ETSI TS 146 020 V18.0.0 (2024-05)

TECHNICAL SPECIFICATION

Digital cellular telecommunications system (Phase 2+) (GSM); Half rate speech; Half rate speech transcoding (3GPP TS 46.020 version 18.0.0 Release 18)

A GLOBAL INITIATIVE

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Reference

RTS/TSGS-0446020vi00

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

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. These shall be interpreted as being references to the corresponding ETSI deliverables.

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

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

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

The present document specifies the speech codec to be used for the GSM half rate channel for the digital cellular telecommunications system. The present document is part of a series covering the half rate speech traffic channels as described below:

GSM 06.02 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Half rate speech processing functions".

GSM 06.06 "Digital cellular telecommunications system (Phase 2+); Half rate speech; ANSI-C code for the GSM half rate speech codec".

GSM 06.07 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Test sequences for the GSM half rate speech codec".

GSM 06.20 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Half rate speech transcoding".

GSM 06.21 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Substitution and muting of lost frames for half rate speech traffic channels".

GSM 06.22 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Comfort noise aspects for half rate speech traffic channels".

GSM 06.41 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Discontinuous Transmission (DTX) for half rate speech traffic channels".

GSM 06.42 "Digital cellular telecommunications system (Phase 2+); Half rate speech; Voice Activity Detector (VAD) for half rate speech traffic channels".

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 specifies the speech codec to be used for the GSM half rate channel. It also specifies the test methods to be used to verify that the codec implementation complies with the present document.

The requirements are mandatory for the codec to be used either in GSM Mobile Stations (MS)s or Base Station Systems (BSS)s that utilize the half rate GSM speech traffic channel.

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 06.02: "Digital cellular telecommunications system (Phase 2+); Half rate speech; Half rate speech processing functions".

[2] GSM 06.06: "Digital cellular telecommunications system (Phase 2+); Half rate speech; ANSI-C code for the GSM half rate speech codec".

[3] GSM 06.07: "Digital cellular telecommunications system (Phase 2+); Half rate speech; Test sequences for the GSM half rate speech codec".

3 Definitions, symbols and abbreviations

3.1 Definitions

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

adaptive codebook: adaptive codebook is derived from the long term filter state. The lag value can be viewed as an index into the adaptive codebook.

adaptive pitch prefilter: in the GSM half rate speech decoder, this filter is applied to the excitation signal to enhance the periodicity of the reconstructed speech. Note that this is done prior to the application of the short term filter.

adaptive spectral postfilter: in the GSM half rate speech decoder, this filter is applied to the output of the short term filter to enhance the perceptual quality of the reconstructed speech.

allowable lags: set of lag values which may be coded by the GSM half rate speech encoder and transmitted to the GSM half rate speech decoder. This set contains both integer and fractional values (see table 3).

analysis window: for each frame, the short term filter coefficients are computed using the high pass filtered speech samples within the analysis window. The analysis window is 170 samples in length, and is centered about the last 100 samples in the frame.

basis vectors: set of M, M1, or M2 vectors of length Ns used to generate the VSELP codebook vectors. These vectors are not necessarily orthogonal.

closed loop lag search: process of determining the near optimal lag value from the weighted input speech and the long term filter state.

closed loop lag trajectory: for a given frame, the sequence of near optimal lag values whose elements correspond to each of the four subframes as determined by the closed loop lag search.

codebook: set of vectors used in a vector quantizer.

Codeword (OR Code): M, M1, or M2 bit symbol indicating the vector to be selected from a VSELP codebook.

Delta (LAG) code: four bit code indicating the change in lag value for a subframe relative to the previous subframe's coded lag. For frames in which the long term predictor is enabled (MODE 1, 2, or 3), the lag for subframe 1 is independently coded using eight bits, and delta codes are used for subframes 2, 3, and 4.

direct form coefficients: one of the formats for storing the short term filter parameters. All filters which are used to modify speech samples use direct form coefficients.

fractional lags: set of lag values having sub-sample resolution. Note that not every fractional lag value considered in the GSM half rate speech encoder is an allowable lag value.

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

harmonic noise weighting filter: this filter exploits the noise masking properties of the spectral peaks which occur at harmonics of the pitch frequency by weighting the residual error less in regions near the pitch harmonics and more in regions away from them. Note that this filter is only used when the long term filter is enabled (MODE = 1, 2 or 3).

high pass filter: this filter is used to de-emphasize the low frequency components of the input speech signal.

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

interpolating filter: FIR filter used to estimate sub-sample resolution samples, given an input sampled with integer sample resolution.

lag: long term filter delay. This is typically the pitch period, or a multiple or sub-multiple of it.

long term filter: this filter is used to generate the periodic component in the excitation for the current subframe. This filter is only enabled for MODE = 1, 2 or 3.

LPC coefficients: Linear Predictive Coding (LPC) coefficients is a generic descriptive term for describing the short term filter coefficients.

open loop lag search: process of estimating the near optimal lag directly from the weighted speech input. This is done to narrow the range of lag values over which the closed loop lag search shall be performed.

open loop lag trajectory: for a given frame, the sequence of near optimal lag values whose elements correspond to the four subframes as determined by the open loop lag search.

reflection coefficients: alternative representation of the information contained in the short term filter parameters.

residual: output signal resulting from an inverse filtering operation.

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

soft interpolation: process wherein a decision is made for each frame to use either interpolated or uninterpolated short term filter parameters for the four subframes in that frame.

soft interpolation bit: one bit code indicating whether or not interpolation of the short term parameters is to be used in the current frame.

spectral noise weighting filter: this filter exploits the noise masking properties of the formants (vocal tract resonances) by weighting the residual error less in regions near the formant frequencies and more in regions away from them.

subframe: time interval equal to 5 ms, or 40 samples at an 8 kHz sampling rate.

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

GSP0 vector quantizer: process of vector quantization, its intermediate parameters (GS and P0) for the coding of the excitation gains β and γ.

VSELP codebook: Vector-Sum Excited Linear Predictive (VSELP) codebook, used in the GSM half rate speech coder, wherein each codebook vector is constructed as a linear combination of the fixed basis vectors.

zero input response: output of a filter due to all 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)	Short term spectral filter.
α_i	The LPC coefficients.
$b_L(n)$	The output of the long term filter state (adaptive codebook) for lag L.
β	The long term filter coefficient.
C(z)	Second weighting filter.
e(n)	Weighted error signal
$f_j(i)$	The coefficients of the j^{th} phase of the 10th order interpolating filter used to evaluate candidate fractional lag values; i ranges from 0 to P_f-1.
$g_j(i)$	The coefficients of the j^{th} phase of the 6th order interpolating filter used to interpolate C's and G's as well as fractional lags in the harmonic noise weighting; i ranges from 0 to Pg-1.
γ	The gain applied to the vector(s) selected from the VSELP codebook(s).
H	A M2 bit code indicating the vector to be selected from the second VSELP codebook (when operating in mode 0).
I	A M or M1 bit code indicating the vector to be selected from one of the two first VSELP codebooks.
L	The long term filter lag value.
L_{max}	142 (samples), the maximum possible value for the long term filter lag.
L_{min}	21 (samples), the minimum possible value for the long term filter lag.
M	9, the number of basis vectors, and the number of bits in a codeword, for the VSELP codebook used in modes 1, 2, and 3.
M1	7, the number of basis vectors, and the number of bits in a codeword, for the first VSELP codebook used in mode 0.
M2	7, the number of basis vectors, and the number of bits in a codeword, for the second VSELP codebook used in mode 0.
MODE	A two bit code indicating the mode for the current frame (see annex A).
N_A	170, the length of the analysis window. This is the number of high pass filtered speech samples used to compute the short term filter parameters for each frame.
N_F	160, the number of samples per frame (at a sampling rate of 8 kHz).
N_p	10, the short term filter order.
N_S	40, the number of samples per subframe (at a sampling rate of 8 kHz).
P1	6, the number of bits in the prequantizer for the r1 - r3 vector quantizer.
P2	5, the number of bits in the prequantizer for the r4 - r6 vector quantizer.
P3	4, the number of bits in the prequantizer for the r7 - r10 vector quantizer.
P_f	The order of one phase of an interpolating filter used to evaluate candidate fractional lag values. P_f equals 10 for $j \neq 0$ and equal to 1 for $j = 0$.
P_g	The order of one phase of an interpolating filter, $f_j(n)$, used to interpolate C's and G's as well as fractional lags in the harmonic noise weighting, P_g equals 6.
pitch	The time duration between the glottal pulses which result when the vocal chords vibrate during speech production.
Q1	11, the number of bits in the r1 - r3 reflection coefficient vector quantizer.
Q2	9, the number of bits in the r4 - r6 reflection coefficient vector quantizer.
Q3	8, the number of bits in the r7 - r10 reflection coefficient vector quantizer.
R0	A five bit code used to indicate the energy level in the current frame.
r(n)	The long term filter state (the history of the excitation signal); $n < 0$
$r_L(n)$	The long term filter state with the adaptive codebook output for lag L appended.

s'(n)	Synthesized speech.
W(z)	Spectral weighting filter.
λ_{hnw}	The harmonic noise weighting filter coefficient.
ξ	The adaptive pitch prefilter coefficient.
$\lceil x \rceil$	Ceiling function: the largest integer y where y < x + 1,0.
$\lfloor x \rfloor$	Floor function: the largest integer y where y ≤ x.
$\sum_{i=j}^{K} x(i)$	Summation: x(j)+x(j+1)+...+x(K).
$\prod_{i=j}^{K} x(i)$	Product: x(j)(x(j+1))....(x(K))
max(x,y)	Find the larger of two numbers x and y.
min(x,y)	Find the smaller of two numbers x and y.
round(x)	Round the non-integer x to the closest integer $y: y = \lfloor x + 0,5 \rfloor$y: y=x+0,5.

3.3 Abbreviations

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

AFLAT	Autocorrelation Fixed point LAttice Technique
CELP	Code Excited Linear Prediction
FLAT	Fixed Point Lattice Technique
LTP	Long Term Predictor
SST	Spectral Smoothing Technique
VSELP	Vector-Sum Excited Linear Prediction

4 Functional description of the GSM half rate speech codec

The GSM half rate codec uses the VSELP (Vector-Sum Excited Linear Prediction) algorithm. The VSELP algorithm is an analysis-by-synthesis coding technique and belongs to the class of speech coding algorithms known as CELP (Code Excited Linear Prediction).

The GSM half rate codec's encoding process is performed on a 20 ms speech frame at a time. A speech frame of the sampled speech waveform is read and based on the current waveform and the past history of the waveform, the codec encoder derives 18 parameters that describe it. The parameters extracted are grouped into the following three general classes:

- energy parameters (R0 and GSP0);
- spectral parameters (LPC and INT_LPC);
- excitation parameters (LAG and CODE).

These parameters are quantized into 112 bits for transmission as described in annex A and their order of occurrence over Abis is given in annex B.

The GSM half rate codec is an analysis-by-synthesis codec, therefore the speech decoder is primarily a subset of the speech encoder. The quantized parameters are decoded and a synthetic excitation is generated using the energy and excitation parameters. The synthetic excitation is then filtered to provide the spectral information resulting in the generation of the synthesized speech (see figure 1).

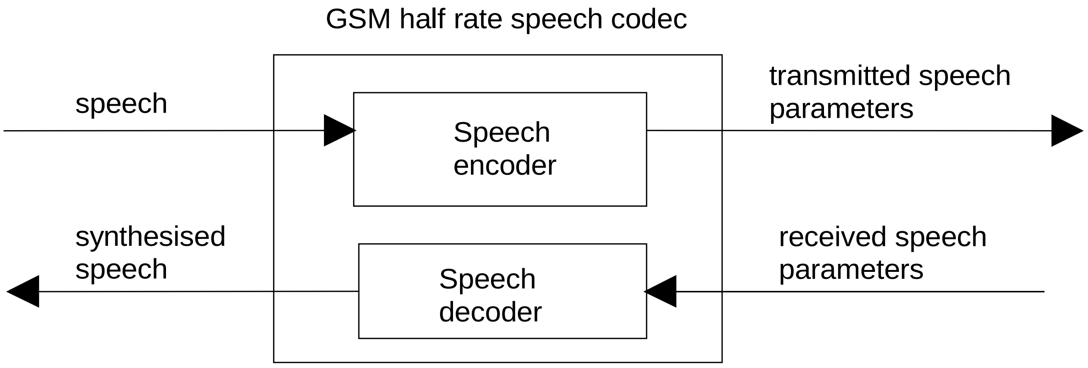
GSM half rate speech codec
speech
Speech encoder
transmitted speech parameters
synthesised speech
Speech decoder
received speech parameters

Figure 1: Block diagram of the GSM half rate speech codec

The ANSI-C code that describes the GSM half rate speech codec is given in GSM 06.06 [2] and the test sequences in GSM 06.07 [3] (see clause 5 for the codec homing test sequences).

4.1 GSM half rate speech encoder

The GSM half rate speech encoder uses an analysis by synthesis approach to determine the code to use to represent the excitation for each subframe. The codebook search procedure consists of trying each codevector as a possible excitation for the Code Excited Linear Predictive (CELP) synthesizer. The synthesized speech s'(n) is compared against the input speech and a difference signal is generated. This difference signal is then filtered by a spectral weighting filter, W(z), (and possibly a second weighting filter, C(z)) to generate a weighted error signal, e(n). The power in e(n) is computed. The codevector which generates the minimum weighted error power is chosen as the codevector for that subframe. The spectral weighting filter serves to weight the error spectrum based on perceptual considerations. This weighting filter is a function of the speech spectrum and can be expressed in terms of the α parameters of the short term (spectral) filter.

$$W(z) = \frac{1 - \sum_{i=1}^{N_p} \alpha_i z^{-i}}{1 - \sum_{i=1}^{N_p} \tilde{\alpha}_i z^{-i}} \qquad (1)$$

The computation of the $\tilde{\alpha}_i$ coefficients is described in subclause 4.1.7.

The second weighting filter C(z), if used, is a harmonic weighting filter and is used to control the amount of error in the harmonics of the speech signal. If the weighting filter(s) are moved to both input paths to the subtracter, an equivalent configuration is obtained as shown in figure 2.

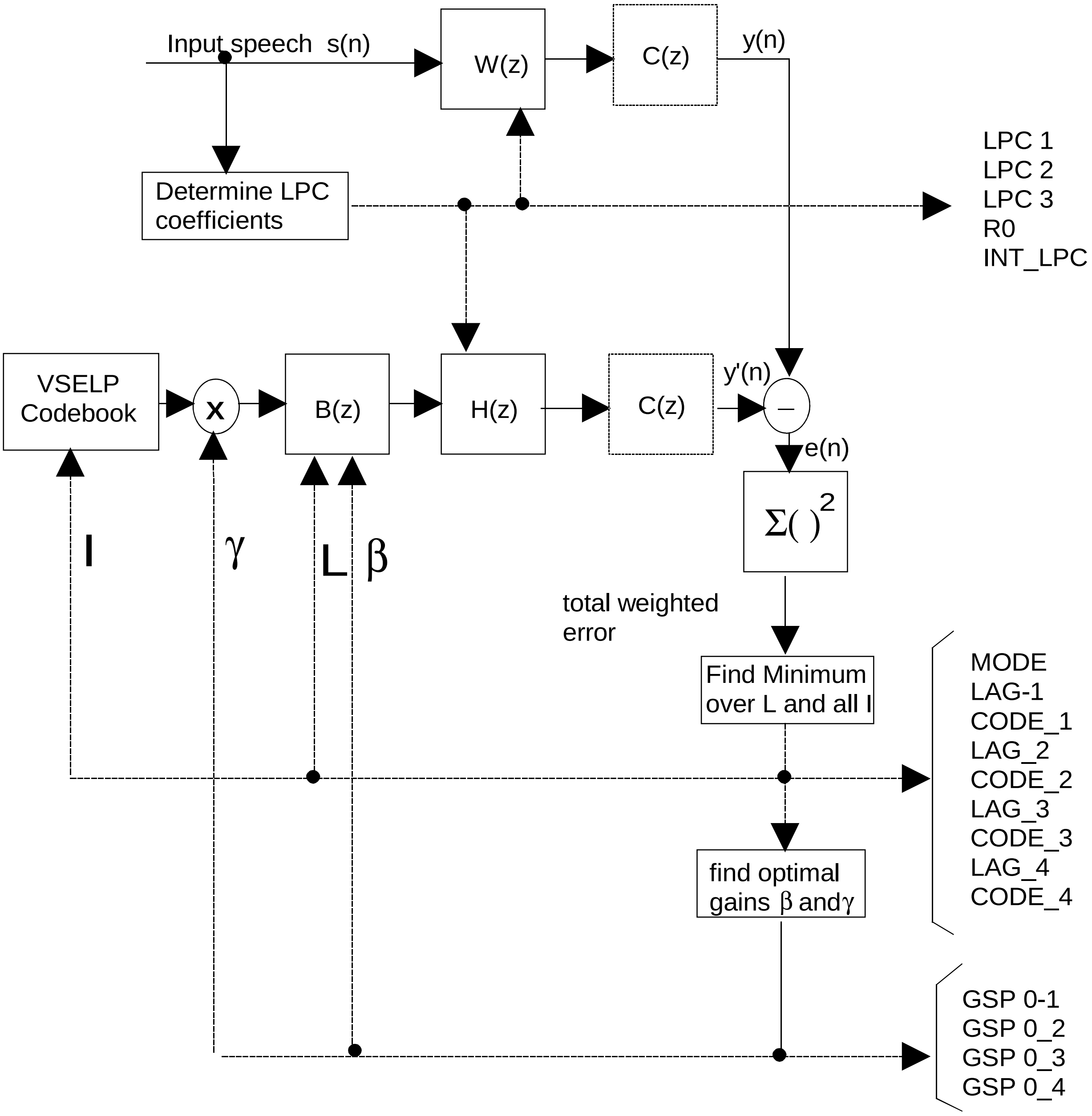

Figure 2: Block diagram of the GSM half rate speech encoder (MODE = 1,2 and 3)

Here H(z) is the combination of A(z), the short term (spectral) filter, and W(z), the spectral weighting filter. These filters are combined since the denominator of A(z) is cancelled by the numerator of W(z).

$$H(z) = \frac{1}{1 - \sum_{i=1}^{N_p} \tilde{\alpha}_i z^{-i}} \qquad (2)$$

There are two approaches that can be used for calculating the gain, γ. The gain can be determined prior to codebook search based on residual energy. This gain would then be fixed for the codebook search. Another approach is to optimize the gain for each codevector during the codebook search. The codevector which yields the minimum weighted error would be chosen and its corresponding optimal gain would be used for γ. The latter approach generally yields better results since the gain is optimized for each codevector. This approach also implies that the gain term needs to be updated at the subframe rate. The optimal code and gain for this technique can be computed as follows.

The input speech is first filtered by a high pass filter as described in subclause 4.1.1. The short term filter parameters are computed from the filtered input speech once per frame. A fast fixed point covariance lattice technique is used. Subclauses 4.1.3 and 4.1.4 describes in detail how the short term parameters are determined and quantized. An overall frame energy is also computed and coded once per frame. Once per frame, one of the four voicing modes is selected. If MODE≠0, the long term predictor is used and the long term predictor lag, L, is updated at the subframe rate. L and a VSELP codeword are selected sequentially. Each is chosen to minimize the weighted mean square error. The long-term filter coefficient, β, and the codebook gain, γ, are optimized jointly. Subclause 4.1.8 describes the technique for selecting from among the voicing modes and, if one of voiced modes is chosen, determining the long-term filter lag. Subclause 4.1.10 describes an efficient technique for jointly optimizing β, γ and the codeword selection. Subclause 4.1.10 also includes the description of the fast VSELP codebook search technique. The β and γ parameters are transformed to equivalent parameters using the frame energy term, and are vector quantized every subframe. The coding of the frame energy and the β and γ parameters is described in subclause 4.1.11.

4.1.1 High-pass filter

The 13 bit linear Pulse Code Modulated (PCM) input speech, x(n), is filtered by a fourth order pole-zero high pass filter. This filter suppresses the frequency components of the input speech which are below 120 Hz. The filter is implemented as a cascade of two second-order Infinite Impulse Response (IIR) filters. Incorporated into the filter coefficients is a gain of 0,5. The difference equation for the first filter is:

$$\tilde{y}(n)=\sum_{i=0}^{2}b_{1,i}x(n-i)+\sum_{j=1}^{2}a_{1,j}\tilde{y}(n-j) \qquad (3)$$

where:

b_{10} = 0,335052

b_{11} = -0,669983 a_{11} = 0,926117

b_{12} = 0,335052 a_{12} = -0,429413

The difference equation for the second filter is:

$$y(n)=\sum_{i=0}^{2}b_{2,i}\tilde{y}(n-i)+\sum_{j=1}^{2}a_{2,j}y(n-j) \qquad (4)$$

where:

b_{20} = 0,335052
b_{21} = -0,669434 a_{21}= 0,965332
b_{22} = 0,335052 a_{22} = -0,469513

4.1.2 Segmentation

A sample buffer containing the previous 195 input high pass filtered speech samples, y(n), is shifted so that the oldest 160 samples are shifted out while the next 160 input samples are shifted in. The oldest 160 samples in the buffer correspond to the next frame of samples to be encoded. The analysis interval comprises the most recent 170 samples in the buffer. The samples in the buffer are labelled as s(n) where 0 ≤ n ≤ 194 and s(0) is the first (oldest) sample.

4.1.3 Fixed Point Lattice Technique (FLAT)

Let r_j represent the j^{th} reflection coefficient. The FLAT algorithm for the determination of the reflection coefficients is stated as follows:

STEP 1 Compute the covariance (autocorrelation) matrix from the input speech:

$$\phi(i,k) = \sum_{n=N_p}^{N_A} s(n+24-i)s(n+24-k) \qquad 0 \le i,\ k \le N_p \qquad (5)$$

STEP 2 The ϕ(i,k) array is modified by windowing

$$\phi'(i,k) = \phi(i,k)w(|i-k|) \qquad 0 \le i,\ k \le N_p \qquad (6)$$

STEP 3

$$F_0(i,k) = \phi'(i,k) \qquad 0 \le i,\ k \le N_p - 1 \qquad (7)$$

$$B_0(i,k) = \phi'(i+1,k+1) \qquad 0 \le i,\ k \le N_p - 1 \qquad (8)$$

$$C_0(i,k) = \phi'(i,k+1) \qquad 0 \le i,\ k \le N_p - 1 \qquad (9)$$

STEP 4 set j = 1

STEP 5 Compute r_j

$$r_j = -2\frac{C_{j-1}(0,0) + C_{j-1}(N_p - j, N_p - j)}{F_{j-1}(0,0) + B_{j-1}(0,0) + F_{j-1}(N_p - j, N_p - j) + B_{j-1}(N_p - j, N_p - j)} \qquad (10)$$

STEP 6 If j = N_P then done.

STEP 7 Update F_j(i,k), B_j(i,k), C_j(i,k) $0 \le$ i, k $\le N_P$-j-1

$$F_j(i,k) = F_{j-1}(i,k) + r_j\left(C_{j-1}(i,k) + C_{j-1}(k,i)\right) + r_j^2 B_{j-1}(i,k) \qquad (11)$$

$$B_j(i,k) = B_{j-1}(i+1,k+1) + r_j\left(C_{j-1}(i+1,k+1) + C_{j-1}(k+1,i+1)\right) + r_j^2 F_{j-1}(i+1,k+1), \qquad (12)$$

$$C_j(i,k) = C_{j-1}(i,k+1) + r_j\left(B_{j-1}(i,k+1) + F_{j-1}(i,k+1)\right) + r_j^2 C_{j-1}(k+1,i) \qquad (13)$$

STEP 8 j = j+1

STEP 9 go to step 5.

The windowing coefficients, w(|i-k|), are found in the table 1.

Table 1: Windowing coefficients

w(0)	0,998966		w(5)	0,974915
w(1)	0,996037		w(6)	0,969054
w(2)	0,991663		w(7)	0,963060
w(3)	0,986399		w(8)	0,956796
w(4)	0.980722		w(9)	0,950127

This algorithm can be simplified by noting that the ϕ', F and B matrices are symmetric such that only the upper triangular part of the matrices need to be computed or updated. Also, step 7 is done so that $F_j(i,k)$, $B_j(i\text{-}1,k\text{-}1)$, $C_j(i,k\text{-}1)$, and $C_j(k,i\text{-}1)$ are updated together and common terms are computed once and the recursion is done in place.

4.1.4 Spectral quantization

A three segment vector quantizer of the reflection coefficients is employed. A reduced complexity search technique is used to select the vector of reflection coefficients for each segment. The reflection coefficient vector quantizer codebooks are stored in compressed form to minimize their memory requirements.

The three segments of the vector quantizer span reflection coefficients r_1 r_3, r_4 r_6, and r_7 - r_{10} respectively. The bit allocations for the vector quantizer segments are:

Q_1	11 bits
Q_2	9 bits
Q_3	8 bits

A reflection coefficient vector prequantizer is used at each segment. The prequantizer size at each segment is:

P_1	6 bits
P_2	5 bits
P_3	4 bits

At a given segment, the residual error due to each vector from the prequantizer is computed and stored in temporary memory. This list is searched to identify the four prequantizer vectors which have the lowest distortion. The index of each selected prequantizer vector is used to calculate an offset into the vector quantizer table at which the contiguous subset of quantizer vectors associated with that prequantizer vector begins. The size of each vector quantizer subset at the k-th segment is given by:

$$S_k = \frac{2^{Q_k}}{2^{P_k}} \qquad (14)$$

The four subsets of quantizer vectors, associated with the selected prequantizer vectors, are searched for the quantizer vector which yields the lowest residual error. Thus at the first segment, 64 prequantizer vectors and 128 quantizer vectors are evaluated, 32 prequantizer vectors and 64 quantizer vectors are evaluated at the second segment, and 16 prequantizer vectors and 64 quantizer vectors are evaluated at the third segment.

4.1.4.1 Autocorrelation Fixed Point Lattice Technique (AFLAT)

An autocorrelation version of the FLAT algorithm, AFLAT, is used to compute the residual error energy for a reflection coefficient vector being evaluated. Compute the autocorrelation sequence R(i), from the optimal reflection coefficients, r_j, over the range $0 \le i \le N_p$.

STEP 1 Define the initial conditions for the AFLAT recursion:

$$\overline{P}_0(i) = R(i), \quad 0 \le i \le N_p - 1 \qquad (15)$$

$$\overline{V}_0(i) = R(|i+1|), \quad 1 - N_p \le i \le N_p - 1 \qquad (16)$$

STEP 2 Initialize k, the vector quantizer segment index:

k = 1 (17)

STEP 3 Let $I_l(k)$ be the index of the first lattice stage in the k-th segment, and $I_h(k)$ be the index of the last lattice stage in the k-th segment.

STEP 4 Initialize j, the index of the lattice stage, to point to the beginning of the k-th segment:

$$j = I_l(k) \qquad (18)$$

STEP 5 Set the initial conditions P_{j-1} and V_{j-1} to:

$$P_{j-1}(i) = \overline{P}_{j-1}(i), \quad 0 \le i \le I_h(k) - I_l(k) \qquad (19)$$

$$V_{j-1}(i) = \overline{V}_{j-i}(i), \quad -I_h(k) + I_l(k) \le i \le I_h(k) - I_l(k) \qquad (20)$$

STEP 6 Compute the values of V_j and P_j arrays using:

$$P_j(i) = \left(1 + \hat{r}_j^2\right) P_{j-1}(i) + \hat{r}_j \left[V_{j-1}(i) + V_{j-1}(-i)\right], \quad 0 \le i \le I_h(k) - j - 1 \qquad (21)$$

$$V_j(i) = V_{j-i}(i+1) + \hat{r}_j^2 V_{j-i}(-i-1) + 2\hat{r}_j P_{j-1}(|i+1|), \quad 1 + j - N_p \le i \le N_p - j - 1 \quad (22)$$

STEP 7 Increment j:

j = j+1

STEP 8 If j < I_h(k) go to STEP 6.

STEP 9 The residual error out of lattice stage I_h(k), given the reflection coefficient vector $\hat{r}$, is computed using equation (21):

$$E_r = P_{I_h(k)}(0) \qquad (23)$$

STEP 10 Using the AFLAT recursion outlined, the residual error due to each vector from the prequantizer at the k-th segment is evaluated, the four subsets of quantizer vectors to be searched are identified, and residual error due to each quantizer vector from the selected four subsets is computed. The index of $\tilde{r}$, the quantizer vector which minimized E_r over all the quantizer vectors in the four subsets, is encoded with Q_k bits.

STEP 11 If k < 3 then the initial conditions for doing the recursion at segment k+1 need to be computed. Set j, the lattice stage index, equal to:

$$j = I_l(k) \qquad (24)$$

STEP 12 Compute:

$$\overline{P}_j(i) = \left(1 + \tilde{r}_j^2\right) \overline{P}_{j-1}(i) + \tilde{r}_j \left[\overline{V}_{j-1}(i) + \overline{V}_{j-1}(-i)\right], \quad 0 \le i \le N_p - j - 1 \qquad (25)$$

$$\overline{V}_j(i) = \overline{V}_{j-i}(i+1) + \tilde{r}_j^2 \overline{V}_{j-i}(-i-1) + 2\tilde{r}_j \overline{P}_{j-1}(|i+1|), \quad 1 + j - N_p \le i \le N_p - j - 1 \quad (26)$$

STEP 13 Increment j,

j = j+1

STEP 14 If j ≤ I_h(k) go to STEP 12

STEP 15 Increment k, the vector quantizer segment index:

k=k+1

STEP 16 If k ≤ 3 go to STEP 4.

Otherwise, the indices of the reflection coefficient vectors for the three segments have been chosen, and the search of the reflection coefficient vector quantizer is terminated.

To minimize the storage requirements for the reflection coefficient vector quantizer, eight bit codes for the individual reflection coefficients are stored in the vector quantizer table, instead of the actual reflection coefficient values. The codes are used to look up the values of the reflection coefficients from a scalar quantization table with 256 entries.

4.1.5 Frame energy calculation and quantization

The unquantized value of R0, R(0), is computed during the computation of the short term predictor parameters.

$$R(0) = \frac{\phi(0,0) + \phi(10,10)}{320} \qquad (27)$$

where ϕ(i,k) is defined by equation (5). R(0) is then converted into dB relative to full scale (full scale, R_{max}, is defined as the square of the maximum sample amplitude).

$$R_{dB} = 10 \log_{10}\left(\frac{R(0)}{R_{\max}}\right) \qquad (28)$$

R_{dB} is then quantized to 32 levels. The 32 quantized values for R_{dB} range from a minimum of -66 (corresponding to a code of 0 for R0) to a maximum of -4 (corresponding to a code of 31 for R0). The step size of the quantizer is 2 (2 dB steps). R0 is chosen as:

R0 which minimizes abs(R0 - (R_{dB} + 66)/2) (29)

where R0 can take on the integer values from 0 to 31 corresponding to the 32 codes for R0.

Decoding of the R0 code is given by:

$$R(0) = R_{\max} 10^{((2R0)-66)/10} \qquad (30)$$

4.1.6 Soft interpolation of the spectral parameters

Interpolation of the short term filter parameters improves the performance of the GSM half rate encoder. The direct form filter coefficients (α_i's), which correspond to quantized reflection coefficients, are the spectral parameters used for interpolation. The GSM half rate speech encoder uses either an interpolated set of α_i's or an uninterpolated set of α_i's, choosing the set which gives better prediction gain for the frame.

Two sets of LPC coefficient vectors are generated: the first corresponds to the interpolated coefficients, the second to the uninterpolated coefficients. The frame's speech samples are inverse filtered using each of the two coefficient sets, and the residual frame energy corresponding to each set is computed. The coefficient set yielding the lower frame residual energy is then selected to be used. If the residual energies are equal, the uninterpolated coefficient set is used. INT_LPC, a soft interpolation bit, is set to 1 when interpolation is selected or to 0 otherwise.

To generate the interpolated coefficient set, the coder interpolates the α_i's for the first, second, and third subframes of each frame. The fourth subframe uses the uninterpolated α_i's for that frame.

The interpolation is done as follows. Let $\alpha_{i,L}$ be the direct-form LPC coefficients corresponding to the last frame, $\alpha_{i,C}$ be the direct-form LPC coefficients corresponding to the current frame, and Del to be the interpolation curve used. The interpolated direct-form LPC coefficient vector at the j-th subframe of the current frame, $\alpha_{i,j}$, is given by:

$$\alpha_{i,j} = \alpha_{i,L} + Del(j, INT\_SOFT)(\alpha_{i,c} - \alpha_{i,L}), \qquad 1 \le i \le Np,\ 1 \le j \le 4 \qquad (31)$$

The values of the interpolation curve Del are given in table 2.

Table 2: Values of the interpolation curve Del

j	Del(j,0)	Del(j,1)
1	0,0	0,30
2	1,0	0,62
3	1,0	0,92
4	1,0	1,00

From this point on, the subframe index j is omitted for simplicity when referring to $\alpha_{i,j}$ coefficients, although it is implied. For interpolated subframes, the α_i's are converted to reflection coefficients to check for filter stability. If the resulting filter is unstable, then uninterpolated coefficients are used for that subframe. The uninterpolated coefficients used for subframe 1 are the previous frame's coefficients. The uninterpolated coefficients used for subframes 2, 3, and 4 are the current frame's coefficients.

4.1.7 Spectral noise weighting filter coefficients

To exploit the noise masking potential of the formants, spectral noise weighting is applied. The computation of the $\tilde{\alpha}_i$ coefficients, used by spectral noise weighting filters W(z) and H(z), is now described. Define an impulse sequence δ(n) over N_s samples:

$$\delta(0) = 1,0$$
$$\delta(n) = 0,0 \qquad (32)$$

where $1 \le n \le N_s-1$ and $h_3(n)$ is the zero-state response of the cascade of three filters to δ(n). The three filters are an LPC synthesis filter, an inverse filter using a weighting factor of 0,93 and a synthesis filter with a weighting factor of 0,7. In equation form:

$$h_1(n) = \delta(n) + \sum_{i=1}^{Np} \alpha_i\, h_1(n-i) \qquad 0 \le n \le N_s-1 \qquad (33)$$

$$h_2(n) = h_1(n) - \sum_{i=1}^{Np} (0,93)^i\, \alpha_i\, h_1(n-i) \qquad 0 \le n \le N_s-1 \qquad (34)$$

$$h_3(n) = h_2(n) + \sum_{i=1}^{Np} (0,7)^i\, \alpha_i h_3(n-i), \qquad 0 \le n \le N_s-1 \qquad (35)$$

where α_i's are the direct form LP coefficients. The autocorrelation sequence of $h_3(n)$ is calculated using:

$$R_{h_3}(i) = \sum_{n=i}^{Ns-1} h_3(n)\, h_3(n-i), \qquad 0 \le i \le N_p \qquad (36)$$

From $R_{h_3}(i)$ the reflection coefficients which define the combined spectrally noise weighted synthesis filter are computed using the AFLAT recursion once per frame.

STEP 1 Define the initial conditions for the AFLAT recursion:

$$P_0(i) = R_{h_3}(i), \qquad 0 \le i \le N_p-1 \qquad (37)$$

$$V_0(i) = R_{h_3}(|i+1|), \qquad 1-N_p \le i \le N_p-1 \qquad (38)$$

STEP 2 Initialize j, the index of the lattice stage, to point to the first lattice stage:

j = 1

STEP 3 Compute r_j, the j-th reflection coefficient, using:

$$r_j = -\frac{V_{j-1}(0)}{P_{j-1}(0)} \qquad (39)$$

STEP 4 Given r_j, update the values of V_j and P_j arrays using:

$$P_j(i) = \left(1+r_j^2\right) P_{j-1}(i) + r_j\left[V_{j-1}(i) + V_{j-1}(-i)\right], \qquad 0 \le i \le N_p - j - 1 \qquad (40)$$

$$V_j(i) = V_{j-1}(i+1) + r_j^2\, V_{j-1}(-i-1) + 2r_j\, P_{j-1}(|i+1|), \qquad 1 + j - N_p \le i \le N_p - j - 1 \; (41)$$

STEP 5 Increment j:

j = j+1

STEP 6 If $j \le N_p$ go to STEP 3, otherwise all N_p reflection coefficients have been obtained.

STEP 7 The reflection coefficients, r_j, are then converted to direct-form LPC filter coefficients, $\tilde{\alpha}_i$ for implementing the combined spectrally noise weighted synthesis filter H(z) and the filter W(z).

The method for the spectral noise weighting filter coefficient update mimicks how the direct form LPC filter coefficients are updated at subframes of a frame (subclause 4.1.6). No stability check of interpolated spectral noise weighting filter coefficients is done at subframes 1, 2, or 3 if the interpolation flag, INT_LPC="1", but if uninterpolated coefficients are used at subframes 1, 2, and/or 3 due to instability of the unweighted coefficients (INT_LPC = "0"), uninterpolated weighting filter coefficients are also used at those subframes.

4.1.8 Long Term Predictor lag determination

Figure 3 illustrates that the long term lag optimization looks just like a codebook search where the codebook is defined by the long term filter state and the specific vector in the codebook is pointed to by the long term predictor lag, L. The input p(n) is the weighted input speech for the subframe minus the zero input response of just the H(z) filter.

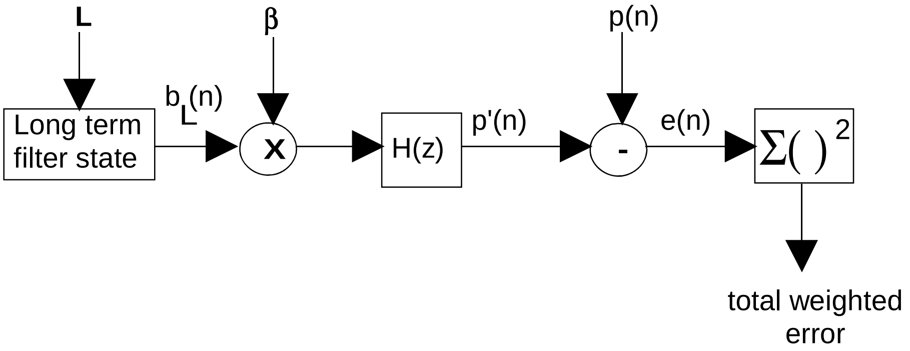

Figure 3: Long term predictor lag search

The GSM half rate speech encoder uses a combination of open loop and closed loop techniques in choosing the long term predictor lag. First an open loop search is conducted to determine "candidate" lags at each subframe. Then at most, two best candidate lags at each subframe are selected, with each serving as an anchor point for constructing an open loop frame lag trajectory, subject to a maximum delta coding constraint. The frame lag trajectory which minimizes the open loop LTP spectrally weighted error energy for the frame is then chosen. The open loop LTP prediction gains corresponding to the winning trajectory are used to select the voicing mode 1, 2 or 3. If MODE≠0, the closed loop lag evaluation is initiated. The winning trajectory has associated with it a list of lags to be searched closed loop at each subframe.

It is possible to allow L to take on fractional values, thus increasing the resolution, and in turn the performance, of the adaptive codebook. Table 3 shows the allowable lags.

Table 3: Allowable lags

Range	Resolution	Number of lags in range
21 to 22 2/3	1/3	6
23 to 34 5/6	1/6	72
35 to 49 2/3	1/3	45
50 to 89 1/2	1/2	80
90 to 142	1	53

The resolution of the long term filter state may be increased by upsampling and filtering the state. In this implementation, a non-causal, zero-phase Finite Impulse Response (FIR) filter is used. Where needed, the future samples for the non-causal filtering operation are replaced by the output of the predictor.

4.1.8.1 Open loop long term search initialization

An open-loop lag search is done to narrow the range of lags over which a closed-loop search will eventually be performed.

The first steps of the open-loop subframe lag search are as follows:

STEP 1 Initialize the subframe counter

m=1

STEP 2 The autocorrelation sequence of y(n) the input speech, s(n), filtered by W, is calculated for all allowable integer lags, and for a few integer lags below and above the lower and upper limits for the current subframe.

$$C(k,m) = \sum_{n=0}^{N_s-1} y\left(n+(m-1)N_s\right) y\left(n+(m-1)N_s-k\right), \quad L_{\min} - \frac{P_g}{2} \le k \le L_{\max} + \frac{P_g}{2} - 1 \quad (42)$$

where L_{min} = 21 and L_{max} = 142.

The value P_g is the order of one phase of the interpolating FIR filter used to interpolate the correlations. The energy of y(n) for the subframe is computed:

$$G(k,m) = \sum_{n=0}^{N_s-1} y^2\left(n+(m-1)N_s-k\right), \qquad L_{\min} - \frac{P_g}{2} \le k \le L_{\max} + \frac{P_g}{2} - 1 \quad (43)$$

STEP 3 These arrays, C(k,m) and G(k,m), are searched for the integer lag which maximizes C^2(k,m)/G(k,m) where C(k,m) and G(k,m) need to be greater than 0.

STEP 4 If a valid maximum is found in step 3, the values for the lag, C, and G corresponding to the maximum are retained in the arrays as L_{peak}(0,m), C_{peak}(0,m), and G_{peak}(0,m).

Otherwise, $L_{peak}(0,m) = L_{min}$ (44)

$C_{peak}(0,m) = 0$ (45)

$G_{peak}(0,m) = 1$ (46)

STEP 5 m=m+1

STEP 6 If m ≤ 4, go to step 2

STEP 7 Calculate the open loop frame LTP prediction gain:

$$P_v = 10\log_{10}\left[\frac{\sum_{m=1}^{4} R(0,m)}{\sum_{m=1}^{4}\left[R(0,m) - \frac{C_{peak}^2(0,m)}{G_{peak}(0,m)}\right]}\right] \qquad (47)$$

where

$$R(0,m) = \sum_{n=0}^{N_s-1} y^2\left(n+(m-1)N_s\right), \qquad (48)$$

STEP 8 Determine if the voicing mode is unvoiced:

If P_V < 1,7 then MODE=0, the long term predictor is disabled and the open loop and closed loop lag searches are aborted. In this case, proceed to subclause 4.1.10.

4.1.8.2 Open loop lag search

When MODE ≠ 0, the lag search processing is continued. The next part of the search finds the allowable lag (see table 3) which maximizes $\frac{C^2}{G}$ in the vicinity of the best open-loop integer resolution lag, $L_{peak}(0,m)$, for values of C > 0.

STEP 1 Initialize the subframe counter

m=1

STEP 2 Initialize the peak index

$L_{p,m} = 0$

STEP 3 Using interpolated versions of the C and G arrays, allowable lag values k' in the range:

$L_{peak}(0,m) - 1 < k' < L_{peak}(0,m) + 1$ (49)

are searched for a k which maximizes

$$\frac{C_I^2(k)}{G_I(k)} \qquad (50)$$

where

$$C_I(k) = \sum_{i=0}^{5} g_j(i)\, C\left(\lceil k \rceil - 3 + i, m\right) \qquad (51)$$

$$G_I(k) = \sum_{i=0}^{5} g_j(i)\, G\left(\lceil k \rceil - 3 + i, m\right) \qquad (52)$$

and

$$j = 6\left(\lceil k \rceil - k\right) \qquad (53)$$

The coefficients of the interpolating filter are $g_j(i)$ for $0 \le i \le 5$.

Only $C_I(k) > 0$ and $G_I(k) > 0$ values are considered. If no positive correlation is found, then set

$\lambda_{hnw,m} = 0$, $L_{peak}(1,m)=L_{min}$, and go to Step 22.

Otherwise, store the information related to the valid best allowable lag k.

$L_{p,m}=L_{p,m}+1$ (54)

$L_{peak}(L_{p,m},m)=k$ (55)

$C_{peak}(L_{p,m},m)=C_I(k)$ (56)

$G_{peak}(L_{p,m},m)=G_I(k)$ (57)

The next part of the search evaluates $\frac{C^2}{G}$, for C > 0 and G > 0, at the submultiples of the lag $L_{peak}(L_{p,m},m)$ to find candidate peaks.

STEP 4 Initialize the divisor

J = 2

STEP 5 Find nearest integer lag corresponding to submultiple of maximum peak

$k_1 = \text{round}[L_{peak}(1,m)/J]$ (58)

STEP 6 Determine if submultiple is within allowable lag range

If $k_1 < L_{min}$

Go to step 12

STEP 7 Find value of k' where $C^2(k',m)/G(k',m)$ is a maximum for

$\max(L_{min},k_1-3) \le k' \le \min(L_{max},k_1+3)$ (59)

If either $C(k',m) \le 0$ or $G(k',m) \le 0$ go to step 11.

STEP 8 Determine if maximum in step 7 is a peak

If

$$\frac{C^2(k'-1,m)}{G(k'-1,m)} > \frac{C^2(k',m)}{G(k',m)} \qquad (60)$$

Go to step 11

If

$$\frac{C^2(k'+1,m)}{G(k'+1,m)} > \frac{C^2(k',m)}{G(k',m)} \qquad (61)$$

Go to step 11

STEP 9 A peak has been found at an integer lag, k'. Using interpolated versions of the C and G arrays, allowable lag values within ± 1 (exclusive) of k' are searched.

Find k where

$$\frac{C_I^2(k)}{G_I(k)} \qquad (62)$$

is a maximum, where

$$C_I(k) = \sum_{i=0}^{5} g_j(i)\,C(\lceil k \rceil - 3 + i, m) \qquad (63)$$

$$G_I(k) = \sum_{i=0}^{5} g_j(i)\,G(\lceil k \rceil - 3 + i, m) \qquad (64)$$

where

$$j = 6(\lceil k \rceil - k) \qquad (65)$$

and k'-1 < k < k'+1 (66)

Only $C_I(k) > 0$ and $G_I(k) > 0$ are considered.

STEP 10 If the prediction gain exceeds a threshold, the corresponding lag, C_I, and G_I are stored in the $L_{peak}()$, $C_{peak}()$, and $G_{peak}()$ arrays; otherwise, these values are not stored.

If

$$\frac{C_I^2(k)}{G_I(k)} > R(0,m) - \frac{R(0,m)}{10^x}, \quad \text{where } x = 7{,}5\log_{10}\left(\frac{R(0,m)}{R(0,m) - \dfrac{C_{peak}^2(0,m)}{G_{peak}(0,m)}}\right) \qquad (67)$$

then

$L_{p,m}=L_{p,m}+1$ (68)

$L_{peak}(L_{p,m},m)=k$ (69)

$C_{peak}(L_{p,m},m)=C_I(k)$ (70)

$G_{peak}(L_{p,m},m)=G_I(k)$ (71)

STEP 11 Increment divisor and check the next submultiple

J=J+1

Go to step 5

STEP 12 A full-resolution search (1/6 sample resolution) is done for a peak within 1 integer lag (exclusive) of the shortest lag.

Find k such that

$$\frac{C_I^2(k)}{G_I(k)} \qquad (72)$$

is a maximum, where

$$C_I(k)=\sum_{i=0}^{5} g_j(i)\,C(\lceil k\rceil-3+i,m) \qquad (73)$$

$$G_I(k)=\sum_{i=0}^{5} g_j(i)\,G(\lceil k\rceil-3+i,m) \qquad (74)$$

$$j=6(\lceil k\rceil-k) \qquad (75)$$

$$\max\left(L_{\min}-\frac{1}{6},L_{peak}(L_{p,m})-1\right)<k<\min\left(L_{\max}+\frac{1}{6},L_{peak}\left(L_{p,m}\right)+1\right) \qquad (76)$$

The fractional lag corresponding to the maximum is referred to as $L_{pitch,m}$. This lag is used by the harmonic noise weighting function C(z) at subframe m. Then

$$C_{pitch,m}=C_I\left(L_{pitch,m}\right) \qquad (77)$$

$$G_{pitch,m}=G_I\left(L_{pitch,m}\right) \qquad (78)$$

STEP 13 The harmonic noise weighting coefficient for subframe m is calculated in this step (see subclause 4.1.9)

$$\lambda_{hnw,m}=0,4\frac{C_{pitch,m}}{G_{pitch,m}} \qquad (79)$$

Once all the correlation peaks associated with submultiples of the $L_{peak}(1,m)$ have been examined, the correlation peaks associated with multiples of $L_{pitch,m}$ are examined.

STEP 14 Initialize the multiplier

J = 2

STEP 15 Find nearest integer lag corresponding to a multiple of the fundamental lag

k_1 = round [$L_{pitch,m}$*J] (80)

STEP 16 Determine if multiple is within allowable lag range

If

$k_1 > L_{max}$

Go to step 22

STEP 17 Find value of k' where C^2(k',m)/G(k',m) is a maximum for

$\max(L_{min}, k_1-3) \le k' \le \min(L_{max}, k_1+3)$ (81)

If either C(k',m) ≤ 0 or G(k',m) ≤ 0 go to step 21.

STEP 18 Determine if maximum in step 17 is a peak

If

$$\frac{C^2(k'-1,m)}{G(k'-1,m)} > \frac{C^2(k',m)}{G(k',m)} \qquad (82)$$

Go to step 21

If

$$\frac{C^2(k'+1,m)}{G(k'+1,m)} > \frac{C^2(k',m)}{G(k',m)} \qquad (83)$$

Go to step 21

STEP 19 A peak has been found at an integer lag, k'. Using interpolated versions of the C and G arrays, allowable lag values within ± 1 (exclusive) of k' are searched.

Find k where

$$\frac{C_I^2(k)}{G_I(k)} \qquad (84)$$

is a maximum, where

$$C_I(k) = \sum_{i=0}^{5} g_j(i)\, C(\lceil k \rceil - 3 + i, m) \qquad (85)$$

$$G_I(k) = \sum_{i=0}^{5} g_j(i)\, G(\lceil k \rceil - 3 + i, m) \qquad (86)$$

where

$$j = 6(\lceil k \rceil - k) \qquad (87)$$

and $k'-1 < k < k'+1$ (88)

Only $C_I(k) > 0$ and $G_I(k) > 0$ are considered.

STEP 20 If the prediction gain exceeds a threshold, the corresponding lag, C_I, and G_I are stored.

If

$$\frac{C_I^2(k)}{G_I(k)} > R(0,m) - \frac{R(0,m)}{10^x}, \quad \text{where } x = 7{,}5\log_{10}\left(\frac{R(0,m)}{R(0,m) - \frac{C_{peak}^2(0,m)}{G_{peak}(0,m)}}\right) \quad (89)$$

then

$L_{p,m}=L_{p,m}+1$ (90)

$L_{peak}(L_{p,m})=k$ (91)

$C_{peak}(L_{p,m})=C_I(k)$ (92)

$G_{peak}(L_{p,m})=G_I(k)$ (93)

STEP 21 Increment multiplier and check the next multiple

J=J+1

Go to step 15

STEP 22 Increment subframe pointer and repeat for all subframes

m=m+1

If m≤4

Go to step 2.

Otherwise, the list of correlation peaks and the harmonic noise weighting filter parameters for each subframe have been found.

4.1.8.3 Frame lag trajectory search (Mode ≠ 0)

The frame lag trajectory search uses the list of potential lag values to determine the one lag value for each subframe which minimizes the open loop prediction error energy for the frame subject to the constraints of the delta lag coding employed for subframes 2, 3 and 4. Several candidate lag trajectories are determined. The trajectory which minimizes the open loop prediction error energy for the frame is chosen.

In subclause 4.1.8.2, the open loop lag search found a list of lags, $L_{peak}(i,m)$, corresponding to the $\frac{C^2}{G}$ peaks, for each subframe. Each trajectory evaluation begins with one of the subframes and selects a lag corresponding to a $\frac{C^2}{G}$ peak for that subframe as the anchor for that candidate trajectory.

A maximum of 2 trajectories are anchored per subframe. From the anchor lag, the trajectory is extended forward and backward to the adjacent subframes in the frame subject to the lag differential coding constraints. The lag for each subframe on the trajectory is chosen to minimize the open loop frame prediction error energy. The trajectory search is described below.

The steps involved in the frame lag trajectory evaluation and selection are:

STEP 1 Set m, the pointer to the selected subframe, equal to 1.

STEP 2 Choose the lag at the selected subframe, m, to be an anchor lag for the frame lag trajectory; i.e., the frame lag trajectory being evaluated needs to pass through that lag. The lag which is chosen, corresponds to the highest peak in the list of $\frac{{C_I}^2}{G_I}$ peaks at subframe m, which has not been crossed by a trajectory evaluated previously. If no peaks qualify, no peaks are left, or two trajectories have already been anchored and evaluated at subframe m, go to step 7. Otherwise, compute the open loop subframe weighted error energy corresponding to the chosen lag, and store the result in the frame weighted error accumulator corresponding to the trajectory currently being evaluated.

STEP 3 If m < 4, begin the forward search:

STEP 3a Define the current subframe to be m+1.

STEP 3b Define the forward search range as -7 to +6 levels relative to the current subframe's lag level.

STEP 3c Check that the lower bound does not point to a level below the lowest allowable lag level, clipping if necessary. Similarly, check that the upper bound does not point past the highest allowable lag level; clip if necessary.

STEP 3d Find the lag within the range which maximizes $\frac{C_I}{\sqrt{G_I}}$.

NOTE: negative values of C_I are allowed. Compute the open loop subframe weighted error energy corresponding to that lag at the current subframe, and add the result to the frame weighted error accumulator corresponding to the trajectory being evaluated.

STEP 3e If the current subframe < 4, increment the pointer to the current subframe, and go to step 3b.

STEP 4 If m > 1, initiate the backward search:

STEP 4a Define the current subframe to be m-1.

STEP 4b Define the backward search range as -6 to +7 levels relative to the current subframe's lag level.

STEP 4c Check that the lower bound does not point to a level below the lowest allowable lag level, clipping if necessary. Similarly, check that the upper bound does not point past the highest allowable lag level; clip if necessary.

STEP 4d Find lag within the range which maximizes. $\frac{C_I}{\sqrt{G_I}}$.

NOTE: negative values of C_I are allowed. Compute the open loop subframe weighted error energy corresponding to that lag at the current subframe, and add the result to the frame weighted error accumulator corresponding to the trajectory being evaluated.

STEP 4e If the current subframe index is > 1, decrement the pointer to the current subframe, and go to step 4a.

STEP 5 Store the lags defining the frame lag trajectory derived and the open loop LTP frame weighted error energy which this trajectory yields. Increment the counter of evaluated frame lag trajectories.

STEP 6 Go to step 2.

STEP 7 If m < 4, increment m and go to step 2.

STEP8 Choose, from the set of constructed frame lag trajectories, a lag trajectory which yields the lowest LTP weighted error energy for the frame, as the selected frame lag trajectory.

4.1.8.4 Voicing mode selection

The frame lag trajectory is specified by a vector K={k_1,k_2,k_3,k_4}, where k_m is the open loop LTP lag at the m-th subframe. Define the interpolated correlation of the input spectrally weighted speech y(n) at the m-th subframe, specified by lag k_m, as C_I(k_m,m) and the interpolated energy of y(n), delayed by k_m samples relative to the m-th subframe, as G_I(k_m,m).

The open loop LTP prediction gain in dB at the m-th subframe is:

$$P_m = 10\log_{10}\left[\frac{R(0,m)}{R(0,m) - \dfrac{C_I^2(k_m,m)}{G_I(k_m,m)}}\right] \qquad (94)$$

The open loop frame LTP prediction gain, is given by:

$$P_v = 10\log_{10}\left[\frac{\sum_{m=1}^{4} R(0,m)}{\sum_{m=1}^{4}\left[R(0,m) - \dfrac{C_{peak}^2(0,m)}{G_{peak}(0,m)}\right]}\right] \qquad (95)$$

The rules for mode selection are specified as follows:

MODE=0 if P_V < 1,7 (96)

MODE=1 if P_V ≥ 1,7 and Pm < 3,5 for any m (97)

MODE=2 if P_m ≥ 3,5 for all m and Pm < 7 for any m (98)

MODE=3 if P_m ≥ 7,0 for all m (99)

4.1.8.5 Closed loop lag search

From the selected frame lag trajectory, develop a list of lags to be searched closed loop. At each subframe, three allowable lag levels centered around the subframe lag, specified by the selected frame lag trajectory, will be searched. If the lag points to the lowest or the highest level in the table of quantized lag values, only two closed loop lag evaluations will be done at that subframe, with the lag outside the quantizer range being eliminated from consideration. The closed loop evaluation of the subframe lags is not performed if MODE=0. What follows is a description of the construction of the output of the long term predictor (adaptive codebook) for a given, possibly fractional lag, L. Defining:

L_{max}	maximum possible value for long term lag L
r(n)	long term filter state; n < 0 (history of the excitation signal)
r_L(n)	long term filter state with adaptive codebook output for L appended
b_L(n)	output of long term filter state (adaptive codebook) for lag L
P_f	order of one phase of the interpolating FIR filter (P_f = 10 except for the special case when j = 0, see below)
$\tilde{f}_j(i)$	coefficients of jth phase of interpolating FIR filter, i=0 to i=P_f -1
N_S	number of samples per subframe (N_S = 40)

The sequence r_L(n) is defined as:

$$r_L(n) = \begin{cases} r(n) & ;-L_{\max} \le n \le -1 \\ \sum\limits_{i=0}^{P_f-1} \tilde{f}_j(i) r_L\left(n - \Lambda - \frac{P_f}{2} + i\right) & ;0 \le n \le N_s - 1 \end{cases} \tag{100}$$

where; $q = \left\lfloor \frac{n + L + \frac{5}{6}}{L} \right\rfloor L, \quad \Lambda = \lfloor q \rfloor, \quad and \quad j = 6(q - \lfloor q \rfloor)$

The portion of the sequence $r_L(n)$ from n=0 to n=N_S-1 shall be calculated in order from 0 to N_S-1, so that the necessary terms in the sum will be available. The 0^{th} phase of the interpolating filter, $\tilde{f}_o(i)$, is a special case and has only one non-zero tap, so that if q is an integer, the summation reduces to the single term, r_L(n-q).

The output of the codebook for lag L is just the last N_S samples in the sequence $r_L(n)$.

$$b_L(n) = r_L(n) \qquad ;0 \le n \le N_S-1 \tag{101}$$

The closed loop search minimizes the weighted error by maximizing the term $\frac{C^2}{G}$, where

$$C = \sum_{n=0}^{N_s-1} b_L^{'}(n) p(n) \tag{102}$$

$$G = \sum_{n=0}^{N_s-1} b_L^{'\,2}(n) \tag{103}$$

The sequence $b'_L(n)$ is the zero state response of H(z) to the adaptive codebook output for lag L. The sequence p(n) is the input speech, weighted by the filter W(z), minus the zero input response of H(z). The error minimization is done over only those lags in the list supplied by the open loop search. The lag L which maximizes $\frac{C}{\sqrt{G}}$ (C is allowed to be negative) is then chosen as the lag for the subframe.

4.1.9 Harmonic noise weighting

If MODE = 1, 2 or 3, then C(z), the harmonic noise weighting transfer function, is activated. The excitation codebook vector and gains codebook vector are selected to minimize the spectrally and harmonically weighted error. The harmonic weighting filter, C(z), can be expressed as:

$$C(z) = 1 - \lambda_{hnw} z^{-L_{pitch}} \tag{104}$$

where

$$\lambda_{hnw} = 0{,}4 \frac{C_{pitch}}{G_{pitch}} \tag{105}$$

C_{pitch}, G_{pitch} and L_{pitch} were determined during the open loop lag search where the subscript m denoting the subframe has been dropped from $L_{pitch,m}$ and $\lambda_{hnw,m}$ for notational convenience. L_{pitch} can take on fractional values so the interpolating filter employed for the open loop lag search is utilized to generate the fractionally delayed samples.

Let x(n) represent the input of the harmonic noise weighting filter and y(n) represent the output. The filter can then be described by equation (106) and equation (107).

$$y(n) = x(n) - \lambda_{hnw}\, x(n - L_{pitch}) \qquad (106)$$

$$x\left(n - L_{pitch}\right) = \sum_{i=0}^{P_g - 1} g_j(i)\, x\left(n - \left\lfloor L_{pitch} \right\rfloor - \frac{P_g}{2} + i\right) \qquad (107)$$

where

$$j = \left(L_{pitch} - \left\lfloor L_{pitch} \right\rfloor\right)6$$

Figure 4 incorporates harmonic noise weighting (for MODE = 1, 2 or 3) and shows the VSELP excitation source. All error minimizations done after lag selection utilize the combination of spectral and harmonic noise weighting.

For MODE ≠ 0, P(n) is the input speech signal weighted by W(z)C(z) minus the input response of H(z)C(z). For MODE = 0, P(n) is the input speech signal weighted by W(z) minus the zero input response of H(z).

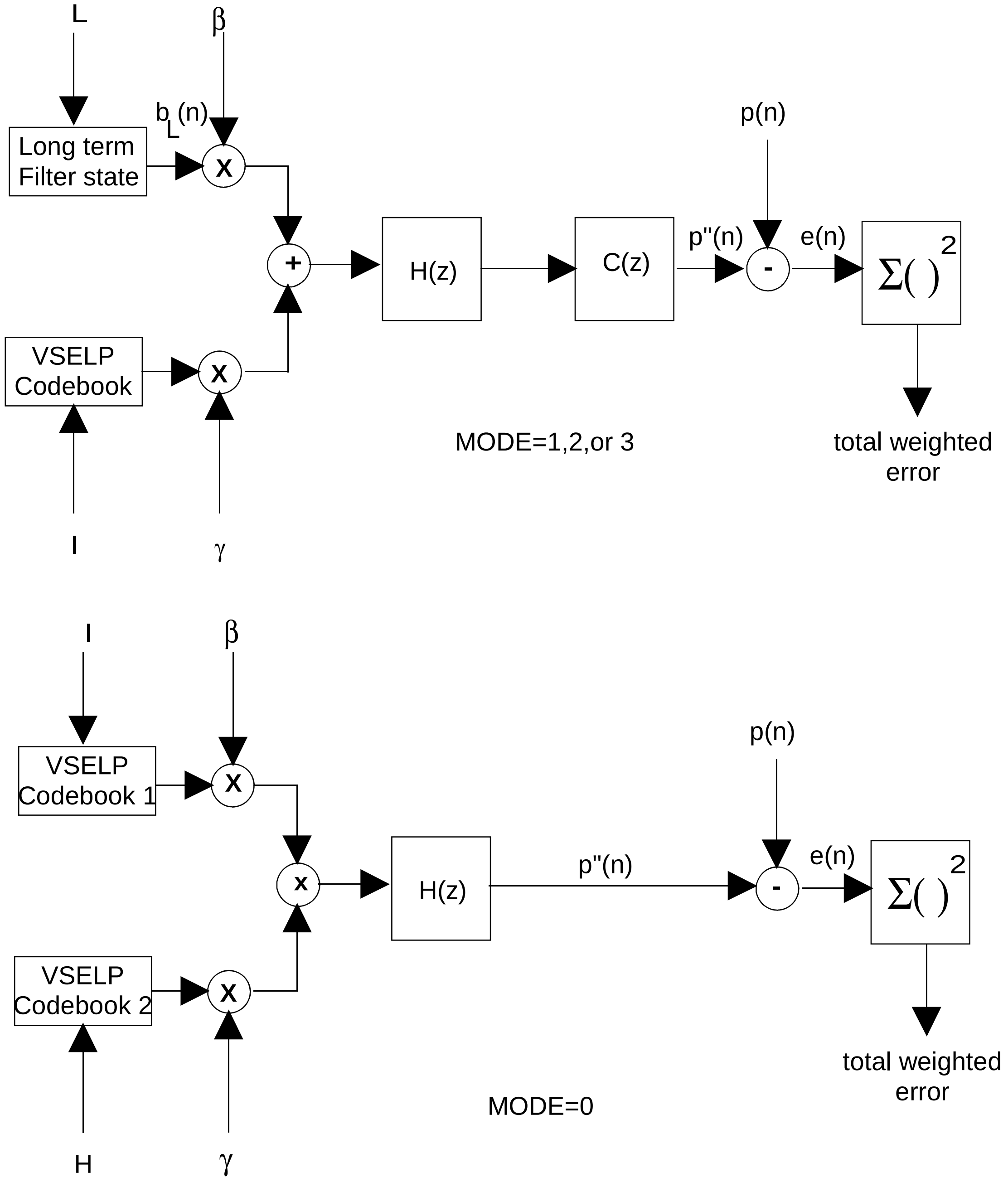

Figure 4: Long term predictor lag and code search

The zero state response of H(z)C(z) to the pitch predictor (adaptive codebook) vector, $b_L(n)$, needs to be computed prior to the search of the excitation codebook. This spectrally and harmonically weighted adaptive codebook vector is represented by $b''_L(n)$.

4.1.10 Code search algorithm

For MODE≠0, the excitation codebook search procedure takes place after the long term predictor lag, L, has been determined. The codebook search procedure then chooses one codevector from the VSELP codebook. The GSM half rate speech encoder uses an excitation codebook of 2^M codevectors which is constructed from M basis vectors. Defining $v_m(n)$ as the m^{th} basis vector and $u_i(n)$ as the i^{th} codevector in the codebook, then:

$$u_i(n) = \sum_{m=1}^{M} \theta_{im} V_m(n) \qquad (108)$$

where $0 \le i \le 2^{M}-1$; $0 \le n \le N_S-1$. In other words, each codevector in the codebook is constructed as a linear combination of the M basis vectors. The linear combinations are defined by the θ parameters.

θ_{im} is defined as:

θ_{im} = + 1 if bit m of codeword i = 1

θ_{im} = - 1 if bit m of codeword i = 0

The codebook construction for the GSM half rate speech encoder can be restated as follows. Codevector i is constructed as the sum of the M basis vectors where the sign (plus or minus) of each basis vector is determined by the state of the corresponding bit in codeword i. The codebook search procedure finds the codevector which will produce the minimum total spectral and harmonic weighted error for the subframe given $b''_L(n)$ (the zero state response of H(z)C(z) to $b_L(n)$) and allowing both the gain, γ, and the long term filter coefficient, β, to be optimized for each codevector being evaluated.

The filtered codevector $f_i(n)$, can be expressed as:

$$f_i(n) = \sum_{m=1}^{M} \theta_{im} q_m(n) \qquad (109)$$

where $q_m(n)$ is the zero state response of H(z)C(z) to basis vector $v_m(n)$.

If MODE=0, two VSELP codebooks are the excitation sources and are searched sequentially to identify the codeword I specifying the codevector selected from the first codebook, and codeword H, identifying the codevector chosen from the second VSELP codebook. Harmonic noise weighting is not used. When searching the second VSELP codebook, each codevector is evaluated assuming optimal gains for the codevector I, and the potential codevector H.

4.1.10.1 Decorrelation of filtered basis vectors

For MODE≠0, each filtered codevector, $f_i(n)$, is decorrelated to the long term predictor vector, $b''_L(n)$. If MODE=0, the $b_L(n)$ vector and the single VSELP codebook excitation are replaced by two VSELP codebook excitations, as the excitation sources. In this case, decorrelation is only performed for the second VSELP codebook. In this case, the orthogonalization is done with respect to the filtered codevector chosen from the first VSELP codebook.

Defining:

$$\Gamma = \sum_{n=0}^{N_s-1} \left(b''_L(n)\right)^2 \qquad (110)$$

and

$$\Psi m = \sum_{n=0}^{N_s-1} b''_L(n) q_m(n) \qquad (111)$$

for $1 \le m \le M$; then $q'_m(n)$, the decorrelated filtered basis vectors, can be computed by:

$$q'_m(n) = q_m(n) - \left(\frac{\Psi_m}{\Gamma}\right) b''_L(n) \qquad (112)$$

for $1 \le m \le M$ and $0 \le n \le N_S-1$.

The decorrelated filtered codevectors can now be expressed as:

$$f_i^{'}(n) = \sum_{m=1}^{M} \theta_{im} q'_m(n) \qquad (113)$$

for $0 \le i \le 2^M-1$ and $0 \le n \le N_S-1$.

4.1.10.2 Fast search technique

The codebook search procedure should find the codeword i which minimizes:

$$E'_i = \sum_{n=0}^{Ns-1} \left(p(n) - \gamma f'_i(n)\right)^2 \qquad (114)$$

Defining :

$$C_i = \sum_{n=0}^{N_s-1} f_i^{'}(n)p(n) \qquad (115)$$

and

$$G_i = \sum_{n=0}^{N_s-1} \left(f_i^{'}(n)\right)^2 \qquad (116)$$

then the best codevector is the one which maximizes:

$$\frac{(C_i)^2}{G_i} \qquad (117)$$

and the corresponding optimal gain is given by:

$$\gamma_i = \frac{C_i}{G_i} \qquad (118)$$

The search process needs to evaluate equation (117) for each codevector. The codevector which maximizes equation (117) is then chosen. Using properties of the VSELP codebook construction, the computations required for computing C_i and G_i can be greatly simplified.

defining:

$$R_m = 2\sum_{n=0}^{N_S-1} q'_m(n)p(n) \qquad (119)$$

for $1 \le m \le M$ and

$$D_{mj} = 4\sum_{n=0}^{N_S-1} q'_m(n)q'_j(n) \qquad (120)$$

for $1 \le m \le j \le M$

C_i can be expressed as:

$$C_i = \frac{1}{2}\sum_{m=1}^{M} \theta_{im} R_m \qquad (121)$$

and G_i can be expressed as:

$$G_i = \frac{1}{2}\sum_{j=2}^{M}\sum_{m=1}^{j-1}\theta_{im}\theta_{ij}D_{mj} + \frac{1}{4}\sum_{j=1}^{M}D_{jj} \qquad (122)$$

Assuming that codeword u differs from codeword i in only one bit position, say position v such that $\theta_{uv} = -\theta_{iv}$ and $\theta_{um} = \theta_{im}$ for $m \neq v$ then:

$$C_u = C_i + \theta_{uv}R_v \qquad (123)$$

and

$$G_u = G_i + \sum_{j-1}^{v-1}\theta_{uj}\theta_{uv}D_{jv} + \sum_{j=v+1}^{M}\theta_{uj}\theta_{uv}D_{vj} \qquad (124)$$

The codebook search is structured such that each successive codeword evaluated differs from the previous codeword in only one bit position, then equation (123) and equation (124) can be used to update C_i and G_i in a very efficient manner. Sequencing of the codewords in this manner is accomplished using a binary Gray code. This updating operation for both equation (123) and equation (124) can be accomplished with a total of only M multiply-accumulates per codevector.

With this technique for computing C_i and G_i for the codevectors in a VSELP codebook, it is necessary to find the i which maximizes equation (117). Note that complementary codewords (see subclause 3.1.10) will have equivalent values for equation (117). Therefore only half of the codevectors need to be evaluated. Once the codevector which maximizes equation (117) is found, the sign of C_i for that codevector will determine whether that codevector or its complement will yield a positive gain, γ.

A running maximum for equation (117) is kept during the code search then for each codevector evaluated, evaluate equation (117) and compare to the running maximum.

$$\frac{(C_i)^2}{G_i} > \frac{(C_{best})^2}{G_{best}} \qquad (125)$$

Evaluating equation (125) directly from C_i and G_i requires one multiply, one divide and one compare operation. By cross multiplying equation (117) can be expressed as:

$$(C_i)^2 G_{best} > (C_{best})^2 G_i \qquad (126)$$

Using equation (126) requires only three multiplies and a compare per evaluation (and no divides) where $(C_{best})^2$ and G_{best} are updated throughout the search to reflect the running best codeword.

4.1.11 Multimode gain vector quantization

A separate GSP0 vector quantizer is derived for each of the four voicing modes. Once the frame voicing mode is selected, the vector quantizer, corresponding to that mode, is searched to select the excitation gains at each subframe of the frame.

Although the interpretation of what the excitation sources are differs between MODE=0 and the remaining MODE values, the procedure for searching the gain vector quantizer is identical. In each case, the P0 term specifies the relative contribution of the first of the two excitation vectors to the total excitation energy at the subframe, where the first excitation vector is the long term prediction vector for MODE=1, 2 or 3, while the vector selected from the first of the two VSELP codebooks is used in the MODE=0 case.

4.1.11.1 Coding GS and P0

Define ex(n) to be the excitation function at a given subframe. For MODE=1, 2 or 3, ex(n) is a linear combination of the pitch prediction vector scaled by β, the long term predictor coefficient, and of the codevector scaled by γ, its gain. In equation form

$$ex(n) = \beta c_0(n) + \gamma c_1(n) \qquad 0 \le n \le N_S\text{-}1 \qquad (127)$$

where for MODE≠0

$c_0(n)$ is the unweighted long term prediction vector, $b_L(n)$

$c_1(n)$ is the unweighted codevector selected, $u_I(n)$

and for MODE=0

$c_0(n)$ is the unweighted codevector selected from the first VSELP codebook, $u_{I,1}(n)$

$c_1(n)$ is the unweighted codevector selected from the second VSELP codebook, $u_{H,2}(n)$

The variable $c'_j(n)$ is a weighted version of $c_j(n)$. The power in each excitation vector is given by

$$R_x(k) = \sum_{n=0}^{N_s-1} c^2{}_k(n) \qquad 0 \le k \le 1 \qquad (128)$$

Let R be the total power in the coder subframe excitation

$$R = \beta^2 R_x(0) + \gamma^2 R_x(1) \qquad (129)$$

P0, the power contribution of the pitch prediction vector as a fraction of the total excitation power at a subframe,

$$P0 = \frac{\beta^2 R_x(0)}{R} \qquad \text{where } 0 \le P0 \le 1 \qquad (130)$$

Define $R'_q(0)$ to be the quantized value of R(0) to be used for the current subframe and $R_q(0)$ to be the quantized value of R(0). Then:

$R'_q(0) = R_q(0)_{\text{previous frame}}$ for subframe 1 (131a)

$R'_q(0) = R_q(0)_{\text{current frame}}$ for subframes 2, 3, 4 (131b)

Let RS be

$$RS = N_s\, R'_q(0) \prod_{i=1}^{N_p} \left(1 - r_i^2\right) \qquad (132)$$

The term GS is the energy tweak parameter defined as

$$R = GS\ RS\ -> GS = \frac{R}{RS} \qquad (133)$$

P0 represents the fraction of the total subframe excitation energy which is due to the first codebook vector, and GS, the energy tweak factor which bridges the gap between R, the actual energy in the coder excitation, and RS, its estimated value.

The gain bias factor χ, formulated to force a better energy match between p(n) and the weighted synthetic excitation, is given below where.

$$\chi = \min\left[\sqrt{2{,}0}, \left\{\max\left[1{,}0, \sqrt{\frac{R_{pp}}{\beta_{opt}^2 R_{cc}(0{,}0) + \gamma_{opt}^2 R_{cc}(1{,}1) + 2\beta_{opt}\,\gamma_{opt} R_{cc}(0{,}1)}}\right]\right\}\right] \qquad (134)$$

The weighted error equation is

$$E = \chi^2 R_{pp} - a\sqrt{GS\,P0} - b\sqrt{GS(1-P0)} + c\,GS\sqrt{P0(1-P0)} + d\,GS\,P0 + e\,GS\,(1-P0) \quad (135)$$

where

$$a = 2\chi R_{pc}(0)\sqrt{\frac{RS}{R_x(0)}} \qquad (136)$$

$$b = 2\chi R_{pc}(1)\sqrt{\frac{RS}{R_x(1)}} \qquad (137)$$

$$c = \frac{2R_{cc}(0,1)RS}{\sqrt{R_x(0)R_x(1)}} \qquad (138)$$

$$d = \frac{RS\,R_{cc}(0,0)}{R_x(0)} \qquad (139)$$

$$e = \frac{RS\,R_{cc}(1,1)}{R_x(1)} \qquad (140)$$

$$R_{pc}(k) = \sum_{n=0}^{N_S-1} p(n)c'_k(n) \qquad \text{k=0,1} \qquad (141)$$

$$R_{cc}(k,j) = \sum_{n=0}^{N_S-1} c'_k(n)c'_j(n) \qquad \text{k=0,1, j=k,1} \qquad (142)$$

$$R_{cc}(k,j) = R_{cc}(j,k) \qquad (143)$$

$$R_{pp} = \sum_{n=0}^{N_s-1} p^2(n) \qquad (144)$$

Four separate vector quantizers for jointly coding P0 and GS are defined, one for each of the four voicing modes. The first step in quantizing of P0 and GS consists of calculating the parameters required by the error equation:

$R_{cc}(k,j)$ k = 0, 1, j = k, 1

$R_x(k)$ k = 0, 1

RS

$R_{pc}(k)$ k = 0, 1

a, b, c, d, e

Next equation (135) is evaluated for each of the 32 vectors in the {P0,GS} codebook, corresponding to the selected voicing mode, and the vector which minimizes the weighted error is chosen. Note that in conducting the code search $\chi^2 R_{pp}$ may be ignored in equation (135), since it is a constant. β_q, the quantized long term predictor coefficient, and γ_q, the quantized gain, are reconstructed from

$$\beta_q = \sqrt{\frac{RS\,GS_{vq}\,P0_{vq}}{R_x(0)}} \qquad (145)$$

$$\gamma_q = \sqrt{\frac{RS\,GS_{vq}\left(1-P0_{vq}\right)}{R_x(1)}} \qquad (146)$$

where $P0_{vq}$ and GS_{vq} are the elements of the vector chosen from the {P0,GS} codebook.

A special case occurs when the long term predictor is disabled for a certain subframe, but voicing MODE≠0. This will occur when the state of the long term predictor is populated entirely by zeroes.

For that case, the following error equation is used:

$$E \cong \chi^2 R_{pp} - b\sqrt{GS} + eGS \qquad (147)$$

For this case the quantized codevector gains are:

$$\beta_q = 0 \qquad (148)$$

$$\gamma_q = \sqrt{\frac{RS\,GS_{vq}\left(1 - P0_{vq}\right)}{R_x(1)}} \qquad (149)$$

4.2 GSM half rate speech decoder

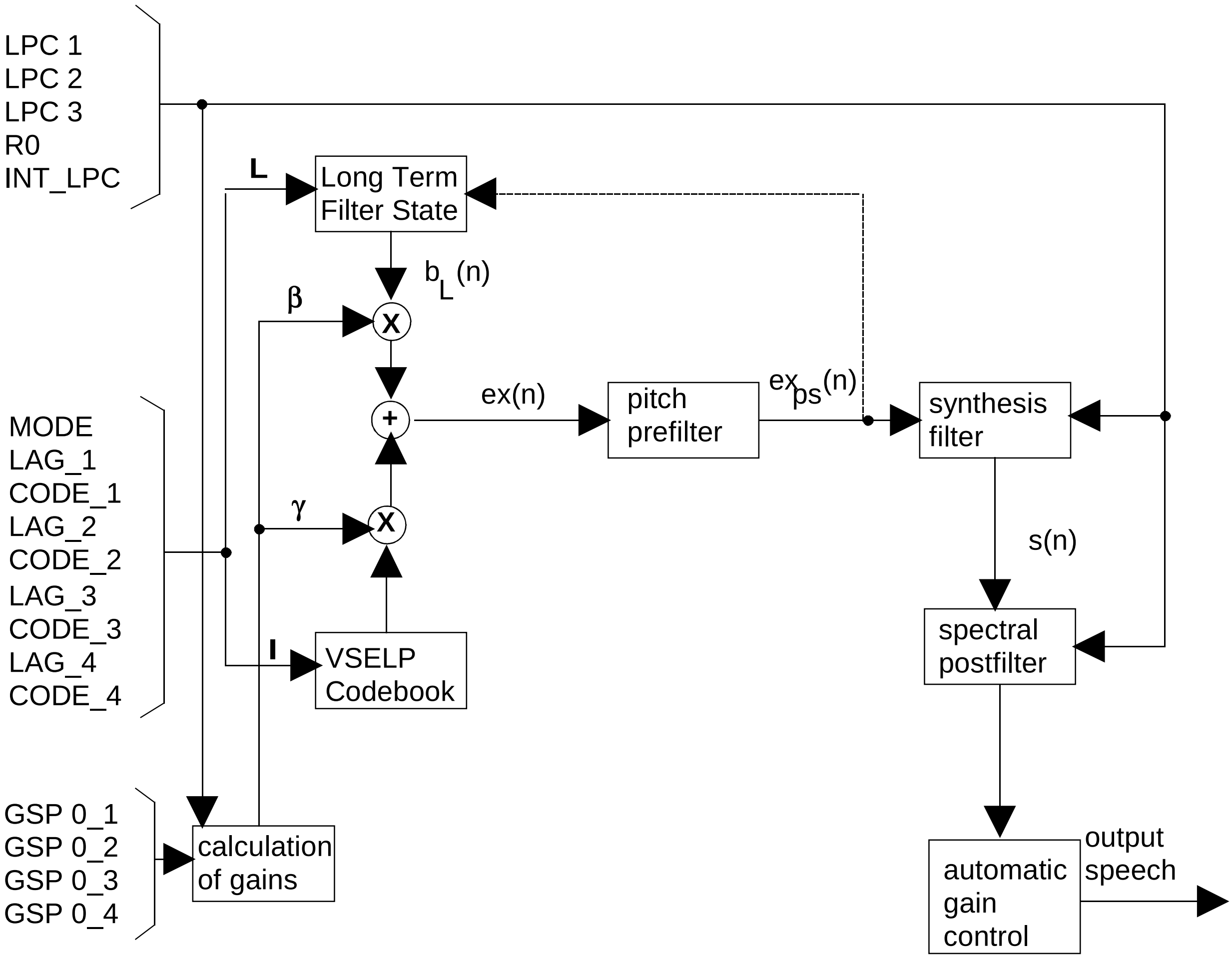

Figure 5: The GSM half rate speech decoder for MODE = 1, 2 or 3

A block diagram of the GSM half rate speech decoder for MODE=1, 2 or 3 is given in figure 5. The speech decoder creates the combined excitation signal, ex(n), from the long term filter state and the VSELP codevector. For MODE=0, the long term filter state is replaced by another VSELP codebook and the pitch prefilter is not used. The combined excitation is then processed by an adaptive pitch prefilter and gain. The prefiltered excitation is applied to the LPC synthesis filter. After reconstructing the speech signal with the synthesis filter, an adaptive spectral postfilter is applied followed by an automatic gain control which is the final processing step in the speech decoder.

4.2.1 Excitation generation

The combined excitation, ex(n), shall be computed as shown in equation (127)

The combined excitation, ex(n), is filtered by the synthesis filter to generate the speech signal. The synthesis filter is a tenth order all pole filter. The filter coefficients for the subframe are the α_i's defined in subclause 4.1.6. The filter coefficients will change from subframe to subframe. The filter state shall be preserved from subframe to subframe. A direct form filter shall be used for the synthesis filter.

4.2.2 Adaptive pitch prefilter

Given ex(n) as the input, $ex_p(n)$, the pitch prefiltered output, is defined by

$$ex_p(n) = ex(n) + \xi ex_p(n-L) \qquad ;\text{for } 0 \le n \le N_S\text{-}1 \qquad (150)$$

where

$$\xi = \begin{cases} 0{,}3 Min\left[\beta, \sqrt{P0}\right] & ; MODE \ne 0 \\ 0 & ; MODE = 0 \end{cases} \qquad (151)$$

Since L can be fractional in value, an interpolating filter is used. This is the same interpolating filter which is used for the open loop lag search. A gain scale factor is computed and is used to scale the pitch prefiltered excitation, prior to applying it to the LPC synthesis filter. P_{scale}, the gain scale factor, is

$$P_{scale} = \sqrt{\frac{\sum_{n=0}^{N-1} ex^2(n)}{\sum_{n=0}^{N-1} ex_p{}^2(n)}} \qquad (152)$$

Thus $ex_{ps}(n)$, the gain corrected pitch prefiltered excitation which drives the LPC synthesis filter, is given by

$$ex_{ps}(n) = P_{scale} ex_p(n) \qquad ;\text{for } 0 \le n \le N_S\text{-}1 \qquad (153)$$

4.2.3 Synthesis Filter

A direct form synthesis filter is used:

$$s(n) = ex_{ps}(n) + \sum_{i=1}^{10} \alpha_i \, s(n-i) \qquad ,0 \le n \le N_S\text{-}1 \qquad (154)$$

4.2.4 Adaptive spectral postfilter

The perceptual quality of the synthetic speech is enhanced by using an adaptive postfilter as the final processing step. The general form of the postfilter is given by:

$$\bar{s}(n) = s(n) - \sum_{i=1}^{Np} \eta^i \alpha_i s(n-i) \qquad ,0 \le n \le N_S\text{-}1 \qquad (155)$$

$$\tilde{s}(n) = \bar{s}(n) + \sum_{i=1}^{Np} (0{,}75)^i \alpha_i \tilde{s}(n-i) \qquad ,0 \le n \le N_S\text{-}1 \qquad (156)$$

$$\hat{s}(n) = \tilde{s}(n) - 0,2\,\tilde{s}(n-1) \qquad 0 \le n \le N_S\text{-}1 \qquad (157)$$

The adaptive spectral postfilter numerator polynomial equation (155) is replaced by a spectrally smoothed version of the adaptive spectral postfilter denominator polynomial equation (156). To derive the coefficients of the numerator polynomial, the denominator polynomial coefficients are converted to the autocorrelation coefficients R(i). The SST bandwidth expansion function is then applied to the autocorrelation sequence,

$$R_{sst}(i) = R(i) W_{sst}(i) \qquad , 0 \le i \le N_p \qquad (158)$$

and the numerator polynomial coefficients are calculated from the modified autocorrelation sequence via the AFLAT recursion.

From R_{sst}(i) the reflection coefficients which define the combined spectrally noise weighted synthesis filter are computed using the AFLAT recursion once per frame.

STEP 1 Define the initial conditions for the AFLAT recursion:

$$P_o(i) = R_{sst}(i) \qquad ,0 \le i \le N_p \qquad (159)$$

$$V_o(i) = R_{sst}(|i+1|) \qquad ,1\text{-}N_p \le i \le N_p\text{-}1 \qquad (160)$$

STEP 2 Initialize j, the index of the lattice stage, to point to the first lattice stage:

j=1 (161)

STEP 3 Compute r_j, the j-th reflection coefficient, using:

$$r_j = -\frac{V_{j-1}(0)}{P_{j-1}(0)} \qquad (162)$$

STEP 4 Given r_j, update the values of V_j and P_j arrays using:

$$P_j(i) = \left(1 + r_j^2\right) P_{j-1}(i) + r_j \left[V_{j-1}(i) + V_{j-1}(-i)\right] \qquad .0 \le i \le Np - j - 1 \qquad (163)$$

$$V_j(i) = V_{j-1}(i+1) + r_j^2 V_{j-1}(-i-1) + 2 r_j P_{j-1}(|i+1|) \quad ,1+j\text{-}Np \le i \le Np\text{-}j\text{-}1 \qquad (164)$$

STEP 5 Increment j:

j = j +1

STEP 6 If j ≤ N_p go to step 3, otherwise all N_p reflection coefficients have been obtained.

STEP 7 The reflection coefficients, r_j, are then converted to $\bar{\alpha}_i$, the direct-form LP filter coefficients for use in the adaptive spectral postfilter numerator polynomial.

The resultant adaptive spectral postfilter is derived from equations 155, 156 and 157:

$$\bar{s}(n) = s(n) - \sum_{i=1}^{N_p} \bar{\alpha}_i s(n-i) \qquad ,0 \le n \le N_S\text{-}1 \qquad (165)$$

$$\tilde{s}(n) = \bar{s}(n) + \sum_{i=1}^{N_p} (0,75)^i \alpha_i \tilde{s}(n-i) \qquad ,0 \le n \le N_S\text{-}1 \qquad (166)$$

$$\hat{s}(n) = \tilde{s}(n) - 0,2\,\tilde{s}(n-1) \qquad ,0 \le n \le N_S\text{-}1 \qquad (167)$$

In order to reduce the computations needed to compute the spectrally smoothed numerator coefficients, the spectral smoothing operation is performed once per frame on the denominator coefficients corresponding to the uninterpolated coefficients. This will yield the coefficients for the numerator of the spectral postfilter for subframe four. The numerator coefficients for subframes one, two, and three are interpolated using the same interpolation scheme that is used for the LPC synthesis coefficients (see subclause 4.1.6).

As in the case of the pitch prefilter, a means of automatic gain control is needed to ensure unity gain through the spectral postfilter. A scale factor, S_{scale}, is given by:

$S'_{scale}(n) = (0,9875\ S'_{scale}(n-1)\ ) + (0,0125\ S_{scale}\ )$ (168)

Scale factor, S_{scale}, is the square root of the ratio of the input signal energy to the output signal energy over the subframe.

The output of the spectral postfilter is then multiplied by S'_{scale} as the last step in reconstructing the speech signal in the speech decoder.

4.2.5 Updating decoder states

The long term predictor state, r(n), is updated by:

$r(n) = r(n+40)$ for $-146 \leq n \leq -41$ (169)

$r(n) = ex(n+40)$ for $-40 \leq n \leq -1$ (170)

5 Homing sequences

5.1 Functional description

The half rate speech codec as well as the DTX system and comfort noise generator are described in a bit exact arithmetic to allow for easy type approval as well as general testing purposes of the half 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.

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 on 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.

5.2 Definitions

encoder homing frame: The encoder homing frame consists of 160 identical samples, each 13 bit long, with the least significant bit set to "one" and all other bits set to "zero". When written to 16 bit long words with left justifications, the samples have a value of 0008 hex. Test sequence SEQ05.INP described in GSM 06.07 [3] defines the encoder homing frame. 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 defined in test sequence SEQ05.INP described in GSM 06.07 [3]. 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.

5.3 Encoder homing

Whenever the half 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 submodules 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.

5.4 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 including comfort noise generation 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 submodules 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 58 bits of the decoder homing frame must differ in at least one bit position from the first 58 bits of any of the decoder test sequences. Therefore, if the decoder is in its home state, it is sufficient to check only these first 58 bits to detect a subsequent decoder homing frame. This definition ids made to support a delay optimised implementation in the TRAU uplink direction.

5.5 Encoder home state

In GSM 06.06 [2], a listing of all the encoder state variables with their predefined values when in the home state is given.

5.6 Decoder home state

In GSM 06.06 [2], a listing of all the decoder state variables with their predefined values when in the home state is given.

Annex A (normative): Codec parameter description

A.1 Codec parameter description

The following is a list of all the parameters which are coded for each 20 ms speech frame. The basic data rate of the speech coder is 5,6 kbps. Therefore each 20 ms speech frame consists of 112 bits. These bits are given in table A.1.

Table A.1: Codec parameter description

Parameter	No. of bits	Description
Frame bits:		
MODE	2	voicing mode
R0	5	frame energy
LPC1	11	reflection coefficient vector r_1-r_3
LPC2	9	reflection coefficient vector r_4-r_6
LPC3	8	reflection coefficient vector r_7-r_{10}
INT_LPC	1	the soft interpolation bit for the frame
Subframe bits (MODE = 1,2 or 3):		
LAG_1	8	lag for first subframe
LAG_2	4	lag delta code for second subframe
LAG_3	4	lag delta code for third subframe
LAG_4	4	lag delta code for fourth subframe
CODE_1	9	codebook, I, for first subframe
CODE_2	9	codebook, I, for second subframe
CODE_3	9	codebook, I, for third subframe
CODE_4	9	codebook, I, for fourth subframe
GSP0_1	5	{P0,GS} code for first subframe
GSP0_2	5	{P0,GS} code for second subframe
GSP0_3	5	{P0,GS} code for third subframe
GSP0_4	5	{P0,GS} code for fourth subframe
Subframe bits (MODE=0):		
CODE1_1	7	codebook code, I, for first subframe
CODE2_1	7	codebook code, H, for first subframe
CODE1_2	7	codebook code, I, for second subframe
CODE2_2	7	codebook code, H, for second subframe
CODE1_3	7	codebook code, I, for third subframe
CODE2_3	7	codebook code, H for third subframe
CODE1_4	7	codebook code, I, for fourth subframe
CODE2_4	7	codebook code, H, for fourth subframe
GSP0_1	5	{P0,GS} code for first subframe
GSP0_2	5	{P0,GS} code for second subframe
GSP0_3	5	{P0,GS} code for third subframe
GSP0_4	5	{P0,GS} code for fourth subframe

A.1.1 MODE

The speech coder is defined by 4 voicing modes. MODE is a two bit code which specifies which of the four voicing modes is used at the current frame. The MODE indicates which definition of the frame bits to apply to the current frame.

A.1.2 R0

R0 is a code which represents the average signal power of the input speech for the frame. The average signal power is computed using an analysis window which is centered over the last 100 samples of the frame.

A.1.3 LPC1 - LPC3

The 10 reflection coefficients are vector quantized in three vector segments. The first vector segment codes reflection coefficients r_1 - r_3, the second vector segment codes coefficients r_4-r_6, the third vector segment codes coefficients r_7 - r_{10}.

A.1.4 LAG_1 - LAG_4

LAG_1, the lag for the first subframe, can take on the value in the range of 21 to 142. Eight bits are used to encode the lag which may be fractional in value. Each of the remaining lag values (LAG_2 through LAG_4) is delta coded relative to the preceding subframe's coded value of the lag, with a deviation of -8 to +7 allowable lag value levels specified by a four bit code.

A.1.5 CODEx_1 - CODEx_4

If MODE ≠ 0, the code value for the VSELP codebook is the codeword I as derived by the codebook search procedure. If MODE=0, two VSELP codebooks are sequentially searched, with codeword I, specifying the codevector from the first VSELP codebook, assigned onto CODE1_x, and codeword H, specifying the codeword selected from the second VSELP codebook, assigned onto CODE2_x, where x is the subframe number.

A.1.6 GSP0_1 - GSP0_4

The {P0,GS} codebook contains the values needed to determine the gain factors for the excitation vectors of a given subframe. The index of the corresponding codebook entry is assigned to GSP0_x.

The speech coder is a multimode speech coder, defined by four voicing modes:

MODE = 0	unvoiced
MODE = 1	slightly voiced
MODE = 2	moderately voiced
MODE = 3	strongly voiced

If MODE=0, the adaptive codebook (long-term predictor) and the VSELP codebook are replaced by two other VSELP codebooks.

A.2 Basic coder parameters

The following are the basic parameters for the 5 600 bps GSM half rate speech codec system.

	sampling rate	8 kHz
N_F	frame length	160 samples (20 ms)
Ns	subframe length	40 samples (5 ms)
Np	short term predictor order	10

Annex B (normative): Order of occurrence of the codec parameters over Abis

The order of occurrence of the codec parameters over the Abis is defined for unvoiced speech (MODE = 0) and voiced speech (MODE = 1, 2 or 3) in tables B.1 and B.2 respectively.

Table B.1: Occurrence of the codec parameters over Abis for unvoiced speech (MODE = 0)

Parameter	No. of bits	Bit No. (MSB - LSB)
R0	5	b1 - b5
LPC1	11	b6 - b16
LPC2	9	b17 - b25
LPC3	8	b26 - b33
INT_LPC	1	b34
MODE	2	b35 - b36
CODE1_1	7	b37 - b43
CODE2_1	7	b44 - b50
GSP0_1	5	b51 - b55
CODE1_2	7	b56 - b62
CODE2_2	7	b63 - b69
GSP0_2	5	b70 - b74
CODE1_3	7	b75 - b81
CODE2_3	7	b82 - b88
GSP0_3	5	b89 - b93
CODE1_4	7	b94 - b100
CODE2_4	7	b101 - b107
GSP0_4	5	b108 - b112

Table B.2: Occurrence of the codec parameters over Abis for voiced speech (MODE = 1, 2 or 3)

Parameter	No. of bits	Bit No. (MSB - LSB)
R0	5	b1 - b5
LPC1	11	b6 - b16
LPC2	9	b17 - b25
LPC3	8	b26 - b33
INT_LPC	1	b34
MODE	2	b35 - b36
LAG_1	8	b37 - b44
CODE1	9	b45 - b53
GSP0_1	5	b54 - b58
LAG_2	4	b59 - b62
CODE2	9	b63 - b71
GSP0_2	5	b72 - b76
LAG_3	4	b77 - b80
CODE3	9	b81 - b89
GSP0_3	5	b90 - b94
LAG_4	4	b95 - b98
CODE4	9	b99 - b107
GSP0_4	5	b108 - b112

Annex C (informative): Bibliography

M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 937-940, March 1985.

G. Davidson and A. Gersho, "Complexity Reduction Methods for Vector Excitation Coding", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 3055-3058, April 1986.

I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbps", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 461-464, April 1990.

P. Kroon and B. S. Atal, "Pitch Predictors with High Temporal Resolution", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 661-664, April 1990.

I. A. Gerson, "Method and Means of Determining Coefficients for Linear Predictive Coding", U. S. Patent #4,544,919, Oct. 1985.

A. Cumani, "On a Covariance-Lattice Algorithm for Linear Prediction", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 651-654, May 1982.

M. McLaughlin, I. Gerson, F. Hudziak, and K. Kloker, "High Performance Processor for Real-Time Speech Applications", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 859-863, April 1980.

Y. Tohkura, F. Itakura and S. Hashimoto, "Spectral Smoothing Technique in PARCOR Speech Analysis-Synthesis", **IEEE Trans. Acoustics, Speech and Signal Processing**, vol. ASSP-26, pp. 591-596, Dec. 1978.

W. Kleijn, D. Krasinski, and R. Ketchum, "Improved Speech Quality and Efficient Vector Quantization in SELP", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing**, pp. 155-158, April 1988.

Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", **IEEE Trans. Comm.**, vol. COM-28, pp. 84-95, Jan. 1980.

Juin-Hwey Chen and Allen Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering", **Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing,** pp. 2185-2188, 1987.

Annex D (informative): Change history

Change history					
SMG No.	TDoc. No.	CR. No.	Section affected	New version	Subject/Comments
SMG#13				4.0.0	ETSI Publication
SMG#20				5.0.0	Release 1996 version
SMG#22	430/97	A001		5.1.1	UAP 60 comments
SMG#27				6.0.0	Release 1997 version
SMG#29				7.0.0	Release 1998 version
SMG#31				8.0.0	Release 1999 version

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
12-2008	42				Version for Release 8	7.0.0	8.0.0
12-2009	46				Version for Release 9	8.0.0	9.0.0
03-2011	51				Version for Release 10	9.0.0	10.0.0
09-2012	57				Version for Release 11	10.0.0	11.0.0
09-2014	65				Version for Release 12	11.0.0	12.0.0
12-2015	70				Version for Release 13	12.0.0	13.0.0

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
03-2017	SA#75					Version for Release 14	14.0.0
06-2018	SA#80	-	-	-	-	Version for Release 15	15.0.0
2020-07	-	-	-	-	-	Update to Rel-16 version (MCC)	**16.0.0**
2022-04	-	-	-	-	-	Update to Rel-17 version (MCC)	**17.0.0**
2024-03	-	-	-	-	-	Update to Rel-18 version (MCC)	**18.0.0**

History

Document history		
V18.0.0	May 2024	Publication