CELP Coding - MCELP on the MSP58C80

Literature Number: BPRA070
Texas Instruments Europe
December 1997

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Figures

1. Introduction

One of the software functions which are implemented on the MSP58C8X Digital Telephone Answering Device (DTAD) DSP is the MCELP speech coder (Modified Code Excited Linear Prediction). The goal of this algorithm is to provide high quality speech for answering machine applications at a 4.8 Kbps rate while using the internal resources (ROM / RAM /MIPS) of an MSP58C8X type of device and allowing for concurrent operation of the other s/w modules required for DTAD applications. This module is the most time- and memory-demanding of all the s/w routines and it is directly responsible for the speech quality. A high compression factor is applied to be able to record 15 minutes of speech in a 4 Mbit memory.

This document explains the fundamental ideas around CELP coding and gives detailed information on the MCELP encoder implementation. All information concerning this implementation is given for MSP58C8x DTAD system application purposes and is strictly confidential.

NOTICE: MCELP has been developed by DSPSE for TI. TI has the rights to use this s/w on DTAD MSP58C8x products and TI customers have access to MCELP library object files. MCELP cannot be used on any other TI processor without agreement from DSPSE. NO customer can have access to source code.

2. Main ideas about CELP coding

The speech signal has several characteristics that allow for a data compression via speech encoding algorithms.

There are two basic approaches for speech encoding. One is in the time domain, the other is in the frequency domain. The time domain approach is essentially based on the fact that the contiguous speech samples are correlated, which means that we can estimate the amplitude of the future sample based on the value of the present sample. We can hence use differential coding to employ all our available precision to store only the difference between these samples. This approach allows to bring the bitrate down by a 0.5 factor and yet maintaining a good quality. Unfortunately it is not possible to use this approach at lower bit rates and still keep a high speech quality. Indeed starting from a 64 Kbits per second - bps - signal (8 bits per sample at an 8KHz sampling rate), going down to 16 Kbps implies coding each incoming sample with only two bits.

The second approach speech encoding (frequency domain) analyzes speech by frames composed of about 10-30 ms of signal. Here it is assumed that speech is quasi-stationary during this period of time and in this case, speech production can be modeled as a process in which a filter is excited by a source. In linear predictive coding, the type of excitation changes as a function of our vocal folds (hence relatively slowly) and the filter characteristics) change as a function of our vocal tract configuration (hence relatively fast). At the output of the filter we have the speech samples. The speech encoding algorithm mission is to determine the characteristics of the excitation function and to compute a set of filter coefficients that will adequately model the vocal tract.

A natural way to exploit this model is to distinguish two parts of the prediction:

- A long term prediction which corresponds to the excitation periodicity
- A short term prediction which corresponds to the vocal tract filter and the overall spectral shape

Closed loop “analysis by synthesis” systems work in such a way that the parameters that are derived during the analysis are directly used at this stage to generate a synthetic signal. A comparison is then operated between the original and the synthetic signals. A whole set of parameters are tested and the one which yields the best results is the one that is retained by the encoder for storage or for transmitted to the decoder.

Even if this technique allows to select the best set of parameters, there is still a residual difference between the original and the synthetic signal. In order to improve the likelihood, this residual could also be quantized and transmitted to the synthesizer (decoder). However this operation is also bit consuming : If one bit was used per each residual sample, we would already need to add 8 Kbps to our compression rate, when sampling at 8KHz. One way to address this problem is to group the residual values into vectors an to quantify them jointly. Not necessarily by frames, but by subframes.

CELP coding approach consists on computing a unique set of filter coefficients per frame and using vector quantized codebooks to represent the excitation that allows to best model the original speech. Information to the CELP decoder is then the filter coefficients and the codebook indices and gains.

The drawing below shows the way how the speech is synthezised. It consists of three basic components:

1- the adaptive codebook which deals with the long term prediction,

2- the innovation or stochastic codebook which handles the non predictable innovation,

3- the synthesis filter $\frac{1}{A(z)}$ which exploits the short term prediction

The duty of the encoder is to find out:

- which sequences from the adaptive and the stochastic codebook should be chosen
- which gain factors should be used for these sequences
- the best synthesis filter coefficients.

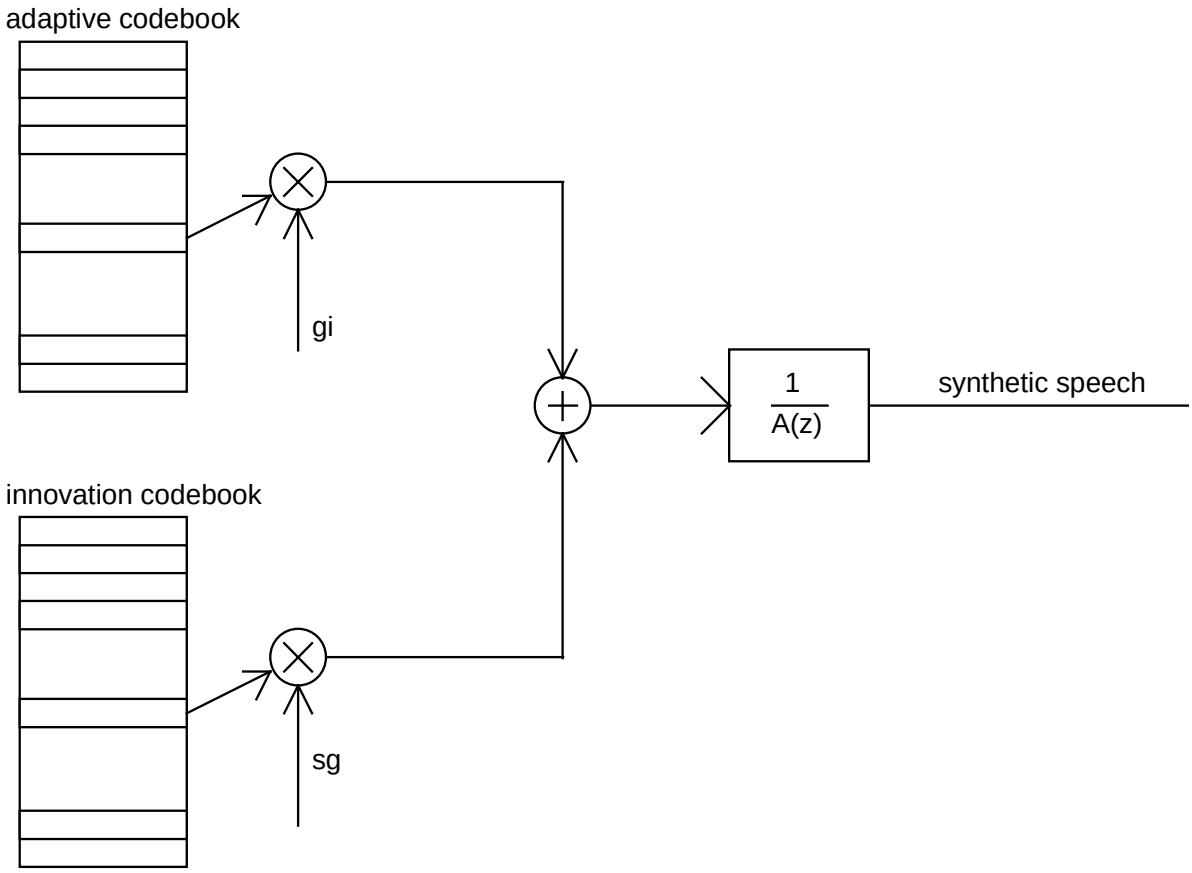


Figure 1: CELP Speech Decoding

3. The basic components of CELP coders

3.1 Short term prediction

To describe the air propagation in the vocal tract we can establish equations for the pressure and the flux which are very similar to the equations of electronic circuits if we replace pressure by tension and flux by current. Much like in the electronic case we will have resonant and anti-resonant frequencies. In the speech case these frequencies are called formants and anti-formants. Since the energy of the excitation is concentrated in the lower frequency range (-12dB/octave average slope), energy at higher formats is low and only the first 3-5 formants are important. Knowing that for each resonance we need two filter coefficients (IIR filter), we can see that a filter length of 10 coefficients will be sufficient. These coefficients represent a configuration of the vocal tract which varies slowly (compared with the sampling frequency) in time. This means that every 20..30ms (NF=160..240 samples) we update the filter. This interval is often called a frame and NF is the number of samples in a frame.

Between two frames an interpolation of the coefficients may be utilized, but we have to avoid intermediate unstable filter states. In most cases and in particular in this telephone answering machine application a delay of about 100ms is still tolerable. In this case we can determine the filter coefficients in a non causal manner: First we wait for the end of a frame, then we determine the best prediction coefficients for this frame. This will best match at about the middle of the current frame, so that at the end

of a frame we code the values from the middle of the last frame to the middle of the current frame since the interpolation can not be done before knowing the coefficients for this frame.

There are many forms to represent an IIR filter. In the last years two representations have been used in most cases : Reflection coefficients and line spectral pairs (LSP). These values can be scalar (34..38bits) or vector quantized (24..28bits) and either of them are used.

To find the best prediction coefficients according to a least squares error criterion, we want to minimize:

$$\sum_{n\in frame} e(n)^2 = \sum_{n\in frame} (s(n) - \hat{s}(n))^2 \qquad \text{Equ. 1}$$

where the prediction is

$$\hat{s}(n) = -\sum_{i=1}^{NP} a_i s(n-i) \qquad \text{Equ. 2}$$

NP designs the number of preceding values which are used for the estimation. The negative sign is only chosen for simplicity, so that the error signal or short term linear prediction residual e(n) can be written as:

$$e(n) = \sum_{i=0}^{NP} a_i s(n-i) \text{ with } a_0 = 1 \qquad \text{Equ. 3}$$

If we derive the sum of squared errors by the coefficients, we obtain a set of linear equations, called Yule-Walker equations.

$$\frac{\delta}{\delta a_j}\Big(\sum_{n\in frame} e(n)^2\Big) = \sum_{n\in frame} 2e(n)\frac{\delta e(n)}{\delta a_j} = 2\sum_{n\in frame}\Big(\sum_{i=0}^{NP} a_i s(n-i)\Big)s(n-j)$$

Equ. 4

$$= 2\sum_{i=0}^{NP} a_i \Big(\sum_{n\in frame} s(n-i)s(n-j)\Big) = 2\sum_{i=0}^{NP} a_i r_{i,j} = 0 \qquad \forall j \in [1, NP]$$

We obtain a set of NP linear equations with NP unknown variables $(a_1..a_{NP})$. We can also use a matrix notation :

$$\begin{pmatrix} r_{0,1} & r_{1,1} & r_{2,1} & \cdots & r_{NP,1} \\ r_{0,2} & r_{1,2} & r_{2,2} & \cdots & r_{NP,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{0,NP} & r_{1,NP} & r_{2,NP} & \cdots & r_{NP,NP} \end{pmatrix} * \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_{NP} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \text{Equ. 5}$$

$$\begin{pmatrix} r_{1,1} & r_{2,1} & \cdots & r_{NP,1} \\ r_{1,2} & r_{2,2} & \cdots & r_{NP,2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,NP} & r_{2,NP} & \cdots & r_{NP,NP} \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{NP} \end{pmatrix} = \begin{pmatrix} -r_{0,1} \\ -r_{0,2} \\ \vdots \\ -r_{0,NP} \end{pmatrix} \qquad \text{Equ. 6}$$

with the estimated autocorrelation values (without normalization by the frame length) as matrix coefficients.

$$r_{i,j} = \sum_{n \in frame} s(n-i)s(n-j) \qquad \text{Equ. 7}$$

Unfortunately this straightforward method (called the covariance method) doesn't guarantee that the filter is stable. In addition the values on the diagonals are not exactly the same, i.e. the matrix is not Toeplitz. Therefore in most cases a similar method is used, which is called the autocorrelation method. This method does not take into account values outside the frame but assumes that they are zero. To prevent that the first values are estimated from the zeroes before the current frame and that the last values serve to predict the zeroes after the current frame, a window is used to lessen the border effects. The autocorrelation method assures that the filter is stable and that the matrix is symmetric and Toeplitz, so that a fast algorithm like Levinson-Durbin [Papa 87] can be used.

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_{NP-1} \\ r_1 & r_0 & \cdots & r_{NP-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{NP-1} & r_{NP-2} & \cdots & r_0 \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{NP} \end{pmatrix} = \begin{pmatrix} -r_0 \\ -r_1 \\ \vdots \\ -r_{NP-1} \end{pmatrix} \qquad \text{Equ. 8}$$

with

$$r_i = \sum_{n=i}^{NF-1} \tilde{s}(n) * \tilde{s}(n-i) \qquad \tilde{s}(n) = s(n) * h(n) \qquad \text{Equ. 9}$$

NF is the number of samples in a frame and h(n) is the window, e.g. a Hamming window is defined by

$$h(n) = 0.54 - 0.46 * \cos\left(\frac{2\pi n}{NF-1}\right) \qquad \text{Equ. 10}$$

Several scientists [Mark 76,Gibs 80] have found out that for speech compression the autocorrelation method's prediction is very close to the covariance method's prediction while offering the mentioned advantages. We can use the correlation coefficients r_i to determine the prediction coefficients a_i and finally another representation which is more bit-efficient, e.g. the reflection coefficients. Another possibility is to calculate the reflection coefficients directly from the correlation coefficients (e.g. Le Roux-Gueguen algorithm [Roux 77,Papa 87]) and then the prediction coefficients from the reflection coefficients. This second method is often applied for fixed point calculations since the

prediction coefficients are less robust to rounding errors and since the decoder only knows the quantized transmitted coefficients so that he has to calculate the direct form prediction coefficients from these values and we want that the coder and the decoder use exactly the same coefficients and this would not be the case if the coder uses the non quantzied prediction coefficients.

Now we could pass the speech signal through the interpolated short term prediction filter

$$A_{\text{int}}(z) = \alpha A_{new}(z) + (1-\alpha)A_{old}(z) \qquad \alpha \in [0,1] \qquad \text{Equ. 11}$$

with

$$A(z) = \sum_{i=0}^{NP} a_i z^{-i} \qquad \text{Equ. 12}$$

"New" and "old" refer to the current and the previous frame respectively. We could then code the short term residual and the decoder could pass the decoded short term residual through the inverse filter. Unfortunately even if the inverse of the new and the old A are guaranteed to be stable, the inverse of the interpolated filter can be very unstable. Indeed experimental work shos that if we limit the absolute value of the poles of A_{old} and A_{new} in the complex z plane by 0.8, the most unstable case seems to be the transition between
$(z-0.8)^{10}$ to $(z+0.8)^{10}$ for which we obtain for α=0.5 a complex pole pair with an absolute value of more than 5 ! Therefore the stability check is a must and we have to do without interpolation in the unstable case. This must be done by both the coder and the decoder. No additional information must be transmitted. The GSM half-rate coder [ETSI 95] also checks if the filter performs better with interpolation than without and sends one bit to indicate this. In this case the decoder doesn't need to check stability.

The efficiency of the short time linear prediction varies from very poor prediction for unvoiced frames to very good prediction for vowels. This technique has been used by the famous LPC10 and a lot of other vocoders which preserve intelligible but often artificial sounding speech at very low bit rates (2400bit/s and less). An important improvement of speech quality has been made by M. Schroeder and Bishnu S. Atal in 1985 [Schr 85] when they proposed to use a combination of closed loop pitch lag and codebook excitation search. Closed loop means that the coder searches the lag or the excitation sequence by testing if it is the best match and not by deduction. This is also called "Analysis by synthesis". This means that the coder minimizes a distortion function depending on the synthetic speech and not on the residual.

The first CELP coders in 1985 had a very high complexity. Even on a Cray 1 the coding took about 125 times real [Schr 85] time so that initially this coder has only served as a proof that it is possible to maintain high quality at data rates below 16kbit/s. One decade later several algorithms have been found to reduce complexity and dedicated signal processors have become faster and cheaper allowing for consumer market applications such as DTAD.

3.2 Long term prediction

We can distinguish between voiced and unvoiced phonemes. Voiced phonemes have a pseudo periodic excitation. Assuming the vocal tract can be modeled by a filter then the speech also is nearly periodic. This is the case for vowels. The voiced consonants also contain a noisy component due to the constrictions in the vocal tract. If the constriction is very tight or total then the noise component will be much more powerful than the harmonic part. In the case of sustained sounds and during the stable part of vowels, we can be optimistic and expect that a long term prediction - i.e. a prediction based on the precedent period - will be very efficient. Since the number of samples between two periods (pitch lag) is smaller than the frame length (20..140 compared with 160..240) and since the pitch lag may change faster than the vocal tract, we have to divide a frame into subframes and to code the pitch information for each subframe. Most coders use 4 subframes so that the subframe length NSF is between 40 and 60. Much like in the short term prediction case the word “prediction” can be misleading since this “prediction” is done after examining the subframe.

If we want to understand the coding strategy we must be aware that the decoder does not exactly know the samples in the previous subframes. If the encoder tells him that he can use a sequence of values of the previous subframes he only can use what he has decoded before. If we desire that this will match best, the encoder also must use these values although he knows the exact values. In other words an identical decoder must in some way be included in the encoder.

In addition we must have a criterion for “best match”. We could use a simple squared error criterion but we will do better if we exploit the frequency masking property of the human perception. It is well established that if you have a powerful signal around a frequency f, then the human cannnot percieve a less powerful signal in the neighborhood of f, though you could hear it if it was f was not there or if the less powerful signal was centered at a different frequency far away from f. This masking property can be exploited : We can force the noise to be situated around the formant frequencies so that the signal will mask it. On the other hand the noise will be less powerful at frequencies at which the speech signal is weak. Though the overall noise increases, the listener will be more satisfied.

Therefore we should minimize the square of the error filtered by a special weighting filter which emphasizes the errors in frequency ranges where the signal is weak and de-emphasizes the errors in the neighborhood of the formants. A simple filter doing this is

$$W(z) = \frac{A_{\text{int}}(z)}{A_{\text{int}}(z/\gamma)} \qquad \text{Equ. 13}$$

with the bandwidth expansion factor $\gamma \in]0;1]$

Since $A_{\text{int}}(z)$ is the filter to pass from the original speech to the short term residual, it is often called analysis filter. By analogy, the inverse of this filter $\frac{1}{A_{\text{int}}(z)}$ is called synthesis filter. The peaks of the synthesis filter will be near the formant frequencies.

Finally $\frac{1}{A_{\text{int}}(z/\gamma)}$ will also have peaks at this frequencies, but they will be smoothed since the poles of the filter are nearer to the origin in the z plane by a γ factor. As a result, the weighting filter has gain minima near the formant frequencies so that the noise is deemphasized there. The influence of γ on the weighting filter is shown in Figure 2.

Now let's come back to the long term prediction. We want to determine the best lag so that the sequence which has been decoded for this lag matches the values to code. To take into account the attack and decay transitions we will not only code the lag but also a lag gain. The figure below shows how the periodic part of the speech can be synthesized. It does not include the innovation part which is necessary to code non periodic speech and to follow the transitions between phonemes and variations during one phoneme.

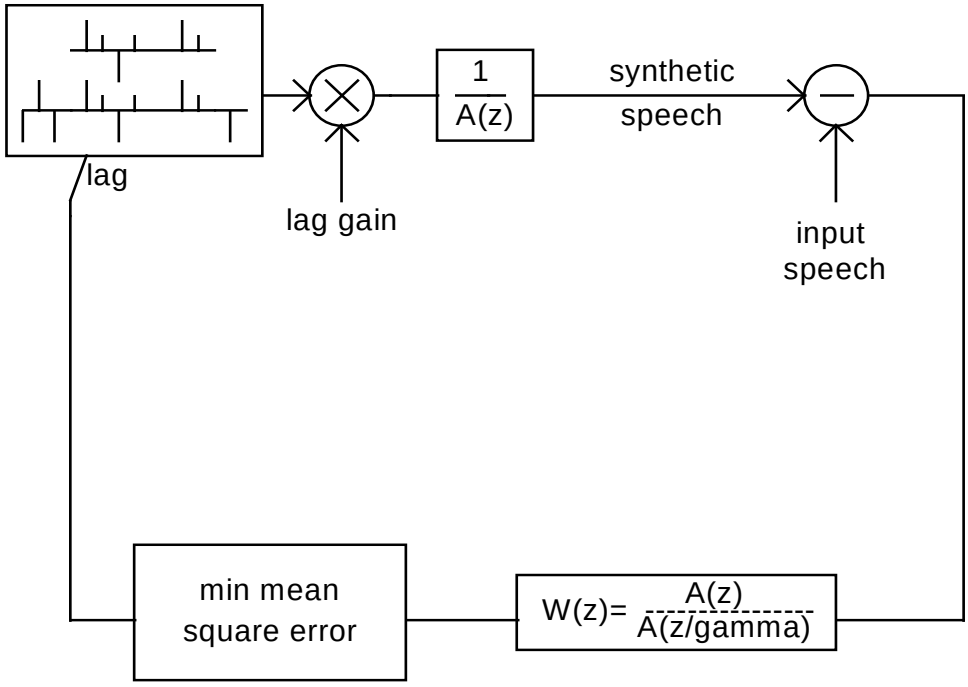


Figure 2: CELP coding principle for voiced sounds

Comparison, min error estimation, perceptual weighting and codebook searching

Note that in Figure 2 we ommit the index int to indicate that we use the interpolated A for all calculations of the subframe processing loop.

The memory which contains the previous excitations is often called "adaptive codebook". Some modifications of this scheme have been proposed to reduce complexity. Firstly we can move the weighting filter W(z) before the difference node. Then we can remove the influence of the previous frames which is caused by the non-zero filter state of the filter behind the adaptive codebook. Since this is a linear system, we can separate this influence and compute it by observing the filter output if the input is zero. This is called the zero input response.

There is no direct method to compute the best lag. If we want to know the optimal value, all we can do is to test all lag values and hold the best match. When searching for the best lag, we can distinguish between successive search (first search lag and then search gain) and joint lag and gain search. This second approach will give optimal values at the expense of complexity.

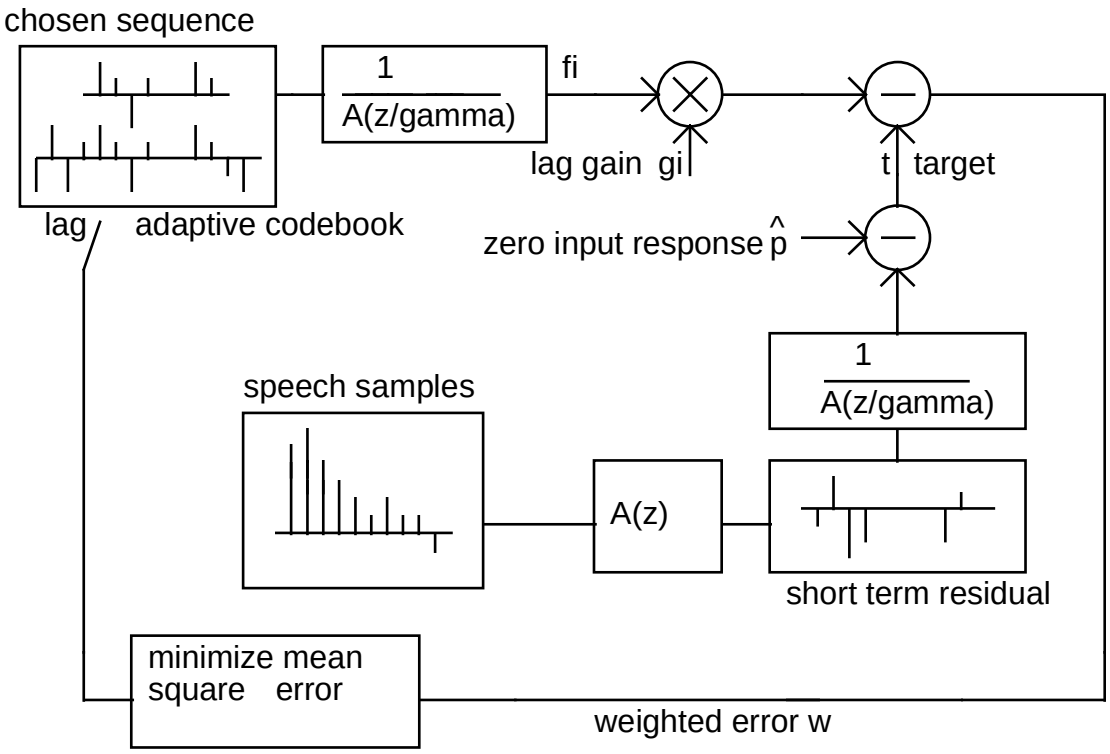


Figure 3: Reduced Complexity CELP computation

Note that the target will only be computed once per subframe, but the fi must be computed for each subframe for all lags i.

To minimize the sum of weighted errors we put

$$E = \sum_{n \in subframe} w(n)^2 \qquad \text{Equ. 14}$$

With $\vec{t}$ regrouping all NSF values of t in a subframe and $\vec{f}_i$ NSF values of the adaptive codebook for the lag i we obtain:

$$E = \left\| \vec{t} - g_i \vec{f}_i \right\|^2 = \left\| \vec{t} \right\|^2 - 2 g_i \left\langle \vec{f}_i, \vec{t} \right\rangle + g_i^2 \left\| \vec{f}_i \right\|^2 \qquad \text{Equ. 15}$$

The best gain, for a fixed vector $\vec{f}_i$, is given by

$$\frac{\delta E}{\delta g_i} = -2 \left\langle \vec{f}_i, \vec{t} \right\rangle + 2 g_i \left\| \vec{f}_i \right\|^2 = 0 \qquad \Rightarrow \qquad g_i = \frac{\left\langle \vec{f}_i, \vec{t} \right\rangle}{\left\| \vec{f}_i \right\|^2} \qquad \text{Equ. 16}$$

If we put this result into equ. 15, we obtain:

$$E = \left\|\vec{t}\right\|^2 - \frac{\left\langle \vec{f}_i, \vec{t} \right\rangle^2}{\left\|\vec{f}_i\right\|^2} \qquad \text{Equ. 17}$$

Since the first term does not depend on the lag i, we will minimize the error if we maximize the second term. The nominator term is often called the square of the correlation, the denominator is the energy. To find the best lag we have to do for all lags:

1- filter the adaptive codebook sequence $\vec{c}_i$ indicated by the lag i to get $\vec{f}_i$
2- calculate the correlation and the energy
3- if the square of the correlation divided by the energy is bigger than best value so far then note i as the best lag so far.

If we exploit the fact that the codebook sequences overlap in all but one position, we can perform the filtering by a recursive algorithm. When we have found the lag i with the highest score we can conclude the best gain by the formula above. If we did not quantize this value the result would be optimal. But the purpose of a speech coder is to compress the data, so that we have only a few quantized gain values. To avoid this we can either compute the gain for each vector and compare the errors (joint optimization) either preselect some lags and only do the exact calculation for them.

3.3 Stochastic Codebook Vector quantization

Once we have removed the main part of the redundancy by the LPC and pitch lag coding, the resulting residual is similar to Gaussian noise. There are vocoders which do not code this noise at all, the decoder has to generate an independent noise and pass it through the short term and long term prediction filter. This produces an intelligible speech of synthetic quality at very low bit rates. To achieve higher quality we cannot do without the information in this non predictible part, but scalar quantization cannot be applied since the data rate becomes prohibitive. In the middle of these extremes we can use a vector quantization which preserves high quality at low data rates at the expense of complexity.

The vector quantization method is very similar to the technique presented in the adaptive lag search : The codebook contains values which are similar to noise, that's why it is often called stochastic codebook. Opposite to the adaptive codebook, we can chose the values in the codebook before execution time. Several techniques can be used to reduce complexity: Overlapping codebooks, sparse codebooks, binary, ternary or other codebooks. Another possibility is to use multiple codebooks, so that we can use faster algorithms at the expense of a slightly increased data rate/quality ratio.

Much like in the adaptive codebook case, we have to evaluate for each index i the correlation and the energy to know if the new vector is better than the previous ones. In this case the vectors needn't be overlapping, but this is often used to reduce calculation time. The gain sg for the best vector is computed in absolute analogy.

The stochastic codebook not only has to model the residual, it is also necessary to change the values in the adaptive codebook. The manner how the adaptive codebook is updated is showed in Figure 4.

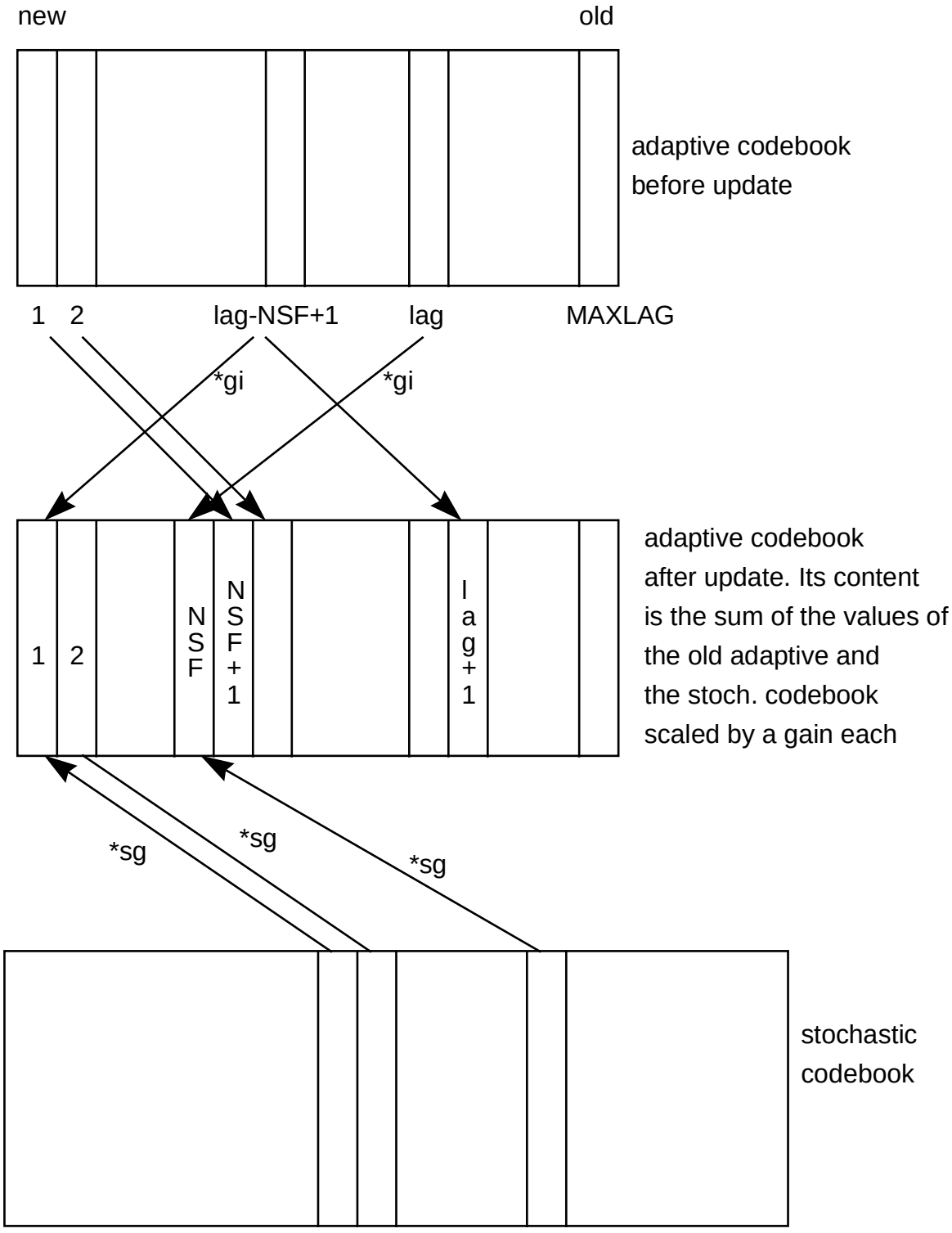


Figure 4: Adaptive Codebook Updating

4. Comparison MCELP vs. FS1016 vs. GSM half-rate

	MCELP	FS1016	GSM half-rate
Frame length	24ms	30ms	20ms
Subframe length	6ms	7.5ms	5ms
LPC transmitted parameters	10 Reflection coeffs. scalar quant. 38bits	10 LSP scalar quant. 34bits	10 Reflection coeffs vector quant. 28bits 1 bit if interpolation
adaptive codebook	127 int lags 20..146 no fractional lags, no delta search, 16 gain values 4*(7bits+4bits)	128 int lag 20..147, 128 fract. lags delta search for odd subframes, 32 gain values, (8+6+8+6+4*5) bits	122 int lags 21..142 134 fractional lags delta search for subframes 1,2,3 gain coded jointly with stoch. gain, if mode=1,2 or 3 (8+4+4+4) bits
modes	voice activity or not	-	4 voicing modes, voice activity or not
stochastic codebook	low-pass filtered undersampling for even subframes, 32 gain values 2*(12+5)bits	ternary overlapping codebook of 512 vectors, 32 gain values, 4*(9bits+5bits)	if mode=1,2 or 3 then 1 VSELP codebook is used (4*9bits), else 2 codebooks (2*4*7)bits
other functions	-	1 bit synchronization, 4 bits forward error correction (included in bit rate below)	forward error correction (not included in bit rate below)
data rate	4875bits/s	4800bits/s	5600bits/s

This comparison shows that the MCELP is a coder which is designed for cost-effective applications, in which low processor resources (RAM/ROM/Mips) are an issue. The reflection coefficients are not vector quantized, the additional data rate (about 10bits/frame => 417bit/s) is accepted to maintain low complexity.

The long term prediction lag can only take integer values, which is an important reduction of complexity since no interpolation of the adaptive codebook must be calculated. On the other hand the pitch frequency resolution is low, above all for high frequencies (steps of 5%).

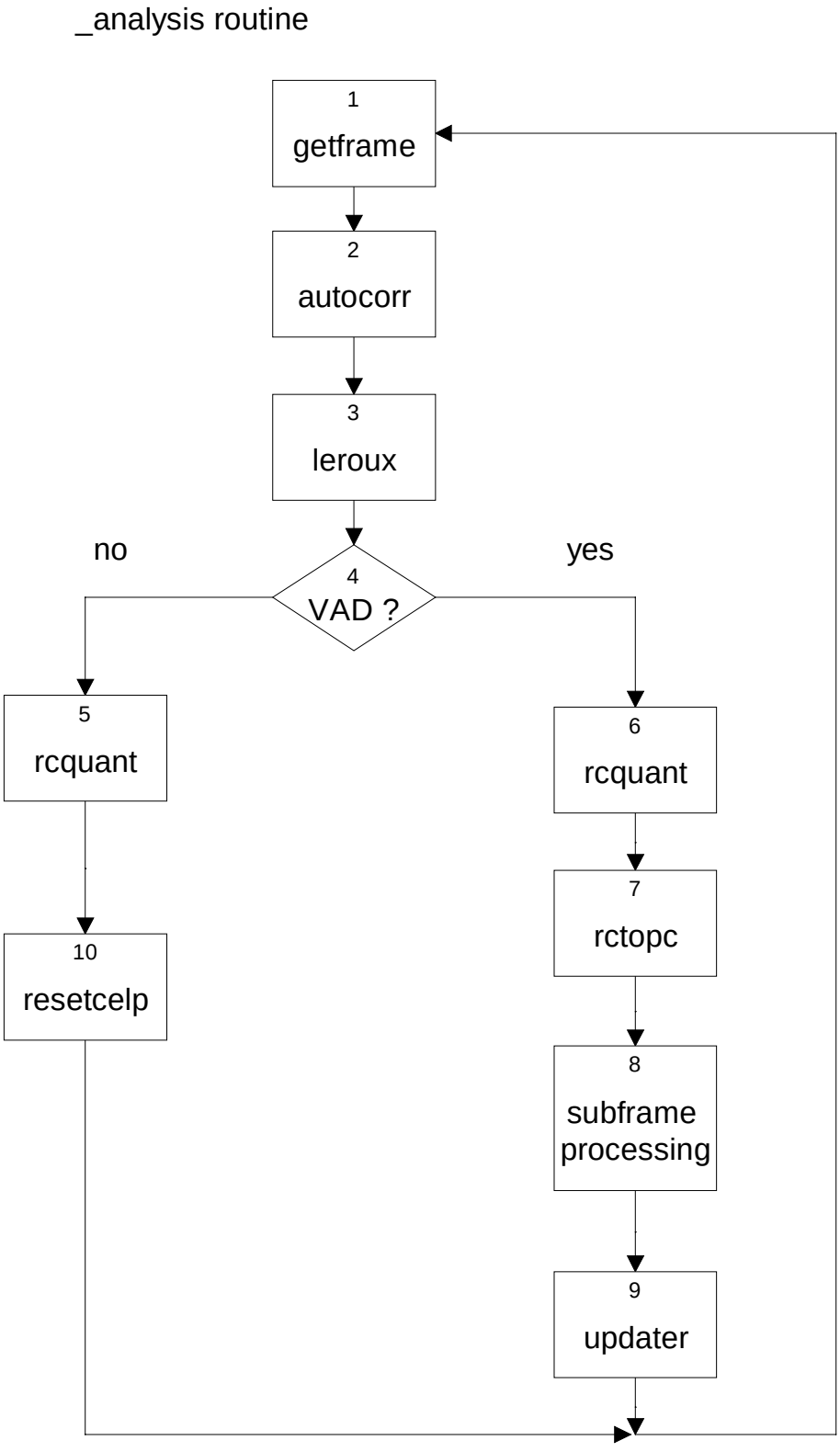
The MCELP coder does not include a real stochastic codebook. Instead of vector quantizing the long term prediction residual, a bit is transmitted for each sequence of 4 values of the backward filtered residual, which indicates if their sum is positive or negative. This can be seen as low-pass filtered subsampling. Comparing with a vector quantization, we can see that this procedure maximizes the correlation without taking into account the energy denominator. In the sequence (or vector) to approximate the residual each code bit is replaced by 4 samples of +1 or -1, then multiplied by a common gain and finally passed through the short term synthesis filter. Without the filtering operation each sequence would have the same energy, but this is no more true with filtering, so that this proceeding will have a suboptimal result compared to real vector quantization. In addition this coding technique is not very bit-efficient, instead of coding an index of 7-9 bits which points to a sequence in a codebook, we have to transmit NSF/4 values for each subframe, this means in the MCELP case we have 12bits. To avoid high data rates, only one of two subframes is coded in this way,

in the other one there is no coded innovation, it relies exclusively on the prediction, so that eventually the data rate is even lower than for the vector quantization. In addition this coding technique is of extremely low complexity. On the other hand we cannot expect to achieve very high quality.

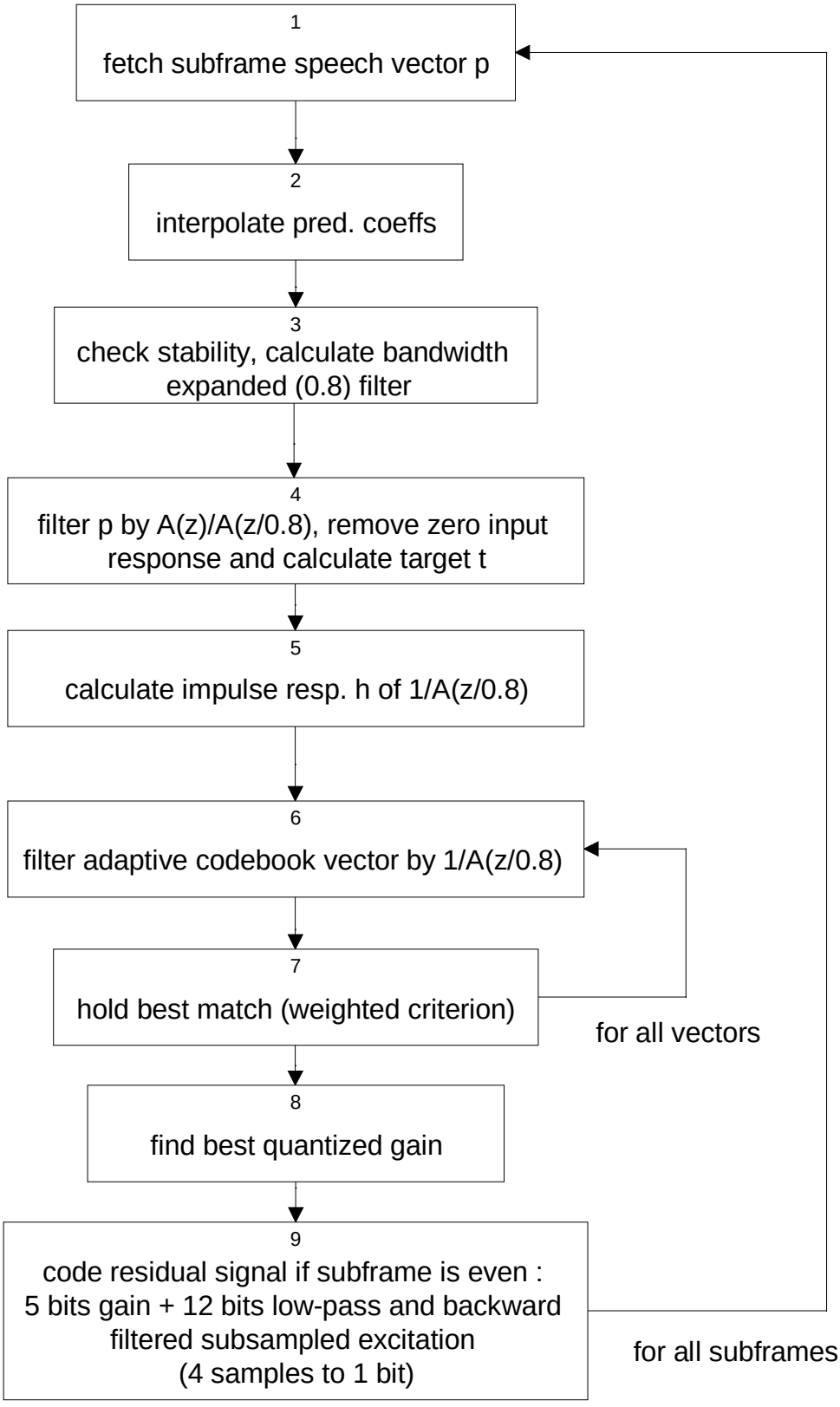
We will now introduce the concept of speech signal “modes”. If we take a look on the speech signal, we can see that we have to face very different situations : From highly periodic vowel sounds to almost random noisy consonants. If we want we can also include the difference between silent intervals and intervals with voice activity. The latter difference is used by the MCELP to lessen the bit rate in the silent intervals. In order to avoid the clicking on-off switch, the MCELP coder transmits in these intervals a simplified short term linear prediction parameter set and a code to indicate the energy of the background sound, so that the decoder can generate a sound which is called “comfort noise”. The voice activity detection algorithm is very simple in the MCELP implementation, it only takes into account the energy of the sampled speech.

The MCELP coder does not make any difference between quasiperiodic intervals and quasirandom intervals. In the first case an efficient coder could allocate more bits and more time to calculate and code the long term prediction coefficients, in the latter case the innovation (codebook or subsampling) should be emphasized. The GSM half-rate coder even distinguishes 4 modes from quasiperiodic to quasirandom.

5. MCELP 58C80 implementation Block Diagram



subframe processing :



References

Articles

[Roux 77] A fixed point computation of partial correlation coefficients
Le Roux (J.), Gueguen (C.)
ASSP 25 pp. 257-259, June 1977

[Gibs 80] Adaptive prediction in speech differential encoding systems
Gibson (J.D.)
Proc. IEEE, pp. 488-525, April 1980

[Schr 85] Code excited linear prediction (CELP) : High quality at very low bit rates
Schroeder (M.R.), Atal (B.S.)
Proc. ICASSP, Vol. 3, pp. 937-940, 1985

[Adou 87] Fast CELP coding based on algebraic codes
Adoul (J.-P.), Mabilleau (P.), Delprat (M.), Morissette (S.)
ICASSP Vol. 49, No. 4, pp. 1957-1960, 1987

[Fran 90] Variable rate speech coding with online segmentation and fast algebraic codes
Di Francesco (R.), Lamblin (C.), Le Guyader (A.), Massaloux (D.)
ICASSP vol. 1, pp. 233-236, 1990

[Klei 90] Fast methods for the CELP Speech Coding Algorithm
Kleijn (W. B.), Krasinsky (D. J.), Ketchum (R. H.)
IEEE Transactions on Acoustics, Speech and Signal Processing
Vol. 38, No. 8, August 1990

[Kroo 91] On the use of pitch predictors with high temporal resolution
Kroon (P.), Atal (B. S.)
IEEE Transactions on signal processing, Vol. 39, No. 3, March 1991

[Cupe 91] Speech Coding - Cuperman (V.)
Advances in Electronics and Electron Physics, Vol. 82, pp. 97-189, 1991

[More 91] Codage prédictif du signal de parole à débit reduit: une présentation unifiée
Moreau (N.)
Annales de télécommunications, Vol. 46, No. 3-4, pp. 223-239, 1991

[Shoh 91] Constrained stochastic excitation coding of speech at 4.8kb/s
Shoham (Y.)
Advances in Speech Coding
Atal (B.S.), Cuperman (V.), Gersho (A.) Eds.
Norwood, MA : Kluwer Academic Publishers, pp. 339-348, 1991

[Tani 91] Speech coding with dynamic bit allocation (Multimode coding)
Taniguchi (T.), Tanaka (Y.), Gray (R.M.)
Advances in Speech Coding
Atal, Cuperman,Gersho Eds.
Dordrecht, Netherlands : Kluwer Academic Publishers, pp. 329-338, 1991

[Yong 91] Efficient encoding of the long term predictor in vector excitation coders
Yong (M.), Gersho (A.)
Advances in Speech coding
Atal, Cuperman,Gersho Eds.
Dordrecht, Netherlands : Kluwer Academic Publishers, pp. 329-338, 1991

[Camp 91] The DoD 4.8 kbps standard (Proposed federal standard 1016)
Campbell (J. P. Jr), Tremain (T. E.), Welch (V. C.)
Advances in speech coding, ed. Atal, Cuperman, Gersho
Kluwer Academic Publishers, Chapter 12, pp. 121-133, 1991

[FS1016] Federal Standard 1016

Telecommunications : Analog to digital conversion of radio voice by 4800bit/s code excited linear prediction (CELP)
Febr. 1992

[Gers 92] Techniques for Improving the Performance of CELP-Type Speech Coders
Gerson (I. A.), Jasiuk (M. A.)
IEEE Journal of selected areas in communications
Vol. 10, No. 5, June 1992

[Wang 92] An objective measure for predicting subjective quality of speech coders
Wang (S.), Sekey (A.), Gersho (A.)
IEEE Journal on selected areas in communications
Vol. 10, No. 5, pp.819-829, June 1992

[Jaya 92] Signal Compression : Technology targets and research directions
Jayant (N.)
IEEE Journal on selected areas in communications,
Vol. 10, No. 5, June 1992

[Paks 93] Variable rate speech coding with phonetic segmentation
Paksoy (E.), Srinivasan (K.), Gersho (A.)
ICASSP vol. 2, pp. 155-158, 1993

[Span 94] Speech Coding : A tutorial review
Spanias (A. S.)
Proceedings of the IEEE, Vol. 82, No. 10, pp. 1541-1582, October 1994

[Gers 94] Advances in speech and audio compression
Gersho (A.)
Proceedings of the IEEE, Vol. 82, No. 6, pp. 900-918, June 1994

[ETSI 95] Draft GSM 06.20 version 0.0.3
Half-rate speech transcoding
European Telecommunications Standard Institute (ETSI) 1995

Books

[Mark 76] Linear Prediction of Speech
Markel (J.D.), Gray (A.H. Jr.)
Springer Verlag, New York 1976

[Rabi 78] Digital processing of speech signals
Rabiner (L.R.), Schafer (R.W.)
Prentice Hall, Englewood Cliffs, New Jersey 1978

[Zwic 81] Psychoacoustique
Zwicker (E.), Feldtkeller (R.)
Masson Paris et CNET-ENST 1981

[Jaya 84] Digital Coding of waveforms
Jayant (N.S.), Noll (P.)
Prentice Hall, Englewood Cliffs, New Jersey 1984

[Papa 87] Practical approach to speech coding
Papamichalis (P.E.)
Prentice Hall, Englewood Cliffs, New Jersey 1987

[More 94] Techniques de compression des signaux
Moreau (N.)
Masson, Paris et CNET-ENST 1994

Appendix : Abbreviations, keywords, and symbols

Abbreviations and keywords

CELP	code excited linear prediction
MCELP	modified CELP, for low memory, low processor resources
LPC	linear predicting coding, used for short term prediction
pitch lag	number of samples for a speech period for voiced speech
lag	examined candidate for the pitch lag in the search procedure
q12,q15 ...	fix point number representation format indicating the position of the virtual comma in the binary number (e.g. 12 means 12 bits behind comma)
VAD	voice activity detection
frame	segment of speech in which the vocal tract is nearly stationary (20..30ms)
subframe	subdivision of a frame (often frame/4)
onset	transition from a low to a high energy segment
innovation	non predictable part of the speech
$\langle \vec{a}, \vec{b} \rangle$	scalar product
$\|\vec{a}\|$	norm with $\|\vec{a}\|^2 = \langle \vec{a}, \vec{a} \rangle$
xxxx bit/s	always voice activity supposed

Note that some symbols may sometimes appear in *italics* and indices as lower case characters (in particular in drawings) without a difference in signification.

Symbols

s(n)	sampled speech input signal
NP	number of samples used for short term prediction (MCELP : 10)
NF	number of samples in a frame (MCELP : 192)
$\hat{s}(n)$	short term linear prediction for s(n)
e(n)	short term prediction error or short term residual
a_i	short term prediction coefficient
$r_{i,j}$	estimated autocorrelation for 1 frame using covariance method
r_i	estimated autocorrelation for 1 frame using autocorrelation method
h(n) (1)	window for autocorrelation function
$\tilde{s}(n)$	speech weighted by the window
A(z)	short term analysis filter or short term linear prediction filter
$A_{int}(z)$	interpolated analysis filter
α	analysis filter interpolation coefficient
NSF	number of samples in a subframe (MCELP : 48)
W(z)	perceptual weighting filter
γ	bandwidth expansion coefficient (MCELP : 0.8)
$\vec{c}_i$	excitation vector of one of the two codebooks
fi(n)	filtered excitation of one codebook
$\vec{f}_i$	candidate shape vector to model one of the two targets, its elements are fi(n)
g_i or gi	pitch gain factor
s_g or sg	stochastic gain factor
$\vec{t}$	target vector of the adaptive codebook search
$\vec{t}_2$	target vector of the stochastic codebook search
$\hat{p}$	zero input response of the speech synthesis filter
w(n)	weighted error signal
E	sum of squares weighted errors for one subframe
h (2), H	FIR filter to model the IIR filter 1/A(z/γ)
r(i)	adaptive codebook, -MAXLAG ≤ i ≤ -1
zl	intermediate vector to calculate the filter output for lag=MINLAG
S	shift matrix
d(k)	backward filtered target
q_i	coefficients for the adaptive codebook interpolation
m_1	score threshold for end of search after test of “nearlags”
m_2	score threshold for voiced to unvoiced transition
mdiffmin	amount that a score for a farlag must be better than for a nearlag to be chosen