Moriya and H. Suda: "An 8 kbit/s Transform Coder for Noisy Channels", *Proc. ICASSP'89*, pp.196-199, 1989.