Efficient VQ Techniques and General Noise Shaping in Noise Feedback Coding

Jes Thyssen and Juin-Hwey Chen

Broadcom Corporation, Irvine, California, USA
jthyssen@broadcom.com, rchen@broadcom.com

Abstract

This paper describes new efficient Vector Quantization (VQ) techniques that enable low complexity implementations of VQ-based Noise Feedback Coding (NFC). These methods offer mathematical equivalence to higher complexity methods. Furthermore, the paper presents efficient codec structures for general noise shaping as used in BroadVoice®16 – a new SCTE® (Society of Cable Telecommunications Engineers) and PacketCable™ speech coding standard for Cable Telephony.
Index Terms: speech coding, noise shaping, efficient VQ.

1. Introduction

In order for speech coding paradigms to find successful applications practical issues such as computational complexity must be addressed. For the Code Excited Linear Prediction (CELP) coding paradigm [1] efficient techniques for excitation quantization have been developed since its initial conception. Examples of efficient CELP methods include the vector excitation coding techniques in [2], and the algebraic codebook techniques in ACELP [3]. Both of these approaches have found their way into many speech coding standards, e.g. [4].

Efficient techniques can generally be divided into four categories: Bit-exact techniques; mathematically equivalent techniques; perceptually equivalent techniques, and techniques offering increased efficiency at a measured degradation. This paper focuses on mathematically equivalent techniques for VQ in NFC [5] that enable implementations at much lower complexity compared to [6] and [7]. The mathematical equivalence guarantees the speech quality to also be equivalent.

The paper is organized as follows. Section 2 provides a brief introduction to VQ in NFC. Section 3 presents the efficient VQ techniques and general noise shaping followed by practical numbers in Section 4. Section 5 concludes the paper.

2. VQ in noise feedback coding

Noise feedback coding [8], [9] typically operates on a sample-by-sample basis. However, recently, in [6], VQ was proposed to improve the coding efficiency for the NFC prediction residual (excitation), and techniques utilizing the superposition principle were proposed to reduce the complexity of the corresponding VQ codebook search.

Noise feedback coding can be implemented with two fundamental structures as described in [8] and [9], and in [6] in the context of VQ – shown in Figure 1 and Figure 2. The first fundamental structure of NFC with VQ is shown in Figure 1. The resulting spectral shape of the coding noise, $e(n) = s_q(n) - s(n)$, from quantizing $u(n)$ with $u_q(n)$ by choosing the codebook entry that minimizes the mean squared error (MSE), i.e. the energy of $q(n)$, is given by (1).

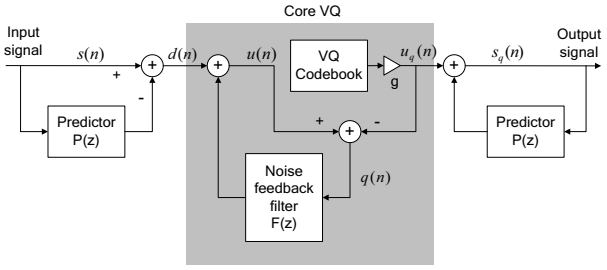

Figure 1 *First fundamental structure of NFC with VQ.*

$$W(z) = \frac{1 - F(z)}{1 - P(z)} \tag{1}$$

If $P(z)$ is the typical short-term predictor of LPC,

$$P(z) = \sum_{i=1}^{J} \alpha_i z^{-i} \ . \tag{2}$$

and the noise feedback filter is defined as $F(z) = P(z/\gamma)$, $0 < \gamma < 1$, then noise shaping similar to CELP is achieved [1].

The second fundamental structure of noise feedback coding utilizing VQ is shown in Figure 2.

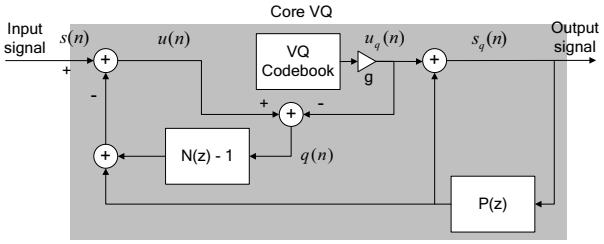

Figure 2 *Second fundamental structure of NFC with VQ.*

The resulting spectral shape of the coding noise by minimizing the MSE between $u(n)$ and $u_q(n)$ during VQ is given by $N(z)$ [8], [6]. Hence, the spectral shape of the coding noise can be controlled directly by the feedback filter $N(z) - 1$. Choosing $N(z) = W(z)$ results in a noise shaping equivalent to the first fundamental structure.

The shaded area in Figure 1 and Figure 2 represents the core VQ, which comprises the VQ codebook search loop. It is obvious from studying and comparing Figure 1 and Figure 2 that the increased flexibility in controlling the noise shape with the structure of Figure 2 comes with a higher complexity of the core VQ. The cost function for the VQ is the power of the error signal $q(n)$. Hence, the codebook entry providing the minimum MSE between $u(n)$ and $u_q(n)$ is selected. Conceptually, every codevector is passed through the filter structure and the one minimizing the energy of the error signal $q(n)$ is selected.

10.21437/Interspeech.2006-62

However, as described in [6], it is advantageous to calculate the error signal as the superposition of two contributions, the Zero State Response (ZSR), $q_{ZSR}(n)$, and the Zero Input Response (ZIR), $q_{ZIR}(n)$:

$$q(n) = q_{ZSR}(n) + q_{ZIR}(n). \quad (3)$$

The ZIR excludes the contribution of the codevector but includes the contributions of the input signal and filter memories to $q(n)$. The ZSR excludes the contributions of the filter memories and the input signal but includes the contribution of the codevector to $q(n)$.

3. Efficient VQ techniques

The VQ method in [6] adds the ZSR and ZIR contribution for every codevector and calculates the MSE of the VQ as

$$E_k = \sum_{n=0}^{N-1} q^2(k,n) = \sum_{n=0}^{N-1} \left(q_{ZSR}(k,n) + q_{ZIR}(n)\right)^2, \quad (4)$$

where k denotes the codevector, and N is the vector size. Typically, the vector size, N, is smaller than a frame length, L, for which the filters of the structures are invariant [6]. A practical example has $L = 40$ samples and $N = 4$ samples [7]. Hence, there would be $M = L/N = 10$ consecutive input vectors for which the ZSRs of the codevectors would remain the same. With a codebook of K codevectors it would require $2K \cdot N \cdot M$ adds/macs/multiplies to calculate all error terms for a frame. This does not include the computations required to derive the ZIRs and ZSRs. All example frame and vector sizes in the following apply to the BroadVoice16 standard [7].

3.1. The VQ error term

It is important to recognize that the ZIR for a VQ is independent of the codevectors, and that the ZSRs of the codevectors are invariant throughout a frame. With this in mind, the error term of (4) is expanded as follows

$$\begin{aligned} E_k &= \sum_{n=0}^{N-1} \left[q_{ZSR}^2(k,n) + q_{ZIR}^2(n) + 2 \cdot q_{ZSR}(k,n) \cdot q_{ZIR}(n)\right] \\ &= E_{ZSR}(k) + E_{ZIR} + R\left(q_{ZSR}(k,n), q_{ZIR}(n)\right) \end{aligned} \quad (5)$$

where

$$E_{ZIR} = \sum_{n=0}^{N-1} q_{ZIR}^2(n), \quad (6)$$

$$E_{ZSR}(k) = \sum_{n=0}^{N-1} q_{ZSR}^2(k,n), \text{ and} \quad (7)$$

$$R\left(q_{ZSR}(k,n), q_{ZIR}(n)\right) = 2\sum_{n=0}^{N-1} q_{ZSR}(k,n) \cdot q_{ZIR}(n). \quad (8)$$

Since the ZIR is independent of the codevectors, minimizing E_k of (5) is equivalent to minimizing

$$\widetilde{E}_k = E_{ZSR}(k) + R\left(q_{ZSR}(k,n), q_{ZIR}(n)\right). \quad (9)$$

From (9) it can be seen that the energy of the codevectors only needs to be calculated once as the ZSRs are invariant for the M vectors in a frame. Hence, the only "search loop" calculation is the cross-correlation between the ZIR and ZSRs. Consequently, the complexity would be $K \cdot (N \cdot M + N + M)$. Reusing the example numbers from above, with a codebook size of $K = 32$, this technique would result in 1728 compared to 2560 ($2K \cdot N \cdot M$ - see Section 3). There is an equal overhead for both for calculation of per-frame ZSRs and per-vector ZIRs.

3.2. A sign-shape structured codebook

Exploiting a structured codebook offers further reductions. A sign-shape codebook can be suitable [7]. Inspecting the error term of (9), it is evident that a signed codebook will result in pairs of error terms of the form

$$\widetilde{E}_k = E_{ZSR}(k) \overset{s}{\pm} R\left(q_{ZSR}(k,n), q_{ZIR}(n)\right), \quad (10)$$

where $k = 0.,1...,K/2-1$ and s identifies the sign of the codevector. Note that a signed codevector translates to a signed ZSR as the ZSR is a linear filtering of the codevector. Utilizing the sign-shape structured codebook, the complexity would be $K/2 \cdot (N \cdot M + N + M)$, corresponding to 864 operations for the example numbers above. It is sufficient to only examine the error term corresponding to one of the signs for each of the $K/2$ shape codevectors as the cross-correlation term of (8) automatically identifies the optimal sign for a given codevector. If the cross-correlation term is negative, then from (10), it is evident that the positive sign is optimal and vice-versa. It should be noted that a sign-shape codebook additionally offers savings in calculating the ZSRs as only half need to be calculated. The other half is given by simple negation.

3.3. VQ in TSNFC with pole-zero noise feedback filter

While the former two techniques apply to both structures the next technique addresses the structure of Figure 2 specifically. The benefit of this structure is the ability to directly specify the desired spectral noise shape. A transfer function of

$$N(z) = \frac{1 - P(z/\gamma_1)}{1 - P(z/\gamma_2)}, \quad 0 < \gamma_1 < \gamma_2 < 1, \quad (11)$$

is proposed. It offers a more general noise shaping, comparable to the traditional perceptual weighting of CELP coders [4]. It represents the short-term noise feedback filter adopted in [7]. The techniques are presented in the context of Two-Stage Noise Feedback Coding (TSNFC) [6] which includes both short-term and long-term prediction and noise shaping. However, it applies equally to the regular NFC structure of Figure 2.

Figure 3 shows the TSNFC structure with a pole-zero short-term noise feedback filter (NFF) $F_{sz}(z)/F_{sp}(z)$, short-term prediction $P_s(z)$, long-term noise feedback filter $N_l(z)-1$, and long-term prediction $P_l(z)$.

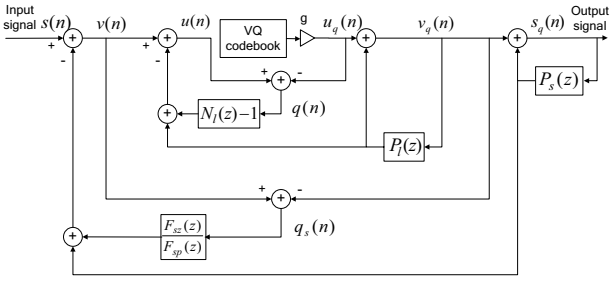

Figure 3 *TSNFC with pole-zero noise feedback filter*.

According to (11) and (2), and Figure 2 and Figure 3, the pole-zero short-term noise feedback filter is given by

$$\begin{aligned} N(z)-1 &= \frac{1-P(z/\gamma_1)}{1-P(z/\gamma_2)}-1 \\ &= \frac{P(z/\gamma_2)-P(z/\gamma_1)}{1-P(z/\gamma_2)} \\ &= \frac{\sum_{i=1}^{J_{NFF}} \alpha_i \cdot \left(\gamma_2{}^i - \gamma_1{}^i\right) \cdot z^{-i}}{1-\sum_{i=1}^{J_{NFF}} \alpha_i \cdot \gamma_2{}^i \cdot z^{-i}} \end{aligned} \quad (12)$$

where J_{NFF} is the order of the pole and zero sections of the noise feedback filter. Hence, the pole and zero sections of the noise feedback filter in Figure 3 are given by

$$F_{sp}(z) = 1-\sum_{i=1}^{J_{NFF}} \alpha_i \cdot \gamma_2{}^i \cdot z^{-i} \quad \text{and} \quad (13)$$

$$F_{sz}(z) = \sum_{i=1}^{J_{NFF}} \alpha_i \cdot \left(\gamma_2{}^i - \gamma_1{}^i\right) \cdot z^{-i}, \quad (14)$$

respectively. For $M>1$, efficient VQ would comprise:

1. ZSR calculation of codevectors
2. Loop over the M vectors in a frame:
   A. ZIR calculation
   B. Codebook search
   C. Filter memory update

3.3.1. ZSR calculation of codevectors

If the smallest lag of the long-term (pitch) predictor and the long-term noise feedback filter is greater than the maximum of the order of the zero section and pole section of the noise feedback filter and the order of the short-term predictor, then the ZSR calculation for the structure in Figure 3 reduces to the structure in Figure 4.

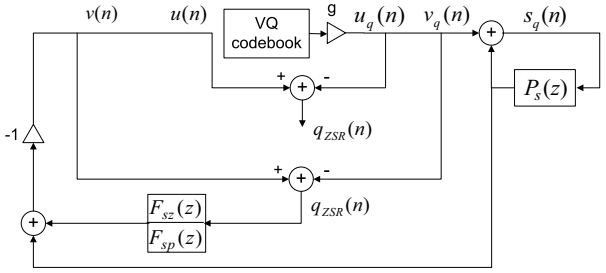

Figure 4 *Structure for ZSR calculation.*

The structure in Figure 4 can be further reduced to the structure in Figure 5 where $H(z)$ is given by (15), in which $\alpha_{s,i}$, $i=0,1,...,J-1$, are the short-term predictor coefficients of $P_s(z)$.

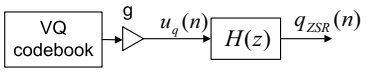

Figure 5 *Compact structure for ZSR calculation.*

Typically, the short-term predictor would use quantized short-term predictor coefficients while the noise feedback filter would use unquantized coefficients for more accurate noise shaping. The order of the pole and zero sections of the noise feedback filter need not necessarily be the same as the order of the short-term predictor, but it could be for practical reasons.

$$\begin{aligned} H(z) &= \frac{Q_{ZSR}(z)}{U_Q(z)} = -\frac{1}{N(z)\cdot\left(1-P_s(z)\right)} \\ &= -\frac{1-P(z/\gamma_2)}{\left(1-P(z/\gamma_1)\right)\cdot\left(1-P_s(z)\right)} \\ &= -\frac{1-\sum_{i=1}^{J_{NFF}} \alpha_i \cdot \gamma_2{}^i \cdot z^{-i}}{\left(1-\sum_{i=1}^{J_{NFF}} \alpha_i \cdot \gamma_1{}^i \cdot z^{-i}\right)\cdot\left(1-\sum_{i=1}^{J} \alpha_{s,i} \cdot z^{-i}\right)} \end{aligned} \quad (15)$$

The codevector dimension N determines the number of coefficients of $H(z)$ that must be calculated. Typically $N<J$ and $N<J_{NFF}$. Consistent with previous examples typical numbers could be $N=4$, $J_{NFF}=J=8$ [7]. Once $h(j)$, $j=0,1,...,N-1$, is calculated by passing an impulse through the filter given by (15), the ZSRs of all codevectors are calculated according to

$$q_{ZSR}(k,n) = h(n) * u_q(k,n), \quad (16)$$

for $k=0,1,...,K-1$, $n=0,1,...,N-1$. Equivalent to Section 3.2, a sign-shape codebook can be utilized to reduce the calculations to half, i.e. only for $k=0,1,...,K/2-1$.

3.3.2. ZIR calculation

The ZIR calculation for the structure in Figure 3 reduces to the structure shown in Figure 6. The ZIR calculation is performed prior to each of the M codebook searches.

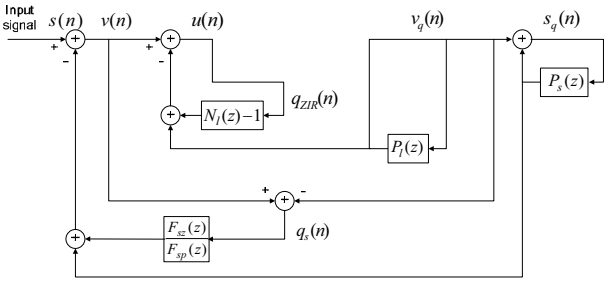

Figure 6 *Structure for ZIR calculation.*

3.3.3. Codebook search

Based on the ZSRs of the codevectors and the ZIR, each of the M codebook searches is carried out by selecting the corresponding codevector that minimizes

$$E_k = \sum_{n=0}^{N-1} q(k,n)^2 = \sum_{n=0}^{N-1} \left(q_{ZSR}(k,n)+q_{ZIR}(n)\right)^2. \quad (17)$$

This codevector is denoted $u_q(n)$. Note that the efficient techniques presented in sections 3.1 and 3.2 can be used to implement the codebook search efficiently.

3.3.4. Filter memory update

Updating the filter memories of the structure in Figure 3 involves updating the memory of:

- the short-term predictor, $p_s(n)$,
- the long-term predictor, $p_l(n)$,
- the long-term noise feedback filter, $n_l(n)$,

- the zero section of the short-term NFF, $f_{sz}(n)$,
- the pole section of the short-term NFF, $f_{sp}(n)$.

An alternative and more efficient method compared to straightforward filtering according to the structure of Figure 3 with the selected codevector as $u_q(n)$, is to calculate the five filter memory updates as the superposition of the contributions from the ZSR and ZIR components. The contributions from the ZSR component to the five filter memories are denoted $p_{s,ZSR}(n)$, $p_{l,ZSR}(n)$, $n_{l,ZSR}(n)$, $f_{sz,ZSR}(n)$, and $f_{sp,ZSR}(n)$, respectively, and the contributions from the ZIR component are denoted $p_{s,ZIR}(n)$, $p_{l,ZIR}(n)$, $n_{l,ZIR}(n)$, $f_{sz,ZIR}(n)$, and $f_{sp,ZIR}(n)$, respectively. Conceptually, the structure to calculate the contributions to the five filter memories from the ZSR component is identical to the structure in Figure 4. It reduces to the structure depicted in Figure 7.

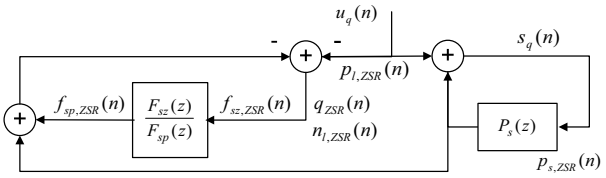

Figure 7 *ZSR contribution to filter memories*.

It can be seen that

$$p_{l,ZSR}(n) = u_q(n), \tag{18}$$

$$n_{l,ZSR}(n) = q_{ZSR}(n), \text{ and} \tag{19}$$

$$f_{sz,ZSR}(n) = q_{ZSR}(n), \tag{20}$$

which are all available from the ZSR calculation corresponding to the selected codevector $u_q(n)$, and hence require no calculations. The filter memory update for the short-term predictor, $p_{s,ZSR}(n)$, must be calculated. From Figure 7 it is evident that this calculation is independent of any of the other filter memories. Furthermore, it can be shown that the ZSR contribution to the pole-section of the short-term noise feedback filter can be expressed as

$$f_{sp,ZSR}(n) = -q_{ZSR}(n) - p_{s,ZSR}(n). \tag{21}$$

Referring to Figure 6 it is evident that the ZIR contributions to the five filter memories are all available from the ZIR calculation prior to the codebook search, and consequently, no additional calculations are necessary. From the ZSR and ZIR contributions, the final updates of the filter memories are calculated as

$$p_s(n) = p_{s,ZSR}(n) + p_{s,ZIR}(n), \tag{22}$$

$$p_l(n) = p_{l,ZSR}(n) + p_{l,ZIR}(n), \tag{23}$$

$$n_l(n) = n_{l,ZSR}(n) + n_{l,ZIR}(n), \tag{24}$$

$$f_{sz}(n) = f_{sz,ZSR}(n) + f_{sz,ZIR}(n), \tag{25}$$

$$f_{sp}(n) = f_{sp,ZSR}(n) + f_{sp,ZIR}(n). \tag{26}$$

Obviously, this offers a much more efficient method to update the filter memories compared to simply filtering the selected codevector through the structure of Figure 3.

4. Practical numbers

Since the presented techniques addressing complexity are mathematical equivalent they provide equivalent speech quality. Section 3 contains equations and example numbers of the complexity savings. Table 1 below compares the number of adds/macs/multiplies per 5 ms for the methods in [6] that are applicable via extension or directly comparable. The numbers represent the TSNFC configuration with short-term pole-zero noise feedback filter of BroadVoice16 [7].

Table 1. *Complexity comparison*.

	Codebook search	Filter memory update
According to [6]	$2K \cdot N \cdot M$ =2560	$(3J+11) \cdot N \cdot M$ =1400
Proposed	$K/2 \cdot (N \cdot M + N + M)$ =864	$(7N + N/2 \cdot (N-1)) \cdot M$ =340

5. Conclusion

The paper presented methods to implement VQ in NFC and TSNFC in a very efficient manner. It also presented the method of the PacketCable and SCTE BroadVoice16 standard to achieve noise shaping similar to the typical perceptual weighting in CELP coders, including associated efficient methods. The complexity savings were demonstrated using the PacketCable and SCTE BroadVoice16 codec parameters. All methods are part of Broadcom's efficient implementation of TSNFC in BroadVoice16 – a new speech coding standard for Voice over Cable in North America.

6. References

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

[2] G. Davidson and A. Gersho "Complexity Reduction Methods for Vector Excitation Coding," Proc. ICASSP, 1986, pp. 3055-3058.

[3] C. Laflamme, J-P. Adoul, H.Y. Su, and S. Morissette, "On Reducing Computational Complexity of Codebook Search in CELP Coder Through the Use of Algebraic Codes," Proc. ICASSP, 1990, pp. 177-180.

[4] R. Salami, C. Laflamme, J-P. Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, "Design and Description of CS-ACELP: A Toll Quality 8 kb/s Speech Coder," IEEE Trans. Speech and Audio Processing, Vol. 6, No. 2, March 1998, pp. 116-130.

[5] J. Thyssen and J.-H. Chen, US Patent No. 6,751,587, "Efficient Excitation Quantization in Noise Feedback Coding with General Noise Shaping," June 2004, and other issued patents/pending patent applications.

[6] J.-H. Chen, "Novel Codec Structures for Noise Feedback Coding of Speech," Proc. ICASSP, 2006.

[7] J.-H. Chen, BroadVoice®16 Speech Codec Specification, Version 1.2, October 2003. (For further information, contact PacketCable, Cable Television Laboratories, Inc.)

[8] J. D. Makhoul and M. Berouti, "Adaptive Noise Spectral Shaping and Entropy Coding in Predictive Coding of Speech," IEEE Trans. ASSP, February 1979, pp. 63-73.

[9] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," IEEE Trans. ASSP, June 1979, pp. 247-254.