ETSI TS 146 060 V7.0.0 (2007-06)

Technical Specification

Digital cellular telecommunications system (Phase 2+); Enhanced Full Rate (EFR) speech transcoding (3GPP TS 46.060 version 7.0.0 Release 7)

Reference
RTS/TSGS-0446060v700

Keywords
GSM

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The present document describes the detailed mapping between input blocks of 160 speech samples in 13-bit uniform PCM format to encoded blocks of 244 bits and from encoded blocks of 244 bits to output blocks of 160 reconstructed speech samples within the digital cellular telecommunications system.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes the detailed mapping between input blocks of 160 speech samples in 13-bit uniform PCM format to encoded blocks of 244 bits and from encoded blocks of 244 bits to output blocks of 160 reconstructed speech samples. The sampling rate is 8 000 sample/s leading to a bit rate for the encoded bit stream of 12,2 kbit/s. The coding scheme is the so-called Algebraic Code Excited Linear Prediction Coder, hereafter referred to as ACELP.

The present document also specifies the conversion between A-law or µ-law (PCS 1900) PCM and 13-bit uniform PCM. Performance requirements for the audio input and output parts are included only to the extent that they affect the transcoder performance. This part also describes the codec down to the bit level, thus enabling the verification of compliance to the part to a high degree of confidence by use of a set of digital test sequences. These test sequences are described in GSM 06.54 [7] and are available on disks.

In case of discrepancy between the requirements described in the present document and the fixed point computational description (ANSI-C code) of these requirements contained in GSM 06.53 [6], the description in GSM 06.53 [6] will prevail.

The transcoding procedure specified in the present document is applicable for the enhanced full rate speech traffic channel (TCH) in the GSM system.

In GSM 06.51 [5], a reference configuration for the speech transmission chain of the GSM enhanced full rate (EFR) system is shown. According to this reference configuration, the speech encoder takes its input as a 13-bit uniform PCM signal either from the audio part of the Mobile Station or on the network side, from the PSTN via an 8-bit/A-law or µ-law (PCS 1900) to 13-bit uniform PCM conversion. The encoded speech at the output of the speech encoder is delivered to a channel encoder unit which is specified in GSM 05.03 [3]. In the receive direction, the inverse operations take place.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] GSM 01.04: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".

[2] 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".

[3] GSM 05.03: "Digital cellular telecommunications system (Phase 2+); Channel coding".

[4] GSM 06.32: "Digital cellular telecommunications system (Phase 2+); Voice Activity Detection (VAD)".

[5] GSM 06.51: "Digital cellular telecommunications system (Phase 2+); Enhanced Full Rate (EFR) speech processing functions General description".

[6] GSM 06.53: "Digital cellular telecommunications system (Phase 2+); ANSI-C code for the GSM Enhanced Full Rate (EFR) speech codec".

[7] GSM 06.54: "Digital cellular telecommunications system (Phase 2+); Test vectors for the GSM Enhanced Full Rate (EFR) speech codec".

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

[9] 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 the present document, the following terms and definitions apply:

adaptive codebook: 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 GSM enhanced full rate codec, the adaptive postfilter is a cascade of two filters: a formant postfilter and a tilt compensation filter.

algebraic codebook: 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.

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 GSM enhanced full 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 GSM enhanced full rate codec, all filters which are used to modify speech samples use direct form coefficients.

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

fractional lags: set of lag values having sub-sample resolution. In the GSM enhanced full rate codec a sub-sample resolution of 1/6th of a sample is used.

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

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

interpolating filter: FIR filter used to produce an estimate of sub-sample 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: long term filter delay. This is typically the true pitch period, or a multiple or sub-multiple of it.

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 GSM enhanced full rate codec, the length of the analysis window is 240 samples. For each frame, two asymmetric windows are used to generate two sets of LP coefficients. No samples of the future frames are used (no lookahead).

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

open-loop pitch search: 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 GSM enhanced full rate codec, open-loop pitch search is performed every 10 ms.

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

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

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

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$A(z)$	The inverse filter with unquantized coefficients
$\hat{A}(z)$	The inverse filter with quantified coefficients
$H(z) = \frac{1}{\hat{A}(z)}$	The speech synthesis filter with quantified 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)
γ_1, γ_2	The perceptual weighting factors
$F_E(z)$	Adaptive pre-filter
T	The nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe
β	The adaptive pre-filter coefficient (the quantified pitch gain)
$H_f(z) = \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)}$	The formant postfilter
γ_n	Control coefficient for the amount of the formant post-filtering
γ_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
$\mu = \gamma_t k_1'$	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 $h_f(n)$
$\hat{A}(z/\gamma_n)$	The inverse filter (numerator) part of the formant postfilter
$1/\hat{A}(z/\gamma_d)$	The synthesis filter (denominator) part of the formant postfilter
$\hat{r}(n)$	The residual signal of the inverse filter $\hat{A}(z/\gamma_n)$
$h_t(z)$	Impulse response of the tilt compensation filter
$\beta_{sc}(n)$	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 *i*th iteration of the Levinson algorithm
k_i	The *i*th reflection coefficient
$a_j^{(i)}$	The *j*th direct form coefficient in the *i*th 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 *i*th subframe of the frame *n*
ω_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 ω

λ_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 $t_i, i = 1,\ldots,3$
$H(z)W(z) = \frac{A(z/\gamma_1)}{\hat{A}(z)A(z/\gamma_2)}$	The weighted synthesis filter
$A(z/\gamma_1)$	The numerator of the perceptual weighting filter
$1/A(z/\gamma_2)$	The denominator of the perceptual weighting filter
T_1	The nearest integer 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 $\mathbf{d}$
$\phi(i,j)$	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 *i* th pulse
ϑ_i	The amplitude of the *i* th 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 sum of the normalized $d(n)$ vector and normalized long-term prediction residual $res_{LTP}(n)$
$s_b(n)$	The sign signal for the algebraic codebook search
$d^{'}(n)$	Sign extended backward filtered target
$\phi^{'}(i,j)$	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
η	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

$\gamma_{gc} = g_c / g_c'$	A correction factor between the gain g_c and the estimated one g_c'
$\hat{\gamma}_{gc}$	The optimum value for γ_{gc}
γ_{sc}	Gain scaling factor

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply. Further GSM related abbreviations may be found in GSM 01.04 [1].

ACELP	Algebraic Code Excited Linear Prediction
AGC	Adaptive Gain Control
CELP	Code Excited Linear Prediction
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

The present document is structured as follows.

Subclause 4.1 contains a functional description of the audio parts including the A/D and D/A functions. Subclause 4.2 describes the conversion between 13-bit uniform and 8-bit A-law or µ-law (PCS 1900) samples. Subclauses 4.3 and 4.4 present a simplified description of the principles of the GSM EFR encoding and decoding process respectively. In clause 4.5, the sequence and subjective importance of encoded parameters are given.

Clause 5 presents the functional description of the GSM EFR encoding, whereas clause 6 describes the decoding procedures. Clause 7 describes variables, constants and tables of the C-code of the GSM EFR codec.

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 μ-law (PCS 1900) compounded format, based on a standard A-law or μ-law (PCS 1900) codec/filter according to ITU-T Recommendations G.711 [8] and G.714, followed by the 8-bit to 13-bit conversion as specified in clause 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, the present document does not on its own ensure sufficient out-of-band attenuation. The specification of out-of-band signals is defined in GSM 03.50 [2] 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 μ-law (PCS 1900) compressed data and linear data with 13-bit resolution at the speech encoder input shall be as defined in ITU-T Rec. G.711 [8].

ITU-T Recommendation G.711 [8] specifies the A-Law or μ-law (PCS 1900) 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 Recommendation G.726 [9]. Subclause 4.2.1 of G.726 [9] describes A-Law and μ-law (PCS 1900) to linear expansion and clause 4.2.7 of G.726 [9] provides a solution for linear to A-Law and μ-law (PCS 1900) compression.

4.3 Principles of the GSM enhanced full rate speech encoder

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:

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

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:

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

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:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \qquad (3)$$

where $A(z)$ is the unquantized LP filter and $0 < \gamma_2 < \gamma_1 \leq 1$ are the perceptual weighting factors. The values $\gamma_1 = 0.9$ and $\gamma_2 = 0.6$ are used. The weighting filter uses the unquantized LP parameters while the formant synthesis filter uses the quantified ones.

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. The two sets of LP parameters are converted to line spectrum pairs (LSP) and jointly quantified using split matrix quantization (SMQ) with 38 bits. 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 two sets of quantified and unquantized LP filters are used for the second and fourth subframes while in the first and third subframes interpolated LP filters are used (both quantified and unquantized). An open-loop pitch lag is estimated twice per frame (every 10 ms) 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 of a sample resolution is used. The pitch lag is encoded with 9 bits in the first and third subframes and relatively encoded with 6 bits in the second and fourth subframes.
>
> 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). An algebraic codebook with 35 bits is used for the innovative excitation.
>
> The gains of the adaptive and fixed codebook are scalar quantified with 4 and 5 bits respectively (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 codec is shown in table 1. In each 20 ms speech frame, 244 bits are produced, corresponding to a bit rate of 12.2 kbit/s. More detailed bit allocation is available in table 6. Note that the most significant bits (MSB) are always sent first.

Table 1: Bit allocation of the 12.2 kbit/s coding algorithm for 20 ms frame

Parameter	1st & 3rd subframes	2nd & 4th subframes	total per frame
2 LSP sets			38
Pitch delay	9	6	30
Pitch gain	4	4	16
Algebraic code	35	35	140
Codebook gain	5	5	20
Total			244

4.4 Principles of the GSM enhanced full rate speech decoder

The signal flow at the decoder is shown in figure 4. At the decoder, 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 two LSP vectors, the 4 fractional pitch lags, the 4 innovative codevectors, and the 4 sets of 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 6, the sequence of output bits s1 to s244 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. Before being submitted to the channel encoding function the bits have to be rearranged in the sequence of importance as given in table 6 in 05.03 [3].

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

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:

$$H_{h1}(z) = \frac{0.92727435 - 1.8544941z^{-1} + 0.92727435z^{-2}}{1 - 1.9059465z^{-1} + 0.9114024z^{-2}}. \qquad (4)$$

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

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).

5.2.1 Windowing and auto-correlation computation

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:

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

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:

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

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.

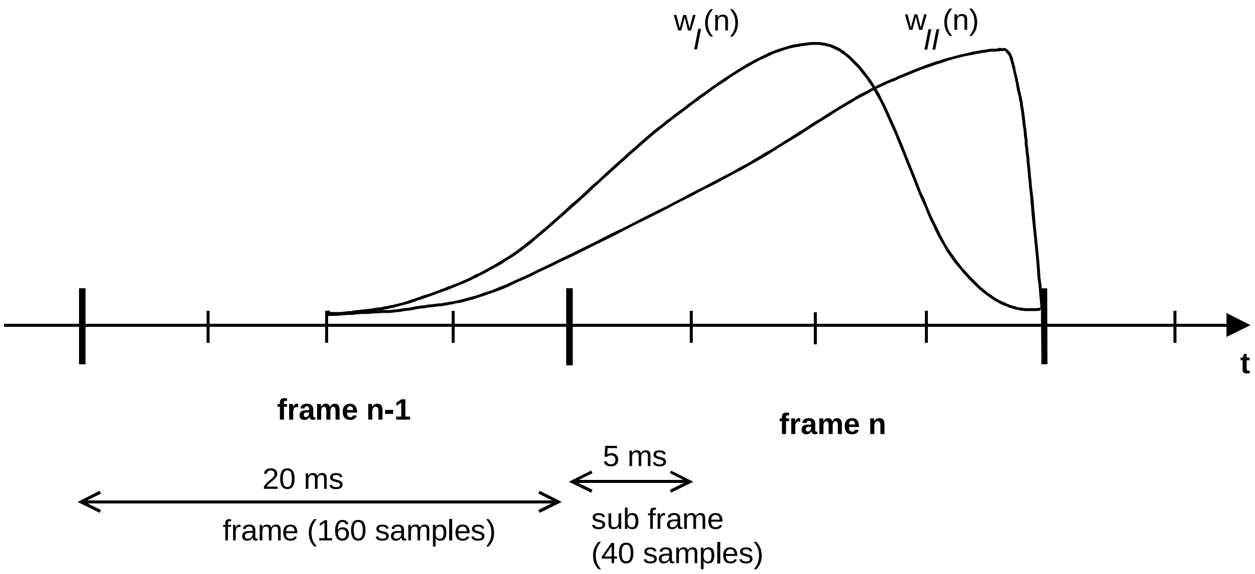

Figure 1: LP analysis windows

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:

$$w_{lag}(i) = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 i}{f_s}\right)^2\right], \qquad i = 1,\ldots,10, \tag{8}$$

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.

5.2.2 Levinson-Durbin algorithm

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

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:

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

and

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

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 $z = -1$ $(\omega = \pi)$ and $F_2'(z)$ has a root $z = 1$ $(\omega = 0)$. To eliminate these two roots, we define the new polynomials:

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

and

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

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

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

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 ω_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 π 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 $z = e^{j\omega}$ can be written as:

$$F(\omega) = 2e^{-j5\omega}C(x),$$

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 $T_m(x) = cos(m\omega)$ 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 $x = cos(\omega)$ using the recursive relation:

```
for $k = 4$ down to 1
  $\lambda_k = 2x\lambda_{k+1} - \lambda_{k+2} + f(5-k)$
end
$C(x) = x\lambda_1 - \lambda_2 + f(5)/2,$
```

with initial values $\lambda_5 = 1$ and $\lambda_6 = 0.$ The details of the Chebyshev polynomial evaluation method are found in P. Kabal and R.P. Ramachandran [6].

5.2.4 LSP to LP conversion

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
    $f_1(i) = -2q_{2i-1}f_1(i-1) + 2f_1(i-2)$
    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:

$$\begin{aligned} f_1^{'}(i) &= f_1(i)+f_1(i-1), \qquad i=1,\dots,5, \\ f_2^{'}(i) &= f_2(i)-f_2(i-1), \qquad i=1,\dots,5. \end{aligned} \tag{18}$$

Finally the LP coefficients are found by:

$$a_i = \begin{cases} 0.5 f_1^{'}(i) + 0.5 f_2^{'}(i), & i=1,\dots,5, \\ 0.5 f_1^{'}(11-i) - 0.5 f_2^{'}(11-i), & i=6,\dots,10. \end{cases} \tag{19}$$

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

5.2.5 Quantization of the LSP coefficients

The two sets of LP filter coefficients per frame are quantified using the LSP representation in the frequency domain; that is:

$$f_i = \frac{f_s}{2\pi} arccos\left(q_i\right), \qquad i=1,\dots,10, \tag{20}$$

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 $\mathbf{f}^t=\left[f_1 \; f_2 \dots f_{10}\right]$, 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:

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

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

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

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.

5.2.6 Interpolation of the LSPs

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:

$$\begin{aligned} \hat{\mathbf{q}}_1^{(n)} &= 0.5\hat{\mathbf{q}}_4^{(n-1)} + 0.5\hat{\mathbf{q}}_2^{(n)}, \\ \hat{\mathbf{q}}_3^{(n)} &= 0.5\hat{\mathbf{q}}_2^{(n)} + 0.5\hat{\mathbf{q}}_4^{(n)}. \end{aligned} \tag{25}$$

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 clause 5.2.4.

5.3 Open-loop pitch analysis

Open-loop pitch analysis is performed twice per frame (each 10 ms) to find two estimates of the pitch lag in each frame. This is done 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 $W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}$. That is, in a subframe of size L, the weighted speech is given by:

$$s_w(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i) - \sum_{i=1}^{10} a_i \gamma_2^i s_w(n-i), \quad n = 0,\ldots,L-1. \tag{26}$$

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

$$O_k = \sum_{n=0}^{79} s_w(n) s_w(n-k) \tag{27}$$

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.

5.4 Impulse response computation

The impulse response, $h(n)$, of the weighted synthesis filter $H(z)W(z) = A(z/\gamma_1)/\left[\hat{A}(z)A(z/\gamma_2)\right]$ 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 $A(z/\gamma_1)$ extended by zeros through the two filters $1/\hat{A}(z)$ and $1/A(z/\gamma_2)$.

5.5 Target signal computation

The target signal for adaptive codebook search is usually computed by subtracting the zero input response of the weighted synthesis filter $H(z)W(z) = A(z/\gamma_1)/\left[\hat{A}(z)A(z/\gamma_2)\right]$ 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 the present document, 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 $A(z/\gamma_1)/A(z/\gamma_2)$. 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 clause 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:

$$res_{LP}(n)=s(n)+\sum_{i=1}^{10}\hat{a}_i s(n-i). \qquad (28)$$

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.

In the first and third subframes, a fractional pitch delay is used with resolutions: 1/6 in the range $\left[17\frac{3}{6}, 94\frac{3}{6}\right]$ 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 $\left[T_1-5\frac{3}{6}, T_1+4\frac{3}{6}\right]$, 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:

$$R(k)=\frac{\sum_{n=0}^{39}x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39}y_k(n)y_k(n)}}, \qquad (29)$$

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:

$$y_k(n)=y_{k-1}(n-1)+u(-k)h(n), \qquad (30)$$

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 (30) valid for all delays.

Once the optimum integer pitch delay is determined, the fractions from $-\frac{3}{6}$ to $\frac{3}{6}$ with a step of $\frac{1}{6}$ around that integer are tested. The fractional pitch search is performed by interpolating the normalized correlation in equation (29) 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 $-\frac{3}{6}$ to $\frac{3}{6}$ are obtained using the interpolation formula:

$$R(k)_t = \sum_{i=0}^{3} R(k-i)\, b_{24}(t+i\cdot 6) + \sum_{i=0}^{3} R(k+1+i) b_{24}(6-t+i\cdot 6), \quad t=0,\ldots,5, \tag{31}$$

where $t=0,\ldots,5$ corresponds to the fractions 0, $\frac{1}{6}$, $\frac{2}{6}$, $\frac{3}{6}$, $-\frac{2}{6}$, and $-\frac{1}{6}$, respectively. Note that it is necessary to compute the correlation terms in equation (29) 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 :

$$v(n) = \sum_{i=0}^{9} u(n-k-i) b_{60}(t+i\cdot 6) + \sum_{i=0}^{9} u(n-k+1+i) b_{60}(6-t+i\cdot 6), \quad n=0,\ldots,39, \; t=0,\ldots,5. \tag{32}$$

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:

$$g_p = \frac{\sum_{n=0}^{39} x(n)y(n)}{\sum_{n=0}^{39} y(n)y(n)}, \quad \text{bounded by} \quad 0 \le g_p \le 1.2, \tag{33}$$

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].

5.7 Algebraic codebook structure and search

The algebraic codebook structure is based on interleaved single-pulse permutation (ISPP) design. 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 2.

Table 2: Potential positions of individual pulses in the algebraic codebook

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.

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:

$$x_2(n)=x(n)-\hat{g}_p y(n), \quad n=0,\ldots,39, \tag{34}$$

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:

$$A_k=\frac{(C_k)^2}{E_{Dk}}=\frac{\left(\mathbf{d}^t\mathbf{c}_k\right)^2}{\mathbf{c}_k^t\Phi\mathbf{c}_k} \tag{35}$$

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

$$d(n)=\sum_{i=n}^{39}x_2(i)h(i-n), \quad n=0,\ldots,39, \tag{36}$$

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

$$\phi(i,j)=\sum_{n=j}^{39}h(n-i)h(n-j), \quad (j\geq i). \tag{37}$$

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 (35) is given by:

$$C=\sum_{i=0}^{N_p-1}\vartheta_i d(m_i) \tag{38}$$

where m_i is the position of the i th pulse, ϑ_i is its amplitude, and N_p is the number of pulses *(N_p = 10)*. The energy in the denominator of equation (35) is given by:

$$E_D=\sum_{i=0}^{N_p-1}\phi(m_i,m_i)+2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1}\vartheta_i\vartheta_j\phi(m_i,m_j). \tag{39}$$

To simplify the search procedure, the pulse amplitudes are preset by the mere quantization of an appropriate signal. In this case the signal $b(n)$, which is a sum of the normalized $d(n)$ vector and normalized long-term prediction residual $res_{LTP}(n)$:

$$b(n)=\frac{res_{LTP}(n)}{\sqrt{\sum_{i=0}^{39} res_{LTP}(i)\, res_{LTP}(i)}}+\frac{d(n)}{\sqrt{\sum_{i=0}^{39} d(i)\, d(i)}}, \quad n=0,\ldots,39, \tag{40}$$

is used. 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 $s_b(n)=\mathrm{sign}[b(n)]$ 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, $\phi^{'}(i,j)=s_b(i)s_b(j)\phi(i,j)$. The correlation in equation (38) is now given by:

$$C=\sum_{i=0}^{N_p-1} d^{'}(m_i) \tag{41}$$

and the energy in equation (39) is given by:

$$E_D=\sum_{i=0}^{N_p-1}\phi^{'}(m_i,m_i)+2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1}\phi^{'}(m_i,m_j). \tag{42}$$

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 (35) 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 i0 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 i1 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 {i2,i3}, {i4,i5}, {i6,i7} and {i8,i9} 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 i1 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 $F_E(z)=1/(1-\beta z^{-T})$ is used, where T is the nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe, and β is a pitch gain. In the present document, β 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, $h(n)=h(n)+\beta h(n-T),\quad n=T,\ldots,39$.

The fixed codebook gain is then found by:

$$g_c=\frac{\mathbf{x}_2^t\mathbf{z}}{\mathbf{z}^t\mathbf{z}} \tag{43}$$

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

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

5.8 Quantization of the fixed codebook gain

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:

$$E(n)=10\log\left(\frac{1}{N}g_c^2\sum_{i=0}^{N-1}c^2(i)\right)-\overline{E}, \tag{45}$$

where $N=40$ is the subframe size, $c(i)$ is the fixed codebook excitation, and $\overline{E}=36$ dB is the mean of the innovation energy. The predicted energy is given by:

$$\tilde{E}(n)=\sum_{i=1}^{4}b_i\,\hat{R}(n-i), \tag{46}$$

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 (45) (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:

$$E_I=10\log\left(\frac{1}{N}\sum_{j=0}^{N-1}c^2(j)\right) \tag{47}$$

and then the predicted gain $g_c^{'}$ is found by:

$$g_c^{'}=10^{0.05(\tilde{E}(n)+\overline{E}-E_I)}. \tag{48}$$

A correction factor between the gain g_c and the estimated one $g_c^{'}$ is given by:

$$\gamma_{gc}=g_c\,/\,g_c^{'}. \tag{49}$$

Note that the prediction error is given by:

$$R(n)=E(n)-\tilde{E}(n)=20\log(\gamma_{gc}). \tag{50}$$

The correction factor γ_{gc} is quantified using a 5-bit codebook. The quantization table search is performed by minimizing the error:

$$E_Q=(g_c-\hat{\gamma}_{gc}g_c^{'})^2. \tag{51}$$

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

5.9 Memory update

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:

$$u(n)=\hat{g}_p v(n)+\hat{g}_c c(n), \quad n=0,\ldots,39, \tag{52}$$

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 $A(z/\gamma_1)/A(z/\gamma_2)$ 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 $A(z/\gamma_1)/A(z/\gamma_2)$ 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:

$$e_w(n)=x(n)-\hat{g}_p y(n)-\hat{g}_c z(n). \tag{53}$$

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 (53) for $n=30,\ldots,39$. This saves two filterings.

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 two quantified LSP vectors. The interpolation described in clause 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 clause 5.6.

2) **Decoding of the adaptive codebook gain:** The received index is used to readily find the quantified adaptive codebook gain, $\hat{g}_p$ from the quantization table.

3) **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 is less than the subframe size 40, the pitch sharpening procedure is applied which translates into modifying $c(n)$ by $c(n) = c(n) + \beta c(n-T)$, where β is the decoded pitch gain, $\hat{g}_p$, bounded by [0.0,1.0].

4) **Decoding of the fixed codebook gain:** The received index gives the fixed codebook gain correction factor $\hat{\gamma}_{gc}$. The estimated fixed codebook gain g'_c is found as described in clause 5.7. First, the predicted energy is found by:

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

and then the mean innovation energy is found by:

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

The predicted gain g'_c is found by:

$$g'_c = 10^{0.05(\tilde{E}(n)+\bar{E}-E_I)}. \tag{56}$$

The quantified fixed codebook gain is given by:

$$\hat{g}_c = \hat{\gamma}_{gc}\, g'_c \tag{57}$$

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

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \tag{58}$$

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:

$$\hat{u}(n) = \begin{cases} u(n) + 0.25\beta\hat{g}_p v(n), & \hat{g}_p > 0.5 \\ u(n), & \hat{g}_p \le 0.5 \end{cases} \tag{59}$$

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 η for the emphasized excitation is computed by:

$$\eta = \begin{cases} \sqrt{\dfrac{\sum_{n=0}^{39} u^2(n)}{\sum_{n=0}^{39} \hat{u}^2(n)}}, & \hat{g}_p > 0.5 \\ 1.0 & \hat{g}_p \le 0.5 \end{cases} \tag{60}$$

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

$$\hat{u}'(n) = \hat{u}(n)\eta \tag{61}$$

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

$$\hat{s}(n) = \hat{u}'(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i), \qquad n = 0,\ldots,39. \tag{62}$$

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

Post-processing consists of two functions: adaptive post-filtering and signal up-scaling.

6.2.1 Adaptive post-filtering

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:

$$H_f(z) = \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)} \tag{63}$$

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:

$$H_t(z) = (1 - \mu z^{-1}) \tag{64}$$

where $\mu = \gamma_t k_1'$ 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 $\hat{A}(z/\gamma_n)/\hat{A}(z/\gamma_d)$. k_1' is given by:

$$k_1' = \frac{r_h(1)}{r_h(0)}; \qquad r_h(i) = \sum_{j=0}^{L_h-i-1} h_f(j) h_f(j+i) \tag{65}$$

The post-filtering process is performed as follows. First, the synthesized speech $\hat{s}(n)$ is inverse filtered through $\hat{A}(z/\gamma_n)$ to produce the residual signal $\hat{r}(n)$. The signal $\hat{r}(n)$ is filtered by the synthesis filter $1/\hat{A}(z/\gamma_d)$. Finally, the signal at the output of the synthesis filter $1/\hat{A}(z/\gamma_d)$ 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 γ_{sc} for the present subframe is computed by:

$$\gamma_{sc} = \sqrt{\frac{\sum_{n=0}^{39} \hat{s}^2(n)}{\sum_{n=0}^{39} \hat{s}_f^2(n)}} \tag{66}$$

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

$$\hat{s}'(n) = \beta_{sc}(n)\hat{s}_f(n) \tag{67}$$

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

$$\beta_{sc}(n) = \alpha\beta_{sc}(n-1) + (1-\alpha)\gamma_{sc} \tag{68}$$

where α is a AGC factor with value of 0.9.

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

$$\gamma_t = \begin{cases} 0.8, & k_1' > 0 \\ 0, & otherwise \end{cases}. \tag{69}$$

6.2.2 Up-scaling

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 Variables, constants and tables in the C-code of the GSM EFR codec

The various components of the 12,2 kbit/s GSM enhanced full rate codec are described in the form of a fixed-point bit-exact ANSI C code, which is found in GSM 06.53 [6]. This C simulation is an integrated software of the speech codec, VAD/DTX, comfort noise and bad frame handler functions. In the fixed-point ANSI C simulation, all the computations are performed using a predefined set of basic operators.

Two types of variables are used in the fixed-point implementation. These two types are signed integers in 2's complement representation, defined by:

`Word16` 16 bit variables

`Word32` 32 bit variables

The variables of the **`Word16`** type are denoted *var1, var2,..., varn,* and those of type **`Word32`** are denoted *L_var1, L_var2,..., L_varn.*

7.1 Description of the constants and variables used in the C code

The ANSI C code simulation of the codec is, to a large extent, self-documented. However, a description of the variables and constants used in the code is given to facilitate the understanding of the code. The fixed-point precision (in terms of Q format, double precision (DP), or normalized precision) of the vectors and variables is given, along with the vectors dimensions and constant values.

Table 3 gives the coder global constants and table 4 describes the variables and vectors used in the encoder routine with their precision. Table 5 describes the fixed tables in the codec.

Table 3: Codec global constants

Parameter	Value	Description
L_TOTAL	240	size of speech buffer
L_WINDOW	240	size of LP analysis window
L_FRAME	160	size of speech frame
L_FRAME_BY2	80	half the speech frame size
L_SUBFR	40	size of subframe
M	10	order of LP analysis
MP1	11	M+1
AZ_SIZE	44	4*M+4
PIT_MAX	143	maximum pitch lag
PIT_MIN	18	minimum pitch lag
L_INTERPOL	10	order of sinc filter for interpolating the excitations is 2*L_INTERPOL*6+1
PRM_SIZE	57	size of vector of analysis parameters
SERIAL_SIZE	245	number of speech bits + bfi
MU	26214	tilt compensation filter factor (0.8 in Q15)
AGC_FAC	29491	automatic gain control factor (0.9 in Q15)

Table 4: Description of the coder vectors and variables

Parameter	Size	Precision	Description
speech	-80..159	Q0	speech buffer
wsp	-143..159	Q0	weighted speech buffer
exc	-(143+11)..159	Q0	LP excitation
F_gamma1	0..9	Q15	spectral expansion factors
F_gamma2	0..9	Q15	spectral expansion factors
lsp_old	0..9	Q15	LSP vector in past frame
lsp_old_q	0..9	Q15	quantified LSP vector in past frame
mem_syn	0..9	Q0	memory of synthesis filter
mem_w	0..9	Q0	memory of weighting filter (applied to input)
mem_wO	0..9	Q0	memory of weighting filter (applied to error)
error	-10..39	Q0	error signal (input minus synthesized speech)
r_1 & r_h	0..10	normalized DP	correlations of windowed speech (low and hi)
A_t	11x4	Q12	LP filter coefficients in 4 subframes
Aq_t	11x4	Q12	quantified LP filter coefficients in 4 subframes
Ap1	0..10	Q12	LP coefficients with spectral expansion
Ap2	0..10	Q12	LP coefficients with spectral expansion
lsp_new	0..9	Q15	LSP vector in 4th subframe
lsp_new_q	0..9	Q15	quantified LSP vector in 4th subframe
lsp_mid	0..9	Q15	LSP vector in 2nd subframe
lsp_mid_q	0..9	Q15	quantified LSP vector in 2nd subframe
code	0..39	Q12	fixed codebook excitation vector
h1	0..39	Q12	impulse response of weighted synthesis filter
xn	0..39	Q0	target vector in pitch search
xn2	0..39	Q0	target vector in algebraic codebook search
dn	0..39	scaled max < 8192	backward filtered target vector
y1	0..39	Q0	filtered adaptive codebook vector
y2	0..39	Q12	filtered fixed codebook vector
zero	0..39		zero vector
res2	0..39		long-term prediction residual
gain_pit	scalar	Q12	adaptive codebook gain
gain_code	scalar	Q0	algebraic codebook gain

Table 5: Codec fixed tables

Parameter	Size	Precision	Description
grid [ ]	61	Q15	grid points at which Chebyshev polynomials are evaluated
lag_h [ ] and lag_1 [ ]	10	DP	higher and lower parts of the lag window table
window_160_80 [ ]	240	Q15	1st LP analysis window
window_232_8 [ ]	240	Q15	2nd LP analysis window
table [ ] in Lsf_lsp ( )	65	Q15	table to compute cos(x) in Lsf_lsp ( )
slope [ ] in Lsp_lsf ( )	64	Q12	table to compute acos(x) in LSP_lsf ( )
table [ ] in Inv_sqrt ( )	49		table used in inverse square root computation
table [ ] in Log2 ( )	33		table used in base 2 logarithm computation
table [ ] in Pow2 ( )	33		table used in 2 to the power computation
mean_lsf [ ]	10	Q15	LSF means in normalized frequency [0.0, 0.5]
dico1_lsf [ ]	128 x 4	Q15	1st LSF quantizer in normalized frequency [0.0, 0.5]
dico2_lsf [ ]	256 x 4	Q15	2nd LSF quantizer in normalized frequency [0.0, 0.5]
dico3_lsf [ ]	256 x 4	Q15	3rd LSF quantizer in normalized frequency [0.0, 0.5]
dico4_lsf [ ]	256 x 4	Q15	4th LSF quantizer in normalized frequency [0.0, 0.5]
dico5_lsf [ ]	64 x 4	Q15	5th LSF quantizer in normalized frequency [0.0, 0.5]
qua_gain_pitch [ ]	16	Q14	quantization table of adaptive codebook gain
qua_gain_code [ ]	32	Q11	quantization table of fixed codebook gain
inter_6 [ ] in Interpol_6 ( )	25	Q15	interpolation filter coefficients in Interpol_6 ( )
inter_6 [ ] in Pred_lt_6 ( )	61	Q15	interpolation filter coefficients in Pred_lt_6 ( )
b [ ]	3	Q12	HP filter coefficients (numerator) in Pre_Process ( )
a [ ]	3	Q12	HP filter coefficients (denominator) in Pre_Process ( )
bitno [ ]	57	Q0	number of bits corresponding to transmitted parameters

Table 6: Source Encoder output parameters in order of occurrence and bit allocation within the speech frame of 244 bits/20 ms

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

8 Homing sequences

8.1 Functional description

The enhanced full rate speech codec is described in a bit-exact arithmetic to allow for easy type approval as well as general testing purposes of the enhanced full 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 table7. 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.

Table7: Parameter values for the decoder homing frame

Parameter	Value (LSB=b0)
LPC 1	0x0004
LPC 2	0x002F
LPC 3	0x00B4
LPC 4	0x0090
LPC 5	0x003E
LTP-LAG 1	0x0156
LTP-LAG 2	0x0036
LTP-LAG 3	0x0156
LTP-LAG 4	0x0036
LTP-GAIN 1	0x000B
LTP-GAIN 2	0x0001
LTP-GAIN 3	0x0000
LTP-GAIN 4	0x000B
FCB-GAIN 1	0x0003
FCB-GAIN 2	0x0000
FCB-GAIN 3	0x0000
FCB-GAIN 4	0x0000
PULSE 1_1	0x0000
PULSE 1_2	0x0001
PULSE 1_3	0x000F
PULSE 1_4	0x0001
PULSE 1_5	0x000D
PULSE 1_6	0x0000
PULSE 1_7	0x0003
PULSE 1_8	0x0000
PULSE 1_9	0x0003
PULSE 1_10	0x0000
PULSE 2_1	0x0008
PULSE 2_2	0x0008
PULSE 2_3	0x0005
PULSE 2_4	0x0008
PULSE 2_5	0x0001
PULSE 2_6	0x0000
PULSE 2_7	0x0000
PULSE 2_8	0x0001
PULSE 2_9	0x0001
PULSE 2_10	0x0000
PULSE 3_1	0x0000
PULSE 3_2	0x0000
PULSE 3_3	0x0000
PULSE 3_4	0x0000
PULSE 3_5	0x0000
PULSE 3_6	0x0000
PULSE 3_7	0x0000
PULSE 3_8	0x0000
PULSE 3_9	0x0000
PULSE 3_10	0x0000
PULSE 4_1	0x0000
PULSE 4_2	0x0000
PULSE 4_3	0x0000
PULSE 4_4	0x0000
PULSE 4_5	0x0000
PULSE 4_6	0x0000
PULSE 4_7	0x0000
PULSE 4_8	0x0000
PULSE 4_9	0x0000
PULSE 4_10	0x0000

8.3 Encoder homing

Whenever the enhanced full 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 DTX 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 DTX 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 in the TRAU uplink direction.

8.5 Encoder home state

In table 8, a listing of all the encoder state variables with their predefined values when in the home state is given.

Table 8: Initial values of the encoder state variables

File	Variable	Initial value
cod_12k2.c	old_speech[0:319]	All set to 0
	old_exc[0:153]	All set to 0
	old_wsp[0:142]	All set to 0
	mem_syn[0:9]	All set to 0
	mem_w[0:9]	All set to 0
	mem_w0[0:9]	All set to 0
	mem_err[0:9]	All set to 0
	ai_zero[11:50]	All set to 0
	hvec[0:39]	All set to 0
	lsp_old[0], lsp_old_q[0]	30000
	lsp_old[1], lsp_old_q[1]	26000
	lsp_old[2], lsp_old_q[2]	21000
	lsp_old[3], lsp_old_q[3]	15000
	lsp_old[4], lsp_old_q[4]	8000
	lsp_old[5], lsp_old_q[5]	0
	lsp_old[6], lsp_old_q[6]	-8000
	lsp_old[7], lsp_old_q[7]	-15000
	lsp_old[8], lsp_old_q[8]	-21000
	lsp_old[9], lsp_old_q[9]	-26000
levinson.c	old_A[0]	4096
	old_A[1:10]	All set to 0
pre_proc.c	y2_hi, y2_lo, y1_hi, y1_lo, x1, x0	All set to 0
q_plsf_5.c	past_r2_q[0:9]	All set to 0
q_gains.c	past_qua_en[0:3]	All set to -2381
	pred[0]	44
	pred[1]	37
	pred[2]	22
	pred[3]	12
dtx.c	txdtx_hangover	7
	txdtx_N_elapsed	0x7fff
	txdtx_ctrl	0x0003
	old_CN_mem_tx[0:5]	All set to 0
	lsf_old_tx[0:6][0]	1384
	lsf_old_tx[0:6][1]	2077
	lsf_old_tx[0:6][2]	3420
	lsf_old_tx[0:6][3]	5108
	lsf_old_tx[0:6][4]	6742
	lsf_old_tx[0:6][5]	8122
	lsf_old_tx[0:6][6]	9863
	lsf_old_tx[0:6][7]	11092
	lsf_old_tx[0:6][8]	12714
	lsf_old_tx[0:6][9]	13701
	gain_code_old_tx[0:27]	All set to 0
	L_pn_seed_tx	0x70816958
	buf_p_tx	0

Initial values for variables used by the VAD algorithm are listed in GSM 06.32 [4].

8.6 Decoder home state

In table 9, a listing of all the decoder state variables with their predefined values when in the home state is given.

Table 9: Initial values of the decoder state variables

File	Variable	Initial value
decoder.c	synth_buf[0:9]	All set to 0
dec_12k2.c	old_exc[0:153]	All set to 0
	mem_syn[0:9]	All set to 0
	lsp_old[0]	30000
	lsp_old[1]	26000
	lsp_old[2]	21000
	lsp_old[3]	15000
	lsp_old[4]	8000
	lsp_old[5]	0
	lsp_old[6]	-8000
	lsp_old[7]	-15000
	lsp_old[8]	-21000
	lsp_old[9]	-26000
	prev_bf	0
	state	0
agc.c	past_gain	4096
d_plsf_5.c	past_r2_q[0:9]	All set to 0
	past_lsf_q[0], lsf_p_CN[0], lsf_old_CN[0],lsf_new_CN[0]	1384
	past_lsf_q[1], lsf_p_CN[1], lsf_old_CN[1],lsf_new_CN[1]	2077
	past_lsf_q[2], lsf_p_CN[2], lsf_old_CN[2],lsf_new_CN[2]	3420
	past_lsf_q[3], lsf_p_CN[3], lsf_old_CN[3],lsf_new_CN[3]	5108
	past_lsf_q[4], lsf_p_CN[4], lsf_old_CN[4],lsf_new_CN[4]	6742
	past_lsf_q[5], lsf_p_CN[5], lsf_old_CN[5],lsf_new_CN[5]	8122
	past_lsf_q[6], lsf_p_CN[6], lsf_old_CN[6],lsf_new_CN[6]	9863
	past_lsf_q[7], lsf_p_CN[7], lsf_old_CN[7],lsf_new_CN[7]	11092
	past_lsf_q[8], lsf_p_CN[8], lsf_old_CN[8],lsf_new_CN[8]	12714
	past_lsf_q[9], lsf_p_CN[9], lsf_old_CN[9],lsf_new_CN[9]	13701
d_gains.c	pbuf[0:4]	All set to 410
	gbuf[0:4]	All set to 1
	past_gain_pit	0
	past_gain_code	0
	prev_gp	4096
	prev_gc	1
	gcode0_CN	0
	gain_code_old_CN	0
	gain_code_new_CN	0
	gain_code_muting_CN	0
	past_qua_en[0:3]	All set to -2381
	pred[0]	44
	pred[1]	37
	pred[2]	22
	pred[3]	12

(continued)

Table 9 (concluded): Initial values of the decoder state variables

File	Variable	Initial value
dtx.c	rxdtx_aver_period rxdtx_N_elapsed rxdtx_ctrl lsf_old_rx[0:6][0] lsf_old_rx[0:6][1] lsf_old_rx[0:6][2] lsf_old_rx[0:6][3] lsf_old_rx[0:6][4] lsf_old_rx[0:6][5] lsf_old_rx[0:6][6] lsf_old_rx[0:6][7] lsf_old_rx[0:6][8] lsf_old_rx[0:6][9] gain_code_old_rx[0:27] L_pn_seed_rx rx_dtx_state prev_SID_frames_lost buf_p_rx	7 0x7fff 0x0001 1384 2077 3420 5108 6742 8122 9863 11092 12714 13701 All set to 0 0x70816958 23 0 0
dec_lag6.c	old_T0	40
preemph.c	mem_pre	0
pstfilt2.c	mem_syn_pst[0:9] res2[0:39]	All set to 0 All set to 0

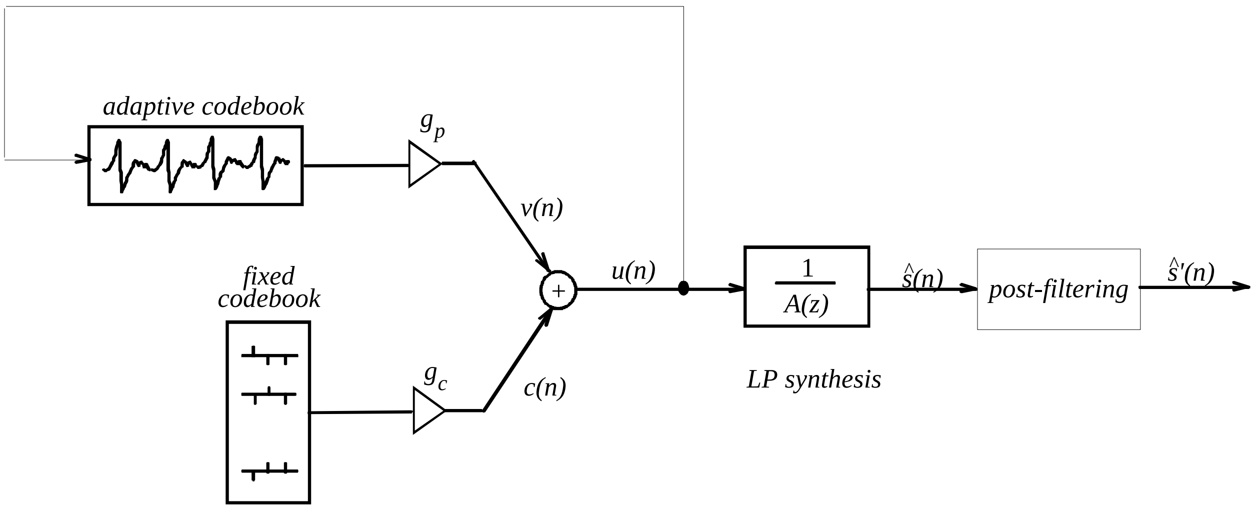

Figure 2: Simplified block diagram of the CELP synthesis model

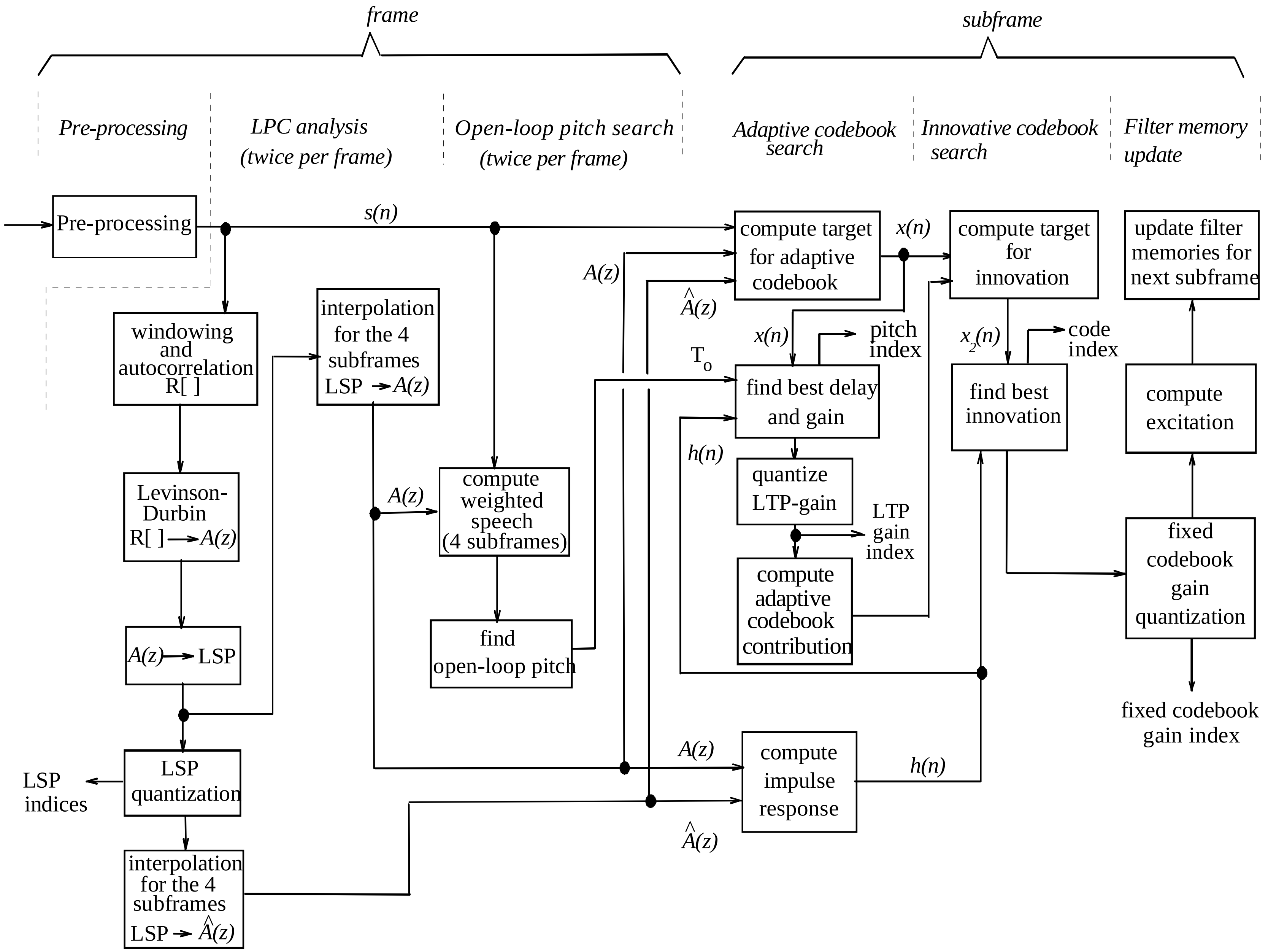
frame
subframe
Pre-processing
LPC analysis
(twice per frame)
Open-loop pitch search
(twice per frame)
Adaptive codebook search
Innovative codebook search
Filter memory update
Pre-processing
s(n)
compute target for adaptive codebook
x(n)
compute target for innovation
update filter memories for next subframe
A(z)
Â(z)
windowing and autocorrelation R[ ]
interpolation for the 4 subframes LSP → A(z)
x(n)
pitch index
x2(n)
code index
T0
find best delay and gain
find best innovation
compute excitation
h(n)
Levinson-Durbin R[ ] → A(z)
A(z)
compute weighted speech (4 subframes)
quantize LTP-gain
LTP gain index
compute adaptive codebook contribution
fixed codebook gain quantization
A(z) → LSP
find open-loop pitch
fixed codebook gain index
A(z)
LSP indices
LSP quantization
compute impulse response
h(n)
Â(z)
interpolation for the 4 subframes LSP → Â(z)

Figure 3: Simplified block diagram of the GSM enhanced full rate encoder

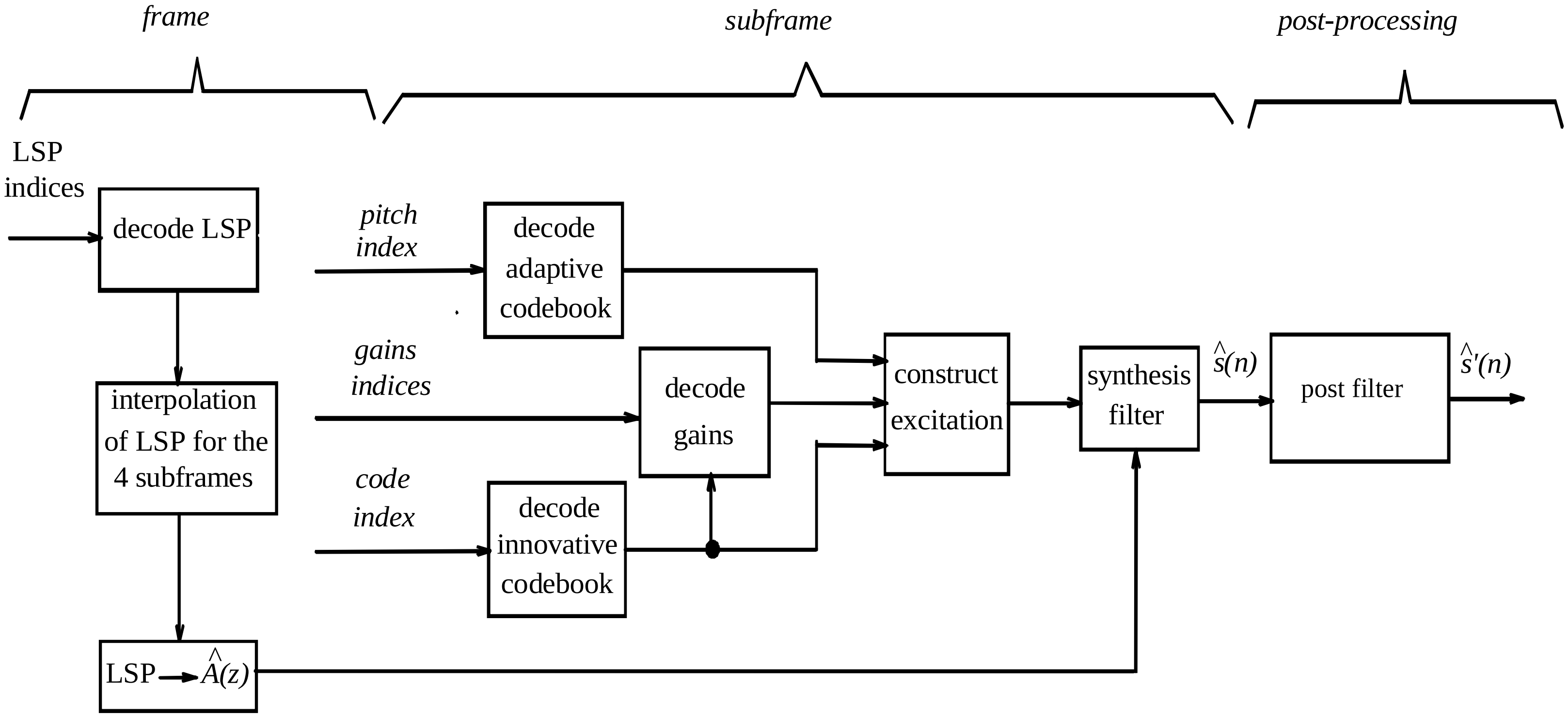

Figure 4: Simplified block diagram of the GSM enhanced full rate decoder

9 Bibliography

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

2) Y. Tohkura and F. Itakura, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. on ASSP*, vol. 26, no. 6, pp. 587-596, Dec. 1978.

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

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

5) F.K. Soong and B.H. Juang, "Line spectrum pair (LSP) and speech data compression", *Proc. ICASSP'84*, pp. 1.10.1-1.10.4, 1984.

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) C. Laflamme, J-P. Adoul, R. Salami, S. Morissette, and P. Mabilleau, "16 kpbs wideband speech coding technique based on algebraic CELP" *Proc. ICASSP'91*, pp. 13-16.

Annex A (informative): Change history

SMG	SPEC	CR	PH	VER	NEW_VE	SUBJECT
s23	06.60	A003	2	4.0.0	4.0.1	Vote 115 comments
s25	06.60	A005	2	4.0.1	4.1.0	Corrections to GSM 06.60
s28	06.60			4.1.0	6.0.0	Release 1997 version
s28	06.60	A007		6.0.0	7.0.0	Addition of mu-Law (PCS 1900)
	06.60			7.0.1	7.0.2	Update to Version 7.0.2 for Publication
s31	06.60			7.0.2	8.0.0	Release 1999 version
	06.60			8.0.0	8.0.1	Update to Version 8.0.1 for Publication

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
03-2001	11				Version for Release 4		4.0.0
06-2002	16				Version for Release 5	4.0.0	5.0.0
12-2004	26				Version for Release 6	5.0.0	6.0.0
06-2007	36				Version for Release 7	6.0.0	7.0.0

History

Document history		
V7.0.0	June 2007	Publication