Technical Specification Group Services and System Aspects
Meeting #3, Yokohama, Japan, April 26-28 1999

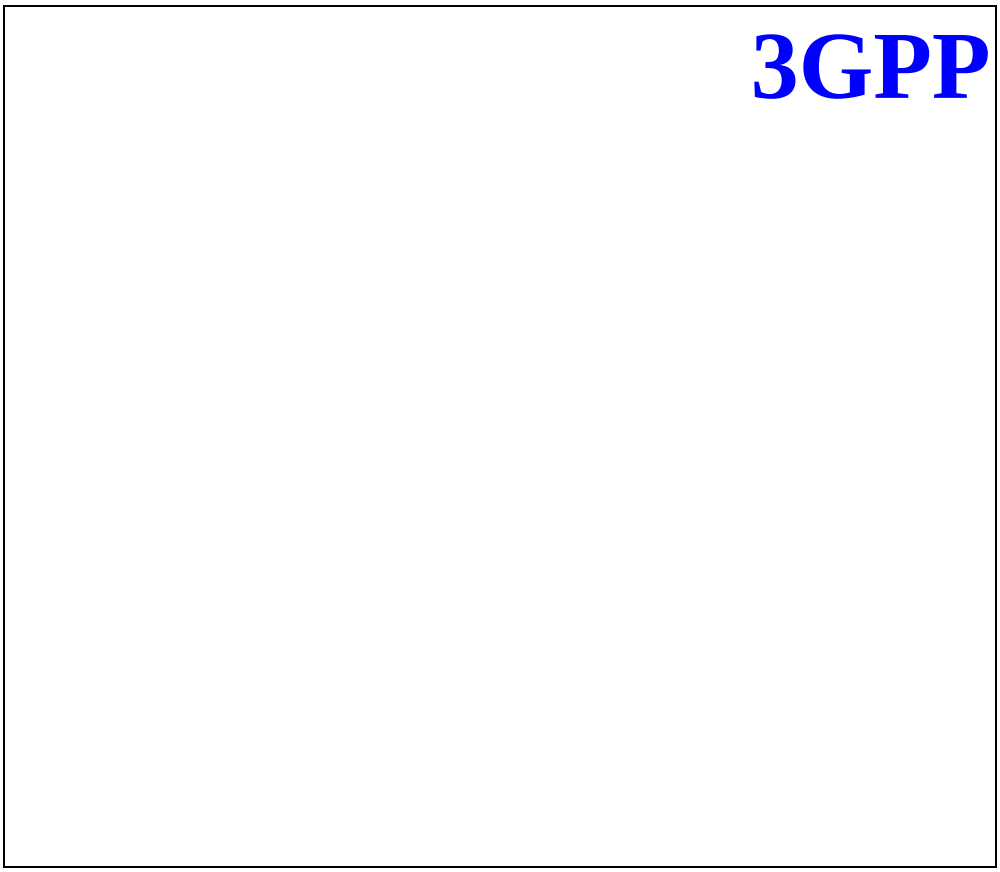
TSGS#3(99)197
(replaces Tdoc SP-99132)

Source: TSG-S4
Agenda Item: 5.4.3

TS 26.090 V1.0.0 (1999-04)

Technical Specification

3rd Generation Partnership Project (3GPP)
TSG-SA Codec Working Group
Mandatory Speech Codec speech processing functions
AMR speech codec; Transcoding functions



Reference
TSG-SA4-W1 (<Shortfilename>.PDF)

Keywords
Adaptive Multi-Rate, Mandatory speech coder

3GPP

Postal address

Office address

Internet
secretariat@3gpp.org
Individual copies of this deliverable
can be downloaded from
http://www.3gpp.org

Contents

Intellectual Property Rights

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Working Group 4 (Codec).

The contents of this informal TS may be subject to continuing work within the 3GPP and may change following formal TSG-S4 approval. Should TSG-S4 modify the contents of this TS, it will be re-released with an identifying change of release date and an increase in version number as follows:

Version m.t.e

where:

m indicates [major version number]

x the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

y the third digit is incremented when editorial only changes have been incorporated into the specification.

1 Scope

This Telecommunication Standard (TS) describes the detailed mapping from input blocks of 160 speech samples in 13-bit uniform PCM format to encoded blocks of 95, 103, 118, 134, 148, 159, 204, and 244 bits and from encoded blocks of 95, 103, 118, 134, 148, 159, 204, and 244 bits to output blocks of 160 reconstructed speech samples. The sampling rate is 8 000 samples/s leading to a bit rate for the encoded bit stream of 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2 or 12.2 kbit/s. The coding scheme for the multi-rate coding modes is the so-called Algebraic Code Excited Linear Prediction Coder, hereafter referred to as ACELP. The multi-rate ACELP coder is referred to as MR-ACELP.

In the case of discrepancy between the requirements described in this TS and the fixed point computational description (ANSI-C code) of these requirements contained in [4], the description in [4] will prevail. The ANSI-C code is not described in this TS, see [4] for a description of the ANSI-C code.

The transcoding procedure specified in this TS is mandatory for systems using the AMR speech codec.

2 Normative references

This TS incorporates by dated and undated reference, provisions from other publications. These normative references are cited in the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this TS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1] GSM 03.50: " Digital cellular telecommunications system (Phase 2); Transmission planning aspects of the speech service in the GSM Public Land Mobile Network (PLMN) system"

[2] TS 26.101 : "AMR Speech Codec; Frame structure".

[3] TS 26.094: "AMR Speech Codec; Voice Activity Detection (VAD)".

[4] TS 26.073: "AMR Speech Codec; ANSI-C code".

[5] TS 26.074: "AMR Speech Codec; Test sequences".

[6] ITU-T Recommendation G.711 (1988): "Coding of analogue signals by pulse code modulation Pulse code modulation (PCM) of voice frequencies".

[7] ITU-T Recommendation G.726: "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of this TS, the following definitions apply:

adaptive codebook: The adaptive codebook contains excitation vectors that are adapted for every subframe. The adaptive codebook is derived from the long-term filter state. The lag value can be viewed as an index into the adaptive codebook.

adaptive postfilter: This filter is applied to the output of the short-term synthesis filter to enhance the perceptual quality of the reconstructed speech. In the adaptive multi-rate codec, the adaptive postfilter is a cascade of two filters: a formant postfilter and a tilt compensation filter.

algebraic codebook: A fixed codebook where algebraic code is used to populate the excitation vectors (innovation vectors). The excitation contains a small number of nonzero pulses with predefined interlaced sets of positions..

anti-sparseness processing: An adaptive post-processing procedure applied to the fixed codebook vector in order to reduce perceptual artifacts from a sparse fixed codebook vector.

closed-loop pitch analysis: This is the adaptive codebook search, i.e., a process of estimating the pitch (lag) value from the weighted input speech and the long term filter state. In the closed-loop search, the lag is searched using error minimization loop (analysis-by-synthesis). In the adaptive multi-rate codec, closed-loop pitch search is performed for every subframe.

direct form coefficients: One of the formats for storing the short term filter parameters. In the adaptive multi-rate codec, all filters which are used to modify speech samples use direct form coefficients.

fixed codebook: The fixed codebook contains excitation vectors for speech synthesis filters. The contents of the codebook are non-adaptive (i.e., fixed). In the adaptive multi-rate codec, the fixed codebook is implemented using an algebraic codebook.

fractional lags: A set of lag values having sub-sample resolution. In the adaptive multi-rate codec a sub-sample resolution of 1/6th or 1/3rd of a sample is used.

frame: A time interval equal to 20 ms (160 samples at an 8 kHz sampling rate).

integer lags: A set of lag values having whole sample resolution.

interpolating filter: An FIR filter used to produce an estimate of subsample resolution samples, given an input sampled with integer sample resolution.

inverse filter: This filter removes the short term correlation from the speech signal. The filter models an inverse frequency response of the vocal tract.

lag: The long term filter delay. This is typically the true pitch period, or its multiple or sub-multiple.

Line Spectral Frequencies: (see Line Spectral Pair)

Line Spectral Pair: Transformation of LPC parameters. Line Spectral Pairs are obtained by decomposing the inverse filter transfer function A(z) to a set of two transfer functions, one having even symmetry and the other having odd symmetry. The Line Spectral Pairs (also called as Line Spectral Frequencies) are the roots of these polynomials on the z-unit circle.

LP analysis window: For each frame, the short term filter coefficients are computed using the high pass filtered speech samples within the analysis window. In the adaptive multi-rate codec, the length of the analysis window is always 240 samples. For each frame, two asymmetric windows are used to generate two sets of LP coefficient in the 12.2 kbit/s mode. For the other modes, only a single asymmetric window is used to generate a single set of LP coefficients. In the 12.2 kbit/s mode, no samples of the future frames are used (no lookahead). The other modes use a 5 ms lookahead.

LP coefficients: Linear Prediction (LP) coefficients (also referred as Linear Predictive Coding (LPC) coefficients) is a generic descriptive term for the short term filter coefficients.

mode: When used alone, refers to the source codec mode, i.e., to one of the source codecs employed in the AMR codec.

open-loop pitch search: A process of estimating the near optimal lag directly from the weighted speech input. This is done to simplify the pitch analysis and confine the closed-loop pitch search to a small number of lags around the open-loop estimated lags. In the adaptive multi-rate codec, an open-loop pitch search is performed in every other subframe.

residual: The output signal resulting from an inverse filtering operation.

short term synthesis filter: This filter introduces, into the excitation signal, short term correlation which models the impulse response of the vocal tract.

perceptual weighting filter: This filter is employed in the analysis-by-synthesis search of the codebooks. The filter exploits the noise masking properties of the formants (vocal tract resonances) by weighting the error less in regions near the formant frequencies and more in regions away from them.

subframe: A time interval equal to 5 ms (40 samples at 8 kHz sampling rate).

vector quantization: A method of grouping several parameters into a vector and quantizing them simultaneously.

zero input response: The output of a filter due to past inputs, i.e. due to the present state of the filter, given that an input of zeros is applied.

zero state response: The output of a filter due to the present input, given that no past inputs have been applied, i.e., given that the state information in the filter is all zeroes.

3.2 Symbols

For the purposes of this TS, the following symbols apply:

$A(z)$	The inverse filter with unquantized coefficients
$\hat{A}(z)$	The inverse filter with quantized coefficients
$H(z) = \frac{1}{\hat{A}(z)}$	The speech synthesis filter with quantized coefficients
a_i	The unquantized linear prediction parameters (direct form coefficients)
$\hat{a}_i$	The quantified linear prediction parameters
m	The order of the LP model

$\frac{1}{B(z)}$	The long-term synthesis filter
$W(z)$	The perceptual weighting filter (unquantized coefficients)
g_1, g_2	The perceptual weighting factors
$F_E(z)$	Adaptive pre-filter
T	The integer pitch lag nearest to the closed-loop fractional pitch lag of the subframe
β	The adaptive pre-filter coefficient (the quantified pitch gain)
$H_f(z) = \frac{\hat{A}(z/g_n)}{\hat{A}(z/g_d)}$	The formant postfilter
γ_n	Control coefficient for the amount of the formant post-filtering
g_d	Control coefficient for the amount of the formant post-filtering
$H_t(z)$	Tilt compensation filter
γ_t	Control coefficient for the amount of the tilt compensation filtering
$m = g_t k_1'$	A tilt factor, with k_1' being the first reflection coefficient
$h_f(n)$	The truncated impulse response of the formant postfilter
L_h	The length of $h_f(n)$
$r_h(i)$	The auto-correlations of $h_f(n)$
$\hat{A}(z/g_n)$	The inverse filter (numerator) part of the formant postfilter
$1/\hat{A}(z/g_d)$	The synthesis filter (denominator) part of the formant postfilter
$\hat{r}(n)$	The residual signal of the inverse filter $\hat{A}(z/g_n)$
$h_t(n)$	Impulse response of the tilt compensation filter
$b_{sc}(n)$	The AGC-controlled gain scaling factor of the adaptive postfilter
α	The AGC factor of the adaptive postfilter
$H_{h1}(z)$	Pre-processing high-pass filter
$w_I(n)$, $w_{II}(n)$	LP analysis windows
$L_1^{(I)}$	Length of the first part of the LP analysis window $w_I(n)$
$L_2^{(I)}$	Length of the second part of the LP analysis window $w_I(n)$

$L_1^{(II)}$	Length of the first part of the LP analysis window $w_{II}(n)$
$L_2^{(II)}$	Length of the second part of the LP analysis window $w_{II}(n)$
$r_{ac}(k)$	The auto-correlations of the windowed speech $s'(n)$
$w_{lag}(i)$	Lag window for the auto-correlations (60 Hz bandwidth expansion)
f_0	The bandwidth expansion in Hz
f_s	The sampling frequency in Hz
$r'_{ac}(k)$	The modified (bandwidth expanded) auto-correlations
$E_{LD}(i)$	The prediction error in the ith iteration of the Levinson algorithm
k_i	The ith reflection coefficient
$a_j^{(i)}$	The jth direct form coefficient in the ith iteration of the Levinson algorithm
$F_1'(z)$	Symmetric LSF polynomial
$F_2'(z)$	Antisymmetric LSF polynomial
$F_1(z)$	Polynomial $F_1'(z)$ with root $z = -1$ eliminated
$F_2(z)$	Polynomial $F_2'(z)$ with root $z = 1$ eliminated
q_i	The line spectral pairs (LSPs) in the cosine domain
$\mathbf{q}$	An LSP vector in the cosine domain
$\hat{\mathbf{q}}_i^{(n)}$	The quantified LSP vector at the ith subframe of the frame n
w_i	The line spectral frequencies (LSFs)
$T_m(x)$	A mth order Chebyshev polynomial
$f_1(i), f_2(i)$	The coefficients of the polynomials $F_1(z)$ and $F_2(z)$
$f_1'(i), f_2'(i)$	The coefficients of the polynomials $F_1'(z)$ and $F_2'(z)$
$f(i)$	The coefficients of either $F_1(z)$ or $F_2(z)$
$C(x)$	Sum polynomial of the Chebyshev polynomials
x	Cosine of angular frequency w
l_k	Recursion coefficients for the Chebyshev polynomial evaluation
f_i	The line spectral frequencies (LSFs) in Hz

$\mathbf{f}^t = [f_1\ f_2 \ldots f_{10}]$ The vector representation of the LSFs in Hz

$\mathbf{z}^{(1)}(n)$, $\mathbf{z}^{(2)}(n)$ The mean-removed LSF vectors at frame n

$\mathbf{r}^{(1)}(n)$, $\mathbf{r}^{(2)}(n)$ The LSF prediction residual vectors at frame n

$\mathbf{p}(n)$ The predicted LSF vector at frame n

$\hat{\mathbf{r}}^{(2)}(n-1)$ The quantified second residual vector at the past frame

$\hat{\mathbf{f}}^k$ The quantified LSF vector at quantization index k

E_{LSP} The LSP quantization error

$w_i, i = 1,\ldots,10,$ LSP-quantization weighting factors

d_i The distance between the line spectral frequencies f_{i+1} and f_{i-1}

$h(n)$ The impulse response of the weighted synthesis filter

O_k The correlation maximum of open-loop pitch analysis at delay k

$O_{t_i}, i=1,\ldots,3$ The correlation maxima at delays $t_i, i = 1,\ldots,3$

$(M_i, t_i), i = 1,\ldots,3$ The normalized correlation maxima M_i and the corresponding delays $t_i, i = 1,\ldots,3$

$H(z)W(z) = \dfrac{A(z/g_1)}{\hat{A}(z)A(z/g_2)}$ The weighted synthesis filter

$A(z/g_1)$ The numerator of the perceptual weighting filter

$1/A(z/g_2)$ The denominator of the perceptual weighting filter

T_1 The integer nearest to the fractional pitch lag of the previous (1st or 3rd) subframe

$s'(n)$ The windowed speech signal

$s_w(n)$ The weighted speech signal

$\hat{s}(n)$ Reconstructed speech signal

$\hat{s}'(n)$ The gain-scaled post-filtered signal

$\hat{s}_f(n)$ Post-filtered speech signal (before scaling)

$x(n)$ The target signal for adaptive codebook search

$x_2(n)$, $\mathbf{x}_2^t$ The target signal for algebraic codebook search

$res_{LP}(n)$ The LP residual signal

$c(n)$ The fixed codebook vector

$v(n)$	The adaptive codebook vector
$y(n) = v(n) * h(n)$	The filtered adaptive codebook vector
$y_k(n)$	The past filtered excitation
$u(n)$	The excitation signal
$\hat{u}(n)$	The emphasized adaptive codebook vector
$\hat{u}'(n)$	The gain-scaled emphasized excitation signal
T_{op}	The best open-loop lag
t_{min}	Minimum lag search value
t_{max}	Maximum lag search value
$R(k)$	Correlation term to be maximized in the adaptive codebook search
b_{24}	The FIR filter for interpolating the normalized correlation term $R(k)$
$R(k)_t$	The interpolated value of $R(k)$ for the integer delay k and fraction t
b_{60}	The FIR filter for interpolating the past excitation signal $u(n)$ to yield the adaptive codebook vector $v(n)$
A_k	Correlation term to be maximized in the algebraic codebook search at index k
C_k	The correlation in the numerator of A_k at index k
E_{Dk}	The energy in the denominator of A_k at index k
$\mathbf{d} = \mathbf{H}^t\mathbf{x}_2$	The correlation between the target signal $x_2(n)$ and the impulse response $h(n)$, i.e., backward filtered target
$\mathbf{H}$	The lower triangular Toepliz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1),\ldots,h(39)$
$\Phi = \mathbf{H}^t\mathbf{H}$	The matrix of correlations of $h(n)$
$d(n)$	The elements of the vector $\mathbf{d}$
$f(i, j)$	The elements of the symmetric matrix Φ
$\mathbf{c}_k$	The innovation vector
C	The correlation in the numerator of A_k
m_i	The position of the ith pulse
J_i	The amplitude of the ith pulse

N_p	The number of pulses in the fixed codebook excitation
E_D	The energy in the denominator of A_k
$res_{LTP}(n)$	The normalized long-term prediction residual
$b(n)$	The signal used for presetting the signs in algebraic codebook search
$s_b(n)$	The sign signal for the algebraic codebook search
$d'(n)$	Sign extended backward filtered target
$f^{'}(i,j)$	The modified elements of the matrix Φ, including sign information
$\mathbf{z}^t$, $z(n)$	The fixed codebook vector convolved with $h(n)$
$E(n)$	The mean-removed innovation energy (in dB)
$\overline{E}$	The mean of the innovation energy
$\tilde{E}(n)$	The predicted energy
$[b_1\ b_2\ b_3\ b_4]$	The MA prediction coefficients
$\hat{R}(k)$	The quantified prediction error at subframe k
E_I	The mean innovation energy
$R(n)$	The prediction error of the fixed-codebook gain quantization
E_Q	The quantization error of the fixed-codebook gain quantization
$e(n)$	The states of the synthesis filter $1/\hat{A}(z)$
$e_w(n)$	The perceptually weighted error of the analysis-by-synthesis search
h	The gain scaling factor for the emphasized excitation
g_c	The fixed-codebook gain
g_c'	The predicted fixed-codebook gain
$\hat{g}_c$	The quantified fixed codebook gain
g_p	The adaptive codebook gain
$\hat{g}_p$	The quantified adaptive codebook gain
$g_{gc} = g_c / g_c'$	A correction factor between the gain g_c and the estimated one g_c'
$\hat{g}_{gc}$	The optimum value for g_{gc}
g_{sc}	Gain scaling factor

3.3 Abbreviations

For the purposes of this TS, the following abbreviations apply.

ACELP	Algebraic Code Excited Linear Prediction
AGC	Adaptive Gain Control
AMR	Adaptive Multi-Rate
CELP	Code Excited Linear Prediction
EFR	Enhanced Full Rate
FIR	Finite Impulse Response
ISPP	Interleaved Single-Pulse Permutation
LP	Linear Prediction
LPC	Linear Predictive Coding
LSF	Line Spectral Frequency
LSP	Line Spectral Pair
LTP	Long Term Predictor (or Long Term Prediction)
MA	Moving Average

4 Outline description

This TS is structured as follows:

Section 4.1 contains a functional description of the audio parts including the A/D and D/A functions. Section 4.2 describes the conversion between 13-bit uniform and 8-bit A-law or m -law samples. Sections 4.3 and 4.4 present a simplified description of the principles of the AMR codec encoding and decoding process respectively. In subclause 4.5, the sequence and subjective importance of encoded parameters are given.

Section 5 presents the functional description of the AMR codec encoding, whereas clause 6 describes the decoding procedures. In section 7, the detailed bit allocation of the AMR codec is tabulated.

4.1 Functional description of audio parts

The analogue-to-digital and digital-to-analogue conversion will in principle comprise the following elements:

1) Analogue to uniform digital PCM

   – microphone;

   – input level adjustment device;

   – input anti-aliasing filter;

   – sample-hold device sampling at 8 kHz;

   – analogue–to–uniform digital conversion to 13–bit representation.

   The uniform format shall be represented in two's complement.

2) Uniform digital PCM to analogue

   - conversion from 13–bit/8 kHz uniform PCM to analogue;
   - a hold device;
   - reconstruction filter including x/sin(x) correction;
   - output level adjustment device;
   - earphone or loudspeaker.

In the terminal equipment, the A/D function may be achieved either

- by direct conversion to 13-bit uniform PCM format;

- or by conversion to 8-bit A-law or m -law compounded format, based on a standard A-law or m -law codec/filter according to ITU-T Recommendations G.711 [6] and G.714, followed by the 8-bit to 13-bit conversion as specified in subclause 4.2.1.

For the D/A operation, the inverse operations take place.

In the latter case it should be noted that the specifications in ITU-T G.714 (superseded by G.712) are concerned with PCM equipment located in the central parts of the network. When used in the terminal equipment, this TS does not on its own ensure sufficient out-of-band attenuation. The specification of out-of-band signals is defined in [1] in clause 2.

4.2 Preparation of speech samples

The encoder is fed with data comprising of samples with a resolution of 13 bits left justified in a 16-bit word. The three least significant bits are set to '0'. The decoder outputs data in the same format. Outside the speech codec further processing must be applied if the traffic data occurs in a different representation.

4.2.1 PCM format conversion

The conversion between 8-bit A-Law or m -law compressed data and linear data with 13-bit resolution at the speech encoder input shall be as defined in ITU-T Rec. G.711 [6].

ITU-T Rec. G.711 [6] specifies the A-Law or m -law to linear conversion and vice versa by providing table entries. Examples on how to perform the conversion by fixed-point arithmetic can be found in ITU-T Rec. G.726 [7]. Section 4.2.1 of G.726 [7] describes A-Law or m -law to linear expansion and subclause 4.2.8 of G.726 [7] provides a solution for linear to A-Law or m -law compression.

4.3 Principles of the adaptive multi-rate speech encoder

The AMR codec consists of eight source codecs with bit-rates of 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15 and 4.75 kbit/s.

The codec is based on the code-excited linear predictive (CELP) coding model. A 10th order linear prediction (LP), or short-term, synthesis filter is used which is given by:

$$H(z) = \frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{m} \hat{a}_i z^{-i}}, \quad (1)$$

where $\hat{a}_i, i = 1, \ldots, m,$ are the (quantified) linear prediction (LP) parameters, and $m = 10$ is the predictor order. The long-term, or pitch, synthesis filter is given by:

$$\frac{1}{B(z)} = \frac{1}{1 - g_p z^{-T}}, \quad (2)$$

where T is the pitch delay and g_p is the pitch gain. The pitch synthesis filter is implemented using the so-called adaptive codebook approach.

The CELP speech synthesis model is shown in figure 2. In this model, the excitation signal at the input of the short-term LP synthesis filter is constructed by adding two excitation vectors from adaptive and fixed (innovative) codebooks. The speech is synthesized by feeding the two properly chosen vectors from these codebooks through the short-term synthesis filter. The optimum excitation sequence in a codebook is chosen using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure.

The perceptual weighting filter used in the analysis-by-synthesis search technique is given by:

$$W(z) = \frac{A(z/g_1)}{A(z/g_2)}, \quad (3)$$

where $A(z)$ is the unquantized LP filter and $0 < \gamma_2 < \gamma_1 \leq 1$ are the perceptual weighting factors. The values $\gamma_1 = 0.9$ (for the 12.2 and 10.2 kbit/s mode) or $g_1 = 0.94$ (for all other modes) and $\gamma_2 = 0.6$ are used. The weighting filter uses the unquantized LP parameters.

The coder operates on speech frames of 20 ms corresponding to 160 samples at the sampling frequency of 8 000 sample/s. At each 160 speech samples, the speech signal is analysed to extract the parameters of the CELP model (LP filter coefficients, adaptive and fixed codebooks' indices and gains). These parameters are encoded and transmitted. At the decoder, these parameters are decoded and speech is synthesized by filtering the reconstructed excitation signal through the LP synthesis filter.

The signal flow at the encoder is shown in figure 3. LP analysis is performed twice per frame for the 12.2 kbit/s mode and once for the other modes. For the 12.2 kbit/s mode, the two sets of LP parameters are converted to line spectrum pairs (LSP) and jointly quantized using split matrix quantization (SMQ) with 38 bits. For the other modes, the single set of LP parameters is converted to line spectrum pairs (LSP) and vector quantized using split vector quantization (SVQ). The speech frame is divided into 4 subframes of 5 ms each (40 samples). The adaptive and fixed codebook parameters are transmitted every subframe. The quantized and unquantized LP parameters or their interpolated versions are used depending on the subframe. An open-loop pitch lag is estimated in every other subframe (except for the 5.15 and 4.75 kbit/s modes for which it is done once per frame) based on the perceptually weighted speech signal.

Then the following operations are repeated for each subframe:

The target signal $x(n)$ is computed by filtering the LP residual through the weighted synthesis filter $W(z)H(z)$ with the initial states of the filters having been updated by filtering the error between LP residual and excitation (this is equivalent to the common approach of subtracting the zero input response of the weighted synthesis filter from the weighted speech signal).

The impulse response, $h(n)$ of the weighted synthesis filter is computed.

Closed-loop pitch analysis is then performed (to find the pitch lag and gain), using the target $x(n)$ and impulse response $h(n)$, by searching around the open-loop pitch lag. Fractional pitch with 1/6th or 1/3rd of a sample resolution (depending on the mode) is used.

The target signal $x(n)$ is updated by removing the adaptive codebook contribution (filtered adaptive codevector), and this new target, $x_2(n)$, is used in the fixed algebraic codebook search (to find the optimum innovation).

The gains of the adaptive and fixed codebook are scalar quantified with 4 and 5 bits respectively or vector quantified with 6-7 bits (with moving average (MA) prediction applied to the fixed codebook gain).

Finally, the filter memories are updated (using the determined excitation signal) for finding the target signal in the next subframe.

The bit allocation of the AMR codec modes is shown in table 1. In each 20 ms speech frame, 95, 103, 118, 134, 148, 159, 204 or 244 bits are produced, corresponding to a bit-rate of 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2 or 12.2 kbit/s. More detailed bit allocation among the codec parameters is given in tables 9a-9h. Note that the most significant bits (MSB) are always sent first.

Table 1: Bit allocation of the AMR coding algorithm for 20 ms frame

Mode	Parameter	1st subframe	2nd subframe	3rd subframe	4th subframe	total per frame
12.2 kbit/s (GSM EFR)	2 LSP sets					38
	Pitch delay	9	6	9	6	30
	Pitch gain	4	4	4	4	16
	Algebraic code	35	35	35	35	140
	Codebook gain	5	5	5	5	20
	Total					244
10.2 kbit/s	LSP set					26
	Pitch delay	8	5	8	5	26
	Algebraic code	31	31	31	31	124
	Gains	7	7	7	7	28
	Total					204
7.95 kbit/s	LSP sets					27
	Pitch delay	8	6	8	6	28
	Pitch gain	4	4	4	4	16
	Algebraic code	17	17	17	17	68
	Codebook gain	5	5	5	5	20
	Total					159
7.40 kbit/s (TDMA EFR)	LSP set					26
	Pitch delay	8	5	8	5	26
	Algebraic code	17	17	17	17	68
	Gains	7	7	7	7	28
	Total					148
6.70 kbit/s (PDC EFR)	LSP set					26
	Pitch delay	8	4	8	4	24
	Algebraic code	14	14	14	14	56
	Gains	7	7	7	7	28
	Total					134
5.90 kbit/s	LSP set					26
	Pitch delay	8	4	8	4	24
	Algebraic code	11	11	11	11	44
	Gains	6	6	6	6	24
	Total					118
5.15 kbit/s	LSP set					23
	Pitch delay	8	4	4	4	20
	Algebraic code	9	9	9	9	36
	Gains	6	6	6	6	24
	Total					103
4.75 kbit/s	LSP set					23
	Pitch delay	8	4	4	4	20
	Algebraic code	9	9	9	9	36
	Gains	8		8		16
	Total					95

4.4 Principles of the adaptive multi-rate speech decoder

The signal flow at the decoder is shown in figure 4. At the decoder, based on the chosen mode, the transmitted indices are extracted from the received bitstream. The indices are decoded to obtain the coder parameters at each transmission frame. These parameters are the LSP vectors, the fractional pitch lags, the innovative codevectors, and the pitch and innovative gains. The LSP vectors are converted to the LP filter coefficients and interpolated to obtain LP filters at each subframe. Then, at each 40-sample subframe:

- the excitation is constructed by adding the adaptive and innovative codevectors scaled by their respective gains;

- the speech is reconstructed by filtering the excitation through the LP synthesis filter.

Finally, the reconstructed speech signal is passed through an adaptive postfilter.

4.5 Sequence and subjective importance of encoded parameters

The encoder will produce the output information in a unique sequence and format, and the decoder must receive the same information in the same way. In table 9a-9h, the sequence of output bits and the bit allocation for each parameter is shown.

The different parameters of the encoded speech and their individual bits have unequal importance with respect to subjective quality. The output and input frame formats for the AMR speech codec are given in [2], where a reordering of bits take place.

5 Functional description of the encoder

In this clause, the different functions of the encoder represented in figure 3 are described.

5.1 Pre-processing (all modes)

Two pre-processing functions are applied prior to the encoding process: high-pass filtering and signal down-scaling.

Down-scaling consists of dividing the input by a factor of 2 to reduce the possibility of overflows in the fixed-point implementation.

The high-pass filter serves as a precaution against undesired low frequency components. A filter with a cut off frequency of 80 Hz is used, and it is given by:

$$H_{h1}(z) = \frac{0.927246093 - 1.8544941 z^{-1} + 0.927246903 z^{-2}}{1 - 1.906005859 z^{-1} + 0.911376953 z^{-2}}. \quad (4)$$

Down-scaling and high-pass filtering are combined by dividing the coefficients at the numerator of $H_{h1}(z)$ by 2.

5.2 Linear prediction analysis and quantization

12.2 kbit/s mode

Short-term prediction, or linear prediction (LP), analysis is performed twice per speech frame using the auto-correlation approach with 30 ms asymmetric windows. No lookahead is used in the auto-correlation computation.

The auto-correlations of windowed speech are converted to the LP coefficients using the Levinson-Durbin algorithm. Then the LP coefficients are transformed to the Line Spectral Pair (LSP) domain for quantization and interpolation purposes. The interpolated quantified and unquantized filter coefficients are converted back to the LP filter coefficients (to construct the synthesis and weighting filters at each subframe).

10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbit/s modes

Short-term prediction, or linear prediction (LP), analysis is performed once per speech frame using the auto-correlation approach with 30 ms asymmetric windows. A lookahead of 40 samples (5 ms) is used in the auto-correlation computation.

The auto-correlations of windowed speech are converted to the LP coefficients using the Levinson-Durbin algorithm. Then the LP coefficients are transformed to the Line Spectral Pair (LSP) domain for quantization and interpolation purposes. The interpolated quantified and unquantized filter coefficients are converted back to the LP filter coefficients (to construct the synthesis and weighting filters at each subframe).

5.2.1 Windowing and auto-correlation computation

12.2 kbit/s mode

LP analysis is performed twice per frame using two different asymmetric windows. The first window has its weight concentrated at the second subframe and it consists of two halves of Hamming windows with different sizes. The window is given by:

$$w_I(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{pn}{L_1^{(I)} - 1}\right), & n = 0, \ldots, L_1^{(I)} - 1, \\ 0.54 + 0.46\cos\left(\dfrac{p(n - L_1^{(I)})}{L_2^{(I)} - 1}\right), & n = L_1^{(I)}, \ldots, L_1^{(I)} + L_2^{(I)} - 1. \end{cases} \quad (5)$$

The values $L_1^{(I)} = 160$ and $L_2^{(I)} = 80$ are used. The second window has its weight concentrated at the fourth subframe and it consists of two parts: the first part is half a Hamming window and the second part is a quarter of a cosine function cycle. The window is given by:

$$w_{II}(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2pn}{2L_1^{(II)} - 1}\right), & n = 0, \ldots, L_1^{(II)} - 1, \\ \cos\left(\dfrac{2p(n - L_1^{(II)})}{4L_2^{(II)} - 1}\right), & n = L_1^{(II)}, \ldots, L_1^{(II)} + L_2^{(II)} - 1 \end{cases} \quad (6)$$

where the values $L_1^{(II)} = 232$ and $L_2^{(II)} = 8$ are used.

Note that both LP analyses are performed on the same set of speech samples. The windows are applied to 80 samples from past speech frame in addition to the 160 samples of the present speech frame. No samples from future frames are used (no lookahead). A diagram of the two LP analysis windows is depicted below.

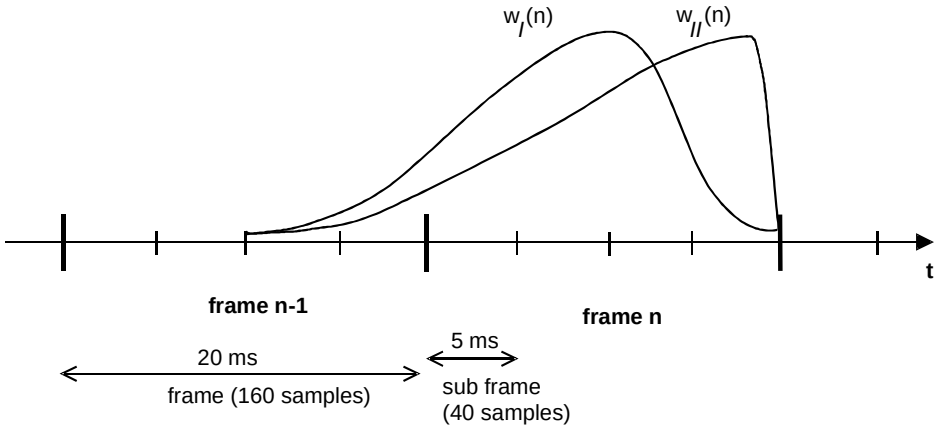


Figure 1: LP analysis windows

The auto-correlations of the windowed speech $s'(n), n = 0,\ldots 239$, are computed by:

$$r_{ac}(k) = \sum_{n=k}^{239} s'(n)s'(n-k) , \qquad k = 0,\ldots,10, \tag{7}$$

and a 60 Hz bandwidth expansion is used by lag windowing the auto-correlations using the window:

$$w_{lag}(i) = \exp\left[-\frac{1}{2}\left(\frac{2pf_0 i}{f_s}\right)^2\right], \quad i = 1,\ldots 10, \tag{8}$$

where $f_0 = 60$ Hz is the bandwidth expansion and $f_s = 8000$ Hz is the sampling frequency. Further, $r_{ac}(0)$ is multiplied by the white noise correction factor 1.0001 which is equivalent to adding a noise floor at -40 dB.

10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbit/s modes

LP analysis is performed once per frame using an asymmetric window. The window has its weight concentrated at the fourth subframe and it consists of two parts: the first part is half a Hamming window and the second part is a quarter of a cosine function cycle. The window is given by equation (6) where the values $L_1 = 200$ and $L_2 = 40$ are used.

The auto-correlations of the windowed speech $s'(n), n = 0,\ldots 239$, are computed by equation (7) and a 60 Hz bandwidth expansion is used by lag windowing the auto-correlations using the window of equation (8). Further, $r_{ac}(0)$ is multiplied by the white noise correction factor 1.0001 which is equivalent to adding a noise floor at -40 dB.

5.2.2 Levinson-Durbin algorithm (all modes)

The modified auto-correlations $r'_{ac}(0) = 1.0001\, r_{ac}(0)$ and $r'_{ac}(k) = r_{ac}(k)w_{lag}(k),\ k = 1,\ldots 10,$ are used to obtain the direct form LP filter coefficients $a_k, k = 1,\ldots,10,$ by solving the set of equations.

$$\sum_{k=1}^{10} a_k r'_{ac}(|i-k|) = -r'_{ac}(i), \qquad i = 1,\ldots,10. \tag{9}$$

The set of equations in (9) is solved using the Levinson-Durbin algorithm. This algorithm uses the following recursion:

$E_{LD}(0) = r_{ac}'(0)$
for $i = 1$ to 10 do
  $a_0^{(i-1)} = 1$
  $k_i = -\left[\sum_{j=0}^{i-1} a_j^{(i-1)} r_{ac}'(i-j)\right] / E_{LD}(i-1)$
  $a_i^{(i)} = k_i$
  for $j = 1$ to $i-1$ do
    $a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}$
  end
  $E_{LD}(i) = (1-k_i^2) E_{LD}(i-1)$
end

The final solution is given as $a_j = a_j^{(10)}, j = 1,\ldots,10$.

The LP filter coefficients are converted to the line spectral pair (LSP) representation for quantization and interpolation purposes. The conversions to the LSP domain and back to the LP filter coefficient domain are described in the next clause.

5.2.3 LP to LSP conversion (all modes)

The LP filter coefficients $a_k, k = 1,\ldots,10$, are converted to the line spectral pair (LSP) representation for quantization and interpolation purposes. For a 10th order LP filter, the LSPs are defined as the roots of the sum and difference polynomials:

$$F_1'(z) = A(z) + z^{-11} A\left(z^{-1}\right) \tag{10}$$

and

$$F_2'(z) = A(z) - z^{-11} A\left(z^{-1}\right), \tag{11}$$

respectively. The polynomial $F_1'(z)$ and $F_2'(z)$ are symmetric and anti-symmetric, respectively. It can be proven that all roots of these polynomials are on the unit circle and they alternate each other. $F_1'(z)$ has a root $z = -1$ ($w = p$) and $F_2'(z)$ has a root $z = 1$ ($w = 0$). To eliminate these two roots, we define the new polynomials:

$$F_1(z) = F_1'(z) \Big/ \left(1 + z^{-1}\right) \tag{12}$$

and

$$F_2(z) = F_2'(z) \Big/ \left(1 - z^{-1}\right) \tag{13}$$

Each polynomial has 5 conjugate roots on the unit circle $\left(e^{\pm j\omega_i}\right)$, therefore, the polynomials can be written as

$$F_1(z) = \prod_{i=1,3,..,9} \left(1 - 2q_i z^{-1} + z^{-2}\right) \tag{14}$$

and

$$F_2(z) = \prod_{i=2,4,\ldots,10} \left(1 - 2q_i z^{-1} + z^{-2}\right), \tag{15}$$

where $q_i = \cos(\omega_i)$ with w_i being the line spectral frequencies (LSF) and they satisfy the ordering property $0 < \omega_1 < \omega_2 < \ldots < \omega_{10} < \pi$. We refer to q_i as the LSPs in the cosine domain.

Since both polynomials $F_1(z)$ and $F_2(z)$ are symmetric only the first 5 coefficients of each polynomial need to be computed. The coefficients of these polynomials are found by the recursive relations (for $i = 0$ to 4):

$$\begin{aligned} f_1(i+1) &= a_{i+1} + a_{m-i} - f_1(i) \\ f_2(i+1) &= a_{i+1} - a_{m-i} + f_2(i) \end{aligned} \tag{16}$$

where $m = 10$ is the predictor order.

The LSPs are found by evaluating the polynomials $F_1(z)$ and $F_2(z)$ at 60 points equally spaced between 0 and p and checking for sign changes. A sign change signifies the existence of a root and the sign change interval is then divided 4 times to better track the root. The Chebyshev polynomials are used to evaluate $F_1(z)$ and $F_2(z)$. In this method the roots are found directly in the cosine domain $\{q_i\}$. The polynomials $F_1(z)$ or $F_2(z)$ evaluated at $z = e^{jw}$ can be written as:

$$F(w) = 2e^{-j5w} C(x),$$

with:

$$C(x) = T_5(x) + f(1)T_4(x) + f(2)T_3(x) + f(3)T_2(x) + f(4)T_1(x) + f(5)/2, \tag{17}$$

where $T_m(x) = \cos(mw)$ is the mth order Chebyshev polynomial, and $f(i), i = 1,\ldots,5$ are the coefficients of either $F_1(z)$ or $F_2(z)$, computed using the equations in (16). The polynomial $C(x)$ is evaluated at a certain value of $x = \cos(w)$ using the recursive relation:

for $k = 4$ down to 1
  $l_k = 2x l_{k+1} - l_{k+2} + f(5-k)$
end
$C(x) = x l_1 - l_2 + f(5)/2,$

with initial values $l_5 = 1$ and $l_6 = 0.$ The details of the Chebyshev polynomial evaluation method are found in P. Kabal and R.P. Ramachandran [4].

5.2.4 LSP to LP conversion (all modes)

Once the LSPs are quantified and interpolated, they are converted back to the LP coefficient domain $\{a_k\}$. The conversion to the LP domain is done as follows. The coefficients of $F_1(z)$ or $F_2(z)$ are found by expanding equations (14) and (15) knowing the quantified and interpolated LSPs q_i, $i = 1,\ldots,10$. The following recursive relation is used to compute $f_1(i)$:

for $i = 1$ to 5
  $f_1(i) = -2q_{2i-1} f_1(i-1) + 2 f_1(i-2)$
  for $j = i - 1$ down to 1
    $f_1(j) = f_1(j) - 2q_{2i-1} f_1(j-1) + f_1(j-2)$
  end
end

with initial values $f_1(0)=1$ and $f_1(-1)=0$. The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-1} by q_{2i}.

Once the coefficients $f_1(i)$ and $f_2(i)$ are found, $F_1(z)$ and $F_2(z)$ are multiplied by $1+z^{-1}$ and $1-z^{-1}$, respectively, to obtain $F_1'(z)$ and $F_2'(z)$; that is:

$$\begin{aligned} f_1(i) &= f_1(i) + f\ (i \quad 1), \quad i = 1,\ldots,5 \\ f_2'(i) &\quad f_2(i) - f\ (i-1), \quad i \quad 1,\ldots,5 \end{aligned} \tag{18}$$

Finally the LP coefficients are found by:

$$a_i = \begin{cases} .\quad '(i) \quad . f_2'(\ ), & i = \ ,\ldots, \\ . f_1(\quad - \ ) - . f\ (\quad - i), & = \ ,\ldots, \end{cases} \tag{19}$$

This is directly derived from the relation $A(\ ) = \left(F_1'(z) + \quad '(z)\right)/2$, and considering the fact that $F_1'(z)$ and $F_2'(z)$ are symmetric and anti-symmetric polynomials, respectively.

Quantization of the LSP coefficients

12.2 kbit/s mode

domain; that is:

$$f \quad \frac{}{2\pi} arccos(q\ ), \qquad i=1,\ldots,10, \tag{20}$$

where f_i are the line spectral frequencies (LSF) in Hz [0,4000] and $f_s = 8000$ is the sampling frequency. The LSF vector is given by $\mathbf{f}^t = [\, f_1\ f_2 \ldots f_{10}\,]$, with t denoting transpose.

A 1st order MA prediction is applied, and the two residual LSF vectors are jointly quantified using split matrix quantization (SMQ). The prediction and quantization are performed as follows. Let $\mathbf{z}^{(1)}(n)$ and $\mathbf{z}^{(2)}(n)$ denote the mean-removed LSF vectors at frame n. The prediction residual vectors $\mathbf{r}^{(1)}(n)$ and $\mathbf{r}^{(2)}(n)$ are given by:

$$\begin{aligned} \mathbf{r}^{(1)}(n) &= \mathbf{z}^{(1)}(n) - \mathbf{p}(n), \quad \text{and} \\ \mathbf{r}^{(2)}(n) &= \mathbf{z}^{(2)}(n) - \mathbf{p}(n), \end{aligned} \tag{21}$$

where $\mathbf{p}(n)$ is the predicted LSF vector at frame n. First order moving-average (MA) prediction is used where:

$$\mathbf{p}(n) = 0.65\hat{\mathbf{r}}^{(2)}(n-1), \tag{22}$$

where $\hat{\mathbf{r}}^{(2)}(n-1)$ is the quantified second residual vector at the past frame.

The two LSF residual vectors $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ are jointly quantified using split matrix quantization (SMQ). The matrix $\left(\mathbf{r}^{(1)}\ \mathbf{r}^{(2)}\right)$ is split into 5 submatrices of dimension 2 x 2 (two elements from each vector). For example, the first submatrix consists of the elements $r_1^{(1)}$, $r_2^{(1)}$, $r_1^{(2)}$, and $r_2^{(2)}$. The 5 submatrices are quantified with 7, 8, 8+1, 8, and 6 bits, respectively. The third submatrix uses a 256-entry signed codebook (8-bit index plus 1-bit sign).

A weighted LSP distortion measure is used in the quantization process. In general, for an input LSP vector $\mathbf{f}$ and a quantified vector at index k, $\hat{\mathbf{f}}^k$, the quantization is performed by finding the index k which minimizes:

$$E_{LSP} = \sum_{i=1}^{10} \left[f_i w_i - \hat{f}_i^k w_i \right]^2 . \tag{23}$$

The weighting factors $w_i, i=1,\ldots,10$, are given by

$$\begin{aligned} w_i &= 3.347 - \frac{1.547}{450} d_i \quad \text{for} \quad d_i < 450, \\ &= 1.8 - \frac{0.8}{1050}(d_i - 450) \quad \text{otherwise,} \end{aligned} \tag{24}$$

where $d_i = f_{i+1} - f_{i-1}$ with $f_0 = 0$ and $f_{11} = 4000$. Here, two sets of weighting coefficients are computed for the two LSF vectors. In the quantization of each submatrix, two weighting coefficients from each set are used with their corresponding LSFs.

10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbit/s modes

The set of LP filter coefficients per frame is quantified using the LSP representation in the frequency domain using equation (20).

A 1st order MA prediction is applied, and the residual LSF vector is quantified using split vector quantization. The prediction and quantization are performed as follows. Let $\mathbf{z}(n)$ denote the mean-removed LSF vectors at frame n. The prediction residual vectors $\mathbf{r}(n)$ is given by:

$$\mathbf{r}(n) = \mathbf{z}(n) - \mathbf{p}(n) \tag{25}$$

where $\mathbf{p}(n)$ is the predicted LSF vector at frame n. First order moving-average (MA) prediction is used where:

$$p_j(n) = a_j \hat{r}_j(n-1) \quad j = 1,\ldots,10 , \tag{26}$$

where $\hat{\mathbf{r}}(n-1)$ is the quantified residual vector at the past frame and a_j is the prediction factor for the jth LSF.

The LSF residual vectors $\mathbf{r}$ is quantified using split vector quantization. The vector $\mathbf{r}$ is split into 3 subvectors of dimension 3, 3, and 4. The 3 subvectors are quantified with 7-9 bits according to table 2.

Table 2. Bit allocation split vector quantization of LSF residual vector.

Mode	Subvector 1	Subvector 2	Subvector 3
10.2 kbit/s	8	9	9
7.95 kbit/s	9	9	9
7.40 kbit/s	8	9	9
6.70 kbit/s	8	9	9
5.90 kbit/s	8	9	9
5.15 kbit/s	8	8	7
4.75 kbit/s	8	8	7

The weighted LSP distortion measure of equation (23) with the weighting of equation (24) is used in the quantization process.

5.2.6 Interpolation of the LSPs

12.2 kbit/s mode

The two sets of quantified (and unquantized) LP parameters are used for the second and fourth subframes whereas the first and third subframes use a linear interpolation of the parameters in the adjacent subframes. The interpolation is performed on the LSPs in the $\mathbf{q}$ domain. Let $\hat{\mathbf{q}}_4^{(n)}$ be the LSP vector at the 4th subframe of the present frame n, $\hat{\mathbf{q}}_2^{(n)}$ be the LSP vector at the 2nd subframe of the present frame n, and $\hat{\mathbf{q}}_4^{(n-1)}$ the LSP vector at the 4th subframe of the past frame $n-1$. The interpolated LSP vectors at the 1st and 3rd subframes are given by:

$$\begin{aligned} \hat{\mathbf{q}}_1^{(n)} &= 0.5\hat{\mathbf{q}}_4^{(n-1)} + 0.5\hat{\mathbf{q}}_2^{(n)}, \\ \hat{\mathbf{q}}_3^{(n)} &= 0.5\hat{\mathbf{q}}_2^{(n)} + 0.5\hat{\mathbf{q}}_4^{(n)}. \end{aligned} \tag{27}$$

The interpolated LSP vectors are used to compute a different LP filter at each subframe (both quantified and unquantized coefficients) using the LSP to LP conversion method described in subclause 5.2.4.

10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbit/s modes

The set of quantified (and unquantized) LP parameters is used for the fourth subframe whereas the first, second, and third subframes use a linear interpolation of the parameters in the adjacent subframes. The interpolation is performed on the LSPs in the $\mathbf{q}$ domain. The interpolated LSP vectors at the 1st, 2nd, and 3rd subframes are given by:

$$\begin{aligned} \hat{\mathbf{q}}_1^{(n)} &= 0.75\hat{\mathbf{q}}_4^{(n-1)} + 0.25\hat{\mathbf{q}}_4^{(n)}, \\ \hat{\mathbf{q}}_2^{(n)} &= 0.5\hat{\mathbf{q}}_4^{(n-1)} + 0.5\hat{\mathbf{q}}_4^{(n)}, \\ \hat{\mathbf{q}}_2^{(n)} &= 0.25\hat{\mathbf{q}}_4^{(n-1)} + 0.75\hat{\mathbf{q}}_4^{(n)}. \end{aligned} \tag{28}$$

The interpolated LSP vectors are used to compute a different LP filter at each subframe (both quantified and unquantized coefficients) using the LSP to LP conversion method described in subclause 5.2.4.

5.3 Open-loop pitch analysis

Open-loop pitch analysis is performed in order to simplify the pitch analysis and confine the closed-loop pitch search to a small number of lags around the open-loop estimated lags.

Open-loop pitch estimation is based on the weighted speech signal $s_w(n)$ which is obtained by filtering the input speech signal through the weighting filter $W(z) = A(z/g_1)/A(z/g_2)$. That is, in a subframe of size L, the weighted speech is given by:

$$s_w(n) = s(n) + \sum_{i=1}^{10} a_i g_1^i s(n-i) - \sum_{i=1}^{10} a_i g_2^i s_w(n-i), \quad n = 0,\ldots,L-1 \tag{29}$$

12.2 kbit/s mode

Open-loop pitch analysis is performed twice per frame (each 10 ms) to find two estimates of the pitch lag in each frame.

Open-loop pitch analysis is performed as follows. In the first step, 3 maxima of the correlation:

$$O_k = \sum_{n=0}^{79} s_w(n) s_w(n-k) \tag{30}$$

are found in the three ranges:

$i = 3$: $18,\ldots,35,$
$i = 2$: $36,\ldots,71,$
$i = 1$: $72,\ldots,143.$

The retained maxima $O_{t_i}, i=1,\ldots,3$, are normalized by dividing by $\sqrt{\sum_n s_w^2(n-t_i)}, i=1,\ldots,3$, respectively. The normalized maxima and corresponding delays are denoted by $(M_i, t_i), i=1,\ldots,3$. The winner, T_{op}, among the three normalized correlations is selected by favouring the delays with the values in the lower range. This is performed by weighting the normalized correlations corresponding to the longer delays. The best open-loop delay T_{op} is determined as follows:

$T_{op} = t_1$
$M(T_{op}) = M_1$
if $M_2 > 0.85 M(T_{op})$
  $M(T_{op}) = M_2$
  $T_{op} = t_2$
end
if $M_3 > 0.85 M(T_{op})$
  $M(T_{op}) = M_3$
  $T_{op} = t_3$
end

This procedure of dividing the delay range into 3 clauses and favouring the lower clauses is used to avoid choosing pitch multiples.

10.2 kbit/s mode

Open-loop pitch analysis is performed twice per frame (every 10 ms) to find two estimates of the pitch lag in each frame.

The open-loop pitch analysis is performed as follows. First, the correlation of weighted speech is determined for each pitch lag value d by:

$$C(d) = \sum_{n=0}^{79} s_w(n)s_w(n-d)w(d), \quad d = 20,\ldots,143, \tag{31}$$

where $w(d)$ is a weighting function. The estimated pitch-lag is the delay that maximises the weighted correlation function $C(d)$. The weighting emphasises lower pitch lag values reducing the likelihood of selecting a multiple of the correct delay. The weighting function consists of two parts: a low pitch lag emphasis function, $w_l(d)$, and a previous frame lag neighbouring emphasis function, $w_n(d)$:

$$w(d) = w_l(d)w_n(d). \tag{32}$$

The low pitch lag emphasis function is a given by:

$$w_l(d) = cw(d) \tag{33}$$

where $cw(d)$ is defined by a table in the fixed point computational computational description (ANSI-C code) in [4]. The previous frame lag neighbouring emphasis function depends on the pitch lag of previous speech frames:

$$w_n(d) = \begin{cases} cw\left(\left|T_{old} - d\right| + d_L\right), & v > 0.3, \\ 1.0, & \text{otherwise}, \end{cases} \tag{34}$$

where $d_L = 20$, T_{old} is the median filtered pitch lag of 5 previous voiced speech half-frames, and v is an adaptive parameter. If the frame is classified as voiced by having the open-loop gain $g > 0.4$, the v-value is set to 1.0 for the next frame. Otherwise, the v-value is updated by $v = 0.9v$. The open loop gain is given by:

$$g = \frac{\sum_{n=0}^{79} s_w(n)s_w\left(n - d_{\max}\right)}{\sum_{n=0}^{79} s_w^2(n)} \tag{35}$$

where $d_{\max}$ is the pitch delay that maximizes $C(d)$. The median filter is updated only during voiced speech frames. The weighting depends on the reliability of the old pitch lags. If previous frames have contained unvoiced speech or silence, the weighting is attenuated through the parameter v.

7.95, 7.40, 6.70, 5.90 kbit/s modes

Open-loop pitch analysis is performed twice per frame (each 10 ms) to find two estimates of the pitch lag in each frame.

Open-loop pitch analysis is performed as follows. In the first step, 3 maxima of the correlation in equation (30) are found in the three ranges:

$i = 3$: $20,\ldots,39,$
$i = 2$: $40,\ldots,79,$
$i = 1$: $80,\ldots,143.$

The retained maxima $O_{t_i}, i=1,\ldots,3$, are normalized by dividing by $\sqrt{\sum_n s_w^2(n-t_i)}, i=1,\ldots,3$, respectively. The normalized maxima and corresponding delays are denoted by $(M_i,t_i), i=1,\ldots,3$. The winner, T_{op}, among the three normalized correlations is selected by favouring the delays with the values in the lower range. This is performed by weighting the normalized correlations corresponding to the longer delays. The best open-loop delay T_{op} is determined as follows:

$T_{op} = t_1$
$M(T_{op}) = M_1$
if $M_2 > 0.85M(T_{op})$
$\quad M(T_{op}) = M_2$
$\quad T_{op} = t_2$
end
if $M_3 > 0.85M(T_{op})$
$\quad M(T_{op}) = M_3$
$\quad T_{op} = t_3$
end

This procedure of dividing the delay range into 3 clauses and favouring the lower clauses is used to avoid choosing pitch multiples.

5.15, 4.75 kbit/s modes

Open-loop pitch analysis is performed once per frame (each 20 ms) to find an estimate of the pitch lag in each frame.

Open-loop pitch analysis is performed as follows. In the first step, 3 maxima of the correlation in equation (30) are found in the three ranges:

$i=3$: 20,…,39,
$i=2$: 40,…,79,
$i=1$: 79,…,143.

The retained maxima $O_{t_i}, i=1,\ldots,3$, are normalized by dividing by $\sqrt{\sum_n s_w^2(n-t_i)}, i=1,\ldots,3$, respectively. The normalized maxima and corresponding delays are denoted by $(M_i,t_i), i=1,\ldots,3$. The winner, T_{op}, among the three normalized correlations is selected by favouring the delays with the values in the lower range. This is performed by weighting the normalized correlations corresponding to the longer delays. The best open-loop delay T_{op} is determined as follows:

$T_{op} = t_1$
$M(T_{op}) = M_1$
if $M_2 > 0.85 M(T_{op})$
$M(T_{op}) = M_2$
$T_{op} = t_2$
end
if $M_3 > 0.85 M(T_{op})$
$M(T_{op}) = M_3$
$T_{op} = t_3$
end

This procedure of dividing the delay range into 3 clauses and favouring the lower clauses is used to avoid choosing pitch multiples.

5.4 Impulse response computation (all modes)

The impulse response, $h(n)$, of the weighted synthesis filter $H(z)W(z) = A(z/g_1)/[\hat{A}(z)A(z/g_2)]$ is computed each subframe. This impulse response is needed for the search of adaptive and fixed codebooks. The impulse response $h(n)$ is computed by filtering the vector of coefficients of the filter $A(z/g_1)$ extended by zeros through the two filters $1/\hat{A}(z)$ and $1/A(z/g_2)$.

5.5 Target signal computation (all modes)

The target signal for adaptive codebook search is usually computed by subtracting the zero input response of the weighted synthesis filter $H(z)W(z) = A(z/g_1)/[\hat{A}(z)A(z/g_2)]$ from the weighted speech signal $s_w(n)$. This is performed on a subframe basis.

An equivalent procedure for computing the target signal, which is used in this standard, is the filtering of the LP residual signal $res_{LP}(n)$ through the combination of synthesis filter $1/\hat{A}(z)$ and the weighting filter $A(z/g_1)/A(z/g_2)$. After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and excitation. The memory update of these filters is explained in subclause 5.9.

The residual signal $res_{LP}(n)$ which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 40 as will be explained in the next clause. The LP residual is given by:

$$res_{LP}(n) = s(n) + \sum_{i=1}^{10} \hat{a}_i s(n-i). \qquad (36)$$

5.6 Adaptive codebook search

Adaptive codebook search is performed on a subframe basis. It consists of performing closed-loop pitch search, and then computing the adaptive codevector by interpolating the past excitation at the selected fractional pitch lag.

The adaptive codebook parameters (or pitch parameters) are the delay and gain of the pitch filter. In the adaptive codebook approach for implementing the pitch filter, the excitation is repeated for delays less than the subframe length. In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search.

12.2 kbit/s mode

In the first and third subframes, a fractional pitch delay is used with resolutions: 1/6 in the range $[17\,3/6, 94\,3/6]$ and integers only in the range [95, 143]. For the second and fourth subframes, a pitch resolution of 1/6 is always used in the range $[T_1 - 5\,3/6, T_1 + 4\,3/6]$, where T_1 is nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe, bounded by 18...143.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first (and third) subframe the range $T_{op} \pm 3$, bounded by 18...143, is searched. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. The pitch delay is encoded with 9 bits in the first and third subframes and the relative delay of the other subframes is encoded with 6 bits.

The closed-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term:

$$R(k) = \frac{\sum_{n=0}^{39} x(n) y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n) y_k(n)}}, \qquad (37)$$

where $x(n)$ is the target signal and $y_k(n)$ is the past filtered excitation at delay k (past excitation convolved with $h(n)$). Note that the search range is limited around the open-loop pitch as explained earlier.

The convolution $y_k(n)$ is computed for the first delay $t_{\min}$ in the searched range, and for the other delays in the search range $k = t_{\min} + 1, \ldots, t_{\max}$, it is updated using the recursive relation:

$$y_k(n) = y_{k-1}(n-1) + u(-k)h(n), \qquad (38)$$

where $u(n), n = -(143+11), \ldots, 39$, is the excitation buffer. Note that in search stage, the samples $u(n), n = 0, \ldots, 39$, are not known, and they are needed for pitch delays less than 40. To simplify the search, the LP residual is copied to $u(n)$ in order to make the relation in equation (38) valid for all delays.

Once the optimum integer pitch delay is determined, the fractions from –3/6 to 3/6 with a step of 1/6 around that integer are tested. The fractional pitch search is performed by interpolating the normalized correlation in equation (37) and searching for its maximum. The interpolation is performed using an FIR filter b_{24} based on a Hamming windowed $\sin(x)/x$ function truncated at ± 23 and padded with zeros at ± 24 ($b_{24}(24) = 0$). The filter has its cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain. The interpolated values of $R(k)$ for the fractions –3/6 to 3/6 are obtained using the interpolation formula:

$$R(k)_t = \sum_{i=0}^{3} R(k-i)\, b_{24}(t + i \cdot 6) + \sum_{i=0}^{3} R(k+1+i)\, b_{24}(6 - t + i \cdot 6), \quad t = 0, \ldots, 5, \qquad (39)$$

where $t = 0, \ldots, 5$ corresponds to the fractions 0, 1/6, 2/6, 3/6, -2/6, and –1/6, respectively. Note that it is necessary to compute the correlation terms in equation (37) using a range $t_{\min} - 4, t_{\max} + 4,$ to allow for the proper interpolation.

Once the fractional pitch lag is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given integer delay k and phase (fraction) t :

$$v(n)=\sum_{i=0}^{9}u(n-k-i)b_{60}(t+i\cdot 6)+\sum_{i=0}^{9}u(n-k+1+i)b_{60}(6-t+i\cdot 6),\quad n=0,\ldots,39,\ t=0,\ldots,5. \quad (40)$$

The interpolation filter b_{60} is based on a Hamming windowed $\sin(x)/x$ function truncated at ± 59 and padded with zeros at ± 60 ($b_{60}(60)=0$). The filter has a cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain.

The adaptive codebook gain is then found by:

$$g_p=\frac{\sum_{n=0}^{39}x(n)y(n)}{\sum_{n=0}^{39}y(n)y(n)},\quad \text{bounded by}\quad 0\le g_p\le 1.2 \quad (41)$$

where $y(n)=v(n)*h(n)$ is the filtered adaptive codebook vector (zero state response of $H(z)W(z)$ to $v(n)$).

The computed adaptive codebook gain is quantified using 4-bit non-uniform scalar quantization in the range [0.0,1.2].

7.95 kbit/s mode

In the first and third subframes, a fractional pitch delay is used with resolutions: 1/3 in the range $[19\,1/3, 84\,2/3]$ and integers only in the range [85, 143]. For the second and fourth subframes, a pitch resolution of 1/3 is always used in the range $[T_1-10\,2/3, T_1+9\,2/3]$, where T_1 is nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe, bounded by 20...143.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first (and third) subframe the range $T_{op}\pm 3$, bounded by 20...143, is searched. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. The pitch delay is encoded with 8 bits in the first and third subframes and the relative delay of the other subframes is encoded with 6 bits.

The closed-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term of equation (37). Note that the search range is limited around the open-loop pitch as explained earlier.

The convolution $y_k(n)$ is computed for the first delay $t_{\min}$ in the searched range, and for the other delays in the search range $k=t_{\min}+1,\ldots,t_{\max}$, it is updated using the recursive relation of equation (38).

Once the optimum integer pitch delay is determined, the fractions from –2/3 to 2/3 with a step of 1/3 around that integer are tested. The fractional pitch search is performed by interpolatingthe normalized correlation in equation (37) and searching for its maximum. Once the fractional pitch lag is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given integer delay and phase (fraction). The interpolation is performed using two FIR filters (Hamming windowed sinc functions); one for interpolating the term in equation (37) with the sinc truncated at ± 11 and the other for interpolating the past excitation with the sinc truncated at ± 29. The filters have their cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain.

The adaptive codebook gain is then found as in equation (41).

The computed adaptive codebook gain is quantified using 4-bit non-uniform scalar quantization as described in section 5.8.

10.2, 7.40 kbit/s mode

In the first and third subframes, a fractional pitch delay is used with resolutions: 1/3 in the range $\left[19\,1/3, 84\,2/3\right]$ and integers only in the range [85, 143]. For the second and fourth subframes, a pitch resolution of 1/3 is always used in the range $\left[T_1 - 5\,2/3, T_1 + 4\,2/3\right]$, where T_1 is nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe, bounded by 20...143.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first (and third) subframe the range $T_{op} \pm 3$, bounded by 20...143, is searched. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. The pitch delay is encoded with 8 bits in the first and third subframes and the relative delay of the other subframes is encoded with 5 bits.

The closed-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term of equation (37). Note that the search range is limited around the open-loop pitch as explained earlier.

The convolution $y_k(n)$ is computed for the first delay $t_{\min}$ in the searched range, and for the other delays in the search range $k = t_{\min} + 1, \ldots, t_{\max}$, it is updated using the recursive relation of equation (38).

Once the optimum integer pitch delay is determined, the fractions from –2/3 to 2/3 with a step of 1/3 around that integer are tested. The fractional pitch search is performed by interpolatingthe normalized correlation in equation (37) and searching for its maximum. Once the fractional pitch lag is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given integer delay and phase (fraction). The interpolation is performed using two FIR filters (Hamming windowed sinc functions); one for interpolating the term in equation (37) with the sinc truncated at ± 11 and the other for interpolating the past excitation with the sinc truncated at ± 29. The filters have their cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain.

The adaptive codebook gain is then found as in equation (41).

The computed adaptive codebook gain (and the fixed codebook gain) is quantified using 7-bit non-uniform vector quantization as described in section 5.8.

6.70, 5.90 kbit/s modes

In the first and third subframes, a fractional pitch delay is used with resolutions: 1/3 in the range $\left[19\,1/3, 84\,2/3\right]$ and integers only in the range [85, 143]. For the second and fourth subframes, integer pitch resolution is used in the range $\left[T_1 - 5, T_1 + 4\right]$, where T_1 is nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe, bounded by 20...143. Additionally, a fractional resolution of 1/3 is used in the range $\left[T_1 - 1\,2/3, T_1 + 2/3\right]$.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first (and third) subframe the range $T_{op} \pm 3$, bounded by 20...143, is searched. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. The pitch delay is encoded with 8 bits in the first and third subframes and the relative delay of the other subframes is encoded with 4 bits.

The closed-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term of equation (37). Note that the search range is limited around the open-loop pitch as explained earlier.

The convolution $y_k(n)$ is computed for the first delay $t_{\min}$ in the searched range, and for the other delays in the search range $k = t_{\min} + 1, \ldots, t_{\max}$, it is updated using the recursive relation of equation (38).

Once the optimum integer pitch delay is determined, the fractions from –2/3 to 2/3 with a step of 1/3 around that integer are tested. The fractional pitch search is performed by interpolatingthe normalized correlation in equation (37) and searching for its maximum. Once the fractional pitch lag is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given integer delay and phase (fraction). The interpolation is performed using two FIR filters (Hamming windowed sinc functions); one for interpolating the term in equation (37) with the sinc truncated at ± 11 and the other for interpolating the past excitation with the sinc truncated at ± 29. The filters have their cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain.

The adaptive codebook gain is then found as in equation (41).

The computed adaptive codebook gain (and the fixed codebook gain) is quantified using vector quantization as described in section 5.8.

5.15, 4.75 kbit/s modes

In the first subframe, a fractional pitch delay is used with resolutions: 1/3 in the range $\left[19\,1/3, 84\,2/3\right]$ and integers only in the range [85, 143]. For the second, third, and fourth subframes, integer pitch resolution is used in the range $\left[T_1 - 5, T_1 + 4\right]$, where T_1 is nearest integer to the fractional pitch lag of the previous subframe, bounded by 20...143. Additionally, a fractional resolution of 1/3 is used in the range $\left[T_1 - 1\,2/3, T_1 + 2/3\right]$.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first subframe the range $T_{op} \pm 5$, bounded by 20...143, is searched. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. The pitch delay is encoded with 8 bits in the first subframe and the relative delay of the other subframes is encoded with 4 bits.

The closed-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term of equation (37). Note that the search range is limited around the open-loop pitch as explained earlier.

The convolution $y_k(n)$ is computed for the first delay $t_{\min}$ in the searched range, and for the other delays in the search range $k = t_{\min} + 1, \ldots, t_{\max}$, it is updated using the recursive relation of equation (38).

Once the optimum integer pitch delay is determined, the fractions from –2/3 to 2/3 with a step of 1/3 around that integer are tested. The fractional pitch search is performed by interpolatingthe normalized correlation in equation (37) and searching for its maximum. Once the fractional pitch lag is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given integer delay and phase (fraction). The interpolation is performed using two FIR filters (Hamming windowed sinc functions); one for interpolating the term in equation (37) with the sinc truncated at ± 11 and the other for interpolating the past excitation with the sinc truncated at ± 29. The filters have their cut-off frequency (-3 dB) at 3 600 Hz in the over-sampled domain.

The adaptive codebook gain is then found as in equation (41).

The computed adaptive codebook gain (and the fixed codebook gain) is quantified using vector quantization as described in section 5.8.

5.7 Algebraic codebook

5.7.1 Algebraic codebook structure

The algebraic codebook structure is based on interleaved single-pulse permutation (ISPP) design.

12.2 kbit/s mode

In this codebook, the innovation vector contains 10 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 5 tracks, where each track contains two pulses, as shown in table 3.

Table 3: Potential positions of individual pulses in the algebraic codebook, 12.2 kbit/s.

Track	Pulse	Positions
1	i_0, i_5	0, 5, 10, 15, 20, 25, 30, 35
2	i_1, i_6	1, 6, 11, 16, 21, 26, 31, 36
3	i_2, i_7	2, 7, 12, 17, 22, 27, 32, 37
4	i_3, i_8	3, 8, 13, 18, 23, 28, 33, 38
5	i_4, i_9	4, 9, 14, 19, 24, 29, 34, 39

Each two pulse positions in one track are encoded with 6 bits (total of 30 bits, 3 bits for the position of every pulse), and the sign of the first pulse in the track is encoded with 1 bit (total of 5 bits).

For two pulses located in the same track, only one sign bit is needed. This sign bit indicates the sign of the first pulse. The sign of the second pulse depends on its position relative to the first pulse. If the position of the second pulse is smaller, then it has opposite sign, otherwise it has the same sign than in the first pulse.

All the 3-bit pulse positions are Gray coded in order to improve robustness against channel errors. This gives a total of 35 bits for the algebraic code.

10.2 kbit/s mode

In this codebook, the innovation vector contains 8 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 4 tracks, where each track contains two pulses, as shown in table 4.

Table 4: Potential positions of individual pulses in the algebraic codebook, 10.2 kbit/s.

Track	Pulse	Positions
1	i_0, i_4	0, 4, 8, 12, 16, 20, 24, 28, 32, 36
2	i_1, i_5	1, 5, 9, 13, 17, 21, 25, 29, 33, 37
3	i_2, i_6	2, 6, 10, 14, 18, 22, 26, 30, 34, 38
4	i_3, i_7	3, 7, 11, 15, 19, 23, 27, 31, 35, 39

The pulses are grouped into 3, 3, and 2 pulses and their positions are encoded with 10, 10, and 7 bits, respectively (total of 27 bits). The sign of the first pulse in each track is encoded with 1 bit (total of 4 bits).

For two pulses located in the same track, only one sign bit is needed. This sign bit indicates the sign of the first pulse. The sign of the second pulse depends on its position relative to the first pulse. If the position of the second pulse is smaller, then it has opposite sign, otherwise it has the same sign than in the first pulse.

This gives a total of 31 bits for the algebraic code.

7.95, 7.40 kbit/s modes

In this codebook, the innovation vector contains 4 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 4 tracks, where each track contains one pulse, as shown in table 5.

Table 5: Potential positions of individual pulses in the algebraic codebook, 7.95, 7.40 kbit/s.

Track	Pulse	Positions
1	i_0	0, 5, 10, 15, 20, 25, 30, 35
2	i_1	1, 6, 11, 16, 21, 26, 31, 36
3	i_2	2, 7, 12, 17, 22, 27, 32, 37
4	i_3	3, 8, 13, 18, 23, 28, 33, 38, 4, 9, 14, 19, 24, 29, 34, 39

The pulse positions are encoded with 3, 3, 3, and 4 bits (total of 13 bits), and the sign of the each pulse is encoded with 1 bit (total of 4 bits). This gives a total of 17 bits for the algebraic code.

6.70 kbit/s mode

In this codebook, the innovation vector contains 3 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 3 tracks, where each track contains one pulse, as shown in table 6.

Table 6: Potential positions of individual pulses in the algebraic codebook, 6.70 kbit/s.

Track	Pulse	Positions
1	i_0	0, 5, 10, 15, 20, 25, 30, 35
2	i_1	1, 6, 11, 16, 21, 26, 31, 36, 3, 8, 13, 18, 23, 28, 33, 38
3	i_2	2, 7, 12, 17, 22, 27, 32, 37, 4, 9, 14, 19, 24, 29, 34, 39

The pulse positions are encoded with 3, 4, and 4 bits (total of 11 bits), and the sign of the each pulse is encoded with 1 bit (total of 3 bits). This gives a total of 14 bits for the algebraic code.

5.90 kbit/s mode

In this codebook, the innovation vector contains 2 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 2 tracks, where each track contains one pulse, as shown in table 7.

Table 7: Potential positions of individual pulses in the algebraic codebook, 5.90 kbit/s.

Track	Pulse	Positions
1	i_0	1, 6, 11, 16, 21, 26, 31, 36, 3, 8, 13, 18, 23, 28, 33, 38
2	i_1	0, 5, 10, 15, 20, 25, 30, 35, 1, 6, 11, 16, 21, 26, 31, 36, 2, 7, 12, 17, 22, 27, 32, 37, 4, 9, 14, 19, 24, 29, 34, 39

The pulse positions are encoded with 4 and 5 bits (total of 9 bits), and the sign of the each pulse is encoded with 1 bit (total of 2 bits). This gives a total of 11 bits for the algebraic code.

5.15, 4.75 kbit/s modes

In this codebook, the innovation vector contains 2 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 40 positions in a subframe are divided into 5 tracks. Two subsets of 2 tracks each are used for each subframe with one pulse in each track. Different subsets of tracks are used for each subframe. The pulse positions used in each subframe are shown in table 8.

Table 8: Potential positions of individual pulses in the algebraic codebook, 5.15, 4.75 kbit/s.

Subframe	Subset	Pulse	Positions
1	1	i_0	0, 5, 10, 15, 20, 25, 30, 35
		i_1	2, 7, 12, 17, 22, 27, 32, 37
	2	i_0	1, 6, 11, 16, 21, 26, 31, 36
		i_1	3, 8, 13, 18, 23, 28, 33, 38
2	1	i_0	0, 5, 10, 15, 20, 25, 30, 35
		i_1	3, 8, 13, 18, 23, 28, 33, 38
	2	i_0	2, 7, 12, 17, 22, 27, 32, 37
		i_1	4, 9, 14, 19, 24, 29, 34, 39
3	1	i_0	0, 5, 10, 15, 20, 25, 30, 35
		i_1	2, 7, 12, 17, 22, 27, 32, 37
	2	i_0	1, 6, 11, 16, 21, 26, 31, 36
		i_1	4, 9, 14, 19, 24, 29, 34, 39
4	1	i_0	0, 5, 10, 15, 20, 25, 30, 35
		i_1	3, 8, 13, 18, 23, 28, 33, 38
	2	i_0	1, 6, 11, 16, 21, 26, 31, 36
		i_1	4, 9, 14, 19, 24, 29, 34, 39

One bit is needed to encoded the subset used. The two pulse positions are encoded with 3 bits each (total of 6 bits), and the sign of the each pulse is encoded with 1 bit (total of 2 bits). This gives a total of 9 bits for the algebraic code.

5.7.2 Algebraic codebook search

The algebraic codebook is searched by minimizing the mean square error between the weighted input speech and the weighted synthesized speech. The target signal used in the closed-loop pitch search is updated by subtracting the adaptive codebook contribution. That is:

$$x_2(n) = x(n) - \hat{g}_p y(n), \quad n = 0,\ldots,39 \tag{42}$$

where $y(n) = v(n) * h(n)$ is the filtered adaptive codebook vector and $\hat{g}_p$ is the quantified adaptive codebook gain. If $\mathbf{c}_k$ is the algebraic codevector at index k, then the algebraic codebook is searched by maximizing the term:

$$A_k = \frac{(C_k)^2}{E_{Dk}} = \frac{\left(\mathbf{d}^t \mathbf{c}_k\right)^2}{\mathbf{c}_k^t \Phi \mathbf{c}_k}, \tag{43}$$

where $\mathbf{d} = \mathbf{H}^t \mathbf{x}_2$ is the correlation between the target signal $x_2(n)$ and the impulse response $h(n)$, $\mathbf{H}$ is a the lower triangular Toepliz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1),\ldots,h(39)$, and $\Phi = \mathbf{H}^t\mathbf{H}$ is the matrix of correlations of $h(n)$. The vector $\mathbf{d}$ (backward filtered target) and the matrix Φ are computed prior to the codebook search. The elements of the vector $\mathbf{d}$ are computed by

$$d(n) = \sum_{i=n}^{39} x_2(n) h(i-n), \quad n = 0,\ldots,39, \tag{44}$$

and the elements of the symmetric matrix Φ are computed by:

$$f(i,j) = \sum_{n=j}^{39} h(n-i)h(n-j), \quad (j \geq i). \tag{45}$$

The algebraic structure of the codebooks allows for very fast search procedures since the innovation vector $\mathbf{c}_k$ contains only a few nonzero pulses. The correlation in the numerator of Equation (43) is given by:

$$C = \sum_{i=0}^{N_p-1} J_i\, d(m_i), \tag{46}$$

where m_i is the position of the i th pulse, J_i is its amplitude, and N_p is the number of pulses ($N_p = 10$). The energy in the denominator of equation (43) is given by:

$$E_D = \sum_{i=0}^{N_p-1} f(m_i,m_i) + 2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1} J_i J_j f(m_i,m_j). \tag{47}$$

To simplify the search procedure, the pulse amplitudes are preset by the mere quantization of an appropriate signal $b(n)$. This is simply done by setting the amplitude of a pulse at a certain position equal to the sign of $b(n)$ at that position. The simplification proceeds as follows (prior to the codebook search). First, the sign signal $s_b(n) = \text{sign}[b(n)]$ and the signal $d'(n) = d(n)s_b(n)$ are computed. Second, the matrix Φ is modified by including the sign information; that is, $f'(i,j) = s_b(i)s_b(j)f(i,j)$. The correlation in equation (46) is now given by:

$$C = \sum_{i=0}^{N_p-1} d'(m_i) \tag{48}$$

and the energy in equation (47) is given by:

$$E_D = \sum_{i=0}^{N_p-1} f'(m_i,m_i) + 2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1} f'(m_i,m_j). \tag{49}$$

12.2 kbit/s mode

In this case the signal $b(n)$, used for presetting the amplitudes, is a sum of the normalized $d(n)$ vector and normalized long-term prediction residual $res_{LTP}(n)$:

$$b(n) = \frac{res_{LTP}(n)}{\sqrt{\sum_{i=0}^{39} res_{LTP}(i)\, res_{LTP}(i)}} + \frac{d(n)}{\sqrt{\sum_{i=0}^{39} d(i)\, d(i)}}, \quad n = 0,\ldots,39, \tag{50}$$

is used. Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an efficient non-exhaustive analysis-by-synthesis search technique. In this technique, the term in equation (43) is tested for a small percentage of position combinations.

First, for each of the five tracks the pulse positions with maximum absolute values of $b(n)$ are searched. From these the global maximum value for all the pulse positions is selected. The first pulse i_0 is always set into the position corresponding to the global maximum value.

Next, four iterations are carried out. During each iteration the position of pulse i_1 is set to the local maximum of one track. The rest of the pulses are searched in pairs by sequentially searching each of the pulse pairs $\{i_2,i_3\}$,

$\{i_4,i_5\}$, $\{i_6,i_7\}$ and $\{i_8,i_9\}$ in nested loops. Every pulse has 8 possible positions, i.e., there are four 8x8-loops, resulting in 256 different combinations of pulse positions for each iteration.

In each iteration all the 9 pulse starting positions are cyclically shifted, so that the pulse pairs are changed and the pulse i_1 is placed in a local maximum of a different track. The rest of the pulses are searched also for the other positions in the tracks. At least one pulse is located in a position corresponding to the global maximum and one pulse is located in a position corresponding to one of the 4 local maxima.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z)=1/(1-b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain bounded by [0.0,1.0]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by:

$$g_c = \frac{\mathbf{x}_2^t \mathbf{z}}{\mathbf{z}^t \mathbf{z}} \tag{51}$$

where $\mathbf{x}_2$ is the target vector for fixed codebook search and $\mathbf{z}$ is the fixed codebook vector convolved with $h(n)$,

$$z(n) = \sum_{i=0}^{n} c(i)\,h(n-i), \qquad n = 0,\ldots,39. \tag{52}$$

10.2 kbit/s mode

In this case the signal $b(n)$, used for presetting the amplitudes, is given by eq. (50). Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an efficient non-exhaustive analysis-by-synthesis search technique. In this technique, the term in equation (43) is tested for a small percentage of position combinations.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z)=1/(1-b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain bounded by [0.0,0.8]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by equation (51).

7.95, 7.40 kbit/s modes

In this case the signal $b(n)$, used for presetting the amplitudes, is equal to the signal $d(n)$. Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an efficient non-exhaustive analysis-by-synthesis search technique. In this technique, the term in equation (43) is tested for a small percentage of position combinations.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z)=1/(1-b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain

bounded by [0.0,0.8]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by equation (51).

6.70 kbit/s mode

In this case the signal $b(n)$, used for presetting the amplitudes, is equal to the signal $d(n)$. Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an efficient non-exhaustive analysis-by-synthesis search technique. In this technique, the term in equation (43) is tested for a small percentage of position combinations.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z) = 1/(1 - b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain bounded by [0.0,0.8]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by equation (51).

5.90 kbit/s mode

In this case the signal $b(n)$, used for presetting the amplitudes, is equal to the signal $d(n)$. Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an exhaustive analysis-by-synthesis search technique.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z) = 1/(1 - b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain bounded by [0.0,0.8]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by equation (51).

5.15, 4.75 kbit/s modes

In this case the signal $b(n)$, used for presetting the amplitudes, is equal to the signal $d(n)$. Having preset the pulse amplitudes, as explained above, the optimal pulse positions are determined using an exhaustive analysis-by-synthesis search technique. Note that both subsets are searched.

A special feature incorporated in the codebook is that the selected codevector is filtered through an adaptive pre-filter $F_E(z)$ which enhances special spectral components in order to improve the synthesized speech quality. Here the filter $F_E(z) = 1/(1 - b\,z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and b is a pitch gain. In this standard, b is given by the quantified pitch gain bounded by [0.0,0.8]. Note that prior to the codebook search, the impulse response $h(n)$ must include the pre-filter $F_E(z)$. That is, $h(n) = h(n) - bh(n-T), \quad n = T,\ldots,39$.

The fixed codebook gain is then found by equation (51).

5.8 Quantization of the adaptive and fixed codebook gains

Prediction of the fixed codebook gain (all modes)

The fixed codebook gain quantization is performed using MA prediction with fixed coefficients. The 4th order MA prediction is performed on the innovation energy as follows. Let $E(n)$ be the mean-removed innovation energy (in dB) at subframe n, and given by:

$$E(n) = 10\log\left(\frac{1}{nN} g_c^2 \sum_{i=0}^{N-1} c^2(i)\right) - \overline{E}\,, \tag{53}$$

where $N = 40$ is the subframe size, $c(i)$ is the fixed codebook excitation, and $\overline{E}$ (in dB) is the mean of the innovation energy. The predicted energy is given by:

$$\tilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i)\,, \tag{54}$$

where $[b_1\, b_2\, b_3\, b_4] = [0.68\ 0.58\ 0.34\ 0.19]$ are the MA prediction coefficients, and $\hat{R}(k)$ is the quantified prediction error at subframe k. The predicted energy is used to compute a predicted fixed-codebook gain g_c' as in equation (53) (by substituting $E(n)$ by $\tilde{E}(n)$ and g_c by g_c'). This is done as follows. First, the mean innovation energy is found by:

$$E_I = 10\log\left(\frac{1}{N}\sum_{j=0}^{N-1} c^2(j)\right) \tag{55}$$

and then the predicted gain g_c' is found by:

$$g_c' = 10^{0.05\left(\tilde{E}(n) + \overline{E} - E_I\right)}\,. \tag{56}$$

A correction factor between the gain g_c and the estimated one g_c' is given by:

$$g_{gc} = g_c / g_c'\,. \tag{57}$$

Note that the prediction error is given by:

$$R(n) = E(n) - \tilde{E}(n) = 20\log(g_{gc}). \tag{58}$$

12.2 kbit/s mode

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 36$ dB. The correction factor g_{gc} is quantified using a 5-bit codebook. The quantization table search is performed by minimizing the error:

$$E_Q = \left(g_c - \hat{g}_{gc} g_c'\right)^2\,. \tag{59}$$

Once the optimum value $\hat{g}_{gc}$ is chosen, the quantified fixed codebook gain is given by $\hat{g}_c = \hat{g}_{gc} g_c'$.

10.2 kbit/s mode

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 33$ dB. The adaptive codebook gain g_p and the correction factor g_{gc} are jointly vector quantized using a 7-bit codebook. The gain codebook search is performed by minimizing equation (63).

7.95 kbit/s mode

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 36$ dB. The same scalar codebooks as for the 12.2 kbit/s mode is used for quantization of the adaptive codebook gain g_p and the correction factor g_{gc}. The search of the codebooks starts with finding 3 candidates for the adaptive codebook gain. These candidates are the best codebook value in scalar quantization and the two adjacent codebook values. These 3 candidates are searched together with the correction factor codebook minimizing the term of equation (63).

An adaptor based on the coding gain in the adaptive codebook decides if the coding gain is low. If this is the case, the correction factor codebook is searched once more minimizing a modified criterion in order to find a new quantized fixed codebook gain. The modified criterion is given by:

$$E_{\mathrm{mod}} = (1-\alpha)\cdot\|\mathbf{c}\|^2\cdot\left(g_c - \hat{g}_{gc}\cdot g_c'\right)^2 + \alpha\cdot\left(\sqrt{E_{res}} - \sqrt{E_{exc}}\right)^2 \qquad (60)$$

where E_{res} and E_{exc} are the energy (the squared norm) of the LP residual and the total exictation, respectively. The criterion is searched with the already quantized adaptive codebook gain and the correction factor $\hat{g}_{gc}$ that minimizes (60) is selected. The balance factor α decides the amount of energy matching in the modified criterion. This factor is adaptively decided based on the coding gain in the adaptive codebook as computed by:

$$ag = 10\cdot\log_{10}\frac{\|\mathbf{res}_{LP}\|^2}{\|\mathbf{res}_{LP} - \mathbf{v}\|^2}. \qquad (61)$$

If the coding gain ag is less than 1 dB, the modified criterion is employed, except when an onset is detected. An onset is said to be detected if the fixed codebook gain in the current subframe is more than twice the value of the fixed codebook gain in the previous subframe. A hangover of 8 subframes is used in the onset detection so that the modified criterion is not used for the next 7 subframes either if an onset is detected. The balance factor α is computed from the median filtered adaptive coding gain. The current and the ag-values for the previous 4 subframes are median filtered to get ag_m. The α -factor is computed by:

$$\alpha = \begin{cases} 0 & ag_m > 2 \\ 0.5\cdot\left(1 - 0.5\cdot ag_m\right) & 0 < ag_m < 2 \\ 0.5 & ag_m < 0 \end{cases}. \qquad (62)$$

7.40 kbit/s mode

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 30$ dB. The adaptive codebook gain g_p and the correction factor g_{gc} are jointly vector quantized using a 7-bit codebook. The gain codebook search is performed by minimizing the square of the weighted error between original and reconstructed speech which is given by

$$E = \left\|\mathbf{x} - g_p\mathbf{y} - g_c\mathbf{z}\right\| = \mathbf{x}^t\mathbf{x} + g_p^2\mathbf{y}^t\mathbf{y} + g_c^2\mathbf{z}^t\mathbf{z} - 2g_p\mathbf{x}^t\mathbf{y} - 2g_c\mathbf{x}^t\mathbf{z} + 2g_pg_c\mathbf{y}^t\mathbf{z} \qquad (63)$$

where $\mathbf{x}$ is the target vector, $\mathbf{y}$ is the filtered adaptive codebook vector, and $\mathbf{z}$ is the filtered fixed codebook vector.

6.70 kbit/s mode

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 28.75$ dB. The adaptive codebook gain g_p and the correction factor g_{gc} are jointly vector quantized using a 7-bit codebook. The gain codebook search is performed by minimizing equation (63).

5.90, 5.15 kbit/s modes

The correction factor g_{gc} is computed using a mean energy value, $\overline{E} = 33$ dB. The adaptive codebook gain g_p and the correction factor g_{gc} are jointly vector quantized using a 6-bit codebook. The gain codebook search is performed by minimizing equation (63).

4.75 kbit/s mode

The correction factors g_{gc} are computed using a mean energy value, $\overline{E} = 33$ dB. The adaptive codebook gains g_p and the correction factors g_{gc} are jointly vector quantized every 10 ms. This is done by minimizing a weighted sum of the error criterion (63) for each of the two subframes. The default values on the weighing factors are 1. If the energy of the second subframe is more than two times the energy of the first subframe, the weight of the first subrame is set to 2. If the energy of the first subframe is more than four times the energy of the first subframe, the weight of the second subrame is set to 2.

5.9 Memory update (all modes)

An update of the states of the synthesis and weighting filters is needed in order to compute the target signal in the next subframe.

After the two gains are quantified, the excitation signal, $u(n)$, in the present subframe is found by:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n), \quad n = 0,\ldots,39, \tag{64}$$

where $\hat{g}_p$ and $\hat{g}_c$ are the quantified adaptive and fixed codebook gains, respectively, $v(n)$ the adaptive codebook vector (interpolated past excitation), and $c(n)$ is the fixed codebook vector (algebraic code including pitch sharpening). The states of the filters can be updated by filtering the signal $res_{LP}(n) - u(n)$ (difference between residual and excitation) through the filters $1/\hat{A}(z)$ and $A(z/g_1)/A(z/g_2)$ for the 40-sample subframe and saving the states of the filters. This would require 3 filterings. A simpler approach which requires only one filtering is as follows. The local synthesized speech, $\hat{s}(n)$, is computed by filtering the excitation signal through $1/\hat{A}(z)$. The output of the filter due to the input $res_{LP}(n) - u(n)$ is equivalent to $e(n) = s(n) - \hat{s}(n)$. So the states of the synthesis filter $1/\hat{A}(z)$ are given by $e(n), n = 30,\ldots,39$. Updating the states of the filter $e(n) = s(n) - \hat{s}(n)$ can be done by filtering the error signal $e(n)$ through this filter to find the perceptually weighted error $e_w(n)$. However, the signal $e_w(n)$ can be equivalently found by:

$$e_w(n) = x(n) - \hat{g}_p y(n) - \hat{g}_c z(n), \tag{65}$$

Since the signals $x(n)$, $y(n)$, and $z(n)$ are available, the states of the weighting filter are updated by computing $e_w(n)$ as in equation (65) for $n = 30,\ldots,39$. This saves two filterings.

4.75 kbit/s mode

The memory update in the first and third subframes use the unquantized gains in equation (64). After the second and fourth subframes respectively, when the gains are quantized, the state is recalculated using the quantized gains.

6 Functional description of the decoder

The function of the decoder consists of decoding the transmitted parameters (LP parameters, adaptive codebook vector, adaptive codebook gain, fixed codebook vector, fixed codebook gain) and performing synthesis to obtain the reconstructed speech. The reconstructed speech is then post-filtered and upscaled. The signal flow at the decoder is shown in figure 4.

6.1 Decoding and speech synthesis

The decoding process is performed in the following order:

Decoding of LP filter parameters: The received indices of LSP quantization are used to reconstruct the quantified LSP vectors. The interpolation described in subclause 5.2.6 is performed to obtain 4 interpolated LSP vectors (corresponding to 4 subframes). For each subframe, the interpolated LSP vector is converted to LP filter coefficient domain a_k , which is used for synthesizing the reconstructed speech in the subframe.

The following steps are repeated for each subframe:

1) **Decoding of the adaptive codebook vector:** The received pitch index (adaptive codebook index) is used to find the integer and fractional parts of the pitch lag. The adaptive codebook vector $v(n)$ is found by interpolating the past excitation $u(n)$ (at the pitch delay) using the FIR filter described in subclause 5.6.

2) **Decoding of the innovative codebook vector:** The received algebraic codebook index is used to extract the positions and amplitudes (signs) of the excitation pulses and to find the algebraic codevector $c(n)$. If the integer part of the pitch lag, T, is less than the subframe size 40, the pitch sharpening procedure is applied which translates into modifying $c(n)$ by $c(n) = c(n) + bc(n-T)$, where β is the decoded pitch gain, $\hat{g}_p$, bounded by [0.0,1.0] or [0.0,0.8], depending on mode.

3) **Decoding of the adaptive and fixed codebook gains:** In case of scalar quantization of the gains (12.2 kbit/s and 7.95 kbit/s modes) the received indices are used to readily find the quantified adaptive codebook gain, $\hat{g}_p$, and the quantified fixed codebook gain correction factor, $\hat{g}_{gc}$, from the corresponding quantization tables. In case of vector quantization of the gains (all other modes), the received index gives both the quantified adaptive codebook gain, $\hat{g}_p$, and the quantified fixed codebook gain correction factor, $\hat{g}_{gc}$. The estimated fixed codebook gain g_c' is found as described in subclause 5.7. First, the predicted energy is found by:

$$\tilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \qquad (66)$$

and then the mean innovation energy is found by:

$$E_I = 10\log\left(\frac{1}{N}\sum_{j=0}^{N-1} c^2(j)\right). \quad (67)$$

The predicted gain g_c' is found by:

$$g_c' = 10^{0.05\left(\tilde{E}(n)+\bar{E}-E_I\right)}. \quad (68)$$

The quantified fixed codebook gain is given by:

$$\hat{g}_c = g_{gc}\, g_c'. \quad (69)$$

4) **Smoothing of the fixed codebook gain (10.2, 6.70, 5.90, 5.15, 4.75 kbit/s modes):** An adaptive smoothing of the fixed codebook gain is performed to avoid unnatural fluctuations in the energy contour. The smoothing is based on a measure of the stationarity of the short-term spectrum in the **q** domain. The smoothing strength is computed from this measure. An averaged **q**-value is computed for each frame n by:

$$\bar{\mathbf{q}}(n) = 0.84 \cdot \bar{\mathbf{q}}(n-1) + 0.16 \cdot \hat{\mathbf{q}}_4(n). \quad (70)$$

For each subframe m, a difference measure between the averaged vector and the quantized and interpolated vector is computed by:

$$diff_m = \sum_j \sum_m \frac{\left|\bar{q}^{(j)}(n) - \hat{q}_m^{(j)}(n)\right|}{\bar{q}^{(j)}(n)}, \quad (71)$$

where j runs over the 10 LSPs. Furthermore, a smoothing factor, k_m, is computed by:

$$k_m = \min\left(K_2, \max\left(0, diff_m - K_1\right)\right) \Big/ K_2, \quad (72)$$

where the constants are set to $K_1 = 0.4$ and $K_2 = 0.25$. A hangover period of 40 subframes is used where the k_m-value is set 1.0 if the $diff_m$ has been above 0.65 for 10 consecutive frames. A value of 1.0 corresponds to no smoothing. An averaged fixed codebook gain value is computed for each subframe by:

$$\bar{g}(m) = \frac{1}{5}\sum_{i=0}^{4} \hat{g}_c(m-i). \quad (73)$$

The fixed codebook gain used for synthesis is now replaced by a smoothed value given by:

$$\hat{g}_c = \hat{g}_c \cdot k_m + \bar{g}_c \cdot \left(1 - k_m\right). \quad (74)$$

5) **Anti-sparseness processing (7.95, 6.70, 5.90, 5.15, 4.75 kbit/s modes):** An adaptive anti-sparseness post-processing procedure is applied to the fixed codebook vector $c(n)$ in order to reduce perceptual artifacts arising from the sparseness of the algebraic fixed codebook vectors with only a few non-zero samples per subframe. The anti-sparseness processing consists of circular convolution of the fixed codebook vector with an impulse response. Three pre-stored impulse responses are used and a number $impNr = 0,1,2$ is set to select one of them. A value of 2 corresponds to no modification, a value of 1 corresponds to medium modification, while a value of 0 corresponds to strong modification. The selection of the impulse response is performed adaptively from the adaptive and fixed codebook gains. The following procedure is employed:

if $\hat{g}_p < 0.6$ then
  $impNr = 0;$
else if $\hat{g}_p < 0.9$ then
  $impNr = 1;$
else
  $impNr = 2;$

Detect onset by comparing the fixed codebook gain to the previous fixed codebook gain. If the current value is more than twice the previous value an onset is detected.

If not onset and $impNr = 0$, the median filtered value of the current and the previous 4 adaptive codebook gains are computed. If this value is less than 0.6, $impNr = 0$.

If not onset, the $impNr$ -value is restricted to increase by one step from the previous subframe.

If an onset is declared, the $impNr$ -value is increased by one if it is less than 2.

6) **Computing the reconstructed speech:** The excitation at the input of the synthesis filter is given by:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) . \tag{75}$$

Before the speech synthesis, a post-processing of excitation elements is performed. This means that the total excitation is modified by emphasizing the contribution of the adaptive codebook vector:

$$\hat{u}(n) = \begin{cases} u(n) + 0.25 b \hat{g}_p v(n), & \hat{g}_p > 0.5, \text{ 12.2 kbit/s mode} \\ u(n) + 0.5 b \hat{g}_p v(n), & \hat{g}_p > 0.5, \text{ all other modes} \\ u(n) & \hat{g}_p \le 0.5 \end{cases} \tag{76}$$

Adaptive gain control (AGC) is used to compensate for the gain difference between the non-emphasized excitation $u(n)$ and emphasized excitation $\hat{u}(n)$ The gain scaling factor h for the emphasized excitation is computed by:

$$h = \begin{cases} \sqrt{\dfrac{\sum_{n=0}^{39} u^2(n)}{\sum_{n=0}^{39} \hat{u}^2(n)}}, & \hat{g}_p > 0.5, \\ 1.0, & \hat{g}_p \le 0.5. \end{cases} \tag{77}$$

The gain-scaled emphasized excitation signal $\hat{u}'(n)$ is given by:

$$\hat{u}'(n) = \hat{u}(n)h\,. \tag{78}$$

The reconstructed speech for the subframe of size 40 is given by:

$$\hat{s}(n) = \hat{u}'(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i), \quad n = 0,\ldots,39\,. \tag{79}$$

where $\hat{a}_i$ are the interpolated LP filter coefficients.

The synthesized speech $\hat{s}(n)$ is then passed through an adaptive postfilter which is described in the following clause.

6.2 Post-processing

6.2.1 Adaptive post-filtering (all modes)

The adaptive postfilter is the cascade of two filters: a formant postfilter, and a tilt compensation filter. The postfilter is updated every subframe of 5 ms.

The formant postfilter is given by:

$$H_f(z) = \frac{\hat{A}(z/g_n)}{\hat{A}(z/g_d)} \tag{80}$$

where $\hat{A}(z)$ is the received quantified (and interpolated) LP inverse filter (LP analysis is not performed at the decoder), and the factors γ_n and γ_d control the amount of the formant post-filtering.

Finally, the filter $H_t(z)$ compensates for the tilt in the formant postfilter $H_f(z)$ and is given by:

$$H_t(z) = 1 - mz^{-1} \tag{81}$$

where $m = g_t k_1'$ is a tilt factor, with k_1' being the first reflection coefficient calculated on the truncated ($L_h = 22$) impulse response, $h_f(n)$, of the filter $\hat{A}(z/g_n)/\hat{A}(z/g_d)$. k_1' is given by:

$$k_1' = \frac{r_h(1)}{r_h(0)}; \quad r_h(i) = \sum_{j=0}^{L_h-i-1} h_f(j) h_f(j+i). \tag{82}$$

The post-filtering process is performed as follows. First, the synthesized speech $\hat{s}(n)$ is inverse filtered through $\hat{A}(z/g_n)$ to produce the residual signal $\hat{r}(n)$. The signal $\hat{r}(n)$ is filtered by the synthesis filter $1/\hat{A}(z/g_d)$. Finally, the signal at the output of the synthesis filter $1/\hat{A}(z/g_d)$ is passed to the tilt compensation filter $H_t(z)$ resulting in the post-filtered speech signal $\hat{s}_f(n)$.

Adaptive gain control (AGC) is used to compensate for the gain difference between the synthesized speech signal $\hat{s}(n)$ and the post-filtered signal $\hat{s}_f(n)$. The gain scaling factor g_{sc} for the present subframe is computed by:

$$g_{sc} = \sqrt{\frac{\sum_{n=0}^{39} \hat{s}^2(n)}{\sum_{n=0}^{39} \hat{s}_f^2(n)}}. \tag{83}$$

The gain-scaled post-filtered signal $\hat{s}'(n)$ is given by:

$$\hat{s}'(n) = b_{sc}(n)\hat{s}_f(n) \tag{84}$$

where $b_{sc}(n)$ is updated in sample-by-sample basis and given by:

$$b_{sc}(n) = a\,b_{sc}(n-1) + (1-a)g_{sc} \tag{85}$$

where α is a AGC factor with value of 0.9.

12.2, 10.2 kbit/s modes

The adaptive post-filtering factors are given by: $\gamma_n = 0.7$, $\gamma_d = 0.75$ and

$$g_t = \begin{cases} 0.8, & k_1' > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{86}$$

7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbit/s modes

The adaptive post-filtering factors are given by: $g_n = 0.55$, $g_d = 0.7$ and $g_t = 0.8$.

6.2.2 High-pass filtering and up-scaling (all modes)

The high-pass filter serves as a precaution against undesired low frequency components. A filter cut-off frequency of 60 Hz is used, and the filter is given by

$$H_{h2}(z) = \frac{0.939819335 - 1.879638672z^{-1} + 0.939819335z^{-2}}{1 - 1.933105469z^{-1} + 0.935913085z^{-2}}. \tag{87}$$

Up-scaling consists of multiplying the post-filtered speech by a factor of 2 to compensate for the down-scaling by 2 which is applied to the input signal.

7 Detailed bit allocation of the adaptive multi-rate codec

The detailed allocation of the bits in the adaptive multi-rate speech encoder is shown for each mode in table 9a-9h. These tables show the order of the bits produced by the speech encoder. Note that the most significant bit (MSB) of each codec parameter is always sent first

Table 9a: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 244 bits/20 ms, 12.2 kbit/s mode.

Bits (MSB-LSB)	Description
s1 - s7	index of 1st LSF submatrix
s8 - s15	index of 2nd LSF submatrix
s16 - s23	index of 3rd LSF submatrix
s24	sign of 3rd LSF submatrix
s25 - s32	index of 4th LSF submatrix
s33 - s38	index of 5th LSF submatrix
subframe 1	
s39 - s47	adaptive codebook index
s48 - s51	adaptive codebook gain
s52	sign information for 1st and 6th pulses
s53 - s55	position of 1st pulse
s56	sign information for 2nd and 7th pulses
s57 - s59	position of 2nd pulse
s60	sign information for 3rd and 8th pulses
s61 - s63	position of 3rd pulse
s64	sign information for 4th and 9th pulses
s65 - s67	position of 4th pulse
s68	sign information for 5th and 10th pulses
s69 - s71	position of 5th pulse
s72 - s74	position of 6th pulse
s75 - s77	position of 7th pulse
s78 - s80	position of 8th pulse
s81 - s83	position of 9th pulse
s84 - s86	position of 10th pulse
s87 - s91	fixed codebook gain
subframe 2	
s92 - s97	adaptive codebook index (relative)
s98 - s141	same description as s48 - s91
subframe 3	
s142 - s194	same description as s39 - s91
subframe 4	
s195 - s244	same description as s92 - s141

Table 9b: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 204 bits/20 ms, 10.2 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s17	index of 2nd LSF subvector
s18 – s26	index of 3rd LSF subvector
subframe 1	
s27 – s34	adaptive codebook index
s35	sign information for 1st and 5th pulses
s36	sign information for 2nd and 6th pulses
s37	sign information for 5th and 7th pulses
s38	sign information for 4th and 8th pulses
s39-s48	position for 1st, 2nd, and 5th pulses
s49-s58	position for 3rd, 6th, and 7th pulses
s59-s65	position for 4th and 7th pulses
s66 – s72	codebook gains
subframe 2	
s73 – s77	adaptive codebook index (relative)
s78 – s115	same description as s35 – s72
subframe 3	
s116 – s161	same description as s27 – s72
subframe 4	
s162 – s204	same description as s73 – s115

Table 9c: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 159 bits/20 ms, 7.95 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s9	index of 1st LSF subvector
s10 - s18	index of 2nd LSF subvector
s19 – s27	index of 3rd LSF subvector
subframe 1	
s28 – s35	adaptive codebook index
s36 – s38	position of 1st pulse
s39 – s41	position of 2nd pulse
s42 – s44	position of 3rd pulse
s45 – s48	position of 4th pulse
s49	sign information for 1st pulse
s50	sign information for 2nd pulse
s51	sign information for 3rd pulse
s52	sign information for 4th pulse
s53 – s56	adaptive codebook gain
s57 – s61	fixed codebook gain
subframe 2	
s62 – s67	adaptive codebook index (relative)
s68 – s93	same description as s36 – s61
subframe 3	
s94 – s127	same description as s28 – s61
subframe 4	
s128 – s159	same description as s62 – s93

Table 9d: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 148 bits/20 ms, 7.40 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s17	index of 2nd LSF subvector
s18 – s26	index of 3rd LSF subvector
subframe 1	
s27 – s34	adaptive codebook index
s35 – s37	position of 1st pulse
s38 – s40	position of 2nd pulse
s41 - s43	position of 3rd pulse
s44 – s47	position of 4th pulse
s48	sign information for 1st pulse
s49	sign information for 2nd pulse
s50	sign information for 3rd pulse
s51	sign information for 4th pulse
s52 – s58	codebook gains
subframe 2	
s59 – s63	adaptive codebook index (relative)
s64 – s87	same description as s35 – s58
subframe 3	
s88 – s119	same description as s27 – s58
subframe 4	
s120 – s148	same description as s59 – s87

Table 9e: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 134 bits/20 ms, 6.70 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s17	index of 2nd LSF subvector
s18 – s26	index of 3rd LSF subvector
subframe 1	
s27 – s34	adaptive codebook index
s35 – s37	position of 1st pulse
s38 – s41	position of 2nd pulse
s42 – s45	position of 3rd pulse
s46	sign information for 1st pulse
s47	sign information for 2nd pulse
s48	sign information for 3rd pulse
s49 – s55	codebook gains
subframe 2	
s56 – s59	adaptive codebook index (relative)
s60 – s80	same description as s35 – s55
subframe 3	
s81 – s109	same description as s27 – s55
subframe 4	
s110 – s134	same description as s56 – s80

Table 9f: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 118 bits/20 ms, 5.90 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s17	index of 2nd LSF subvector
s18 – s26	index of 3rd LSF subvector
subframe 1	
s27 – s34	adaptive codebook index
s35 – s38	position of 1st pulse
s39 – s43	position of 2nd pulse
s44	sign information for 1st pulse
s45	sign information for 2nd pulse
s46 – s51	codebook gains
subframe 2	
s52 – s55	adaptive codebook index (relative)
s56 – s72	same description as s35 – s51
subframe 3	
s73 – s97	same description as s27 – s51
subframe 4	
s98 – s118	same description as s52 – s72

Table 9g: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 103 bits/20 ms, 5.15 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s16	index of 2nd LSF subvector
s17 – s23	index of 3rd LSF subvector
subframe 1	
s24 – s31	adaptive codebook index
s32	position subset
s33 – s35	position of 1st pulse
s36 – s38	position of 2nd pulse
s39	sign information for 1st pulse
s40	sign information for 2nd pulse
s41 – s46	codebook gains
subframe 2	
s47 – s50	adaptive codebook index (relative)
s51 – s65	same description as s32 – s46
subframe 3	
s66 – s84	same description as s47 – s65
subframe 4	
s85 – s103	same description as s47 – s65

Table 9h: Source encoder output parameters in order of occurrence and bit allocation within the speech frame of 95 bits/20 ms, 4.75 kbit/s mode.

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s16	index of 2nd LSF subvector
s17 – s23	index of 3rd LSF subvector
subframe 1	
s24 – s31	adaptive codebook index
s32	position subset
s33 – s35	position of 1st pulse
s36 – s38	position of 2nd pulse
s39	sign information for 1st pulse
s40	sign information for 2nd pulse
s41 – s48	codebook gains
subframe 2	
s49 – s52	adaptive codebook index (relative)
s53 – s61	same description as s32 – s40
subframe 3	
s62 - s65	same description as s49 – s52
s66 – s82	same description as s32– s48
subframe 4	
s83 – s95	same description as s49 – s61

8 Homing sequences

8.1 Functional description

The adaptive multi-rate speech codec is described in a bit-exact arithmetic to allow for easy type approval as well as general testing purposes of the adaptive multi-rate speech codec.

The response of the codec to a predefined input sequence can only be foreseen if the internal state variables of the codec are in a predefined state at the beginning of the experiment. Therefore, the codec has to be put in a so called home state before a bit-exact test can be performed. This is usually done by a reset (a procedure in which the internal state variables of the codec are set to their defined initial values).

To allow a reset of the codec in remote locations, special homing frames have been defined for the encoder and the decoder, thus enabling a codec homing by inband signalling.

The codec homing procedure is defined in such a way, that in either direction (encoder or decoder) the homing functions are called after processing the homing frame that is input. The output corresponding to the first homing frame is therefore dependent on the codec state when receiving that frame and hence usually not known. The response to any further homing frame in one direction is by definition a homing frame of the other direction. This procedure allows homing of both, the encoder and decoder from either side, if a loop back configuration is implemented, taking proper framing into account.

8.2 Definitions

Encoder homing frame: The encoder homing frame consists of 160 identical samples, each 13 bits long, with the least significant bit set to "one" and all other bits set to "zero". When written to 16-bit words with left justification, the samples have a value of 0008 hex. The speech decoder has to produce this frame as a response to the second and any further decoder homing frame if at least two decoder homing frames were input to the decoder consecutively.

Decoder homing frame: The decoder homing frame has a fixed set of speech parameters as described in table 10. It is the natural response of the speech encoder to the second and any further encoder homing frame if at least two encoder homing frames were input to the encoder consecutively.

Table 10: Parameter values for the decoder homing frame.

Parameter	Value (LSB=b0)
f.f.s.	f.f.s.

8.3 Encoder homing

Whenever the adaptive multi-rate speech encoder receives at its input an encoder homing frame exactly aligned with its internal speech frame segmentation, the following events take place:

Step 1: The speech encoder performs its normal operation including VAD and SCR and produces a speech parameter frame at its output which is in general unknown. But if the speech encoder was in its home state at the beginning of that frame, then the resulting speech parameter frame is identical to the decoder homing frame (this is the way how the decoder homing frame was constructed).

Step 2: After successful termination of that operation the speech encoder provokes the homing functions for all sub-modules including VAD and SCR and sets all state variables into their home state. On the reception of the next input frame, the speech encoder will start from its home state.

NOTE: Applying a sequence of N encoder homing frames will cause at least N-1 decoder homing frames at the output of the speech encoder.

8.4 Decoder homing

Whenever the speech decoder receives at its input a decoder homing frame, then the following events take place:

Step 1: The speech decoder performs its normal operation and produces a speech frame at its output which is in general unknown. But if the speech decoder was in its home state at the beginning of that frame, then the resulting speech frame is replaced by the encoder homing frame. This would not naturally be the case but is forced by this definition here.

Step 2: After successful termination of that operation the speech decoder provokes the homing functions for all sub-modules including the comfort noise generator and sets all state variables into their home state. On the reception of the next input frame, the speech decoder will start from its home state.

NOTE 1: Applying a sequence of N decoder homing frames will cause at least N-1 encoder homing frames at the output of the speech decoder.

NOTE 2: By definition (!) the first frame of each decoder test sequence must differ from the decoder homing frame at least in one bit position within the parameters for LPC and first subframe. Therefore, if the decoder is in its home state, it is sufficient to check only these parameters to detect a subsequent decoder homing frame. This definition is made to support a delay-optimized implementation.

8.5 Encoder home state

In [4], a listing of all the encoder state variables in the ANSI-C code with their predefined values when in the home state is given.

8.6 Decoder home state

In [4], a listing of all the decoder state variables in the ANSI-C code with their predefined values when in the home state is given.

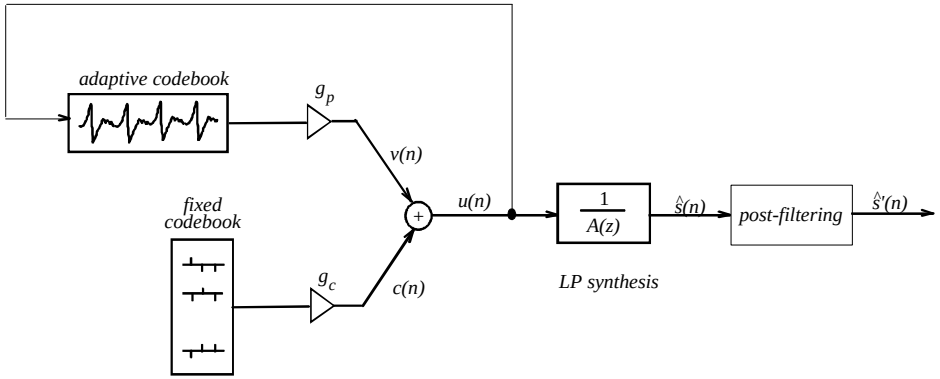


Figure 2: Simplified block diagram of the CELP synthesis model

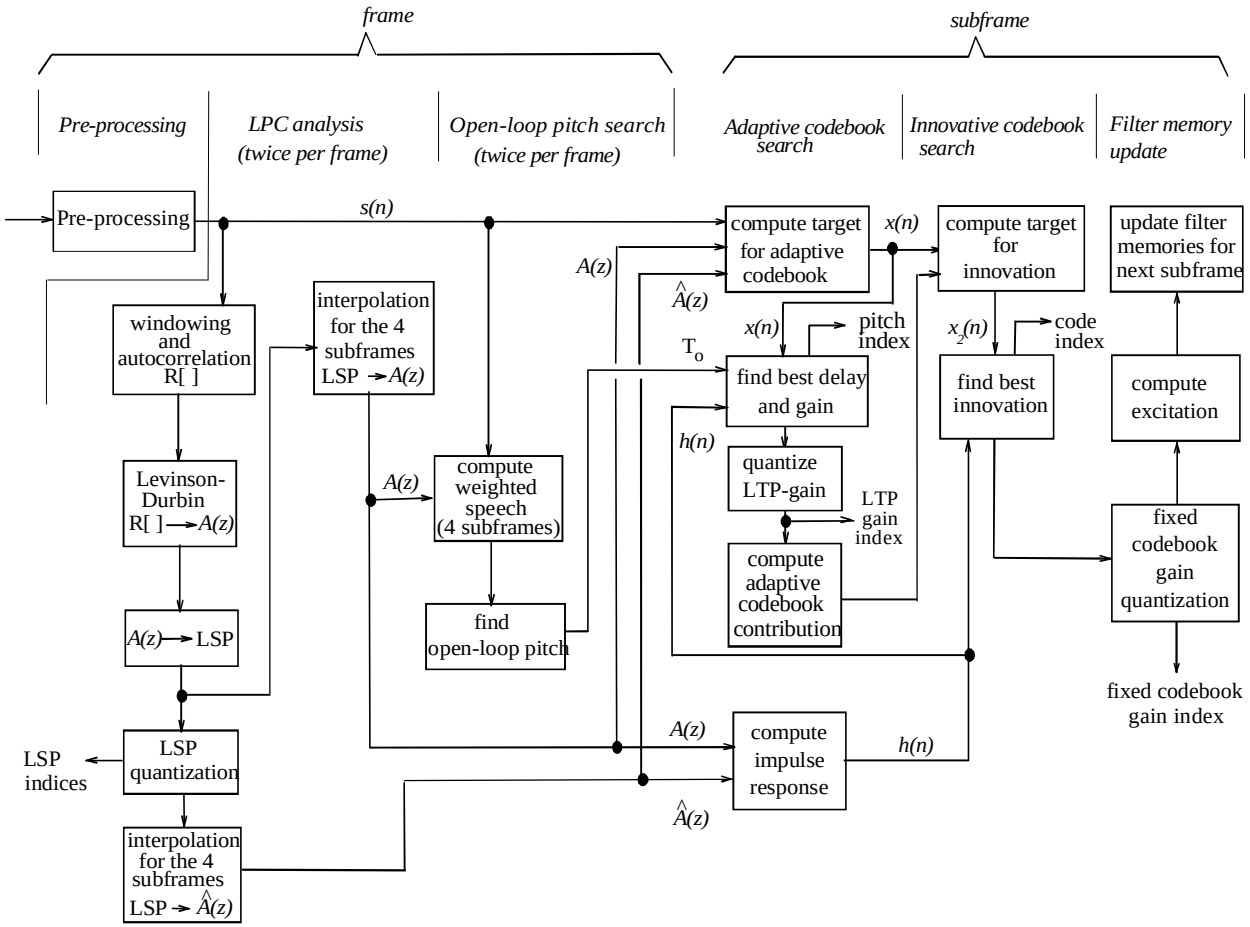


Figure 3: Simplified block diagram of the adaptive multi-rate encoder

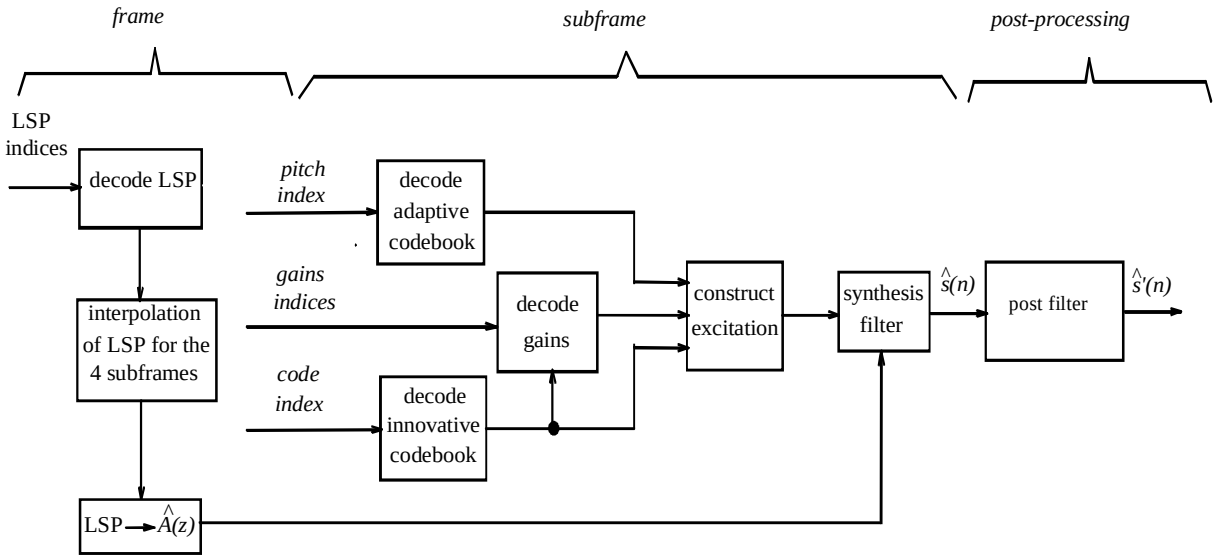


Figure 4: Simplified block diagram of the adaptive multi-rate decoder

9 Bibliography

1) M.R. Schroeder and B.S. Atal, "Code-Excited Linear Prediction (CELP): High quality speech at very low bit rates," in Proc. *ICASSP'85*, pp. 937-940, 1985.

2) L.R. Rabiner and R.W. Schaefer. *Digital processing of speech signals*. Prentice-Hall Int., 1978.

3) F. Itakura, "Line spectral representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Amer*., vol. 57, Supplement no. 1, S35, 1975.

4) F.K. Soong and B.H. Juang, "Line spectrum pair (LSP) and speech data compression", in *Proc. ICASSP'84*, pp. 1.10.1-1.10.4.

5) K.K Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", *IEEE Trans. Speech and Audio Processing*, vol. 1, no 1, pp. 3-14, 1993.

6) P. Kabal and R.P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", *IEEE Trans. on ASSP*, vol. 34, no. 6, pp. 1419-1426, Dec. 1986.

7) K. Järvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, R. Salami, C. Laflamme, and J.-P. Adoul, “GSM enhanced full rate speech codec”, in *Proc. ICASSP’97*, pp. 771-774.

8) T. Honkanen, J. Vainio, K. Järvinen, P. Haavisto, R. Salami, C. Laflamme, and J.-P. Adoul, “Enhanced full rate speech codec for IS-136 digital cellular system”, in *Proc. ICASSP’97*, pp. 731-734.

9) R. Hagen, E. Ekudden, B. Johansson, and W.B. Kleijn, “Removal of sparse-excitation artifacts in CELP”, in *Proc. ICASSP’98*, pp. I-145-I-148.

History

Document history		
V. 0.1.0	March 1999	First Draft based on GSM 06.90 2.0.0, CHC text removed, DTX->SCR
V. 0.1.1	April 1999	References updated, CHE reference removed
V. 1.0.0	April 22, 1999	Minor editorial changes