3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

COMPLEXITY REDUCTION FOR FEDERAL STANDARD 1016 CELP CODER

M. MAUC, G. BAUDOIN and M. JELINEK

ESIEE, BP 99 - 93162 NOISY-LE-GRAND CEDEX - FRANCE

ABSTRACT

In the past recent years, it appeared that sparse code books in CELP coders provide significant computation reduction without degrading the quality of the synthetic speech. We present a method that reduces the calculation complexity of the cross-correlation between the original speech p and the synthetic speech $\hat{p}$ in case of ternary valued samples (-1, 0, +1) code book sequences.

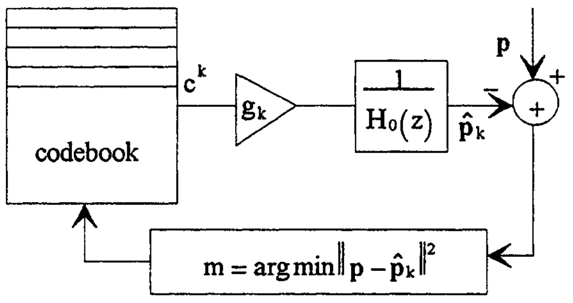
After describing the method, we apply it to the Federal Standard 1016 CELP coder. We show that the computation reduction for calculating the cross-correlation can be up to 5.

keywords: Speech coding, Linear prediction, Algebraic method.

1. INTRODUCTION

Since their introduction by M.R. Schroeder and B.S. Atal [1], CELP coders have proved to be very efficient for the coding of speech at medium and low bit rates. Several methods have been proposed in order to reduce the complexity of these hybrid coders [2] [3] [4].

Figure 1 represents one analysis by synthesis loop.



We use the following notations :

N is the frame size.
N' is the sub frame size with N' = N / 4.
c^k is the kth code book sequence of length N'.
g_j is the optimal corresponding gain.
p is the perceptual memoryless speech signal of length N'.
$\hat{p}_k$ is the synthetic perceptual memoryless signal corresponding to the response of the synthesis filter $1/H_0(z)$ to the code book sequence c^k.
T is the number of sequences of the code book.
P is the prediction order.

The optimum CELP sequence c^j minimizes the Least Square Error E(k) between p and $\hat{p}_j$:

(1) $$E(k) = \|p - \hat{p}_k\|^2$$

or equivalently maximizes the following performance criterion :

(2) $$PC(k) = \frac{\langle p, \hat{p}_k \rangle^2}{\langle \hat{p}_k, \hat{p}_k \rangle} = \frac{\beta_k^2}{\alpha_k^2}$$

where

(3) $$\langle u, v \rangle = \sum_{n=0}^{x} u_n v_n \quad \text{with} \quad x = N' - 1$$

Let j be the index of the best code book sequence then :

(4) $$j = \arg(\max PC(k))$$

The term β_k is the cross-correlation between p the vector of length N' of the original speech and $\hat{p}_j$ the synthetic speech. It can be written :

(5) $$\beta_k = \langle p, \hat{p}_k \rangle = \langle q, c^k \rangle$$

where q is a sequence calculated once per frame and does not depend on the code book sequence c^k .

We propose a new method to lessen the computation task of the calculation of the β_k term in case of ternary valued code book sequences. After having described the method, we apply it in case of FS-1016 [5].

CALCULATION OF β_k TERMS

We have to calculate the scalars products β_k between a real sequence **q** of size N' and T ternary code book sequences of size N'.

If c^k is one code book sequence then :

(6) $$\beta_k = \langle q, c^k \rangle = \sum_{n=0}^{N'-1} q_n c_n^k$$

The **q** sequence can be segmented in N'/ L parts of length L then the first segment is :

(7) $$sq_1 = [q_0 \, q_1 \cdots q_{L-1}]$$

and the jth is :

(8) $$sq_j = [q_{jL} \cdots q_{(j+1)L-1}]$$

and q is the concatenation of the sq_i 's.

(9) $$q = \left[sq_1 \; sq_2 \ldots sq_{\frac{N'}{L}} \right]$$

In the same way, each code word can be segmented in N'/L parts of size L. Let note sc_j^k the jth segment of code word c^k, then we have :

(10) $$c^k = \left[sc_1^k \, sc_2^k \ldots sc_{\frac{N'}{L}}^k \right]$$

The scalar product β_k (6) is thus given by :

(11) $$\beta_k = \sum_{n=1}^{N'/L} sq_n \, sc_n^k$$

If we consider now one particular segment, say sc_1^k (see (10)), knowing that the code book does only contains -1, 0, +1 values, it exists $M=3^L$ different segment configurations for sc_1^k.

Let note l_m (m = 1 ... M) the M different segment configurations. For example, for L = 2, we have : l_1 = [-1,-1] l_2 = [-1,0] , l_3 = [-1,1] , l_4 = [0,-1] , l_5 = [0,0] , l_6 = [0,1] , l_7 = [1, -1] , l_8 = [1,0] , l_9 = [1,1].

The idea is to pre-calculate the partial scalar products (see (3) with x = L) :

$$\langle sq_n, sc_n^k \rangle \quad \text{where} \quad sc_n^k = \{l_1, l_2, \ldots, l_M\}$$

for n = 1... N'/L

Then, for sq_1, we calculate M partial scalar products :

$$b_{1,m} = \langle sq_1, l_m \rangle \quad m = 1 \ldots M$$

For sq_2, we calculate M partial scalar products :

$$b_{2,m} = \langle sq_2, l_m \rangle \quad m = 1 \ldots M$$

and so on until $b_{N'/L,m}$.

The $b_{i,m}$ being calculated, the scalar product (6) for a code word c^k can be evaluated by adding the corresponding partial scalar products :

(12) $$\beta_k = b_{1,m(1)}^k + b_{2,m(2)}^k + \ldots + b_{N'/L,m(N'/L)}^k$$

CALCULATION OF THE PARTIAL SCALAR PRODUCTS.

Let's consider the calculation of the partial scalar products $b_{1,m}$ with m = 1... M for sq_1 (7). To simplify the notation, $b_{1,m}$ will be noted down b(m) in that section. We have to consider all the possible configurations for L sized sc_1 segments from l_1 = [-1 -1 -1 ... -1] to l_M =[1 1 1 ... 1], that is 3^L possible configurations.

Let set L = 3. For that particular case, we have 27 different configurations : [-1 -1 -1] , [-1 -1 0] , ..., [0 0 0] , [0 0 1] , [1 1 0] , [1 1 1].

As a first remark, we can notice that each configuration has a symmetric term except [0 0 0]. For example, the symmetric term of [1 1 1] is [-1 -1 -1] and :
$b(1) = -q_0 - q_1 - q_2 = -(q_0 + q_1 + q_2) = -b(27)$.

Considering that property, we just have to calculate $(3^L - 1)/2$ partial scalar products seeing that $\langle [000], sq_1 \rangle$ is not to be evaluated.

Then, for L= 3, the different configurations to consider are defined in table 1.

The first terms to evaluate are the terms that contains only 0 and one 1. Let i be the place of the 1 in the segment, we have $b(1) = q_2$, $b(3) = q_1$, $b(9) = q_0$. That can be written :

(13) $$b(3^i) = q_{L-1-i}$$

Since we have a similarity with the FFT butterfly, the calculation of the partial scalar products is easy and requires at most one operation for each term.

We have :

$b(2) = b(3) - b(1)$

$b(4) = b(3) + b(1)$

$b(8) = b(9) - b(1)$

$b(10) = b(9) + b(1)$

$b(7) = b(9) - b(2)$

$b(11) = b(9) + b(2)$

Thus, we have :

$$(14) \qquad \begin{cases} b(3^i) = q_{L-1-i} \\ b(3^i - j) = b(3^i) - b(j) \\ b(3^i + j) = b(3^i) + b(j) \end{cases}$$

Table 1

m \ i	0	1	2	b(m)
1	0	0	1	q_2
2	0	1	-1	$q_1 - q_2$
3	0	1	0	q_1
4	0	1	1	$q_1 + q_2$
5	1	-1	-1	$q_0 - q_1 - q_2$
6	1	-1	0	$q_0 - q_1$
7	1	-1	1	$q_0 - q_1 + q_2$
8	1	0	-1	$q_0 - q_2$
9	1	0	0	q_0
10	1	0	1	$q_0 + q_2$
11	1	1	-1	$q_0 + q_1 - q_2$
12	1	1	0	$q_0 + q_1$
13	1	1	1	$q_0 + q_1 -+ q_2$

The overall complexity to calculate the scalar product β_k (6) for k = 1... T is then given by :

- Calculation of the partial scalar products. For sq_1 , we saw that $(3^L - 1) / 2$ addition/ subtraction were needed. We have to calculate the partial scalar products for the other segments $sq_1, sq_2, \ldots, sq_{N'/L}$ then the complexity is $(N'/L)(3^L - 1) / 2$.
- The scalar product β_k is the sum of the N'/L partial scalar products see (12). For k = 1 ... T, we have (N'/L - 1) T additions/ subtraction.

Thus the overall complexity is :

$$(15) \qquad C'_\beta = \frac{N'}{L}\left(\frac{3^L - 1}{2} + T\right) - T$$

And the complexity for the calculation of the β_k^2 is :

$$(16) \qquad C'_{\beta^2} = \frac{N'}{L}\left(\frac{3^L - 1}{2} + T\right)$$

Numerical application : for N' = 60, we can have L = 2, 3, 4, 5, 6, 10, 12, 15, 30.

We give in table 2 the number of operations to calculate C'_{β^2} supposing that L = 1 is the complexity of the classical algorithm :

$$(17) \qquad C_{\beta^2} = (N' + 1)T = C'_{\beta^2}(1)$$

For L = 2, 3, ... we give the relative complexity to C_{β^2}.

Table 2

	T = 1024		T= 512	
L	$C'_{\beta^2(L)}$	$C'_{\beta^2(L)}/C'_{\beta^2(1)}$	$C'_{\beta^2(L)}$	$C'_{\beta^2(L)}/C'_{\beta^2(1)}$
1	62464	1	31232	1
2	30840	0.49	15480	0.49
3	20740	0.33	10500	0.34
4	15960	0.25	8280	0.26
5	13740	0.22	7596	0.24
6	13880	0.22	8760	0.28
10	183288	2.93	180216	5.77
12	1333720	21.35	1331160	42.62

APPLICATION TO THE CODER FEDERAL STANDARD 1016

The FS-1016 [5] uses two code books to model the excitation. One is an adaptive code book VQ of size T'= 256 to model long-term signal periodicity. The second is a fixed stochastic code book of size T = 512. The stochastic code book contains sparse, overlapped (shift by -2) and ternary valued samples

(+1,0, -1) of zero-mean, unit variance white gaussien sequences, centre clipped at 1.2 resulting approximately 77% sparsity (zero values).

The used code book is given in reference [6]. It contains 1082 values. Due to the sparsity, all the configurations for L > 2 are not represented and theoretical results given in table 2 can be reduced if we take into account the particular structure of this code book.

Let's call the weight the number of -1 and +1 in the segment. For example, for L=4, [-1 1 0 1] will be said to be of weight 3. For a segment of size L, we have $2^q C_L^q$ theoretical configurations of weight q. We give in table 3 the number of existing configurations for a certain weight q (up right) compared to the theory (down left). For example, for L=5, we have 80 theoretical configurations of weight 3 and 53 existing configurations in the code book.

Table 3 : number of configurations

L / q	1	2	3	4	5	6
0	1 1	1 1	1 1	1 1	1 1	1 1
1	2 2	4 4	6 6	8 8	10 10	12 12
2	▨	4 4	12 12	24 24	40 40	56 60
3	▨	▨	6 8	14 32	53 80	50 160
4	▨	▨	▨	1 16	11 80	12 240
5	▨	▨	▨	▨	0 32	2 192
6	▨	▨	▨	▨	▨	0 64

For a typical application, we can make the choice to pre-calculate all the theoretical configuration for a given weight q using (14) and to adapt the algorithm for the weight where just few configurations are represented compared to the theory.

For example, for L = 3, (14) can be used for each weight, for L=4, (14) can be used up to q = 3 and the last configuration is calculated apart, etc.

Table 4 gives the complexity obtained by taking account of the particularity of the code book from reference [6].

Table 4 : complexity for code book from [6].

	T = 1024		T= 512	
L	$C'_{\beta^2(L)}$	$C'_{\beta^2(L)}/C'_{\beta^2(1)}$	$C'_{\beta^2(L)}$	$C'_{\beta^2(L)}/C'_{\beta^2(1)}$
4	15855	0.25	8175	0.26
5	13548	0.22	7404	0.24
6	11840	0.19	6720	0.21

CONCLUSION

For ternary valued samples code books, the cross-correlation term (5) can be calculated in two stages. In a first step, we pre-calculate partial scalar products and in a second step we calculate the cross-correlation term by adding the appropriated partial scalar products.

The complexity of this algorithm is evaluated for the general case and for the case of the code book of Federal Standard 1016. Typically, by using partial scalar products of size L= 6, for sub frame of size 60, the reduction of calculation complexity is 5.

REFERENCES

[1] Schroeder M.R., Atal B.S. "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates ". Proc. ICASSP, vol. 3,pp. 937-940 1985

[2] Kleijn W.B., Krasinski D.J., Ketchum R.H. "Fast Methods For The CELP Speech Coding Algorithm ". IEEE Trans. on Acous.,Sp.,and Sig. Proc,Vol. 38,No. 8,pp. 1330-1342,August 1990

[3] Gerson I.A., Jasiuk M.A. "Vector Sum Excited Linear Prediction (VSELP) Speech Coding At 8Kbps ". IEEE-ICASSP,pp. 461-464, 1990

[4] Trancoso I.M., Atal B.S. "Efficient Procedures For Finding The Optimum Innovation In Stochastic coders ". ICASSP,tokyo,pp.2375-2378, 1986

[5] Campbell Jr. J.P., Tremain T.E., Welch V.C. "The Federal Standard 1016 4800 Bps CELP Voice Coder ". Digital Processing I,pp. 145-155, 1991

[6] Fenichel R., Bodson D. "Details to Assist in Implementation of Federal Standard 1016 CELP ". Technical Information Bulletin 92-1,National Communication system 1992