ENHANCING THE CODEBOOK FOR IMPROVING THE SPEECH QUALITY OF CELP CODERS

Bernt Ribbum, Andrew Perkis and K.K. Paliwal

ELAB
The Norwegian Institute of Technology
Trondheim, Norway

ABSTRACT - A Code Excited Linear Predictive (CELP) coder with a stochastic-multipulse (STMP) codebook is presented. The LPC residual exhibits a certain structure due to non-linearities in the glottal excitation. This structure can be exploited by a refinement of the STMP excitation signal, as a training procedure for the codebook. The algorithms are described and results are reported, both in terms of segmental SNR and subjective preference.

INTRODUCTION

Code Excited Linear Predictive (CELP) coders have shown considerable promise for speech coding at bit rates as low as 4.8 kbps. A CELP coder (Schroeder & Atal, 1985) comprises the Linear Predictive (LP) filter for pitch analysis P(z), and the short-term filter A(z) to account for the formants. A weighting filter $A(z/\gamma)$ is important for utilizing the human ear auditory masking properties, by moving quantization noise into frequency regions with a high signal level. The appropriate LP excitation signal is found using an analysis-by-synthesis algorithm performing a search through the codevectors $c_i(k)$.

Our basic structure is shown in figure 1, where the weighting filter has been moved out of the main loop to reduce the synthesis complexity. Also, the effect of the short term filter memory is subtracted from the incoming speech signal, to allow the synthesis filters to be zeroed for each block.

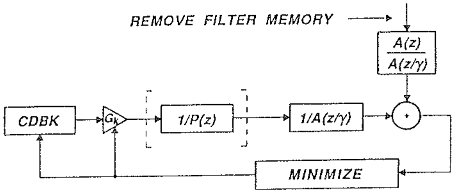

FIGURE 1 - Basic CELP coder

Restricting the pitch lag downwards to 40 samples will have the same effect on the pitch filter, which can also be removed from the synthesis loop. The coder used in our simulations uses a "recursive codebook" or self excitation codebook to take care of the signal pitch, and keeping the lag ≥40 makes this codebook search an independent task (Paliwal, 1987).

Despite those complexity reduction methods the main problem in realising a CELP coder is the computational load involved at the coder side, and different methods for further reducing the complexity have recently been proposed.

By moving the search for the best codevector from time domain to frequency domain the convolution operations may be replaced by (complex) multiplications, as the error signal is invariant under orthonormal transformations (Trancoso & Atal, 1986). The cost of computing the error signal can therefore be reduced by a factor 10 at the expense of performing DFT's and increasing the need for stored information (i.e. storing frequency representations of the codevectors). Another, non-ideal, method for reducing the computational load is to perform some pre-selection of the codevectors prior to creating the synthesized speech, as is covered in (Paliwal, 1988).

The cost involved in synthesizing the speech may also be reduced by restricting the codebook to contain overlapping data. Filtering of any codevector except the very first will then be reduced to filtering only one or two new samples and subtracting the contribution from the oldest one or two. This will as well reduce the need for storage. It is shown that stochastic codevectors sharing all samples but one or two will perform as well as a standard codebook of non-overlapping vectors (Kleijn, Krasinski & Ketchum, 1988).

A final method used for complexity reduction is to restrict the codevectors to containing only a few non-zero entries. The sparse vector or stochastic-multipulse approach will reduce the number of multiplications needed to filter the excitation signal by a factor of approx. 10, when a 40-sample codevector contains 4 pulses only.

This paper describes our simulations run with the stochastic-multipulse self excited linear predictive speech coder and the experiments performed to enhance the quality of the codebook.

In the next chapter the stochastic-multipulse codebook will be described with implications to enhancements. The following chapter will describe our methods chosen for codebook training, and the results are reported in the last chapter.

STOCHASTIC-MULTIPULSE CODEBOOK

The generation of stochastic-multipulse (STMP) codebooks is motivated by the success of multipulse-excited (MP) LPC-based coders. For the MP coder to perform well at least 1 pulse per ms is required, and the bit rate will be in excess of 10 kbit/s. The pulse positions and amplitudes will also be successively optimized, with a possible non-global optimum as the result. When the LP filters are excited from codevectors, the pulses will be optimized simultaneously, and the bit rate will be reduced. A saturation in quality is shown to occur (Paliwal, 1987) when a 40-sample codevector contains 4 stochastically generated pulses, or one pulse per 1.25 ms. We will in this paper concentrate on codevectors with this property. The pulse amplitudes are Gaussian zero-mean, unit-variance, and the positions uniformly distributed within the codevector.

Upon examining the residual signal after removing the pitch and inverse-filtering with the LPC-filter A(z) we are faced with the fact that a certain structure is still present in the signal. This structure has been described (Sreenivas, 1987) as a result from the vocal-tract excitation, the glottis. Apart from the large pitch pulses we will have a slowly varying smooth component in the speech signal that is due to the non-linear glottal wave shape. When the speech is inverse-filtered the 2nd derivative of this glottal wave will be present, in form of large pulses followed shortly after by a second pulse of (possibly) opposite sign.

The double-pulse nature found in the LPC residual is worth exploiting. The pulses may be parameterised, but the result would be expensive in terms of bitrate. Our approach is to try to collect multipulse codevectors that will reflect this nature as a means of generating more accurate synthetic speech at no extra computational cost.

Using a stochastically populated multipulse 1024-codebook the plots in figure 2 show the most-used codevectors in coding approximately 26 seconds of Norwegian, male/female speech.

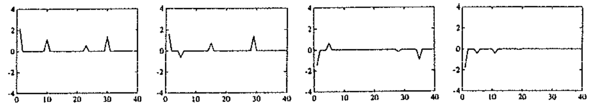

Figure 2 - Codevector plots

This clearly shows that most of the chosen codevectors exhibit the double-pulse nature. It is also worth noticing that the larger pulses are as a rule located in the beginning of the codevectors; this being a result of a single-frame-only error signal generation, and worth exploiting.

We are therefore faced with a situation where a trained codebook reflecting some of the signal properties most likely will give a speech coder with increased quality. At the same time it is favourable not to give up the interesting features of a non-trained, stochastic codebook in terms of the signal-independent performance.

Training a codebook is normally done by using the K-means algorithm (Linde, Buzo & Gray, 1980), which will produce codevectors as the centroids of accumulated training data. This in our case has two major drawbacks: the algorithm would not be able to generate the desired multipulse codebook, and the enormous amount of training data required makes the procedure non-feasible in terms of computer cost.

A different, non-ideal, training algorithm may however be used, which will result in a codebook combining both the trained-codebook properties and also stochastic-codebook features.

TRAINING PROCEDURES

We have considered two basic methods for training a STMP codebook, both based on a selection from existing codebooks as a result of running training data through the coder. For each vector in the codebook a log is updated when the vector gives the minimum weighted error for a speech segment. The statistics we collect from our simulations are:

- number of times each codevector is selected
- accumulated weighted MSE for the vectors chosen
- min/max MSE

The data obtained gives the possibility for generating a large variety of codebooks based on the different logged measures.

By selecting a specified number of vectors to retain, a new codebook is generated by substituting the remaining vectors by new, purely random entries. The algorithm is then repeated with the new codebook as input, and an increased number of vectors are kept for the next generation. Two basic methods may be used when selecting the vectors for further use, as shown in figure 3.

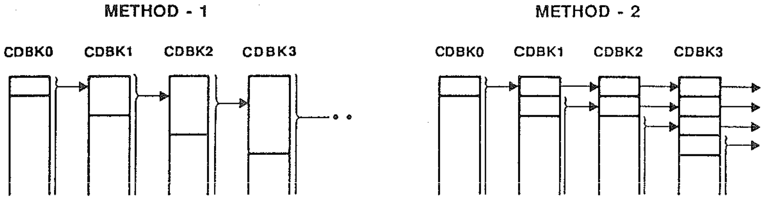

Figure 3 - Selecting vectors

The two methods may perform differently, but will probably give approximately equal results. We have therefore decided to concentrate upon method-1, in which all codevectors compete on an equal basis to be chosen for the next codebook.

By iterating the training sequence as described the codevectors chosen should grow into a codebook of equally "good" vectors expressing the inherent structure of the LPC residual. But as described earlier we will keep part of the codebook untrained to serve as a purely stochastic database.

RESULTS

The training data used in our simulations was made up of approx. 5 minutes of Norwegian speech, equally divided between male and female speakers (generated with normal background noise). Even when we use the STMP codebook, running the training procedure is costly in terms of computer time. The codebook size has therefore been reduced to 512 vectors for our simulation purposes. Our primary aim was to generate a codebook consisting of 256 trained and 256 random vectors. Following method-1 described above we successively selected 32, 64, 96, -- up to 256 vectors from the most popular (i.e. most frequently chosen) using the training data. By using the number of times each vector was selected only, the following plots show the distribution among the vectors in the initial and final codebooks.

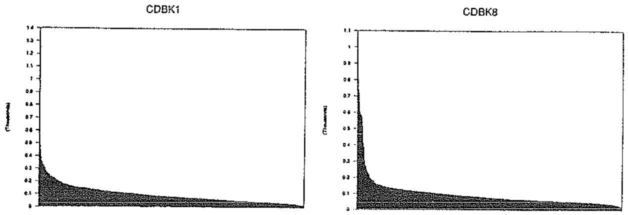

Figure 4 - Vector selection distribution

We can observe a certain flatness of the distribution as we move on to later generation of codebooks; no astonishing change can be spotted, however. By coding two sentences not part of the training data, we find the segmented SNR's given in Table 1a, (CDBK denotes the original STMP codebook; CDBKX is the enhanced codebook). From a paired-comparison test (Sreenivas, 1987) we find the subjective preference scale given in Table 1b.

Original	-	Original	0.00
CDBK	11.50 dB	CDBK	-2.40
CDBKX	11.36 dB	CDBKX	-2.61

Table 1a
Segmental SNR

Table 1b
Subjective Preference

As can be seen from the tables, no increase in quality is observed.

The results from the training procedure can however be further exploited. By selecting the 32 best vectors from each of the generated codebooks, based on the additional statistical information obtained (and assuring against duplicates), new codebooks are generated:

CDBKA - Selection based on minimum MSE
CDBKB - minimum MSE + popularity (weighted)
CDBKC - absolute minimum error obtained

CDBKB is generated by assigning to each vector a "quality" factor

$$f = \alpha \cdot \sqrt{MSE/MSEMAX} + (1-\alpha) \cdot NchosenMax/Nchosen \qquad (1)$$

where the best value of α is found to be 0.5, and "good" vectors correspond to minimum f. The results are given in table 2, where we find minimal differences. (Again, the original codebook CDBK is used as a reference; also, CDBKC is excluded from the subjective test as a result of informal listening.)

Original	-	Original	0.00
CDBK	11.50 dB	CDBK	-2.40
CDBKX	11.36 dB	CDBKX	-2.61
CDBKA	11.27 dB	CDBKA	-4.56
CDBKB	11.75 dB	CDBKB	-2.79
CDBKC	11.33 dB		

Table 2a
Segmental SNR

Table 2b
Subjective Preference

An interesting result from the codebook training is found, however, by using largely diminished versions of the codebooks. By using the first 32 vectors from the codebooks only, we observe a noticeable subjective improvement in both codebooks CDBKX and CDBKB:

Original	-
CDBK	9.88 dB
CDBKA	7.53 dB
CDBKB	9.91 dB
CDBKX	9.48 dB

Table 3a
Segmental SNR

Original	0.00
CDBK	-2.67
CDBKA	-4.94
CDBKB	-2.20
CDBKX	-2.54

Table 3b
Subjective Preference

CONCLUSIONS

Training procedures for the codebook excitation signal in CELP coders have been presented. By running simulations and performing listening tests it may seem as if no increase in quality is found.

By using simulation data it is found, however, that codevectors can be chosen to compose a codebook of minimal size. These codebooks are suitable for low complexity coders at the cost of some subjective degradation.

Design of small codebooks also raise a number of interesting questions for further work.

ACKNOWLEDGEMENTS

This work was supported by the multiclient project in speech processing at ELAB, The Norwegian Institute of Technology, Trondheim, Norway.

REFERENCES

Kleijn, W.B., Krasinski, D.J., & Ketchum, R.H. (1988) *Improved Speech Quality and Efficient Vector Quantization in SELP*, Proc. ICASSP, pp. 155-158, 1988.

Linde, Y., Buzo, A., & Gray, R.M. (1980) *An Algorithm for Vector Quantizer Design*, IEEE Trans. on Comms., pp. 84-95, January 1980.

Paliwal, K.K. (1987) *Stochastic, Multipulse and Self Excited Linear Predictive Coders for Low Bit-Rate Coding of Speech*, ELAB report STF44 F87108.

Paliwal, K.K. (1988) *Reduced-Complexity Stochastic-Excited Coder for Low Bit-Rate Coding of Speech*, Proc. EUSIPCO, pp. 1031-1034, 1988.

Schroeder, M.R. & Atal, B.S. (1985) *Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates*, Proc. ICASSP, pp. 937-940, 1985.

Sreenivas, T.V. (1987) *Modelling LPC-Residue By Components for Good Quality Speech Coding*, ELAB report STF44 F87147.

Trancoso, I.M. & Atal, B.S. (1986) *Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders*, Proc. ICASSP, pp. 2375-2378 1986.