ETSI TS 126 447 V16.1.0 (2020-09)

Universal Mobile Telecommunications System (UMTS);
LTE;
Codec for Enhanced Voice Services (EVS);
Error concealment of lost packets
(3GPP TS 26.447 version 16.1.0 Release 16)

Reference
RTS/TSGS-0426447vg10

Keywords
LTE,UMTS

ETSI

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

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

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

Intellectual Property Rights

Essential patents

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

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

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.

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 http://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 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 defines a frame loss concealment procedure, also termed frame substitution and muting procedure, which is executed by the Enhanced Voice Services (EVS) decoder when one or more frames (speech or audio or SID frames) are unavailable for decoding due to e.g. packet loss, corruption of a packet or late arrival of a packet.

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] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.441: "Codec for Enhanced Voice Services (EVS); General Overview".

[3] 3GPP TS 26.442: "Codec for Enhanced Voice Services (EVS); ANSI C code (fixed-point)".

[4] 3GPP TS 26.444: "Codec for Enhanced Voice Services (EVS); Test Sequences".

[5] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[6] 3GPP TS 26.446: "Codec for Enhanced Voice Services (EVS); AMR-WB Backward Compatible Functions".

[7] R. Martin, Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics, 2001

[8] 3GPP TS 26.443: "Codec for Enhanced Voice Services (EVS); ANSI C code (floating-point)"

[9] 3GPP TS 26.452: "Codec for Enhanced Voice Services (EVS); ANSI C code; Alternative fixed-point using updated basic operators".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Further EVS codec specific definitions are found in clause 3.1 of [5].

3.2 Symbols

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

EVS codec specific symbol definitions may be found in clause 3.2 of [5].

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AMR	Adaptive Multi Rate (codec)
AMR-NB	Adaptive Multi Rate Narrowband (codec) = AMR
AMR-WB	Adaptive Multi Rate Wideband (codec)
EFR	Enhanced Full Rate (codec)
EVS	Enhanced Voice Services
FB	Fullband
FR	(GSM) Full Rate (codec)
HR	(GSM) Half Rate (codec)
JBM	Jitter Buffer Management
MTSI	Multimedia Telephony Service for IMS
NB	Narrowband
PLC	Packet Loss Concealment
PS	Packet Switched
PSTN	Public Switched Telephone Network
SWB	Super Wideband
WB	Wideband

Further EVS codec specific abbreviations may be found in clause 3.3 of [5].

4 General

The purpose of the frame loss concealment procedure is to conceal the effect of any unavailable EVS frame (speech or audio or SID) for decoding. The concealment of erased frames covers both the reconstruction of missing frames and the techniques to ensure smooth and rapid recovery of normal synthesis after erased segments. The frame loss concealment procedure also covers proper strategies including muting or fading to background noise for the case of multiple frame losses in a row. The purpose of muting the output or fading to background noise in the case of several lost frames in a row is to indicate the breakdown of the channel to the user and to avoid generating possible annoying sounds as a result from the frame loss concealment procedure.

Unless stated differently, fading operations (described in subclause 5.3.4 and subclause 5.4.6) start already with the first lost frame.

Given the architecture and features of the EVS codec (details in [EVS Codec Detailed Algorithmic Description]) the frame loss concealment procedure comprises concealment methods for the various major codec modules, such as signal classification, spectral envelope (LPC), ACELP, MDCT and Bandwidth Extension. A particular feature of the EVS codec is 'guided' frame loss concealment for which the encoder provides specific supplementary data guiding the concealment during erased frames and enhancing the convergence and recovery of the decoder after erased frames. The description in this specification is limited on how to apply the 'guided' frame loss concealment data; the corresponding encoding operations are described as part of the EVS codec algorithmic description [5].

The procedure of the present document is mandatory for implementation in all network entities and User Equipment (UE)s supporting the EVS decoder.

The present document does not describe the ANSI C code of this procedure. For a description of the two reference fixed-point ANSI C code implementations, using different sets of basic operators, see [3] and [9] respectively; for a description of the reference floating-point ANSI C code implementation see [8].

In the case of discrepancy between the procedure described in the present document and its ANSI-C code specifications contained in [3] the procedure defined by [3] prevails. In the case of discrepancy between the procedure described in the present document and its ANSI-C code specifications contained in [8] the procedure defined by [8] prevails. In the case of discrepancy between the procedure described in the present document and its ANSI-C code specifications contained in [9] the procedure defined by [9] prevails.

5 Detailed description

5.1 Concealment operation related to signal classification

5.1.1 Overview

Many concealment methods are based on signal classification. The frame class is either transmitted and decoded from the bit stream, or estimated in the decoder. This estimation process is specified subsequently in subclause 5.1.2; it is performed in the following during normal decoding, if the decoded frame is ACELP or MDCT based TCX. In the case of mode or rate switching, the buffer storing the signal history is resampled appropriately.

5.1.2 Signal class estimation

If possible, the class is directly derived from the coding mode in case of UC or VC modes, i.e. the class is UNVOICED_CLAS in case of UC frame and VOICED_CLAS in case of VC frame. Otherwise, it is estimated at the decoder as follows.

The frame classification at the decoder is based on the following parameters: zero-crossing parameter, Z_{cf}, pitch-synchronous normalized correlation, R_T, pitch coherence parameter, p_c, spectral tilt, e', and pitch synchronous relative energy at the end of the frame, dE_T.

The zero-crossing parameter, Z_{cf}, is averaged over the whole frame. That is,

$$Z_{cf} = \frac{1}{L}\sum_{i=0}^{n_{subfr}} Z_c(i) \tag{1}$$

where Z_c is the number of times the signal sign of the synthesized signal, $\hat{S}_{pre}(n)$, changes from positive to negative during subframe i. The number of subframes n_{subfr} depends of the internal sampling frequency which could be 12.8 kHz or 16 kHz. In case of 12.8kHz the number of subframes is 4 otherwise the number of subframes is 5. In case the internal sampling frequency is 16 kHz, Z_c is multiplied by 0.8.

The pitch synchronous normalized correlation is computed based on a pitch lag, T0, where T0 is the integer part of the pitch lag of the last subframe, or the average of the pitch lag of the last two subframe if it is larger than $3L_{subfr}/2$, where L_{subfr} = 64 is the subframe size. That is

$$T_0 = \begin{cases} \lfloor d_{fr}^{[3]} \rfloor & \text{if} \quad \lfloor d_{fr}^{[3]} \rfloor \le 3L_{subfr}/2 \\ \text{round}\left(\lfloor d_{fr}^{[2]} \rfloor + \lfloor d_{fr}^{[3]} \rfloor\right) & \text{if} \quad \lfloor d_{fr}^{[3]} \rfloor > 3L_{subfr}/2 \end{cases} \tag{2}$$

where $d_{fr}^{[i]}$ is the fractional pitch lag at subframe i.

The pitch synchronous normalized correlation computed at the end of the frame is given by

$$R_T = \begin{cases} C_N(T_0) & \textit{for} \quad T_0 > L_{subfr} \\ 0.5C_N(T_0) + 0.5C_N(2T_0) & \textit{for} \quad T_0 \le L_{subfr} \end{cases} \tag{3}$$

where

$$C_N(kT_0) = \frac{\sum_{n=L-kT_0}^{L-1-(k-1)T_0} \hat{s}_{pre}(n)\hat{s}_{pre}(n-T_0)}{\sqrt{\sum_{n=L-kT_0}^{L-1-(k-1)T_0} \hat{s}_{pre}(n)\hat{s}_{pre}(n)} \sqrt{\sum_{n=L-kT_0}^{L-1-(k-1)T_0} \hat{s}_{pre}(n-T_0)\hat{s}_{pre}(n-T_0)}} \quad (4)$$

where L is the frame size and $\hat{S}_{pre}(n)$ is the synthesized speech signal.

The pitch coherence parameter is compute only in the case that the actual frame is not in TCX MDCT mode. The pitch coherence is given by

$$p_c = \left| d_{fr}^{[3]} + d_{fr}^{[2]} - d_{fr}^{[1]} - d_{fr}^{[0]} \right| \quad (5)$$

where $d_{fr}^{[i]}$ is the fractional pitch lag at subframe i. In case the internal sampling frequency is 16 kHz, p_c is multiplied by 0.8.

The spectral tilt parameter, e', is estimated based on the last 3 subframes and given by

$$e' = \frac{\sum_{n=L_{subfr}}^{L-1} \hat{s}_{pre}(n)\hat{s}_{pre}(n-1)}{\sum_{n=L_{subfr}}^{L-1} \hat{s}_{pre}(n)\hat{s}_{pre}(n)} \quad (6)$$

The pitch synchronous relative energy at the end of the frame is given by

$$dE_T = E_T - \overline{E}_T \quad (7)$$

where

$$E_T = 10\log_{10}\left(\frac{1}{T'} \sum_{n=0}^{T'-1} \hat{s}^2(L-T'+n) \right) \quad (8)$$

and $\overline{E}_T$ is the long-term energy. $\overline{E}_T$ is updated only when a current frame is classified as VOICED_CLAS and is of interoperable coding mode or isn't of generic or transition coding mode, and is classified as VOICED_CLAS at the same time, using the relation

$$\overline{E}_T = 0.99\overline{E}_T + 0.01E_T \quad (9)$$

$$dE_T = E_T - \overline{E}_T \quad (10)$$

The pitch lag value, T', over which the energy, $\overline{E}_T$, is computed is given by

$$\overline{E}_T = 0.99\overline{E}_T + 0.01E_T \; p = \text{round}(0.5d_{fr}^{[2]} + 0.5d_{fr}^{[3]})$$

$$T' = p \quad \text{if} \quad p \geq L_{subfr}$$

$$T' = 2p \quad \text{if} \quad p < L_{subfr}$$

$$dE_T = E_T - \overline{E}_T \quad (11)$$

To make the classification more robust, the classification parameters are considered together forming a function of merit, $\hat{f}_m$. For that purpose, the classification parameters are first scaled so that each parameter's typical value for

unvoiced signal translates in 0 and each parameter's typical value for voiced signal translates into 1. A linear function is used between them. The scaled version, p^s, of a certain parameter, p_x, is obtained using

$$p^s = k_p p_x + c_p \tag{12}$$

$$dE_T = E_T - \overline{E}_T \tag{13}$$

and in case of pc the scaled parameter is constrained by $0 \le pc^s \le 1$.

The function coefficients, k_p, and C_p, have been found experimentally for each of the parameters so that the signal distortion due to the concealment and recovery techniques used in the presence of frame erasures is minimal. The values used are summarized in Table 1 below.

Table 1: Signal classification parameters at the decoder

Parameter	Meaning	Kp	cp
R_T	Normalized correlation	0.8547	0.2479
e'	Spectral tilt	0.8333	0.2917
p_c	Pitch coherence	–0.0357	1.6071
dE_T	Relative frame energy	0.04	0.56
Z_{cf}	Zero-crossing counter	–0.04	2.52

The merit function has been defined as

$$\hat{f}_m = \frac{1}{6}(2R_T^s + e'^s + p_c{}^s + dE_T^s + z_{cf}{}^s) \tag{14}$$

where the superscript s indicates the scaled version of the parameters. In the case of 8-kHz sampled output and a decoded bit rate of 9.6kbps, the merit function, f, is further multiplied by 0.9.

In the case that the actual frame is not in TCX MDCT mode, the pitch coherence is not compute therefore the merit function has been defined as

$$\hat{f}_m = \frac{1}{5}(2R_T^s + e'^s + dE_T^s + z_{cf}{}^s) \tag{15}$$

$$dE_T = E_T - \overline{E}_T \tag{16}$$

The classification is performed using the merit function, $\hat{f}_m$, and following the rules summarized in Table 2. The default class is UNVOICED_CLAS. Note that the class ARTIFICIAL ONSET is set at the decoder if the frame follows an erased frame and artificial onset reconstruction is used as described in subclause 5.3.3.4.2.

Table 2: Signal classification rules at the decoder

Previous frame class	Rule	Current frame class
ONSET ARTIFICIAL ONSET VOICED_CLAS VOICED TRANSITION	$\hat{f}_m \ge 0.63$	VOICED_CLAS
	$0.63 > \hat{f}_m \ge 0.39$	VOICED TRANSITION
	$\hat{f}_m < 0.39$	UNVOICED_CLAS
UNVOICED TRANSITION UNVOICED_CLAS INACTIVE_CLAS	$\hat{f}_m > 0.56$	ONSET
	$0.56 \ge \hat{f}_m > 0.45$	UNVOICED TRANSITION
	$\hat{f}_m \le 0.45$	UNVOICED_CLAS

5.2 Concealment operation related to spectral envelope (LPC) representation

When the LSF parameters of the first good frame are not available, the LSF parameters of the concealed frame are extrapolated using the last LSF parameters. The general idea is to fade the last LSF parameters towards an adaptive LSF mean vector. First, an average LSF vector is calculated from the last 3 known LSF vectors as

$$f_{avg} = \frac{\hat{f}^{[-1]} + \hat{f}^{[-2]} + \hat{f}^{[-3]}}{3} \tag{17}$$

Then, the adaptive mean LSF vector is calculated by

$$f' = \beta \cdot \overline{f}_{static} + (1-\beta) \cdot f_{avg} \tag{18}$$

Then the LSF vector used for concealing the lost frame is computed

$$\hat{f}^{[0]} = \alpha \cdot \hat{f}^{[-1]} + (1-\alpha) \cdot f' \tag{19}$$

where $\overline{f}_{static}$ is the mean LSF vector defined according to Table 3.

Table 3: Values of LSF mean vector $\overline{f}_{static}$

LPC Quantization == 0 AVQ	$\overline{f}_{static}$ **=**
$f_s = 32kHz$	$(739.65,\ 1811.71, 2794.79, 3708.53, 4594.87, 5528.75, 6583.99, 7512.05,$ $8455.51,\ 9352.67, 10266.73, 11133.74, 12067.91, 12958.21, 13940.67, 14794.15)$
$f_s = 25.6kHz$	$(614.44, 1437.24, 2259.37, 2994.68, 3732.57, 4420.10, 5187.93, 5985.97,$ $6790.66, 7523.16, 8283.87, 9010.86, 9757.03, 10458.90, 11209.84, 11888.64)$
$f_s = 16.0kHz$	$(355.08, 696.48, 1260.55, 1735.55, 2220.70, 2676.17, 3123.44, 3560.94,$ $3989.45, 4399.61, 4869.14, 5372.66, 5894.53, 6364.45, 6883.20, 7302.73)$
LPC Quantization == 1 ACELP	$\overline{f}_{static}$ =
$f_s = 16.0kHz$	$(355.08, 696.48, 1260.55, 1735.55, 2220.70, 2676.17, 3123.44, 3560.94,$ $3989.45, 4399.61, 4869.14, 5372.66, 5894.53, 6364.45, 6883.20, 7302.73)$
$f_s = 12.8kHz$	$(289.84, 527.34, 919.53, 1365.23, 1736.72, 2131.25, 2513.28, 2863.67$ $3245.70, 3600.78, 3962.89, 4314.06, 4703.91, 5102.73, 5508.20, 5899.61)$
$f_s = 8.0kHz$	$(326.56, 525.00, 881.64, 1274.61, 1630.08, 1965.23, 2324.22, 2619.92,$ $2935.16, 3216.41, 3469.14, 3687.11, 4059.77, 4775.78, 5412.11, 5912.11)$

Furthermore, β is defined according to Table 4.

Table 4: Values of LSF interpolation factor β

Bitrates:	**5.9, 6.8, 8.0, 13.2, 32 and 64 kbps**	**9.6, 16.4, 24.4, 48, 96 and 128 kbps**
plcBackgroundNoiseUpdated == 1	$\beta = 0.75$	$\beta = 0$
plcBackgroundNoiseUpdated == 0	$\beta = 0.75$	$\beta = 0.25$

α depends on the previous coder type and the signal class of the last good frame for the first 3 lost frames. It is determined according to Table5.

Table 5: Values of LSF interpolation factor α

Last good received frame (coder_type)	Additional criteria	α
UNVOICED		1
INACTIVE or AUDIO	Last_GSC_pit_band_idx > 0 and nbLostCmpt > 1	0.8
	else	0.995
UNVOICED_CLAS	Successively lost frames = 1	$\theta \cdot 0.2 + 0.8$
	Successively lost frames = 2	0.6
	Successively lost frames = 3	0.4
UNVOICED TRANSITION		0.8
VOICED_CLAS		1
ONSET		1
ARTIFICIAL ONSET		0.6
All other cases		0.4

Starting from the 4th consecutive lost frame $\alpha = \frac{1}{nb_{cons}}$.

The estimated LSF vector of the concealed frame $\hat{f}^{[0]}$ is converted to LSP representation and interpolated. The interpolation procedure corresponds to the procedure described in subclause 5.1.9.6 of [5]. The interpolation procedure calculates four or five LSP vectors, each for a given subframe of the concealed frame. The interpolation is done between the LSP vector of the last subframe of the last frame (the one before the concealed frame)and the LSP vector derived from $\hat{f}^{[0]}$ during concealment, as described above.

5.2.1 Specifics to rates 9.6, 16.4 and 24.4kbps

Additionally to estimating the LSF vector $\hat{f}^{[0]}$ there is another LSF vector $\hat{f}_{cng}^{[0]}$ computed

$$\hat{f}_{cng}^{[0]} = \alpha \cdot \hat{f}^{[-1]} + (1-\alpha) \cdot f_{cng} \tag{20}$$

where

$\hat{f}_{cng}^{[0]}$ is an estimated LSF vector used in ACELP concealment,

f_{cng} is the LSF representation of the CNG noise estimation on decoder side (see clause 4.3 in [5]).

5.2.2 Specifics to AMR-WB IO mode

The same procedure is performed, but instead of LSFs, ISFs are used for the estimation. The mean LSF vector used for interpolation is

$$\overline{f}_{static} = (288.411774,\ 518.149414,\ 912.352051,\ 1397.743652,\ 1795.418823,\ 2211.536133,$$
$$2621.461182,\ 3019.680176,\ 3417.989746,\ 3809.700928,\ 4181.547363,$$
$$4581.064941,\ 5012.819824,\ 5457.521484,\ 5876.145020,\ 1576.906494)$$

5.2.3 Check for Mid LSF stability

The interpolation of the mid-LSF can create unstable LSFs under packet erasure conditions. Let the k th sub-frame LSFs are given by $x_n^k;\ k = \{1,2,3,4\}$. We denote the last sub-frame LSF of n th frame as x_n^e where $x_n^e = x_n^4$. Let us denote the i th LSF dimension of the k th sub-frame of frame n as $x_{i,n}^k$ where $i = \{1,2,...,M\}$.

The end-LSF quantizer quantizes x_n^e. Then mid-LSF quantizer interpolates the mid-LSFs as follows.

$$x_{i,n}^{m} = w_{i,n} \cdot x_{i,n}^{e} + (1 - w_{i,n}) \cdot x_{i,n-1}^{e} \quad (21)$$

where i-th dimension of the weighting vector w_n is given by $w_{i,n}$. The vector elements $w_{i,n}$ are not constrained. In particular if $0 \le w_{i.n} \le 1$ interpolation generates a mid-LSF $x_{i,n}^{m}$ between $x_{i,n-1}^{e}$ and $x_{i,n}^{e}$. However if $w_{i,n} < 0$ or $w_{i,n} > 1$ interpolation might generate a mid-LSF $x_{i,n}^{m}$ outside $[x_{i,n}^{e}, x_{i,n-1}^{e}]$. This could potentially create LSF clustering that result in an unstable LSF synthesis filter. To remedy this situation, a potential instability is detected as described below.

In the frame that follows the packet loss, the decoder checks whether the computed mid-LSFs are ordered correctly i.e. $(x_{1,n}^{m} + \Delta) < (x_{2,n}^{m} + \Delta) < \cdots < (x_{M,n}^{m} + \Delta)$. If violation of this rule is detected the LSFs are considered as potentially unstable. If potential LSF instability is detected, decoder uses a fixed weighting value α_{fixed} (typically 0.6) for mid LSF interpolation as follows.

$$x_{i,n}^{m} = \alpha_{\text{fixed}} \cdot x_{i,n}^{e} + (1 - \alpha_{\text{fixed}}) \cdot x_{i,n-1}^{e} \quad (22)$$

The mid LSF interpolation based on equation (22) is continued until frame $n+k$ such that the frame $n+k$ is the first frame after frame n that uses safety net quantization for quantizing its end LSF.

5.2.4 Adaptive interpolation of LSFs

The sub-frame LSFs are interpolated based on $x_{i,n-1}^{e}$, $x_{i,n}^{m}$ and $x_{i,n}^{e}$ using fixed interpolation factors given by

$$x_{i,n}^{k} = \alpha_{\text{k}} \cdot x_{i,n}^{e} + \beta_{\text{k}} \cdot x_{i,n-1}^{e} + (1 - \alpha_{\text{k}} - \beta_{\text{k}}) \cdot x_{i,n}^{m} \quad (23)$$

where $x_{i,n}^{m}$ and $x_{i,n}^{e}$ correspond to the mid and end LSFs of frame n respectively. Note that α_{k} and β_{k} such that $0 \le \alpha_{\text{k}}, \beta_{\text{k}} \le 1$, and those are fixed values known to both encoder and decoder. If the frame $(n-1)$ is lost its end LSFs are estimated by the decoder. However the dependence on the estimated end LSFs of the $(n-1)$th frame may adversely affect the speech quality if the estimated end LSFs are not well represent the actual one. This issue is addressed by selecting the interpolation factors α_{k} and β_{k} appropriately by giving more weight to the end LSF of frame n which is not lost.

To adaptively select the LSF interpolation factors, we estimate the gain of the synthesis filter resulting from the LSF vectors $x_{i,n-1}^{e}$ and $x_{i,n}^{e}$ by computing the energy of the impulse response of the corresponding synthesis filters. Let the impulse responses of the synthesis filters corresponding to $x_{i,n-1}^{e}$ and $x_{i,n}^{e}$ are given by $h_{n-1}(k)$ and $h_n(k)$. The truncated energy of the impulse responses are given by E_{n-1} and E_n where

$$E_n = \sum_{i}^{N} h_n^2(i) \quad (24)$$

Note that N is the length of the truncated response. Typically 128 samples are used to compute the truncated impulse response. The interpolation factors α_{k} and β_{k} are picked based on the energy ratio $R = E_n / E_{n-1}$, coder type, FEC classification and the use of safety net quantization for LSF quantization.

These interpolation factors are picked from four different sets of interpolation factors. For example a largest difference in E_n should pick the interpolation factors that give the very little or zero weight to the previous end LSF in the interpolation.

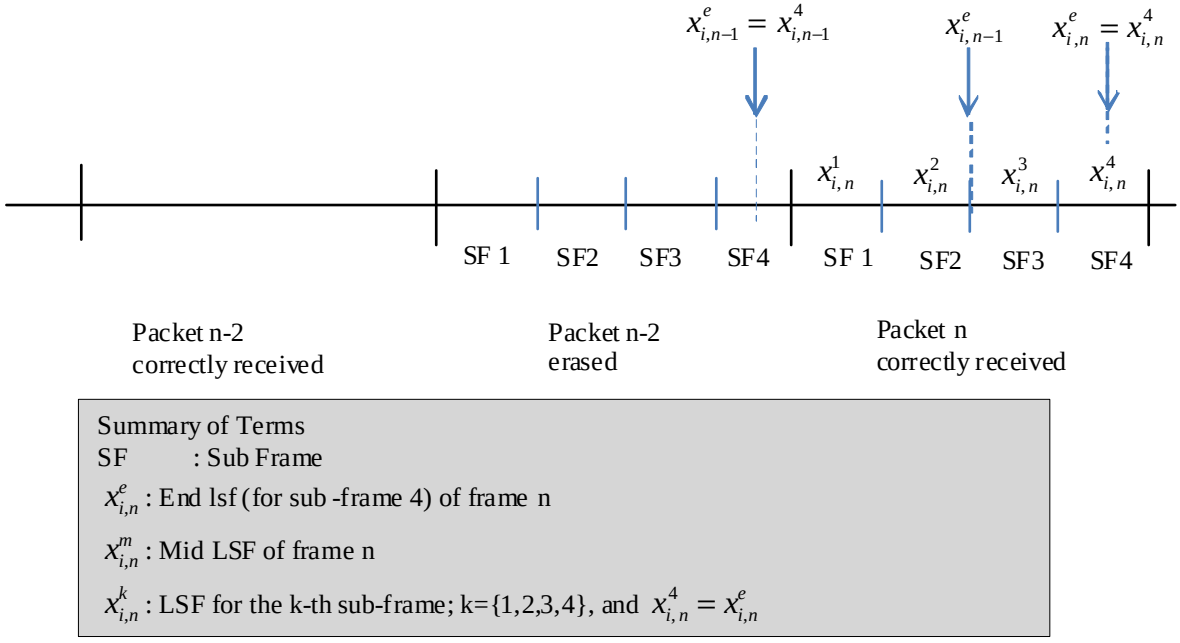

Figure 1: Adaptive interpolation of LSFs

5.2.5 LPC gain compensation

At 9.6, 16.4, 24.4, 48, 96 and 128 kbps, the LPC concealment and interpolation will lead to a change of overall gain of the signal, which is unwanted when targeting a certain background noise level during consecutive frame loss. Therefore the energy of the LPC is measured and stored during decoding of regular frames. In a concealment frame the energy of the concealed LPC is measured and compared to the LPC energy of the last correctly received frame and any change is compensated.

To measure the LPC energy, a vector of length 64 is generated and initialized to all zero. Then the first entry is set to one:

$$imp_{input} = \{1,0,...,<63zeros>,...,0\} \quad (25)$$

imp_{input} is fed into the LPC synthesis filter, where the filter memory is initialized with zeros. The output of the filter (impulse response) is denoted as imp_{LPC}. After filtering, the root mean square energy is calculated by:

$$energ_{\mathrm{LPC}} = \sqrt{\sum_{i=0}^{63} (imp_{\mathrm{LPC}}(i))^2} \quad (26)$$

In correctly received frames the energy is calculated and stored using the latest LPC available.

In case of concealment the compensation differs for ACELP and TCX:

For ACELP there will be 4 or 5 sets of LPC coefficients, depending on the number of subframes to be synthesized. For each set of coefficients the corresponding energy is calculated and divided by the energy derived in the last good frame. The result of the division is used as a factor to be multiplied to each element of the excitation vector of the corresponding subframe. See sub-clause 5.3.4.2.1.

For TCX, there will be one or two sets of coefficients (TCX10/TCX20). For each set of coefficients the corresponding energy is calculated and divided by the energy derived in the previous segment. The segment size equals 10 ms for TCX10 and 20 ms for TCX20. As the fade out is performed in the time domain, the LPC gain compensation is also done in the time domain, by linearly fading from the last compensation factor (would be 1 for the first lost frame) to the derived compensation factor at the end of the segment. See sub-clause 5.4.6.1.3.

5.3 Concealment operation related to ACELP modes

5.3.1 General

In case of frame erasures, the concealment strategy can be summarized as a convergence of the signal energy and the spectral envelope to the estimated parameters of the background noise. A frame erasure is signalled to the decoder by setting the bad frame indicator variable for the current frame active. The periodicity of the signal is converged to zero. The speed of the convergence is dependent on the parameters of the last correctly received frame and the number of consecutive erased frames, and is controlled by an attenuation factor, α. The factor, α, is further dependent on the stability, θ, of the LP filter for UNVOICED_CLAS frames. In general, the convergence is slow if the last good received frame is in a stable segment and is rapid if the frame is in a transition segment. The values of α are summarized in subclause 5.3.4.1 for the excitation concealment of rates: 5.9, 7.2, 8.0, 13.2, 32 and 64 kbps and in subclause 5.3.4.2.3 for the rates: 9.6, 16.4 and 24.4 kbps. Similar values are also defined for LSF concealment as described in subclause 5.2.

5.3.1.1 Extrapolation of future pitch

In case of a frame loss, an estimation of the end-of-frame pitch is done to help keeping the adaptive codebook in sync to the error free case as good as possible. If the error free end-of-frame pitch can be predicted precisely, the recovery after the loss will be a lot quicker. The pitch extrapolation assumes that the encoder uses a smooth pitch contour. The information on the estimated end-of-frame pitch is used by the glottal pulse resynchronization tool described in subclause 5.3.1.2.

The pitch extrapolation is done only if the last good frame was classified as UNVOICED TRANSITION, VOICED TRANSITION or VOICED_CLAS. Also the pitch extrapolation is only performed if the frame before the loss was a good frame. The extrapolation is done based on the pitch lags, d_{fr}^{i}, of the last 5 subframes before the erasure. Also the history of the pitch gains, g_{p}^{i}, of the last 6 subframes before the erasure is needed. The history update of the pitch lags and pitch gains is done after the synthesis of every frame.

First, the difference between the pitch lags is computed:

$$\delta_{dfr}^{[i]} = d_{fr}^{[i]} - d_{fr}^{[i-1]} \text{ for } i = -1,...,-5 \tag{27}$$

where $d_{fr}^{[-1]}$ denotes the last subframe of the previous frame, $d_{fr}^{[-2]}$ denotes the second last sub-frame of the previous frame, and so on.

In case the last good frame contained information about future pitch gains and pitch lags, d_{dfr}^{i} is instead calculated by:

$$\delta_{dfr}^{[i]} = d_{fr}^{[i+2]} - d_{fr}^{[i+1]} \text{ for } i = -1,...,-5 \tag{28}$$

Also in case of information about future pitch gains and pitch lags was contained in the previous frame, the history of pitch gains is shifted by 2 subframes in a way that the $(i+2)$-th pitch gain is moved to the i-th pitch gain, for $i = -1,...,-5$.

Future subframe information might be available if the last good frame was coded with TCX MDCT and there was LTP information available, or the last good frame was coded with ACELP and there was future pitch information transmitted in the bitstream (see subclause 5.3.3.1).

The sum of the differences is computed as

$$s_{\delta} = \sum_{i=-1}^{-5} \delta_{dfr}^{[i]} \tag{29}$$

The position of the maximum absolute difference, $i_{\max} = \arg\left\{ \max_{i=-1}^{-5} \left(abs\left(\delta_{dfr}^{[i]} \right) \right) \right\}$, is found.

If the criterion $\delta_{dfr}^{i_{\max}} < 0.15 \cdot pitch_{old}$ AND $abs\left(\delta_{dfr}^{[i_{\max}]}\right) < abs(s_\delta)$ is met, pitch prediction is performed. Else no prediction is performed and $pitch_{old}$ is used for building the adaptive codebook during concealment.

Pitch prediction is performed by minimizing this error equation.

$$error = \sum_{i=-1}^{-5} g_p^i \left((a + b \cdot i) - \left(d^{[i]}\right)^2 \right) \tag{30}$$

where:

$error$ is the error function,

g_p are the past adaptive codebook gains,

a and b are unknown variables which need to be determined,

$d^{[i]}$ are the adaptive codebook lags from the past frames,

i is the subframe index

The past adaptive codebook gains are multiply by a vector {1.25f, 1.125f, 1.f, 0.875f, .75f};

Minimizing of this function is done by deviating the error function by a and b separately

$$\begin{aligned}
\frac{\Delta error}{\Delta a} &= 8 \cdot g_p^{[-1]} \left(- d^{[-1]} + 4 \cdot b + a \right) + 2 \cdot g_p^{[-2]} \left(- d^{[-2]} + 3 \cdot b + a \right) \\
&\quad + 2 \cdot g_p^{[-3]} \left(- d^{[-3]} + 2 \cdot b + a \right) + 2 \cdot g_p^{[-4]} \left(- d^{[-4]} + b + a \right) + 2 \cdot g_p^{[-5]} \left(- d^{[-5]} + a \right) \\
\frac{\Delta error}{\Delta b} &= 8 \cdot g_p^{[-1]} \left(- d^{[-1]} + 4 \cdot b + a \right) + 6 \cdot g_p^{[-2]} \left(- d^{[-2]} + 3 \cdot b + a \right) \\
&\quad + 4 \cdot g_p^{[-3]} \left(- d^{[-3]} + 2 \cdot b + a \right) + 2 \cdot g_p^{[-4]} \left(- d^{[-4]} + b + a \right)
\end{aligned} \tag{31}$$

By setting the derivatives $Envelope^{[m-1]}$ and $\frac{\Delta error}{\Delta b} = 0$ to zero, this leads to:

$$\begin{aligned}
a = -\Big(& \left(3 \cdot g_p^{[-4]} + 4 \cdot g_p^{[-3]} + 3 \cdot g_p^{[-2]} \right) \cdot g_p^{[-5]} \cdot d^{[-5]} \\
& + \left(\left(2 \cdot g_p^{[-3]} + 2 \cdot g_p^{[-2]} \right) \cdot g_p^{[-4]} - 4 \cdot g_p^{[-4]} \cdot g_p^{[-5]} \right) \cdot d^{[-4]} \\
& + \left(- 8 \cdot g_p^{[-3]} \cdot 3 \cdot g_p^{[-5]} - 3 \cdot g_p^{[-3]} \cdot g_p^{[-4]} + g_p^{[-2]} - 2 \cdot g_p^{[-3]} \right) \cdot d^{[-3]} \\
& + \left(- 12 \cdot g_p^{[-2]} \cdot g_p^{[-5]} - 6 \cdot g_p^{[-2]} - 2 \cdot g_p^{[-4]} - 2 \cdot g_p^{[-2]} - 2 \cdot g_p^{[-3]} \right) \cdot d^{[-2]} \\
& + \left(- 16 \cdot g_p^{[-1]} \cdot g_p^{[-5]} - 9 \cdot g_p^{[-1]} \cdot g_p^{[-4]} - 4 \cdot g_p^{[-1]} \cdot g_p^{[-3]} - g_p^{[-1]} \cdot g_p^{[-2]} \right) \cdot d^{[-1]} \Big) \\
& / denominator
\end{aligned} \tag{32}$$

$$\begin{aligned}
b = -\Big(& \left(g_p^{[-4]} + 2 \cdot g_p^{[-3]} + 3 \cdot g_p^{[-2]} + 4 \cdot g_p^{[-1]} \right) \cdot g_p^{[-5]} \cdot d^{[-5]} \\
& + \left(\left(g_p^{[-3]} + 2 \cdot g_p^{[-2]} + 3 \cdot g_p^{[-1]} \right) \cdot g_p^{[-4]} - g_p^{[-4]} \cdot g_p^{[-5]} \right) \cdot d^{[-4]} \\
& + \left(- 2 \cdot g_p^{[-3]} \cdot g_p^{[-5]} - g_p^{[-3]} \cdot g_p^{[-4]} + \left(g_p^{[-2]} + 2 \cdot g_p^{[-1]} \right) \cdot g_p^{[-3]} \right) \cdot d^{[-3]} \\
& + \left(- 3 \cdot g_p^{[-2]} \cdot g_p^{[-5]} - 2 \cdot g_p^{[-2]} \cdot g_p^{[-4]} - g_p^{[-2]} \cdot g_p^{[-3]} + g_p^{[-1]} \cdot g_p^{[-2]} \right) \cdot d^{[-2]} \\
& + \left(- 4 \cdot g_p^{[-1]} \cdot g_p^{[-5]} - 3 \cdot g_p^{[-1]} \cdot g_p^{[-4]} - 2 \cdot g_p^{[-1]} \cdot g_p^{[-3]} - g_p^{[-1]} \cdot g_p^{[-2]} \right) \cdot d^{[-1]} \Big) \\
& / denominator
\end{aligned} \tag{33}$$

where

$$\begin{aligned} denominator = & \left(g_p^{[-4]} + 4 \cdot g_p^{[-3]} + 9 \cdot g_p^{[-2]} + 16 \cdot g_p^{[-1]}\right) \cdot g_p^{[-5]} \\ & + \left(g_p^{[-3]} + 4 \cdot g_p^{[-2]} + 9 \cdot g_p^{[-1]}\right) \cdot g_p^{[-4]} \\ & + \left(g_p^{[-2]} + 4 \cdot g_p^{[-1]}\right) \cdot g_p^{[-3]} \\ & + g_p^{[-1]} \cdot g_p^{[-2]} \end{aligned} \tag{34}$$

The end-of-frame pitch is determined according to this, if no information about future subframes was available in the previous frame:

$$P_{pred} = a + 4 \cdot b \tag{35}$$

In case there was information about future pitch gains and pitch lags available the end-of-frame pitch is predicted by:

$$P_{pred} = a + 2 \cdot b \tag{36}$$

After this processing, the predicted pitch P_{pred} is limited between $d_{\min}$ and $d_{\max}$.

5.3.1.2 Construction of the periodic part of the excitation

For a concealment of erased frames following a correctly received UNVOICED_CLAS frame, no periodic part of the excitation is generated. For a concealment of erased frames following a correctly received frame other than UNVOICED_CLAS, the periodic part of the excitation is constructed by repeating the low-pass filtered last pitch period of the previous frame. The low-pass filter used is a simple 3-tap linear phase FIR filter with the coefficients equal to 0.18, 0.64 and 0.18. The pitch period, Tc, used to select the last pitch pulse, and hence used during the concealment, is defined so that pitch multiples or submultiples can be avoided or reduced. The following logic is used in determining the pitch period, Tc

if ((T[–1] < 1.8Ts) AND (T[–1] > 0.6Ts)) OR (Tcnt >=5)

tmp_tc = T[–1]

else

tmp_tc = Ts

Tc = round(tmp_tc)

Here, T[–1] = $d_{fr}^{[-1]}$ is the pitch period of the last subframe of the last good received frame and Ts is the pitch period of the last subframe of the last good stable voiced frame with coherent pitch estimates. A stable voiced frame is defined here as a VOICED_CLAS frame, preceded by a frame of voiced type (VOICED TRANSITION, VOICED_CLAS, ONSET). The coherence of pitch is verified by examining whether the closed-loop pitch estimates are reasonably close; i.e. whether the ratio between the 4th subframe pitch, $d_{fr}^{[-1]}$ at 12.8 kHz core sampling frequency or $d_{fr}^{[-2]}$ at 16 kHz core sampling frequency, and the 2nd subframe pitch, $d_{fr}^{[-3]}$ at 12.8 kHz core sampling frequency or $d_{fr}^{[-4]}$ at 16 kHz core sampling frequency, is within the interval [0.7, 1.4], and whether the ratio between the 2nd subframe pitch ($d_{fr}^{[-3]}$ or $d_{fr}^{[-4]}$) and the last subframe pitch of the preceding frame, $d_{fr}^{[-y]}$, is also within that interval, where y = 5 when the core sampling frequency is 12.8 kHz and y = 6 otherwise. The pitch is also assumed cohererent if the coding type is transition.

This determination of the pitch period, Tc, implies that if the pitch at the end of the last good frame and the pitch of the last stable frame are close, the pitch of the last good frame is used. Otherwise, this pitch is considered unreliable and the pitch of the last stable frame is used instead to avoid the impact of erroneous pitch estimates at voiced onsets. This logic

is valid only if the last stable segment is not too far in the past. Hence, a counter, Tcnt, is defined that limits the effect of the last stable segment. If Tcnt is greater than or equal to 5; i.e. if there are at least 5 frames since the last Ts update, the last good frame pitch is used systematically. Tcnt is reset to 0 every time a stable segment is detected and Ts is updated. The period Tc is then maintained constant during the concealment for the entire erased block.

5.3.1.2.1 Particularity of rate 5.9, 7.2, 8.0 and 13.2 kbps

On top of UNVOICED_CLAS, the periodic component of the excitation is not constructed when the last coding mode was GSC AUDIO without a temporal contribution or the last class was INACTIVE without a temporal contribution.

5.3.1.3 Glottal pulse resynchronization

The construction of the periodic part of the excitation, described in the subclause 5.3.1.2, may result in a drift of the glottal pulse position in the concealed frame during voiced segments, since the pitch period used to build the excitation can be different from the encoder pitch period. This will cause the adaptive codebook (or past CELP excitation) to be desynchronized from the actual CELP excitation. Thus, in case a good frame is received after an erased frame, the pitch excitation (or adaptive codebook excitation) will have an error which may persist for several frames and affect the performance of the correctly received frames.

To overcome this problem and improve the decoder convergence, a resynchronization method is used which adjusts the position of the glottal pulses in the concealed frame to be synchronized with the estimated true glottal pulses positions where the positions of the glottal pulses are estimated at the decoder based on the pitch extrapolation performed as in subclause 5.3.1.1. Therefore, this resynchronization procedure is performed based on the estimation of phase information and it aligns the maximum pulse in each pitch period of the concealed frame to the estimated position of the glottal pulse.

The starting point is the constructed periodic part of the excitation src_exc, constructed as described in subclause 5.3.1.2. If $P_{pred} < tmp\_tc$ then samples are removed from src_exc and if $P_{pred} > tmp\_tc$ then samples are added to src_exc. The samples are added or removed at the locations of the minimum energy, between the estimated locations of the glottal pulses as well as the locations of the minimum energy before the estimated location of the first and after the estimated location of the last glottal pulse. The periodic part of the excitation, modified in such way, is stored into dst_exc.

5.3.1.3.1 Condition to perform resynchronisation

The glottal pulse resynchronisation is performed only if some conditions, which describe that a reliable estimation of true pulse positions is available and is different from the actual pulse positions, are met. First the extrapolation of the future pitch as performed in subclause 5.3.1.1 shall have been successful. The pitch period Tc as defined in subclause 5.3.1.2 shall be different than the rounded predicted pitch P_{pred} as defined in subclause 5.3.1.1. The absolute difference between the pitch period Tc and the rounded predicted pitch P_{pred} shall be smaller than $0.15 \cdot T_C$. In order to have enough samples for the pulse resynchronization in the periodic part of the excitation constructed by repeating the last pitch period, the relative pitch change shall be greater than a threshold as described below:

$$\frac{P_{pred}}{tmp\_tc} > 1 - \frac{2}{n_{subfr} + 1} \tag{37}$$

where n_{subfr} is the number of subframes as defined in subclause 5.1.2.

If the conditions to perform the glottal pulse resynchronization are not met, the samples from src_exc are simply copied to dst_exc. In some instances, where the glottal pulse resynchronization is used, this is implemented in a such way that src_exc points to the final location of the modified periodic part of the excitation, and that its contents are first copied to another temporary buffer which is then considered as src_exc inside the pulse resynchronization and the final location which for the caller is src_exc is considered as dst_exc inside the pulse resynchronization.

5.3.1.3.2 Performing glottal pulse resynchronization

First the pitch change per sub-frame δ is calculated as:

$$\delta = \frac{P_{pred} - tmp\_tc}{n_{subfr}} \tag{38}$$

Then the number of samples to be added (to be removed if negative) d is calculated as:

$$d = \delta \frac{L}{Tc} \frac{n_{subfr} + 1}{2} - L\left(1 - \frac{tmp\_tc}{Tc}\right) \tag{39}$$

Then the location of the first maximum pulse $T[0]$, among first T_C samples in src_exc is found using simple search for the maximum absolute value.

The index of the last pulse that will be present in dst_exc is calculated as:

$$k = \left\lceil \frac{L - d - T[0]}{T_C} - 1 \right\rceil \tag{40}$$

The delta of the samples to be added or removed between consecutive pitch cycles a is calculated as:

$$a = \frac{\left|T_C - P_{pred}\right|(L - d) - |d|T_C}{(k+1)\left(T[0] + \frac{k}{2}T_C\right)} \tag{41}$$

The number of samples to be added or removed before the first pulse is calculated as:

$$\Delta_0^p = \left(\left|T_C - P_{pred}\right| - (k+1)a\right)\frac{T[0]}{T_C} \tag{42}$$

The number of samples to be added or removed before the first pulse is rounded down and the fractional part is kept in memory:

$$\begin{aligned} \Delta_0^{'} &= \lfloor \Delta_0^p \rfloor \\ F &= \Delta_0^p - \Delta_0^{'} \end{aligned} \tag{43}$$

For each region between 2 pulses the number of samples to be added or removed is calculated as:

$$\Delta_i = \left|T_C - P_{pred}\right| - (k+1-i)a, \qquad 1 \le i \le k \tag{44}$$

The number of samples to be added or removed between 2 pulses, taking into account the remaining fractional part from the previous rounding, is rounded down:

$$\begin{aligned} \Delta_i^{'} &= \lfloor \Delta_i + F \rfloor \\ F &= \Delta_i - \Delta_i^{'} \end{aligned} \tag{45}$$

If, due to the added F, for some i it happens that $\Delta_i > \Delta_{i-1}$, then the values for Δ_i and Δ_{i-1} are swapped.

The number of samples to be added or removed after the last pulse is calculated as:

$$\Delta_{k+1}^{'} = \lfloor d + 0.5 \rfloor - \sum_{i=0}^{k} \Delta_i^{'} \tag{46}$$

The maximum number of samples to be added or removed among the minimum energy regions is calculated as:

$$\Delta_{\max}^{'} = \max_i \Delta_i^{'} = \begin{cases} \Delta_k^{'}, & \Delta_k^{'} \ge \Delta_{k+1}^{'} \\ \Delta_{k+1}^{'}, & \Delta_k^{'} < \Delta_{k+1}^{'} \end{cases} \tag{47}$$

The location of the minimum energy segment $P_{\min}[1]$ between the first two pulses in src_exc, that has $\Delta'_{\max}$ length, is then found by simple search for minimum in the moving average of length $\Delta'_{\max}$. For every consecutive minimum energy segment between two pulses, the position is calculated as:

$$P_{\min}[i] = P_{\min}[1] + (i-1)T_C, \quad 1 < i \le k \tag{48}$$

If $P_{\min}[1] > T_C$ then the location of the minimum energy segment before the first pulse is calculated using $P_{\min}[0] = P_{\min}[1] - T_C$. Otherwise the location of the minimum energy segment $P_{\min}[0]$ before the first pulse in src_exc is found by simple search for minimum in the moving average of length Δ'_0.

If $P_{\min}[1] + kT_C < L - d$ then the location of the minimum energy segment after the last pulse is calculated using $P_{\min}[k+1] = P_{\min}[1] + kT_C$. Otherwise the location of the minimum energy segment $P_{\min}[k+1]$ after the last pulse in src_exc is found by simple search for minimum in the moving average of length Δ'_{k+1}.

If there is going to be just one pulse in dst_exc, that is if k is equal to 0, the search for $P_{\min}[1]$ is limited to $L - d$. $P_{\min}[1]$ then points to the location of the minimum energy segment after the only pulse in dst_exc.

If $d > 0$ then Δ'_i samples are added at location $P_{\min}[i]$ for $0 \le i \le k+1$ to the signal src_exc and stored in dst_exc, otherwise if $d < 0$ then Δ'_i samples are removed at location $P_{\min}[i]$ for $0 \le i \le k+1$ from the signal src_exc and stored in dst_exc. There are $k+2$ regions where the samples are added or removed.

5.3.1.4 Construction of the random part of the excitation

The innovative (non-periodic) part of the excitation is generated randomly. A simple random generator with approximately uniform distribution is used. Before adjusting the innovation gain, the randomly generated innovation is scaled to some reference value, fixed here to the unitary energy per sample. At the beginning of an erased block, the innovation gain, g_s, is initialized by using the innovative excitation gains of each subframe of the last good frame

for 4 subframes:

$$g_s = 0.1g_c^{[-4]} + 0.2g_c^{[-3]} + 0.3g_c^{[-2]} + 0.4g_c^{[-1]} \tag{49}$$

for 5 subframes:

$$g_s = \frac{1}{15}g_c^{[-5]} + \frac{2}{15}g_c^{[-4]} + \frac{3}{15}g_c^{[-3]} + \frac{4}{15}g_c^{[-2]} + \frac{5}{15}g_c^{[-1]} \tag{49a}$$

where $g_c^{[-5]}$, $\hat{g}_c^{[-4]}$, $\hat{g}_c^{[-3]}$, $\hat{g}_c^{[-2]}$ and $\hat{g}_c^{[-1]}$ are the algebraic codebook gains of the four subframes of the last correctly received frame. The attenuation strategy of the random part of the excitation is somewhat different from the attenuation of the pitch excitation. The reason is that the pitch excitation (and thus the excitation periodicity) is converging to 0 while the random excitation is converging to the CNG excitation energy. The innovation gain attenuation is calculated as

$$g_s^{[1]} = \alpha g_s^{[0]} + (1-\alpha)g_n \tag{50}$$

where $g_s^{[1]}$ is the innovative gain at the beginning of the next frame, $g_s^{[0]}$ is the innovative gain at the beginning of the current frame, g_n is the gain of the excitation used during the comfort noise generation and α is as defined in Table 4. The comfort noise gain, g_n, is given as the square root of the energy $\tilde{E}'$ as described in subclause 5.4.3.6.4. Similarly to the periodic excitation attenuation, the gain is thus attenuated linearly throughout the frame on a sample-by-sample basis starting with, $g_s^{[0]}$, and going to the value of $g_s^{[1]}$ that would be achieved at the beginning of the next frame.

Finally, if the last correctly received frame is different from UNVOICED_CLAS, the innovation excitation is filtered through a linear phase FIR high-pass filter with coefficients –0.0125, –0.109, 0.7813, –0.109, and –0.0125. To decrease the amount of noisy components during voiced segments, these filter coefficients are multiplied by an adaptive factor equal to $(0.75-0.25r_v)$, with r_v denoting the voicing factor as defined in equation (1475) in subclause 6.1.1.3.2 of [5]. The random part of the excitation is then added to the adaptive excitation to form the total excitation signal. If the last good frame is UNVOICED_CLAS, only the innovative excitation is used and it is further attenuated by a factor of 0.8. In this case, the past excitation buffer is updated with the innovation excitation, as no periodic part of the excitation is available. If the last good frame is UNVOICED_CLAS or INACTIVE but it is not coded with UC mode signalling non-stationary unvoiced frame, the innovation excitation is further attenuated by a factor of 0.8.

5.3.1.5 Spectral envelope concealment, synthesis and updates

To synthesize the decoded speech, the LP filter parameters shall be obtained. The spectral envelope is gradually moved to an estimated reference envelope, see clause 5.2. The estimated LSF vector is converted to an LSP vector and interpolated with the last frame's LSP vector for 4 or 5 subframes, depending on the ACELP sampling rate being 12.8 kHz or 16 kHz.

The synthesized signal is obtained by filtering the sum of the adaptive and the random excitation signal through the LP synthesis filter (see clause 6.1.3 of [5]) and post-processed similar to the steps performed in clean channel.

As the LSF quantizers uses prediction, their memories would not be up to date after the normal operation is resumed. To reduce this effect, the quantizers' memories (moving average and auto-regressive) are estimated and updated at the end of each erased frame.

5.3.1.5.1 Specifics for rates 9.6, 16.4 and 24.4 kbps

The coefficients of the filter used in subclause 5.3.1.2 for low pass filtering of the first pitch cycle are dependent on the sampling rate. The pitch period, tmp_tc, is always equal to T[-1], where T[-1] is the pitch period of the last sub-frame of the last good received frame, and Tc, used to select the last pitch pulse is thus equal to round(T[-1]).

The periodic part of the excitation will be generated further by repeatedly copying the last pitch cycle of the dst_exc for an additional half frame, which is used for correctly updating the overlap-add buffers for MDCT recovery.

In contrast to subclause 5.3.1.5, both excitation signals are not added up and filtered. The synthesized signal is obtained by filtering the adaptive excitation through the LP synthesis filter based on the LSF interpolation according to formula (19). The random part of the excitation is filtered through the LP filter based on formula (21). After obtaining two separate synthesis signals, they are added up, post processed and played out like in a correctly received frame. Note, that the memories for both of the LP synthesis filters are initialized with the last known state of the last good frame in the beginning of a frame loss. For consecutive loss, they are updated and stored separately.

5.3.1.6 GSC mode concealment

When the concealment is performed based on the GSC core, the construction of the periodic part of the excitation is performed as described in subclauses 5.3.1.2 and 5.3.1.3. The reconstruction of the periodic part of the excitation corresponds to the time domain contribution of the GSC model. Thus, the reconstructed periodic excitation is converted into the frequency domain using the DCT_{IV} as described in subclause 5.2.3.5.3.1 of [5] and the spectrum above the last known cut-off frequency is smoothed-out to zero.

Then, the spectral concealment is performed by using the last good band energies received. In case of INACTIVE content or active SWB UC mode, the last good decoded spectrum is mixed with random noise at a rate of 4/5 random noise and 1/5 the last decoded spectrum, making the spectrum to become noisy quite fast. In case the last good frame was AUDIO, no noise is added, but the spectrum dynamic is attenuated by 25 %.

The next step consists in adding the spectrum of the reconstructed periodic excitation to the concealed spectrum of the frequency domain contribution and to perform the inverse DCT_{IV} similarly as done in subclause 5.2.3.5.3.1 of [5] to get the final concealed excitation in case of GSC mode.

5.3.1.7 Specifics for AMR-WB IO modes

In case of AMR-WB IO the subclause 5.1.2 is complement with a few more parameters that allow the interoperable decoder to know if the decoded frame contains more likely speech of generic audio and if the current frame contains an

onset. The generic audio can include music, reverberant speech and can also include background music. To determine with good confidence that the current frame can be categorized as generic audio, two parameters are used. The total frame energy E_T as formulated in subclause 5.1.2 and the statistical deviation of the energy variation history σ_E.

First, a mean of the past forty (40) total frame energy variations $\overline{E}_{df}$ is calculated using the following relation:

$$\overline{E}_{df} = \frac{\sum_{t=-40}^{t=-1} \Delta_E^t}{40}; \qquad where \;\; \Delta_E^t = E_T^t - E_T^{(t-1)} \tag{51}$$

Then, a statistical deviation of the energy variation history σ_E over the last fifteen (15) frames is determined using the following relation:

$$\sigma_E = 0.7745967 \sqrt{\sum_{t=-15}^{t=-1} \frac{\left(\Delta_E^t - \overline{E}_{df}\right)}{15}} \tag{52}$$

The resulting deviation σ_E gives an indication on the energy stability of the decoded synthesis. Typically, music has a higher energy stability (lower statistical deviation of the energy variation history) than speech.

Additionally, the first step classification is used to evaluate the interval between two frames classified as unvoiced N_{uv} when the coder type is different from INACTIVE. When a frame is classified as unvoiced and the coder type is different from INACTIVE, meaning that the signal is unvoiced but not silence, if the long term active content energy $\overline{E}_T$, as formulated in subclause 5.1.2, is below 40 dB the unvoiced interval counter is set to 16, otherwise the unvoiced interval counter N_{uv} is decreased by 8 and also limited between 0 and 300 for active signal and between 0 and 125 for inactive signal. It is reminded that, the difference between active and inactive signal is deduced from the voice activity detection VAD information included in the bitstream.

A long term average is derived from this unvoiced frame counter as follow for active signal:

$$N_{uv_{lt}} = 0.9 \cdot N_{uv_{lt}} + 0.1 \cdot N_{uv} \tag{53}$$

And as follows for inactive signal:

$$N_{uv_{lt}} = 0.95 \cdot N_{uv_{lt}} \tag{54}$$

Furthermore, when the long term unvoiced average $N_{uv_{lt}}$ is greater than 100 and the deviation σ_E is greater than 5 and the difference between the current frame energy and the last frame energy Δ_E^t is smaller than 12 dB, the long term average is modified as follow:

$$N_{uv_{lt}} = 0.2 \cdot N_{uv_{lt}} + 80 \tag{55}$$

This parameter on long term average of the number of frames between frames classified as unvoiced is used by the classifier to determine if the frame should be considered as generic audio or not. The more the unvoiced frames are close in time, the more likely the frame has speech characteristics (less probably generic audio). In the illustrative example, the threshold to decide if a frame is considered as generic audio G_A is defined as follows:

A frame is declared G_A if

$$N_{uv_{lt}} > 140 \;\&\; \Delta_E^t < 12 \tag{56}$$

The parameter Δ_E^t , defined at the beginning of this subclause, is added to not classify large energy variation as generic audio, but to keep it as active content. A flag named local attack flag L_{af} and used in subclause 6.8.1.3.5 of [6] is derived from variation of energy parameter Δ_E^t . The local attack flag L_{af} is set to 1 when the energy variation is greater than 6 dB and the frame is classified as GENERIC AUDIO SOUND or when the energy variation is greater than 9 dB.

The modification performed on the excitation depends on the classification of the frame and for some type of frames there is no modification at all. The next table 3 summarizes the case where a modification can be performed or not.

Table 6: Signal category for excitation modification

Frame Classification	Voice activity detected? Y/N	Category	Modification Y/N
ONSET VOICED_CLAS UNVOICED TRANSITION ARTIFICIAL ONSET	Y (VAD=1)	Active voice	N
GENERIC AUDIO SOUND	Y	Generic audio	Y
VOICED TRANSITION UNVOICED_CLAS	Y	Active unvoiced	Y
ONSET VOICED_CLAS UNVOICED TRANSITION ARTIFICIAL ONSET GENERIC AUDIO SOUND VOICED TRANSITION UNVOICED_CLAS	N	Inactive content	Y

The output of the second stage classifier will be used to activate or not different post processing based on content category.

5.3.1.8 Reconstructed excitation

The total excitation from layer 1 in each subframe is constructed by

$$u'(n) = \hat{g}_p v(n) + \hat{g}_c c(n), \qquad for\ 0 \le n < 63 \tag{57}$$

where $c(n)$ is the pre-filtered algebraic code vector. The excitation signal, $u'(n)$, is used to update the contents of the adaptive codebook for the next frame. The excitation signal, $u'(n)$, is then post-processed as described in subclause 6.1.1.3 of [5] to obtain the post-processed excitation signal $u(n)$, which is finally used as an input to the synthesis filter $1/\hat{A}(z)$. The final steps of synthesis, post-processing, de-emphasis and resampling are described in subclauses 6.1.4 of [5].

5.3.1.8.1 Particularity of rate 5.9, 7.2, 8.0 and 13.2 kbps

In case of GSC based concealment, with or without a time domain contribution, the excitation $u'(n)$ corresponds directly to the output of subclause 5.3.1.6.

5.3.2 Concealment for bandwidth extension for ACELP modes

5.3.2.1 Time domain bandwidth extension

5.3.2.1.1 SWB time domain bandwidth extension

The concealment for SWB TD BWE works for 13.2 kbps, 16.4 kbps, 24.4 kbps and 32 kbps. The algorithm aims to reconstruct the high band of the current lost frame for SWB TD BWE. The reconstruction of the lost frame depends on at least one of the following gain adjustment information: the coder type of the previous frame, the frame class of the

last good received frame, the frame class of the current frame, the number of the consecutive lost frame, the energies and the tilts of the low band of both the current frame and the previous frame.

There are gain shapes which are also the subframe gains, global frame gain and LSF should be reconstructed when the current frame is lost. The reconstruction of the LSF information is usually copying from the previous frame. The reconstruction of the subframe gains of the lost frame is based on the subframe gains and the subframe gain gradients of at least one frame before the current frame and adjusted by some of the above gain adjustment information. The reconstruction of the global frame gain of the lost frame is based on the global frame gain of at least one frame before the current frame and the global frame gain gradient of the current frame and adjusted by some of the above gain adjustment information.

The initial high band signal of the current lost frame is synthesized according to the decoding parameters of the frame prior to the current lost frame, specifically it is synthesized by passing the high band excitation through the synthesis filter, where the high band excitation is obtained from the low band excitation and synthesis filter is obtained from the reconstructed LSF parameters. Then the initial synthesized high band signal is adjusted by the reconstructed global frame gain and at least two of the reconstructed subframe gains of the current lost frame. Finally, the high band of the current lost frame is reconstructed.

5.3.2.1.1.1 The reconstruction of the global frame gain

For single frame loss: determining the frame class of the current frame, the tilts of the current frame and the previous frame, the energies of the low parts and high parts from the low band of both the current frame and the previous frame.

Assuming the three following conditions:

- Condition 1: the frame class of the current frame is not UNVOICED_CLAS and UNVOICED_TRANSITION.
- Condition 2: the tilt of the previous frame is less than 8.0.
- Condition 3: the energy of low parts from the low band of the current frame $EnLL$ is more than $0.5*EnLL_{prev}$ and $EnLL$ is less than $2.0*EnLL_{prev}$, or, the energy of high parts from the low band of the current frame $EnHH$ is more than $0.5*EnHH_{prev}$ and $EnHH$ is less than $2.0*EnHH_{prev}$. The $EnLL_{prev}$ is the energy of low parts from the low band of the current frame and the $EnHH_{prev}$ is the energy of high parts from the low band of the previous frame.

If all the above mentioned three conditions are met, the global frame gain of the current lost frame is described as follows:

$$gf = \begin{cases} 0.4*En_{ratio} + 0.6*gf, & En_{ratio} > 4.0*gf \\ 0.8*En_{ratio} + 0.2*gf, & 4.0*gf => En_{ratio} > 2.0*gf \\ 0.2*En_{ratio} + 0.8*gf, & otherwise \end{cases} \quad (58)$$

where the En_{ratio} is calculated by:

$$En_{ratio} = En_{prev} / En \quad (59)$$

$$En = \sqrt{\frac{\sum_{j=1}^{4}\sqrt{\sum_{n=0}^{79}(s\varepsilon(j*80+n)*0.0125)^2}}{4}} \quad (60)$$

where En is the high band excitation energy of the current frame, En_{prev} is the high band excitation energy of the previous frame.

Then if the tilt of the low band of the current frame $tilt$ is more than that of the previous frame $tilt_{prev}$, the global frame gain is updated as follows:

$$gf = \begin{cases} tilt_{ratio} * gf, & tilt_{prev} > 0 \\ gf, & otherhwise \end{cases} \quad (61)$$

$$tilt_{ratio} = \min(5.0, tilt_{prev} / tilt) \quad (62)$$

If the above mentioned three conditions are not met, but the following three conditions are met:

- Condition 4: the frame class of the current frame is not UNVOICED_CLAS or the tilt of the previous frame is more than 8.0,
- Condition 5: En_{ratio} is more than $4.0 * gf$,
- Condition 6: the energy of low parts from the low band of the current frame *EnLL* is more than $0.5 * EnLL_{prev}$, or the energy of high parts from the low band of the current frame *EnHH* is more than $0.5 * EnHH_{prev}$. The $EnLL_{prev}$ is the energy of low parts from the low band of the current frame and the $EnHH_{prev}$ is the energy of high parts from the low band of the previous frame.

The global frame gain would be:

$$gf = 0.2 * En_{ratio} + 0.8 * gf \quad (63)$$

For multiple frame losses:

For 13.2 kbps and 32 kbps, if En_{ratio} is more than $4.0 * gf$, *EnLL* is more than $EnLL_{prev}$ and *EnHH* is more than $EnHH_{prev}$. For 16.4 kbps and 24.4 kbps if En_{ratio} is more than $4.0 * gf$, the global frame gain is as follows:

$$gf = \begin{cases} \min((0.8 * En_{ratio} + 0.2 * gf), 4.0 * gf), & tilt > 10.0, tilt_{prev} > 10.0 \\ \min((0.5 * En_{ratio} + 0.5 * gf), 4.0 * gf), & otherwise \end{cases} \quad (64)$$

Otherwise, for 13.2 kbps and 32 kbps, if En_{ratio} is more than gf , *EnLL* is more than $EnLL_{prev}$ and *EnHH* is more than $EnHH_{prev}$. For 16.4 kbps and 24.4 kbps if En_{ratio} is more than gf , the global frame gain is as follows:

$$gf = \begin{cases} 0.5 * En_{ratio} + 0.5 * gf, & tilt > 10.0, tilt_{prev} > 10.0 \\ 0.2 * En_{ratio} + 0.8 * gf, & otherwise \end{cases} \quad (65)$$

5.3.2.1.1.2 The reconstruction of the gain attenuation factor

Reconstruct the gain attenuation factor according to the following conditions: the coder type of the previous frame, the frame class of the last good received frame, and the energies of the low band of both the current frame and the previous frame, the number of the consecutive lost frames. The detail processing is as follows:

For single frame loss, judging the following three conditions:

- Condition 1: the energy of the shaped excitation $s\varepsilon(n)$ of current frame $En_{s\varepsilon}$ is more than the energy of the shaped excitation $s\varepsilon(n)$ of the previous frame $En_{s\varepsilon\_prev}$.
- Condition 2: The coder type of the previous frame is not UNVOICED.
- Condition 3: The frame class of the last good received frame is not UNVOICED_CLAS.

If condition1, 2 and 3 are met:

The gain attenuation factor $factor_{s\varepsilon}$ to the shaped excitation $s\varepsilon(n)$ is as follows:

$$factor_{s\varepsilon} = \sqrt{\frac{En_{s\varepsilon}}{En_{s\varepsilon\_prev}}}$$
$$factor_{s\varepsilon\_temp=(factor_{s\varepsilon})^{0.125}} \quad (66)$$

Otherwise

- Condition 4: the energy of the shaped excitation $s\varepsilon(n)$ of current frame $En_{s\varepsilon}$ is more than 0.5 times the energy of the shaped excitation $s\varepsilon(n)$ of the previous frame $0.5 * En_{s\varepsilon\_prev}$.
- Condition 5: the energy of low parts from the low band of the current frame $EnLL$ is more than $0.5 * EnLL_{prev}$, or the energy of high parts from the low band of the current frame $EnHH$ is more than $0.5 * EnHH_{prev}$. The $EnLL_{prev}$ is the energy of low parts from the low band of the current frame and the $EnHH_{prev}$ is the energy of high parts from the low band of the previous frame.
- Condition 6: The coder type of the previous frame is not UNVOICED, or the type of the last good received is not UNVOICED_CLAS or the tilt of the previous frame is more than 5.0.

If condition 4, 5 and 6 are met, the gain attenuation factor $factor_{s\varepsilon}$ is calculated as follows:

$$factor_{s\varepsilon} = \sqrt{\frac{En_{s\varepsilon}}{En_{s\varepsilon\_prev}}}$$
$$factor_{s\varepsilon\_temp=(factor_{s\varepsilon})^{0.125}} \quad (67)$$

For multiple frame losses:

If the energy of the shaped excitation $s\varepsilon(n)$ of current frame $En_{s\varepsilon}$ is more than the energy of the shaped excitation $s\varepsilon(n)$ of the previous frame $En_{s\varepsilon\_prev}$, the gain attenuation factor $factor_{s\varepsilon}$ is as follows:

$$factor_{s\varepsilon} = \sqrt{\frac{En_{s\varepsilon}}{En_{s\varepsilon\_prev}}}$$
$$factor_{s\varepsilon\_temp} = (factor_{se})^{0.125} \quad (68)$$

Otherwise if condition4, 5 and 6 are met, the gain attenuation factor $factor_{s\varepsilon}$ is as follows:

$$factor_{s\varepsilon} = \min\left(2.0, \sqrt{\frac{En_{s\varepsilon}}{En_{s\varepsilon\_prev}}}\right)$$
$$factor_{s\varepsilon\_temp} = (factor_{se})^{0.125} \quad (69)$$

Use the $factor_{s\varepsilon}$ and $factor_{s\varepsilon\_temp}$ to the subframe gains and the shaped excitation $s\varepsilon(n)$ described as follows:

$$\begin{aligned} &gs(2*j) = gs(2*j) * factor_{s\varepsilon} \\ &gs(2*j+1) = gs(2*j+1) * factor_{s\varepsilon} \\ &s\varepsilon(i+j*40) = s\varepsilon(i+j*40) * factor_{s\varepsilon} \quad j = 0,...,7, i = 0,...,39 \\ &factor_{s\varepsilon} = factor_{s\varepsilon} / factor_{s\varepsilon\_temp} \end{aligned} \quad (70)$$

Use the reconstructed information including subframe gains, global frame gain and LSFs to reconstruct the high band signal of the lost frame.

5.3.2.1.1.3 Specifics for rates 13.2 and 32 kbps

Calculating the subframe gain gradients of the previous frame and the frame immediately prior to the previous frame are described as follows:

$$\begin{aligned} gs_{grad0}(j) &= gs'(j+1) - gs'(j), \ \ j = 0,1,2 \\ gs_{grad1}(j) &= gs''(j+1) - gs''(j), \ \ j = 0,1,2 \\ gs_{gradfec}(j+1) &= 0.4 * gs_{grad0}(j) + 0.6 * gs_{grad1}(j), \ j = 0,1,2 \end{aligned} \tag{71}$$

where $gs''(j), j = 1,2,3,4$ are the subframe gains of the previous frame, $gs'(j), j = 0,1,2,3$ are the subframe gains of the frame immediately prior to the previous frame.

$$gs_{gradfec}(0) \begin{cases} 0.1 * gs_{grad1}(1) + 0.9 * gs_{grad1}(2), if \quad gs_{grad1}(2) > 2.0 * gs_{grad1}(1) \ and \ gs_{grad1}(2) > 2.0 * gs_{grad1}(1) \\ \qquad\qquad\qquad\qquad\qquad\qquad or \quad gs_{grad1}(2) > 2.0 * gs_{grad1}(1) \ and \ gs_{grad1}(2) > 2.0 * gs_{grad1}(1) \\ 0.2 * gs_{grad1}(1) + 0.3 * gs_{grad1}(2) + 0.5 * gs_{grad1}(3), \ otherwise \end{cases} \tag{72}$$

If the coder type of the previous frame is UNVOICED, or the frame class of the last good received frame is UNVOICED_CLASS, and the $gs_{gradfec}(0)$ is positive. Then the first subframe gain template $gs_{template}(0)$ would be:

$$gs_{template}(0) = gs''(3) + gs_{gradfec}(0) \tag{73}$$

Otherwise, if $gs_{gradfec}(0)$ is positive, the subframe gain template $gs_{template}(0)$ would be:

$$gs_{template}(0) = gs''(3) + 0.5 * gs_{gradfec}(0) \tag{74}$$

Otherwise, the subframe gain template $gs_{template}(0)$ would be:

$$gs_{template}(0) = gs''(3) \tag{75}$$

The other gain subframe gain templates $gs_{template}(j), j = 1,2,3$ are determined as follows:

If $gs_{grad1}(2)$ is more than $10 * gs_{grad1}(1)$ and the $gs_{grad1}(1)$ is positive. Then:

$$gs_{template}(j) = gs_{template}(j-1) + 0.8 * gs_{gradfec}(j), \quad j = 1,2,3 \tag{76}$$

Otherwise, if $gs_{grad1}(2)$ is more than $10 * gs_{grad1}(1)$ and $gs_{grad1}(1)$ is negative. Then:

$$gs_{template}(j) = gs_{template}(j-1) + 0.2 * gs_{gradfec}(j), \quad j = 1,2,3 \tag{77}$$

Otherwise

$$gs_{template}(j) = gs_{template}(j-1) + gs_{gradfec}(j), \quad j = 1,2,3 \tag{78}$$

Reconstruct the $gs(j), j = 1,...,15$ use the $gs_{template}(j), j = 1,2,3,4$ according to the coder type of the previous frame, the frame class of the last good received frame and the number of consecutive lost frame. The global frame gain gradient g_{attn} is also determined by the upper three conditions, it is initialized to 1.0:

If the coder type of the previous frame is UNVOICED or the frame class of the last good received frame is UNVOICED_CLAS, and there is single frame loss, then:

$$gs(i*4+j) = gs_{template}(i)*1.2, \quad i = 0,1,2,3; j = 0,1,2,3 \tag{79}$$

$$\hat{g}_{attn} = g_{attn} * 0.95 \tag{80}$$

Otherwise if the coder type of the previous frame is UNVOICED or the frame class of the last good received frame is UNVOICED_CLAS Then:

$$gs(i*4+j) = gs_{template}(i), \quad i = 0,1,2,3; j = 0,1,2,3 \tag{81}$$

$$\hat{g}_{attn} = g_{attn} * 0.95 \tag{82}$$

Otherwise, if there are multiple frame losses, then:

$$gs(i*4+j) = gs_{template}(i)*0.5, \quad i = 0,1,2,3; j = 0,1,2,3 \tag{83}$$

$$\hat{g}_{attn} = g_{attn} * 0.5 \tag{84}$$

Otherwise then:

$$gs(i*4+j) = gs_{template}(i), \quad i = 0,1,2,3; j = 0,1,2,3 \tag{85}$$

$$\hat{g}_{attn} = g_{attn} * 0.85 \tag{86}$$

where $gs(i*4+j)\ i = 0,1,2,3;\ j = 0,1,2,3$ are the subframe gains of the current frame.

The global frame gain of the current frame is calculated with $\hat{g}_{attn}$ and gf_{prev} described as follows:

$$gf = \hat{g}_{attn} * gf_{prev} \tag{87}$$

where the gf_{prev} is the global frame gain of the previous frame.

5.3.3 Guided concealment and recovery

5.3.3.1 Specifics for rate 24.4 kbps

As described in subclause 5.5.4 of [5], the activation flag and differential pitch lag are transmitted as side information to obtain better pitch lag estimates and excitation signal for the future frame to be concealed.

The first 1 bit of the side information is read from the bit-stream yielding the activation flag. In case the activation flag equals 0, no further decoding is performed. If the flag equals 1, additional 4 bits are decoded yielding the differential pitch lag. With 4bits 16 different states are signalled. 15 states are used to represent the differential pitch lag, ranging from -7 to 7. The remaining signalling state is used to signal, that the pitch lag difference was outside the +-7 range on encoder side.
In case the pitch lag difference is inside the signalled valid rage of +-7, the differential pitch lag is added to the pitch lag of the last sub-frame. The result is used as an initial pitch lag estimate of the future 1st and 2nd sub-frame. The initial pitch lag estimates are used as an input to the pitch lag extrapolation procedure described in subclause 5.3.1.1. If the initial pitch lag estimates are available, the history of pitch lags used for the pitch extrapolation is updated with the initial pitch lag estimates. In case the criteria in clause 5.3.1.1 is not met, instead of $pitch_{old}$, the initial pitch lag estimate is used for building the first and second subframe of the adaptive codebook during concealment .

In case the pitch lag difference indicates that the difference is outside the valid range of +-7 the pitch extrapolation is performed like there is no future pitch lag information available in the bitstream.

5.3.3.2 Specifics for rates 9.6, 16.4 and 24.4 kbps

As described in subclause 5.5.5 of [5], side information on the activation of spectral envelope diffuser is transmitted to suppress too sharp peak at LP spectrum at the decoder side, 1 bit is decoded to obtain the activation flag. In case the

value equals to 1, spectral envelope diffuser is activated, otherwise de-activated. When spectral envelope diffuser is active, the following procedure is performed at the recovery frame.

A modified LSF parameter $\tilde{\omega}_j^{(-1)}$ for the previous frame is calculated by placing the low-order coefficients of the LSF parameter of the concealed frame $\dot{\omega}_j^{(-1)}$ at equal space.

$$\tilde{\omega}_j^{(-1)} = \begin{cases} j \cdot \delta & (1 \le j < idx) \\ \dot{\omega}_j^{(-1)} & (idx \le j < 16) \end{cases} \tag{88}$$

Then the LSF parameter for the current frame is replaced by the sum of mean vector of the current coder type and the residual LSF vector obtained in the decoding of the current frame, and bandwidth separation is applied.

This bandwidth separation is applied to ensure stability and suppress too sharp peak in LP spectrum. The distances are wider than the distance used in the normal LSF decoding process. In case the internal sampling frequency is 12.8 kHz, the distances are as follows:

$$\min\_dist = \begin{cases} 150 & (0 \le \omega_j < 1000) \\ 100 & (1000 \le \omega_j < 1900) \\ 50 & (1900 \le \omega_j < 6400) \end{cases} \tag{89}$$

Then, LP coefficients are calculated based on those modified LSF parameters, and used instead of LP coefficients obtained in the ordinary decoding process. The procedure for conversion from LSF to LPC is the same as normal decoding process.

5.3.3.3 Energy control during recovery

Precise control of the speech energy is very important in frame erasure concealment. The importance of the energy control becomes more evident when a normal operation is resumed after an erased block of frames. Since VC and GC modes are heavily dependent on prediction, the actual energy cannot be properly estimated at the decoder. In voiced speech segments, the incorrect energy can persist for several consecutive frames, which can be very annoying, especially when this incorrect-valued energy increases.

The goal of the energy control is to minimize energy discontinuities by scaling the synthesized signal to render the energy of the signal at the beginning of the recovery frame (a first non-erased frame received following frame erasure) to be similar to the energy of the synthesized signal at the end of the last frame erased during the frame erasure. The energy of the synthesized signal in the received first non-erased frame is further made converging to the energy corresponding to the received energy parameter toward the end of that frame while limiting an increase in energy.

If the available bitrate is sufficiently high, the synthesized speech energy information can be estimated in the encoder and transmitted as a side information to the decoder. In EVS, the energy information is transmitted only at 32 and 64 kb/s, using 5 bits. Further, it is transmitted only in the GC mode. In the TC mode, the energy control is not needed as the TC mode does not make use of the adaptive codebook, and memory-less LSF quantization is used. At lower bitrates, the correct energy is estimated at the decoder.

The energy control for LP-based decoding is triggered in the first non-erased frame following frame erasure for other than TC modes. At frames coded at 7.2 and 8 kb/s, in case that this first non-erased frame is using the Autoregressive (AR) prediction for the LP filter quantization, the energy control is continued in all subsequent frames using the AR prediction. The energy control is then maintained in yet another frame as the synthesis filter can still be affected by the filter coefficients interpolation.

The energy control is done in the synthesized speech signal domain. Even if the energy is controlled in the speech domain, it is the LP excitation signal that is scaled. The synthesis is then repeated to smooth the transitions.

The energy control in a recovery frame is done as follows. Let g_0 denote the gain used to scale the 1st sample in the current frame and g_1 the gain used at the end of the frame. The excitation signal is then scaled as follows

$$u_s(n) = g_{AGC}(n)u(n) \qquad n = 0, \ldots, L-1 \tag{90}$$

where $u_s(n)$ is the scaled excitation, $u(n)$ is the excitation before the scaling, L is the frame length and $g_{AGC}(n)$ is the gain starting from g_0 and converging exponentially to g_1. That is

$$g_{AGC}(n) = f_{AGC}\, g_{AGC}(n-1) + (1 - f_{AGC}) g_1 \qquad n = 0, \ldots, L-1 \tag{91}$$

with the initialization $g_{AGC}(-1) = g_0$. The factor f_{AGC} is the attenuation factor set to the value of 0.98. This value has been found experimentally as a compromise of having a smooth transition from the previous (erased) frame on one side, and scaling the last pitch period of the current frame as much as possible to the correct (transmitted) value on the other side. The gains g_0 and g_1 are defined as

$$g_0 = \sqrt{\frac{E_{-1}}{E_0}} \tag{92}$$

$$g_1 = \sqrt{\frac{E_q}{E_1}} \tag{93}$$

where E_{-1} is the energy computed at the end of the previous (erased) frame, E_0 is the energy at the beginning of the current (recovered) frame, E_1 is the energy at the end of the current frame, and E_q is the target energy at the end of the current frame. At higher bitrates, E_q is computed at the encoder, quantized and transmitted. The energy information is quantized using a 5-bit linear quantizer in the range of 0 dB to 96 dB with a step of 3 dB. The quantization index is given by

$$I_E = \left\lfloor \frac{10 \log\left(E_q + 0.001\right)}{3} \right\rfloor \tag{94}$$

The index is limited to the range [0, 31]. At lower bitrates, E_q is estimated at the decoder.

The energy E_1 of the synthesized speech $\hat{s}_{pre}(n)$ at the end of the first non erased frame is first computed as follows. The energy is the maximum of the signal energy for frames classified as VOICED_CLAS or ONSET, or the average energy per sample for all other frames. For VOICED_CLAS or ONSET frames, the maximum signal energy is computed pitch-synchronously at the end of the current frame as follows:

$$E_1 = \max\left(\hat{s}^2_{pre}(n)\right), \qquad n = L - T^{end}, \ldots, L-1 \tag{95}$$

where *L* is the frame length at internal sampling rate. Signal $\hat{s}_{pre}(n)$ is the local synthesis signal sampled at the internal sampling rate. The integer pitch period length T^{end} is the rounded pitch period of the last subframe, i.e. $T^{end} = \left\lfloor d_{fr}^{[last]} + 0.5 \right\rfloor$.

For all other classes, E_1 is the average energy per sample of the last half of the current frame, i.e.

$$E_1 = \frac{1}{L/2} \sum_{n=L/2}^{L-1} \hat{s}_1^2(n). \tag{96}$$

E_{-1} is computed similarly using the synthesized speech signal of the previous (last erased) frame. When E_{-1} is computed pitch synchronously (i.e. if the class of the previous frame was VOICED_CLAS or ONSET), it uses the concealment pitch period T_c.

When E_0 is computed pitch synchronously (the class of the current frame is VOICED_CLAS or ONSET), it is done similarly using the rounded pitch value $T^{[0]}$ of the first subframe:

$$E_0 = \max\left(\hat{s}^2_{pre}(n)\right), \qquad n = 0, \ldots, T^{end} \tag{97}$$

For other frame classes:

$$E_0 = \frac{1}{L/2} \sum_{n=0}^{(L/2)-1} \hat{s}_{pre}^2(n). \quad (98)$$

As mentioned previously, E_q is transmitted from the encoder, but only at high bitrates. If E_q is not available, it is initialized to E_1 and further limited as described below.

The gains g_0 and g_1 are further limited to a maximum allowed value to prevent too strong energy. This value has been set to 1.2 with the exception of very low energy frames (E_q < 1.1). In this case, g_1 is limited to 1. If E_q is not transmitted, further precautions shall be taken because of the possible mismatch between the excitation signal energy and the LP filter gain.

At 7.2 or 8 kb/s, this is done by upper-limiting the energy E_q by a value $E_{1\max}$, scaled by a factor α_{E_1}.

In the recovery frame or in the scaled frames following the recovery frame coded by the GC mode using AR prediction, $\alpha_{E_1} = 1.5$ and $E_{1\max} = E_{-1}$ if the following conditions are met: 1) the end-frame LP filter is resonant in low frequencies (measured by means of the filter tilt), and 2) the evolution of the transmitted pitch is stable within the frame or the mean of the pitch value over all subframes is lower than 34 samples. In the remaining scaled frames following the recovery frame, the scaling factor $\alpha_{E_1} = 2$ and $E_{1\max}$ equals to the larger value between E_{-1} and an average energy of recent voiced frames $\overline{E}_1$. If E_1 is computed pitch synchronously, $\overline{E}_1$ is the running average of pitch-synchronous energy of previous frames. If E_1 is computed as average energy per sample, $\overline{E}_1$ is the running average of average energy per sample of of previous frames. For other bitrates, when the erasure occurs during a voiced speech segment (i.e. the last good frame before the erasure and the first good frame after the erasure are classified as VOICED TRANSITION, VOICED_CLAS or ONSET) and the LP filter impulse response energy of the first frame after an erasure is twice as high as the LP filter impulse response energy of the last frame before the erasure, the energy of the excitation is adjusted to the gain of the new LP filter as follows:

$$E_q = E_1 \frac{2E_{LP0}}{E_{LP1}}. \quad (99)$$

Here E_{LP0} is the energy of the LP filter impulse response of the last good frame before the erasure and E_{LP1} is the energy of the LP filter of the first good frame after the erasure. The LP filters of the last subframes are used. Further, if $E_q > E_{-1}$ (E_q already initialized to E_1), it is further limited as follows:

$$E_q = 0.7E_{-1} + 0.3E_q \quad (100)$$

At 9.6 and 13.2 kb/s, there is however one exception to this energy scaling strategy if the LP filter is found resonant in low frequencies and the frame is classified as UNVOICED_CLAS or INACTIVE_CLAS. This situation indicates a possible error in the classification and the energy is scaled as in the case of the 7.2 or 8 kb/s recovery frame.

The following exceptions, all related to transitions in speech signal of good frames following an erasure, further overwrite the computation of g_0. If artificial onset is used in the current frame, g_0 is set to $0.5g_1$, to make the onset energy increase gradually. In the case of a first good frame after an erasure is classified as ONSET, the gain g_0 is prevented from being higher than g_1. This precaution is taken to prevent a positive gain adjustment at the beginning of the frame from amplifying the voiced onset at the end of the frame. Finally, during a transition from voiced to unvoiced (i.e. the last good frame being classified as VOICED TRANSITION, VOICED_CLAS or ONSET and the current frame being classified UNVOICED_CLAS) or during a transition from a non-active speech period to an active speech period, the value of g_0 is set to g_1.

Additionally, the synthesis energy control is performed also in the erased frames following frames coded at 7.2 or 8 kb/s or using the AMR-WB IO mode. Here the energy control is simpler in the sense that it is just verified that the gain is not increasing. Energies E_{-1}, E_0 and E_1 are computed similarly as in the recovery frames, but E_{-1} is used instead of E_q, and the gains g_0 and g_1 are limited to 1.

After the energy control, the speech signal is resynthesized by filtering the scaled excitation signal through the LP synthesis filter. The running energy average $\overline{E}_1$ is finally updated in good voiced frames as $0.05E_q + 0.95\overline{E}_1$ with initializion to E_q.

5.3.3.4 Specifics for rates 32 and 64 kbps

5.3.3.4.1 Adaptive codebook resynchronization and fast recovery (WB)

Fast recovery is an approach where side information with some bit rate overhead is transmitted to arrest error propagation into future frames, thereby improving performance under frame erasures. Side information includes parameters like energy, frame classification information and phase information. Specifically, the phase side information is used to align the glottal pulse position at the decoder to that of the encoder thereby synchronizing the adaptive codebook content. The information on the lost frame which becomes available on receiving the future frame is used to correct the excitation (pitch) memory before synthesizing the correctly received future frame. This helps to significantly contain the error propagation into future frames and improves decoder convergence when good frames are received after the erased frame. The waveform interpolation technique ([5], clause 5.2.3) is used to avoid abrupt changes in the pitch contour between the error concealed lost frame and the memory corrected future frame.

5.3.3.4.1.1 Decoding glottal pulse position

The glottal pulse position information consists of the position in the past, τ, of the absolute maximum pulse from the beginning of the current frame and its sign. If the first decoded pitch of the current frame is smaller then 128, the received quantized position τ is used as is, else the received quantized position τ is multiplied by 2.

5.3.3.4.1.2 Performing glottal pulse resynchronization

The goal of the resynchronization is to correct the difference between the target transmitted position of the last glottal pulse in the adaptive codebook of the current frame, and its actual position in the concealed adaptive codebook excitation signal. The position T(0) of the maximum pulse in the concealed adaptive excitation, *u(n)*, from the beginning of the frame is determined as described in the previous subclause. If the decoded maximum pulse position is positive, then the maximum positive pulse in the concealed adaptive codebook excitation from the beginning of the frame is determined. If the decoded maximum pulse position is negative, the maximum negative pulse is determined.

The target position of the absolute maximum pulse with respect to the beginning of the current frame is given by:

$$P_{last} = 256 - \tau \tag{101}$$

where τ has been defined as a decoded pulse or estimated as done in subclause 7.11.2.5.2. The error in the pulse position of the last concealed pulse in the frame is found by searching for the pulse $T(i)$ closest to the actual pulse, P_{last}. The error is given by:

$$T_e = P_{last} - T(k) \tag{102}$$

where k is the index of the pulse closest to P_{last} and T_e the difference between the actual pulse and the closest one. If $T_e = 0$, then no resynchronization is required. If $T_e \geq 0$ then T_e samples need to be inserted. If $T_e \leq 0$, then T_e samples need to be removed. Further, the resynchronization is performed only if $T_e < 64$ and $T_e < N_p \times T_{diff}$, where T_{diff} is the absolute difference between T_c and the pitch lag of the first subframe in the future frame, or its extrapolated value if it is not available.

The samples that need to be added or deleted are distributed across the pitch cycles in the frame. The minimum energy regions in the different pitch cycles are determined and the sample deletion or insertion is performed in those regions. The number of pitch pulses in the frame is N_p at positions $T(i)$, $i = 0, \cdots, N_p - 1$. The number of minimum energy regions is $N_p - 1$. If Tc $\leq$ 128, there shall be at least 2 minimum energy regions in the current frame. The minimum energy regions are determined by computing the energy using a sliding 5-sample window. The minimum energy position is set at the middle of the window at which the energy is at a minimum. The search performed between two pitch pulses at position $T(i)$ and $T(i+1)$ is restricted between $T(i) + T_c/8$ and $T(i+1) - T_c/4$.

The sample deletion or insertion is performed around $T_{\min}(i)$, where $T_{\min}(i), i = 0, \cdots, N_{\min} - 1$ are the minimum positions described above and $N_{\min} = N_p - 1$ is the number of minimum energy regions. The samples to be added or deleted are distributed across the different pitch cycles as follows.

If $N_{\min} = 1$, then there is only one minimum energy region and all samples T_e are inserted or deleted at $T_{\min}(0)$.

For $N_{\min} > 1$, a simple algorithm is used to determine the number of samples to be added or removed at each pitch cycle whereby less samples are added/removed at the beginning and more towards the end of the frame. If the total number of pulses to be removed/added is T_e, and the number of minimum energy regions is $N_{\min}$, the number of samples to be removed/added per pitch cycle, $R(i), i = 0, \cdots, N_{\min} - 1$, is found using the following recursive relation:

$$R(i) = round\left(\frac{(i+1)^2}{2} f - \sum_{k=0}^{i-1} R(k) \right) \tag{103}$$

where $f = \frac{2 | T_e |}{N_{\min}^2}$.

Note that at each stage, if $R(i) < R(i-1)$ then the values of $R(i)$ and $R(i-1)$ are interchanged. The values $R(i)$ correspond to pitch cycles starting from the beginning of the frame. $R(0)$ corresponds to $T_{\min}(0)$, $R(1)$ corresponds to $T_{\min}(1)$, …, $R(N_{\min} - 1)$ corresponds to $T_{\min}(N_{\min} - 1)$. Since $R(i)$ are in increasing order, more samples are added/removed towards the cycles at the end of the frame.

Removing samples is straightforward. Adding samples is performed by copying the last $R(i)$ samples after dividing by 20 and inverting the sign. For example, if 5 samples need to be inserted at position $T_{\min}(0)$, the following is performed:

$$u(T_{\min}(0) + i) = -u(T_{\min}(0) + i - R(3)) / 20 \tag{104}$$

Using the above procedure, the last maximum pulse in the concealed adaptive codebook excitation is forced to be aligned with the actual maximum pulse position at the end of the adaptive codebook frame which is transmitted in the current frame.

5.3.3.4.2 Artificial onset reconstruction

If the frame is classified as ARTIFICIAL ONSET, it means that the lost frame probably contained a voice onset, and the transition mode has not taken care of it (e.g., several consecutive frames were erased, containing the voiced onset frame, but also the following frame usually coded with TC mode). If the position of the glottal pulse of the previous frame is in the bitstream of the first good frame (i.e. 32 and 64 kbps), the onset is reconstructed artificially inside the adaptive codebook.

The lost onset case is the most complicated situation related to the use of the long-term prediction in CELP decoding. The lost onset means that the voiced speech onset happened somewhere during the erased block. In this case, the last good received frame was unvoiced and thus no periodic excitation is found in the excitation buffer. The first good frame after the erased block is however voiced, the excitation buffer at the encoder is highly periodic and the adaptive excitation has been encoded using this periodic past excitation. As this periodic part of the excitation is completely missing at the decoder, it can several frames to recover from this loss. It is worth emphasizing that this problem does not occur in the EVS codec for single frame erasures, as the frames following frames with voiced onsets are generally coded with TC mode and, TC mode does not make use of inter-frame long-term prediction.

If an ONSET frame is lost (i.e. a VOICED_CLAS good frame arrives after an erasure), but the last good frame before the erasure was UNVOICED_CLAS, a special technique is used to artificially reconstruct the lost onset and to trigger the voiced synthesis. The position of the last glottal pulse in the concealed frame can be available from the first good frame after the erase (in case the phase information related to the previous frame is transmitted in the bitstream). The artificial onset reconstruction does not affect the concealment of the erased frame; it is only matter of recovery. However, the last pulse of the erased frame is artificially reconstructed based on the position and sign information available in the first good frame after the erasure. This information consists of the position, P_{last}, of the maximum pulse from the end of the frame and its sign. The last glottal pulse in the erased frame is thus constructed as a low-pass filtered pulse placed in the memory of the adaptive excitation buffer (previously initialized to zero), and centred at the

decoded position, P_{last}. If the pulse sign is positive, the low-pass filter used is a simple linear phase FIR filter with the impulse response $h_{low}\{-0.0125, 0.109, 0.7813, 0.109, -0.0125\}$. If the pulse sign is negative, the low-pass filter used is a linear phase FIR filter with the impulse response $-h_{low}$.

Placing the low-pass filtered glottal pulse at the proper position at the end of the concealed frame significantly improves the performance of the consecutive good frames and accelerates the decoder convergence to actual decoder states.

The energy of the periodic part of the artificial onset excitation is then scaled by the gain corresponding to the quantized and transmitted energy E_q, as described in subclause 5.3.3.3, for frame erasure concealment and divided by the gain of the LP synthesis filter.

$$g_{onset} = \sqrt{\frac{1.5 \cdot E_q}{g_{LP}}} \tag{105}$$

The LP synthesis filter gain being computed as:

$$g_{LP} = \sqrt{\sum_{n=0}^{63} h^2(n)} \tag{106}$$

where $h(n)$ is the LP synthesis filter impulse response. Finally, the artificial onset gain is reduced by multiplying the periodic part by 0.96.

The LP filter for the output speech synthesis is not interpolated in the case of an artificial onset construction. Instead, the received LP parameters are used for the synthesis of the whole frame.

5.3.4 Handling of multiple frame losses and muting

5.3.4.1 Specifics for rates 5.9, 6.8, 8.0, 13.2, 32 and 64 kbps

The principal of attenuation in case of packet lost has been introduced in subclause 5.3.1. However, there are exceptions to the general case. The following three exceptions based on the last good frame coding mode that take precedence over table below. All apply only up to 3 consecutive lost frames. First, if the last good received frame is coded with UC mode, α is set to 1. Second, if the last good received frame is coded with VC or is an ONSET, α is set to 1.0. Finally, A stability factor, θ, is computed based on a distance measure between the adjacent LP filters. Here, the factor, θ, is related to the LSF distance measure and it is bounded by $0 \leq \theta \leq 1$, with larger values of θ corresponding to more stable signals. This limits energy and spectral envelope fluctuations when an isolated frame erasure occurs inside a stable unvoiced segment. Note that the class ARTIFICIAL ONSET is set at the decoder if the frame follows an erased frame and artificial onset reconstruction is used as described in subclause 5.3.3.4.2 at bit rate 32 and 64 kbps.

The signal classification is implicit for VC, UC and TC frames. Further, more precise classification can be decoded from the bitstream depending of the bit rate.

Table 7: Values of the PLC attenuation factor α

Last good received frame	Number of successive erased frames	α
ARTIFICIAL ONSET		0.6
ONSET, VOICED_CLAS	≤ 3	1.0
	> 3	0.4
VOICED TRANSITION		0.4
UNVOICED TRANSITION		0.8
UNVOICED_CLAS	= 1	$0.2\theta + 0.8$
	= 2	0.6
	> 2	0.4
AUDIO || INACTVE	>1 if GSC had temporal contribution	0.8
	<=5	0.995
	>5	0.95

5.3.4.2 Specifics for rates 9.6, 16.4 and 24.4 kbps

5.3.4.2.1 Fading to background level

The innovative as well as the harmonic excitation fade to individual target levels by changing the codebook gains.

$$g^{[m]} = alpha \cdot g^{[m-1]} + (1 - alpha) \cdot g^{\text{target}} \tag{107}$$

where: $g^{[m]}$ is the gain of the current frame;

$g^{[m-1]}$ is the gain of the previous frame;

g^{target} is the target gain;

alpha is the fading factor, its derivation is outlined in subclause 5.3.4.2.3.

The fading is performed as follows:

$$sig_{\text{faded}}[i] = \left(g^{[m-1]} - \frac{i}{L_{\text{frame}}} \left(g^{[m-1]} - g^{[m]} \right) \right) \cdot sig[i], i = 0 \ldots L_{\text{frame}} - 1 \tag{107a}$$

Where $sig[i]$ is the input signal, e.g. the harmonic or the innovative excitation, and $sig_{\text{faded}}[i]$ is the faded output signal.

The harmonic excitation is faded towards zero: $g_p^{\text{target}} = 0$.

The innovative excitation is faded towards a target background noise level: $g_c^{\text{target}} = g^{\text{cng}}$. It is derived during the first lost frame based on the background noise spectrum derived by CNG during clean channel decoding (see clause 4.3 of [5]). Its derivation is performed as follows:

a) Derive target level in time domain based on background noise spectrum X^{cng} :

$$\hat{g}^{cng} = \sqrt{\frac{320}{L} \cdot \sum_{i=startBand}^{stopBin} X_i^{cng}} \tag{108}$$

b) Compensate gain of LPC synthesis / de-emphasis (see also subsection 5.2.5):

$$g^{cng} = \hat{g}^{cng} \cdot \frac{1}{\mathrm{n}_{\mathrm{subfr}}} \cdot \sum_{i=0}^{\mathrm{n}_{\mathrm{subfr}}-1} energ_{\mathrm{LPC}}^{[i]} \quad (109)$$

where $energ_{\mathrm{LPC}}^{[i]}$ is derived subframe-wise as stated in equation (26).

5.3.4.2.2 Fading to background spectral shape

Separate LPCs are applied for the innovative and the harmonic excitation as described in subclause 5.3.1. The innovative and the harmonic excitations are faded to individual target spectral shapes by altering the LPC coefficients. The fading from the last good LPC coefficients to the target LPC coefficients is performed in the LSF domain as follows:

$$f^{[m]} = alpha \cdot f^{[m-1]} + (1 - alpha) \cdot f^{\,\mathrm{target}} \quad (110)$$

where: $f^{[m]}$ are LPC coefficients in the LSF domain of the current frame;

$f^{[m-1]}$ are LPC coefficients in the LSF domain of the previous frame;

$f^{\,\mathrm{target}}$ are the target LPC coefficients;

alpha is the fading factor as described in subclause 5.3.4.2.3. In case of the innovative excitation, *alpha* will be minimal 0.8.

The target spectral shape of the harmonic excitation is the short term mean of the last three LPC coefficient sets. Its derivation is performed in the LSF domain as follows:

$$f_p^{\,\mathrm{target}} = \frac{f^{[m-3]} + f^{[m-2]} + f^{[m-1]}}{3} \quad (111)$$

The target spectral shape of the innovative excitation is derived during the first lost frame based on the background noise spectrum derived by CNG during clean channel decoding (see 4.3 of [5]). Its derivation is performed as follows:

a) Compute power spectrum on the background noise spectrum.

b) Apply an inverse Fourier transform with length 640 on the power spectrum to obtain the autocorrelation values $r(k)$ with $k = 0...16$.

c) Do a normalisation of $r(k)$ to obtain $\hat{r}(k)$, if $\hat{r}(0) < 100$ set $\hat{r}(0)$ to 100 and multiply $\hat{r}(0)$ by 1.0005.

d) Execute the Levinson-Durbin algorithm with the order 16 to obtain the LP parameters from $\hat{r}(k)$.

e) Finally, transform the LPC coefficients to the LSF domain to obtain $f_c^{\,\mathrm{target}}$

5.3.4.2.3 Fading speed

The damping factor *alpha* controls the fading speed of the innovative and the harmonic excitation and depends on a bunch of parameters. These are: the number of consecutive lost frames, the LSF stability factor θ, the coder type, the class of the last good frame, the pitch gain g_p and the current coding mode. With this set of parameters the damping factor is determined as follows:

- Firstly, if current coding mode is ACELP_CORE, then
  - in case the coder type is UNVOICED and the number of consecutive lost frames is maximally three, then *alpha* is set to 1
  - else if the last good frame was UNVOICED_CLAS and
    - if it is the first lost frame, then $alpha = 0.8 + 0.2 \cdot \theta$

- else if exactly two frames are lost, then $alpha = 0.6$
- otherwise, if three or more frames are lost, then $alpha = 0.4$

- else if the last good frame was UNVOICED_TRANSITION, then $alpha = 0.8$
- else if the last good frame was ONSET and number of lost frames are maximally three and the coder type is GENERIC, then $alpha = 0.8$
- else if the last good frame was either VOICED_CLAS or the last good frame was ONSET and the number of lost frames is maximally three, then $alpha = 1$
- otherwise, $alpha = 0.4$
- besides that, if the last good frame was not one out of the set of { UNVOICED_CLAS, UNVOICED_TRANSITION, VOICED_TRANSITION }, then
  - in case it is the first erased frame, then $alpha = alpha \cdot \sqrt{g_p}$, whereas $\sqrt{g_p}$ is limited from 0.85 to 0.98
  - else if the number of lost frames are exactly two, then $alpha = (0.6 + 0.35 \cdot \theta) \cdot g_p$
  - otherwise, if more than two frames are consecutive lost, then the pitch of gain is changed to the new gain $g'_p = g_p \cdot (0.7 + 0.2 \cdot \theta)$ and following the damping factor is calculated as $alpha = alpha \cdot g'_p$

- Otherwise, if current coding mode is not ACELP_CORE and
  - if it is the first lost frame, then $alpha = 0.7 + 0.3 \cdot \theta$
  - else if exactly two frames are lost, then $alpha = 0.45 + 0.4 \cdot \theta$
  - otherwise, if three or more frames are lost, then $alpha = 0.35 + 0.4 \cdot \theta$

5.4 Concealment operation related to MDCT modes

5.4.1 PLC method selection

There is a multitude of PLC methods for MDCT coding modes available. Best possible codec performance in error-prone situations with frame losses is obtained through selecting the most suitable method for a given bit rate, coded audio bandwidth, used MDCT mode and signal class.

The main selector is the MDCT mode of the previous frame, i.e. TCX MDCT or HQ MDCT. Second level criteria are described in the following subclauses.

5.4.2 TCX MDCT

5.4.2.1 PLC method selection

In case the last good frame prior to a loss was coded with the MDCT based TCX, a range of different specifically optimized PLC methods are available that are selected based on second level criteria described in this subclause. The PLC methods are:

- TCX time domain concealment
- MDCT frame repetition with sign scrambling
- tonal MDCT concealment using phase prediction
- non-tonal concealment with waveform adjustment

The criteria evaluated in this second level PLC method selection are

- Last MDCT mode: The MDCT mode of the last good frame *lastCore* is obtained by decoding the bitstream in every good frame.
- Number of consecutively lost frames: The number of consecutively lost frames is increased in case of a frame loss and is reset in a good received frame.
- Last unmodified LTP gain: If LTP information is updated in the last good frame, the variable *ltpLastUnmGain* contains the LTP gain, and otherwise it is zero.
- Tonal MDCT peak detection flag: The flag *tonalMdctActive* describes whether tonal MDCT concealment using phase prediction should be done. It is set to zero by default and remains zero if one of the following conditions is true:
  - the last core or the second last core is not mode TCX20
  - the last unmodified LTP gain is bigger than 0.4 and the last pitch is bigger than $L / 2$
  - the last pitch differs from the second last pitch
  - TNS was active in the last or second last frame

  Otherwise, *tonalMdctActive* is set to one if the output of the peak detection of tonal components (see subclause 5.4.2.4.2) matches one of the following criteria:
  - the number of found peaks is higher than 10; or
  - the number of found peaks is higher than 5 and the difference between the 3rd and 2nd last pitch is smaller than 0.5 or
  - at least one peak is found and the last good frame was either UNVOICED_TRANSITION or UNVOICED_CLAS and the difference between the 3rd and 2nd last pitch is smaller than 0.5 and the last unmodified LTP gain is $<= 0.4$.
- Flag enabling non-tonal concealment with waveform adjustment: The flag *enablePlcWaveadjust* is set to one if the bit rate is one out of the set of {48 kbps, 96 kbps, 128 kbps}.
- Intelligent gap filling:
  The intelligent gap filling flag *igf* describes whether intelligent gap filling is active (1) or not (0) (see subclause 5.4.2.6).
- TCX_Tonality flag array:
  array of tonality flags of the last ten received frames (see subclause 5.4.5.3a).

The decision logic of the different PLC methods is done with the criteria shown above. The selection of the PLC is performed only in the first lost frame after a good frame and pertained in subsequently lost frames.

TCX time domain concealment is selected if:

- *tonalMdctActive* flag is zero; and
- *lastCore* is TCX_CORE and *ltpLastUnmGain* > 0.4 and the last good frame was neither UNVOICED_TRANSITION nor UNVOICED_CLAS.

In all other cases, the three MDCT-based concealment methods are selected as described below.

MDCT frame repetition with sign scrambling is selected if:

- *tonalMdctActive* is one (in conjunction with tonal MDCT concealment using phase prediction); or
- *tonalMdctActive* is zero and non-tonal concealment with waveform adjustment is not active.

Tonal MDCT Concealment using phase prediction is selected if:

- *tonalMdctActive* is one

Non-tonal concealment with waveform adjustment is selected if:

- *enablePlcWaveadjust* is one, *tonalMdctActive* is zero and there is no transition having a larger frame size than a normal TCX20 frame; and
- the lost frame is considered to be a non-tonal frame, which requires that the TCX_Tonality flag array contains five or less ones or one out of the last three frames is not TCX20.

If a MDCT-based PLC mode is selected and *igf* is one, some missing information are added with the intelligent gap filling concealment.

5.4.2.2 TCX time domain concealment

The time domain PLC for TCX is called if the core of the last good frame was TCX and if in the PLC method selection as describe in subclause 5.4.2.1 the TCX time domain concealment was chosen. This concealment method has some similarity to the ACELP like concealment described in subclause 5.3.1. Due to the fact, that this method is operating in the excitation domain to shape the noise towards the vocal tract and preventing discontinuities, a local LPC analysis is applied to the synthesized time domain signal $synth(i)$ of the last frame. To improve the LPC analysis, first the $synth(i)$ signal is filtered with the pre-emphasis filter described in subclause 5.1.3 of [5] to obtain $synth_{pre}(i)$. After that, an LPC analysis is applied on $synth_{pre}(i)$ same as in subclause 5.1.5 of [5], but with the frame length L and the analysis window, which first three-quarter part is a hamming window and last quarter part is a cosine window. The residual signal $exc(i)$ is obtained by filtering $synth_{pre}(i)$ through the inverse filter same as in subclause 5.2.2.4.1.1 of [5]. The local LPC parameters and the excitation signal $exc(i)$ are stored for multiple frame loss.

5.4.2.2.1 Construction of the periodic part of the excitation

If the last good frame was neither UNVOICED_CLAS nor UNVOICED_TRANSITION in combination with coder type being GENERIC, a harmonic part and a random part of excitation have to be generated for a concealment of erased frames. Otherwise, only a random part has to be generated. The harmonic part of the excitation is constructed by repeating the last pitch period of the previous frame. If this is the case of the first erased frame after a good frame and the ISF stability factor is lower than one, the first pitch cycle is first low-pass filtered. The filter used is core sampling rate dependant and consists of an 11-tap linear phase FIR filter. The filter coefficients for core sampling rates lower or equal then 16 000 Hz are:

$$\{0.0053, 0.0000, -0.0440, 0.0000, 0.2637, 0.5500, 0.2637, 0.0000, -0.0440, 0.0000, 0.0053\}, \quad (112)$$

for core sampling rate equal to 25600 Hz

$$\{0.0056, 0.0000, -0.0464, 0.0000, 0.2783, 0.5250, 0.2783, 0.0000, -0.0464, 0.0000, 0.0056\} \quad (113)$$

and for higher core sampling rates

$$\{-0.0053, -0.0037, -0.0140, 0.0180, 0.2668, 0.4991, 0.2668, 0.0180, -0.0140, -0.0037, -0.0053\}. \quad (114)$$

The periodic part of the excitation is constructed as described in subclause 5.3.1 including the pitch extrapolation as described in subclause 5.3.1.1 and the glottal pulse resynchronization as described in subclause 5.3.1.3. The pitches used to get the pitch extrapolation are based on the LTP lag and gains coming from the last TCX frames.

These LTP lag and gain are sent in the bitstream as side information. The specific handling as described in subclause 5.3.1.5 is used for TCX time domain concealment at all bitrates, additionally with the specific low-pass filtering of the first pitch cycle as described above.

The gain of pitch, g_p, is calculated on $synth_{pre}(i)$ as follows:

$$g_p = \frac{\sum_{i=0}^{2L_{subfr}-1} (synth_{pre}(i-2L_{subfr}) \cdot synth_{pre}(i-2L_{subfr}-T_c))}{\sum_{i=0}^{2L_{subfr}-1} (synth_{pre}(i-2L_{subfr}-T_c))^2} \quad (115)$$

where $synth_{pre}(i)$ are samples of pre-emphased prior time data, L_{subfr} is the length of a subframe in samples and T_c is the rounded pitch period equal to the LTP lag of the last good frame. The gain of pitch is limited between zero and one to prevent unexpected increase of energy. The formed adaptive excitation is attenuated sample-by-sample throughout the frame starting with one and ending with the damping factor calculated same as in subclause 5.3.4.2.3. To get a proper overlap add in the case the next good frame is a valid TCX frame, half a frame is additional created the same as describe above.

The attenuation strategy of the periodic part of the excitation is the same as done in subclause 5.3.4.2.1.

5.4.2.2.2 Construction of the random part of the excitation

The innovative (non-periodic) part of the excitation is generated by a simple random generator with approximately uniform distribution. If the last good frame was VOICED_CLAS or ONSET, a pre-emphased filtering of the noise is done same as [19] subclause 5.1.3, but with the pre-emphasis factor of 0.2 for core sampling rates lower or equal then 16 kHz and 0.6 for all other rates. The filtering is applied to decrease the amount of noisy components in the lower frequencies speech region. Furthermore, to shift the noise more to higher frequencies, the noise gets filtered by a 10-order high pass FIR filter in case of the first erased frame after a good frame and if the last good frame was neither UNVOICED_CLAS nor UNVOICED_TRANSITION. The filter coefficients are

$$\{0.0000, -0.0205, -0.0651, -0.1256, -0.1792, 0.8028, -0.1792, -0.1256, -0.0651, -0.0205, 0.0000\} \quad (116)$$

for core sampling rates lower or equal than 16000 Hz,

$$\{-0.0300, -0.0521, -0.0837, -0.1142, -0.1362, 0.8557, -0.1362, -0.1142, -0.0837, -0.0521, -0.0300\} \quad (117)$$

for the core sampling rate of 25600 Hz and

$$\{-0.0517, -0.0587, -0.0820, -0.1024, -0.1164, 0.8786, -0.1164, -0.1024, -0.0820, -0.0587, -0.0517\} \quad (118)$$

for all other rates. For the second and further lost frames, the noise is composed via a linear interpolation between the fullband and a highpass-filtered version of it as

$$noise(i) = (1 - cummulative\_damping) \cdot noise(i) + cummulatie\_damping \cdot noise(i)_{hp} \quad (119)$$

where $noise(i)$ are noise samples generated as described in beginning of this subclause, $noise(i)_{hp}$ are $noise(i)$ filtered with the highpass filter above and $cummulative\_damping$ is a frame wise cumulative factor of the damping factors. This ensures that the noise fades to fullband noise with the fading speed dependently on the damping factor.

The innovation gain, g_s, which is used for adjusting the noise level, is calculated as

$$g_s = \sqrt{\frac{\sum_{i=0}^{2L_{subfr}-1} (exc(i-2L_{subfr}) - g_p \cdot exc(i-2L_{subfr}-T_c))^2}{2L_{subfr}}} \quad (120)$$

where g_p is calculated as in equation (115). However, if the last good frame was neither UNVOICED_CLAS nor UNVOICED_TRANSITION in combination with coder type being GENERIC, g_p is set to zero for calculating g_s and the pitch buffer get reset.

The attenuation strategy of the random part of the excitation is somewhat different from the attenuation of the periodic excitation. The reason is that the pitch excitation is converging to zero while the random excitation is fading towards the

background level g^{cng} described in 5.3.4.2.1. The background level is limited to $2 \cdot g_s$. The random part of the excitation is attenuated linearly throughout the frame on a sample-by-sample basis starting with g_s and going to the end of the frame gain which is

$$g^{end} = alpha \cdot g_s + (1 - alpha) \cdot g^{cng} \quad (121)$$

where g^{end} is the gain in the last sample of the noise signal and $alpha$ is the damping factor as calculated in subclause 5.3.4.2.3. Due to the fact that g_s is a relative component, the noise gets normalized. If the last good frame was UNVOICED_CLAS and the coder type is not UNVOICED, the innovative excitation is further attenuated by a factor of 0.8. Otherwise, if the last good frame was not UNVOICED_CLAS and not UNVOICED_TRANSITION, the excitation is further attenuated by $1.1 - 0.75 g_p$.

To get a proper overlap add in the case the next good frame is a valid TCX frame, half a frame is additional created the same as describe above.

5.4.2.2.3 Construction of the total excitation, synthesis and updates

Finally, the random part of the excitation is added to the adaptive excitation to form the total excitation signal. If the last good frame is UNVOICED_CLAS or last good frame is UNVOICED_TRANSITION and coder type is GENERIC, only the innovative excitation is used as mentioned above. The synthesized signal is obtained by filtering the total excitation signal through the LP synthesis filter (see [5] subclause 6.1.3) with the local calculated LPC parameters and post-processed with the de-emphases filter, which is the inverse of [5] subclause 5.1.3.

If LTP information is available in the last good frame and g_s is equal to zero then the *ltpLastUnmGain* is reset to zero. In the end the overlap and add buffers get updated same as in subclause 5.4.5.

5.4.2.3 MDCT frame repetition with sign scrambling

The excitation of the concealed frame (input to FDNS) $\hat{C}_{exc}^{[m]}(k)$ is derived by sign scrambling of the last received excitation spectrum $C_{exc\_lastGood}(k)$:

$$\hat{C}_{exc}^{[m]}(k) = \frac{randomVector(k)}{|randomVector(k)|} \cdot C_{exc\_lastGood}(k), \text{ for } k = [0,...,igfStartLine] \quad (122)$$

igfStartLine is the IGF cross over frequency. The *randomVector* is derived as

$$randomVector(k) = randomVector(k-1) \cdot 31821 + 13849, \text{ for } k = [1,...,igfStartLine] \quad (123)$$

For any lost frame following a received frame, the initial value is reset:

$$randomVector(0) = 1977 \quad (123a)$$

If the last 2 spectra are coded using TCX5, then the one with smaller energy is chosen.

The spectrum $\hat{C}^{[m]}(k)$ is faded towards noise as described in subclause 5.4.6.1.3.2.1.

5.4.2.4 Tonal MDCT concealment using phase prediction

5.4.2.4.1 Overview

The phase prediction described in subclause 5.4.2.4.3 is performed on the spectral coefficients belonging to tonal components found using the peak detection described in subclause 5.4.2. For the spectral coefficients not belonging to the tonal components, the sign scrambling is applied as described in subclause 5.4.2.3.

5.4.2.4.2 Peak detection of tonal components

Peak detection is performed if the current frame is lost but the previous frame has been received.

The peaks are first searched in the power spectrum of frame $m-1$, using predefined thresholds. Based on the location of the peaks in frame $m-1$, the thresholds for the search in the power spectrum of frame $m-1.5$ are adapted, whereas frame $m-1.5$ represents the second 10ms of frame $m-2$ and the first 10ms of frame $m-1$. Thus, peaks existing in both spectra ($m-1$ and $m-1.5$) are found. Their exact location is based on the power spectrum of frame $m-1.5$.

The power spectra $\tilde{P}^{[m-1.5]}(k)$ and $\tilde{P}^{[m-1]}(k)$ are obtained as follows:

$$\tilde{P}^{[y]}(k) = \left|S^{[y]}(k)\right|^2 + \left|C^{[y]}(k)\right|^2, y \in \{m-1.5, m-1\}, k = 0 \ldots nSamples-1 \tag{124}$$

where $S^{[y]}(k)$ represents the MDST coefficients and $C^{[y]}(k)$ represents the MDCT coefficients and *nSamples* being the number of spectral coefficients. A minimum significant value of a spectral line in the power spectrum is assured by this operation:

$$P^{[y]}(k) = \max\left(\frac{nSamples^2}{400}, \tilde{P}^{[y]}(k)\right) \tag{125}$$

$C^{[m-1.5]}(k)$ and $S^{[m-1.5]}(k)$ are derived from the time domain signal via MDCT/MDST. $C^{[m-1]}(k)$ is given and $S^{[m-1]}(k)$ is estimated:

$$\left|S^{[m-1]}(k)\right| = \left|C^{[m-1]}(k+1) - C^{[m-1]}(k-1)\right| \tag{126}$$

If the change of the pitch lag between the last and the second last frame is larger or equal than 0.25 or the pitch lag is smaller than 10ms (corresponding to $F_0 < 100Hz$), the index of the fundamental frequency is set to zero. Otherwise the index of the fundamental frequency is determined as:

$$F_0^{\text{orig}} = F_0 = \frac{2 \cdot FrameSize}{PitchLag}. \tag{128}$$

10 strongest peaks are found at the positions $n_i \cdot F_0, i \in \{1,...,10\}, n_i \in \{2,3,4...\}$. Distance between peaks are calculated as $d_i = n_{i+1} - n_i, i \in \{1,...,9\}$. The most common among differences d_i is d. If there are at least 3 d_i equal to 1 and if $d = 1$; or less than 5 d_i are equal to $d \neq 1$, then F_0 is not changed. If there are more than 5 d_i equal to $d \neq 1$, then F_0 is set to $d \cdot F_0$. Otherwise F_0 is set to 0.

An envelope of each power spectrum is calculated using a moving average filter:

$$Envelope^{[y]}(k) = \frac{7.59}{FL} \cdot \sum_{i=k-\lfloor FL/2 \rfloor}^{k+\lfloor FL/2 \rfloor} P^y(i), y \in \{m-1.5, m-1\}. \tag{129}$$

The filter length FL depends on the index of the fundamental frequency and is limited to the range [11,23], as shown in Table 1. If the fundamental frequency is not available or not reliable, the filter length *FL* is set to 15, otherwise:

$$FL = \max\left(11, \min\left(23{,}1 + 2 \cdot \left\lfloor \frac{F_0}{2} \right\rfloor\right)\right). \tag{130}$$

Table 8: Filter length depending on the fundamental frequency

F0	FL
0	15
<= 10	11
>= 22	23
else	$1+2\cdot\left\lfloor\frac{F_0}{2}\right\rfloor$

The smoothed power spectra are calculated as follows:

$$P_{\text{smoothed}}^{[y]}(k) = 0.75\cdot P^{[y]}(k-1) + P^{[y]}(k) + 0.75\cdot P^{[y]}(k+1), y\in\{m-1.5, m-1\} \quad (131)$$

5.4.2.4.2.1 Detection of the peak candidates

If the smoothed spectrum $P_{\text{smoothed}}^{[m-1]}$ is above the envelope $Envelope^{[m-1]}$ at bin k and the smoothed spectrum at bin k is bigger than at bins $k-1$ and $k+1$, k is treated as peak candidate $i_{\max}$ and the right and left foot of this peak candidate are searched for.

The right foot is defined as the spectral bin with index i_r, for which

$$P^{[m-1]}(i_r) < P^{[m-1]}(i_r+1) \quad (132)$$

and

$$P^{[m-1]}(x) \geq P^{[m-1]}(x+1), \quad for\ x=[k+1,...,i_r-1] \quad (133)$$

It is also allowed for an $x'\in[k+1,...,i_r-1]$ that $P^{[m-1]}(x') < P^{[m-1]}(x'+1)$ is true, but only if $3\cdot P^{[m-1]}(x') \geq P^{[m-1]}(x'+1)$ and if there is a $k<j<i_r$ for which:

$$2\cdot\frac{P^{[m-1]}(x'+1)}{P^{[m-1]}(x')} \leq \frac{P^{[m-1]}(x')}{P^{[m-1]}(j)} \quad (134)$$

and

$$\begin{aligned} &P^{[m-1]}(x) \geq 3P^{[m-1]}(x+1), \quad for\ x=[x'+1,..., j-1] \\ &P^{[m-1]}(j) < 3P^{[m-1]}(j+1) \end{aligned} \quad (135)$$

The left foot is defined in the same way as the right foot, but on the left side of the bin k.

The local maximum $i_{\max}$ is then found between the left and the right foot.

The thresholds for the peak search in $P^{[m-1.5]}$ are set at positions $k\in[i_{\max}-1, i_{\max}+1]$ as:

$$Threshold(k) = \begin{cases} 1.1 & \text{when } P_{\text{smoothed}}^{[m-1]}(k) > Envelope^{[m-1]}(k) \\ 1.5 & \text{otherwise} \end{cases} \quad (136)$$

If the change of the pitch between the last and the second last frame is smaller then 0.5, then for each

- $k=\lfloor n\cdot F_0\rfloor$, for each $n\in[1,N]$, N being the number of the harmonics of F_0, $frac = n\cdot F_0 - k$
- $k=\lfloor m\cdot F_0^{\text{orig}}\rfloor$, for each $m\in\left[1,\left\lfloor F_0 / F_0^{\text{orig}}+0.5\right\rfloor\right]$, $frac = m\cdot F_0^{\text{orig}} - k$

thresholds are updated as follows:

$$\begin{aligned} Threshold(k) &= thresh_{\text{base}} \\ Threshold(k-1) &= thresh_{\text{base}} + 2 \cdot frac \\ Threshold(k+1) &= thresh_{\text{base}} + 2 \cdot (1 - frac) \end{aligned} \tag{137}$$

with

$$thresh_{\text{base}} = \begin{cases} 0.70 & \text{when} \quad F_0^{\text{orig}} > 0 \text{ and } F_0 = 0 \\ 0.35 & \text{when} \quad F_0^{\text{orig}} > 0 \text{ and } F_0 > 0 \end{cases} \tag{138}$$

For all bins not belonging to peaks or harmonics the threshold is set as:

$$Threshold(k) = 16\,. \tag{140}$$

Note: The base threshold 7.59, as given in equation 129, corresponds to $8.8dB$. All other thresholds, represented by $Threshold(k)$, are given relative to this base threshold. Thus,

- 0.35 corresponds to $4.24dB$
- 0.7 corresponds to $7.25dB$
- 1.1 corresponds to $9.21dB$
- 1.5 correspnds to $10.56dB$
- 16 corresponds to $20.84dB$

5.4.2.4.2.2 Final detection of the tonal components

After setting the thresholds $Threshold(k)$ as described in subclause 5.4.2.4.2.1, peaks detected in frame $m-1$ are now searched for in the power spectrum of frame $m-1.5$.

If the following is fulfilled:

$$\begin{aligned} P_{\text{smoothed}}^{[m-1.5]}(k) &> Envelope^{[m-1.5]}(k) \cdot Threshold(k) \\ P_{\text{smoothed}}^{[\text{m-1.5}]}(k) &\geq \max\left(P_{\text{smoothed}}^{[\text{m-1.5}]}(k-1), P_{\text{smoothed}}^{[\text{m-1.5}]}(k+1)\right) \end{aligned} \tag{141}$$

the right and left foot of the peak is searched for in $P^{[m-1.5]}$ around k. The algorithm for the foot search is the same as the one in subclause 5.4.2.4.2.1.

The local maximum $i_{\max}$ is then found between the left and the right foot.

A tonal component is defined as the set of spectral bins $I_{Tone} = [i_{\max} - 3, ..., i_{\max} + 3]$. If two neighboring tonal components would overlap, their surroundings are symmetrically reduced such that each spectral bin belongs only to one tonal component. All tonal components then build the set I_{Tones}.

5.4.2.4.3 Phase prediction

For all found tonal components I_{Tones}, that include spectrum peaks and their surroundings, as described in subclause 5.4.2.4.2.2, the MDCT phase prediction is used. For all other spectrum coefficients sign scrambling described in subclause 5.4.2.3 is used.

The phases are derived for each bin of a tonal component as:

$$\varphi^{[m-1.5]}(k) = \arctan\left(\frac{S^{[m-1.5]}(k)}{C^{[m-1.5]}(k)}\right), \quad k \in I_{Tone}, I_{Tone} \in I_{Tones} \tag{142}$$

The fractional part Δl is given by:

$$\Delta l = \arctan\left(a \cdot \frac{b}{2}\right) \tag{143}$$

with *a* given in Table 2, depending on the neighboring bins around a spectral peak $l = i_{\max}$.

Table 9: Variable a from equation (143)

if	a
$P^{[m-2]}(l-1) > mr \cdot P^{[m-2]}(l+1)$	$\tan\left(\frac{0 \cdot \pi}{b}\right)$
$P^{[m-2]}(l+1) > mr \cdot P^{[m-2]}(l-1)$	$\tan\left(\frac{2 \cdot \pi}{b}\right)$
else	$\frac{\cos\left(\frac{\pi}{b}\right) - \left(\frac{P^{[m-2]}(l-1)}{P^{[m-2]}(l+1)}\right)^G \cdot \cos\left(\frac{3 \cdot \pi}{b}\right)}{\sin\left(\frac{\pi}{b}\right) + \left(\frac{P^{[m-2]}(l-1)}{P^{[m-2]}(l+1)}\right)^G \cdot \sin\left(\frac{3 \cdot \pi}{b}\right)}$

Where the bandwidth *b* is 7, the maximum ratio mr is 44.8 and the constant *G* is $\frac{1}{2 \cdot 1.36}$.
The phase shift, being the same for every spectrum bin in I_{Tone}, is derived as follows

$$\Delta\varphi = \pi \cdot l\%4 + \Delta l\,, \tag{144}$$

where $l = i_{\max}$ is the index of the bin closest to the peak and Δl is the fractional part (i.e. distance of the peak from $i_{\max}$ given as the fractional number of bins).

The current phase $\varphi^{[m]}(k)$ is estimated for each $k \in I_{Tone}$ using:

$$\varphi^{[m]}(k) = \varphi^{[m-1.5]}(k) + n_{\text{frmdist}} \cdot \Delta\varphi \tag{145}$$

where $n_{\text{frmdist}} = 1.5$ for the first concealed frame and n_{frmdist} is increased for 1 for every consecutive frame loss. The correspondingt MDCT bins are estimated as:

$$C^{[m]}(k) = \sqrt{P^{[m-1.5]}(k)} \cdot \cos\left(\varphi^{[m]}(k)\right). \tag{146}$$

5.4.2.5 Non-tonal concealment with waveform adjustment

5.4.2.5.1 Preliminary concealment in frequency domain

The MDCT coefficients of the current lost frame are computed by using the MDCT coefficients of the frame prior to the current lost frame as follows:

The MDCT coefficients at all frequency points of the frame prior to the current lost frame are multiplied by random signs to obtain the MDCT coefficients of all frequency points of the current lost frame. In other words, when the current lost frame is the p^{th} frame,

$$c^{p}(m) = sgn(m) * c^{p-1}(m),\ m = 0, \cdots, M-1 \tag{147}$$

wherein $c^{p}(m)$ is the MDCT coefficient at the frequency point m of the p^{th} frame, M is the total number of the frequency points, and $sgn(m)$ is the random sign at the frequency point m.

The obtained MDCT coefficients of the current lost frame are transformed by an IMDCT to produce the initially compensated signal of the current lost frame.

5.4.2.5.2 Waveform adjustment in time domain

Waveform adjustment is performed on the initially compensated signal of the current lost frame to obtain the compensated signal of the current lost frame. The detailed procedure of the waveform adjustment is described as follows:

When the first lost frame occurs, the pitch period of the current lost frame is estimated as follows:

The pitch search is performed over the time-domain signal of the frame prior to the current lost frame by using the autocorrelation method to obtain the value of the pitch period of the frame prior to the current lost frame. The obtained pitch period value is used as the pitch period value of the current lost frame and to compute the maximum of normalized autocorrelation of the current lost frame. Detailedly, $t \in [T_{\min}, T_{\max}], 0 < T_{\min} < T_{\max} < L$ is searched so that

$$a(t) = \frac{\sum_{i=0}^{L-t-1} s(i)s(i+t)}{L-t} \tag{148}$$

achieves the maximum value, then the resulting t is the value of the pitch period, denoted by T, wherein $T_{\max}$ and $T_{\min}$ are the upper and lower limits for the pitch searching, respectively, and L is the frame length, $s(i)$ with $0 \le i \le L-1$ is the time-domain signal (the signal before TCX long-time prediction and post-processing) over which the pitch search is performed. $T_{\max}$ and $T_{\min}$ are obtained as follows:

$$T_{\max} = \text{round}(0.75L) \tag{149}$$

$$T_{\min} = \text{round}(34L/256) \tag{150}$$

wherein $\text{round}()$ denotes the rounding operation. Define

$$f(t) = \frac{\sum_{i=0}^{L-t-1} s(i)s(i+t)}{(\sum_{i=0}^{L-t-1} s^2(i) \times \sum_{i=t}^{L-1} s^2(i))^{1/2}} \tag{151}$$

then $f(T)$ is the maximum of normalized autocorrelation. When the frame length L is not greater than 320, define:

$$T_H = [T/2] \tag{152}$$

wherein $[x]$ indicates taking the greatest integer value less than or equal to x. Comparing $f(T)$ with $f(T_H)$, the pitch period is reset as $T = T_H$ in case $f(T_H) > f(T)$.

When the frame length L is greater than 320, in the procedure of estimating pitch period the following processing is carried out before pitch searching over the time-domain signal of the frame prior to the current lost frame: the time-domain signal of the frame prior to the current lost frame is down-sampled towards a half sampling rate, and the down-sampled time-domain signal is used to replace the original time-domain signal of the frame prior to the current lost frame for the pitch estimate. Accordingly, the searching limits $T_{\max}$ and $T_{\min}$ herein are obtained specifically as follows:

$$T_{\max} = \text{round}(0.75L/2) \tag{153}$$

$$T_{\min} = \text{round}(17L/256) \tag{154}$$

The following procedure is used to determine whether the pitch period value of the current lost frame estimated by the above method is usable regarding subsequent waveform adjustment:

i. Verify the following conditions to find if any one of them is met. If so, the obtained pitch period value is unusable.

(1) The cross-zero rate of the initially compensated signal of the first lost frame, denoted by $zp\_current$, is greater than a threshold $z1$, wherein $z1 = 70$ for $L \le 256$, and $z1 = 105$ in other cases.

(2) In the frame prior to the current lost frame, the ratio of lower-frequency energy to whole-frame energy, denoted by low_freq_rate, is smaller than a threshold of 0.02. This ratio is defined as

$$low\_freq\_rate = \frac{e_{low}}{e} = \frac{\sum_{m=0}^{low-1} c^2(m)}{\sum_{m=0}^{M-1} c^2(m)} \tag{155}$$

wherein $low = 30$ when the current lost frame is TCX20, $low = 15$ when the current lost frame is TCX10, M is the total number of the frequency points.

(3) In the frame prior to the current lost frame, the spectral tilt, denoted by $tilt$, is smaller than a threshold $TILT$, wherein $TILT = 0.5$ for $L \le 320$ and $TILT = 0.7$ otherwise. This spectral tilt is defined as

$$tilt = \frac{\sum_{i=L/4}^{L-1} s_{LP}(i) s_{LP}(i-2)}{\sum_{i=L/4}^{L-1} s_{LP}^2(i)} \tag{156}$$

wherein $s_{LP}(i), i = 0, \cdots, L-1$ is a low-pass filtered signal of the time-domain signal of the prior frame. The low-pass filter is given by:

$$H_{\mathrm{LP}}(z) = \frac{0.18}{1 - 0.64z^{-1} - 0.18z^{-2}} \tag{157}$$

(4) In the frame prior to the current lost frame, the cross-zero rate of the second half frame $zp2$ is greater than that of the first half frame $zp1$ by four times.

ii. If none of the above-mentioned conditions (i.e. the conditions (1)-(4)) is met, verify whether the obtained pitch period value is usable according to the following criteria:

(a) When the current lost frame is within a silence segment, the obtained pitch period value is considered to be unusable. The silence segment is identified if the logarithm energy of the frame prior to the current lost frame is smaller than a threshold of 50 or the following two conditions are met simultaneously:

(1) The maximum of normalized autocorrelation mentioned above in the pitch estimate procedure is smaller than 0.9.

(2) The result of the current long-time logarithm energy minus the logarithm energy of the frame prior to the current lost frame is greater than 8.0.

The logarithm energy is defined as:

$$ener = 10\log_{10}\left(\frac{1}{L}\sum_{i=0}^{L-1} o^2(i)\right) \tag{158}$$

where $o(i), i = 0, \cdots, L-1$ is the time-domain signal used as the final decoder output.

The long-time logarithm energy is defined as follows:

Set an initial value $e0 \geq 0$. For each frame, if its logarithm energy is greater than 50 and its cross-zero rate is smaller than 100, the long-time logarithm energy is updated as below:

$$ener_{\text{mean}} = a * ener_{\text{mean}} + (1-a) * ener \tag{159}$$

where $e0 = 59.426$ and $a = 0.98$.

(b) When the current lost frame is not within a silence segment and the maximum of normalized autocorrelation mentioned above is greater than 0.8, the obtained pitch period value is considered to be usable.

(c) When the criteria (a) and (b) are not met and the cross-zero rate of the frame prior to the current lost frame is greater than 100, the obtained pitch period value is considered to be unusable,

(d) When the criteria (a), (b), and (c) are not met and the result of the current long-time logarithm energy minus the logarithm energy of the frame prior to the current lost frame is greater than 6.0, the obtained pitch period value is considered to be unusable,

(e) When the criteria (a), (b), (c), and (d) are not met, and the result of the logarithm energy of the frame prior to the current lost frame minus the current long-time logarithm energy is greater than 1.0 and the maximum of normalized autocorrelation mentioned above is greater than 0.6, the obtained pitch period value is considered to be usable,

(f) When the criteria (a), (b), (c), (d), (e), and (f) are not met, the harmonic characteristic of the frame prior to the current lost frame is verified. When a value $harm$ representing the harmonic characteristic is smaller than a threshold H, the obtained pitch period value is considered to be unusable, When the value $harm$ is greater than or equal to the threshold H, the obtained pitch period value is considered to be usable, In this case, $H = 0.7$. $harm$ can be computed as follows:

$$harm = \frac{\sum_{i=1}^{l} c^2(h_i)}{\sum_{i=0}^{L-1} c^2(i)} \tag{160}$$

wherein h_1 is the fundamental frequency point, $h_i, i = 2, \cdots, l$ is the i^{th} harmonic frequency point of h_1, $c(h_i)$ is the MDCT coefficient at the frequency point h_i. Due to the quantitative relation between the pitch period and the pitch frequency, the value of $h_i, i = 1, \cdots, l$ can be computed with the pitch period value mentioned above. When h_i is not an integer, $harm$ is computed with its adjacent one or several frequency points by using rounding.

When the current lost frame is not the first lost frame, the pitch period of the first lost frame is taken as the estimated pitch period of the current lost frame,

If the pitch period of the current lost frame is not usable, the initially compensated signal of the current lost frame is taken as the compensated signal of the current lost frame; if the pitch period is usable, waveform adjustment is performed on the initially compensated signal with the time-domain signal of the frame prior to the current lost frame, that is, the pitch period is adjusted under certain conditions at first, and then the following are conducted:

It is supposed that the current lost frame is the x^{th} lost frame, wherein $x > 0$, and when x is larger than 4, the initially compensated signal of the current lost frame is taken as the compensated signal of the current lost frame, otherwise the following steps are performed;

(a) A buffer is established with a length of L;

(b) When x equals 1, the first $T/4$ samples of the buffer are configured as a first $T/4$-length signal of the initially compensated signal of the current lost frame, wherein T is the pitch period of the current lost frame;

(c) When x equals 1, the last pitch period of time-domain signal of the frame prior to the current lost frame and the first $T/4$-length signal in the buffer are concatenated, and repeatedly copied into the buffer, until the buffer is filled up to obtain a time-domain signal with a length of L, and during each copy, if the length of the existing signal in the buffer is l, the signal is copied to locations from $l - T/4$ to $l + T - 1$ of the buffer, wherein $l > 0$, and for the resultant overlapped area with a length of $T/4$, the signal of the overlapped area is obtained by adding signals of two

overlapping parts after windowing respectively; when x is larger than 1, the last pitch period of compensated signal of the frame prior to the current lost frame is repeatedly copied into the buffer without overlapping, until the buffer is filled up to obtain a time-domain signal with a length of L;

(d) When x is less than 4, the signal in the buffer is taken as the compensated signal of the current lost frame; when x equals 4, overlap-add is performed on the signal in the buffer and the initially compensated signal of the current lost frame, and the obtained signal is taken as the compensated signal of the current lost frame.

For each lost frame without overlap-add processing, an additional signal as a noise is added to the compensated signal of the frame after the compensated signal is obtained. The detailed method of adding additional signal is as follows: firstly, a past signal, namely, the time-domain signal of the frame prior to the first lost frame (in the case of the first lost frame) or the initially compensated signal of the prior lost frame (in the case of the second, third, or fourth lost frame) is passed through a high-pass filter given as follows to obtain an additional signal:

$$H_{\mathrm{HP}}(z) = 1 - 0.68z^{-1} \tag{160a}$$

secondly, additional-signal gain values of the lost frame are estimated as follows:

$$NoiseGain = 0.99\,NoiseGain + 0.01\,GainBob \tag{160b}$$

wherein $NoiseGain$ is updated sample by sample during a series of consecutively lost frames with an initial value of zero at the beginning of the first lost frame and

$$GainBob = 1 - \frac{1}{2} f(T) \tag{160c}$$

where $f(T)$ is the maximum of normalized autocorrelation as described by equation (151); then, the additional signal is multiplied with the estimated additional-signal gain values sample by sample, and the additional signal resulting from multiplication is added to the compensated signal, to obtain a new compensated signal. For each lost frame with overlap-add processing, overlap-add is performed after the additional signal is added to the signal in the buffer.

For the first correctly received frame after the frame loss, if the number of consecutively lost frames is less than 4, a buffer is established with a length of L, the last pitch period of compensated signal of the frame prior to the first correctly received frame is repeatedly copied into the buffer without overlapping until the buffer is filled up, overlap-add is performed on the signal in the buffer and the time-domain signal obtained by decoding the first correctly received frame, and the obtained signal is taken as a time-domain signal of the first correctly received frame. The additional signal described above is added to the signal in the buffer before overlap-add.

5.4.2.6 Intelligent gap filling

The intelligent gap filling tool is applied on the constructed signal, generated from one of the three MDCT-based TCX PLC methods, as described in [5], subclause 6.2.2.3.8. However, with increasing number of lost frames, the tiled IGF signal gets further attenuated by changing the IGF gain factor for each scale factor band.

In case of a lost frame, the IGF gain factors calculated in [5] subclause 6.2.2.3.8.3.8 firstly get limited to the maximum value of 12. After that, the gain factors get changes as follows:

$$g(k) = \begin{cases} g(k) \cdot (1 - nbLostCmpt/8) & \text{if} \quad nbLostCmpt < 5 \\ g(k)/2 & \text{else} \end{cases} \tag{161}$$

where $g(k)$ is the IGF gain factor at scale factor band k and $nbLostCmpt$ are the number of consecutively lost frames.

5.4.3 HQ MDCT

5.4.3.1 Preliminary signal analysis of past synthesis

The buffer containing the past decoded signal is analysed in a preliminary step to prepare the PLC selection method described in clause 5.4.3.2 and the MDCT concealment described in clause 5.4.3.6.

5.4.3.1.1 Resampling to 8 kHz

The last 2 frames of the previous synthesis signal are resampled to 8 kHz using zero-delay low-pass FIR filter with a cutoff frequency at 4 kHz. The FIR filter order is 20, 40, 60 for a sampling frequency of 16, 32, 48 kHz, respectively. The FIR filter coefficients are denoted $h_{4,16k}(i)$ at 16 kHz, $h_{4,32k}(i)$ at 32 kHz and $h_{4,48k}(i)$ at 48 kHz.

Low-pass filtering and downsampling steps are jointly performed with a polyphase approach; the resampled signal at 8kHz, $\hat{s}_8^{prev}(n), n = 0,...,319$, can be computed using the relationship based on the past synthesis $\hat{s}_{16}(n)$, $\hat{s}_{32}(n)$, $\hat{s}_{48}(n)$ at respectively 16, 32 and 48 kHz:

$$\hat{s}_8^{prev}(n) = \begin{cases} \sum_{i=0}^{20} \hat{s}_{16}(2n+10-i)h_{4,16k}(i) & f_{s,out} = 16 \text{ kHz} \\ \sum_{i=0}^{40} \hat{s}_{32}(4n+20-i)h_{4,32k}(i) & f_{s,out} = 32 \text{ kHz} \\ \sum_{i=0}^{60} \hat{s}_{48}(6n+30-i)h_{4,48k}(i) & f_{s,out} = 48 \text{ kHz} \end{cases} \quad (162)$$

Note that in the above summations, the past synthesis outside the last 2 frames is by convention considered to be zero. For instance, at 16 kHz, it is considered that $\hat{s}_{16}(n) = 0$ when $n < 0$ or $n \geq 640$.

5.4.3.1.2 Pitch search by cross-correlation

The past synthesis signal resampled to 8 kHz and of length 40 ms, $\hat{s}_8^{prev}(n), n = 0,...,319$, is used to perform an open-loop pitch search as follows:

- The target signal is defined as the last 6 ms segment from the 40 ms buffer at 8 kHz:
  $t(n) = \hat{s}_8^{prev}(n+271), n = 0,...,47$

- A search vector of the same length (6 ms), $\hat{s}_8^{prev}(n+j), n = 0,...,47$, with sliding starting point $0 \leq j < L_{search}$ is used. The search range L_{search} covers 33 ms when the voicing parameter indicates a voiced segment (i.e. v =1) and 28 ms otherwise; therefore the pitch search range is adapted depending on the voicing indicator v, to use a longer search range in case of voiced signals. The cross-correlation is computed for each index j as:

$$Corr(j) = \frac{\sum_{ni=0}^{47} \hat{s}_8^{prev}(n+j)t(n)}{\sqrt{\sum_{ni=0}^{47} \hat{s}_8^{prev}(n+j)^2 \sum_{ni=0}^{47} t(n)^2}} \quad (163)$$

  To minimize computational complexity the term $\sum_{ni=0}^{47} t(n)^2$ is pre-computed and the term $\sum_{ni=0}^{47} \hat{s}_8^{prev}(n+j)^2$ is updated incrementally by removing the first term and adding a new term in each iteration.

  For each index j, the maximum correlation c and maximum location $j_{\max}$ are updated as follows: If $Corr(j) > c$, $c = Corr(j)$ and $j_{\max} = j$, with the initial conditions $c = 0$ and $j_{\max} = 0$; this loop is stopped whenever v =0 and $Corr(j) > 0.95$.

The pitch is then defined as $T_c = 272 - j_{\max}$, which corresponds to the time offset with respect to the beginning of the target signal (i.e. 34 ms after the beginning of the past synthesis $\hat{s}_8^{prev}(n)$).

5.4.3.2 PLC method selection

In case the last good frame prior to a loss was coded with HQ MDCT a range of different specifically optimized PLC methods is available that are selected based on second level criteria described in this subclause.

The criteria evaluated in this second level PLC method selection are:

- Output sampling rate
  The output sampling rate $f_{s,out}$ in which response the second level PLC method is selected is one out of the set of {8000Hz, 16000Hz, 32000Hz, 48000Hz}.

- Bit rate
  The bit rate r in which response the second level PLC method is selected is one out of the set of the supported bit rates of the EVS default operation mode [5].

- Voicing
  The voicing parameter v in which response the second level PLC method is selected is a binary parameter.

- Correlation
  The correlation parameter c computed as in clause 5.4.3.1.2, in which response the second level PLC method is selected is a correlation coefficient defined in the number range from [0…1].

- Transient condition
  The transient condition $\mathbf{t} = [t_0, t_1]$ in which response the second level PLC method is selected is a vector of dimension 2 of binary parameters t_0, t_1 indicating a transient condition t_0 in the last good frame or in the frame before t_1. The determination of the transient condition for a given HQ MDCT frame is specified in [5], subclause 5.3.2.4.1.1.

- Spectral envelope stability based speech/music classification
  The Spectral envelope stability based speech/music classification S_{plc} in which response the second level PLC method is selected is a binary parameter. This parameter is a post-processed instance of the envelope stability parameter S that is specified in [5], subclause 6.2.3.2.1.3.2.3 (Noise level adjustment). The spectral envelope stability based speech/music classification is calculated during the decoding of the preceding good HQ MDCT frame and stored for use in the context of the PLC method selection during a bad frame.

  The post-processing of this parameter is a Markov smoother with:

- {speech, music} as hidden states,

- the normalized envelope stability parameter,

$$\bar{S} = 1 \Big/ (1 - 2 \cdot 0.003412) \cdot (S - 0.003412) \tag{164}$$

  and its reverse $1 - \bar{S}$
  as direct state observation likelihoods for music and, respectively, speech,

- and the transition probabilities
  $\mathbf{t}_s = \{0.999 \quad 0.5\}$ for going from speech or, respectively, music state to speech state, and
  $\mathbf{t}_m = \{0.001 \quad 0.5\}$ for going from speech or, respectively, music state to music state.

For each good HQ MDCT frame the following sequence of operations is executed:

1) Calculation of the normalized envelope stability parameter $\bar{S}$ and its reverse $1 - \bar{S}$.

2) Calculation of a priori likelihoods $\mathbf{p}_a$ for speech and music states based on the state likelihoods for the instant $\mathbf{p}'$ of the previous (good) frame and the transition probabilities:

$$\mathbf{p}_a = \begin{bmatrix} p_{a,s} \\ p_{a,m} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_s \\ \mathbf{t}_m \end{bmatrix} \cdot \mathbf{p}' \tag{165}$$

3) Element-wise multiplication of the vector of a priori likelihoods $\mathbf{p}_a$ with the vector of direct state observation likelihoods for music and, respectively, speech:

$$\hat{\mathbf{p}} = \begin{bmatrix} p_{a,s} \cdot (1 - \bar{S}) \\ p_{a,m} \cdot \bar{S} \end{bmatrix} \tag{166}$$

Subsequent normalization yield the vector of state likelihoods $\mathbf{p}$ of the current frame:

$$\mathbf{p} = \frac{\hat{\mathbf{p}}}{|\hat{\mathbf{p}}|} \tag{167}$$

4) Finally, the index of the largest element of the state likelihood vector $\mathbf{p}$ is identified and taken as speech/music classification result S_{plc} for the present frame.

$$S_{plc} = \max^{-1}(\mathbf{p}) \tag{168}$$

5) The state likelihood vector $\mathbf{p}$ of the current frame is stored for subsequent use in the next good HQ MDCT frame.

With the above-specified parameters the second level PLC method selection is performed as follows:

- Firstly, if output sampling rate $f_{s,out}$ equals 8000 Hz, the PLC method specified in clauses 5.4.3.3, 5.4.3.4 is applied.
- Otherwise (if output sampling rate $f_{s,out}$ is equal or exceeds 16000 Hz), then:
  - in case the bit rate r is less or equal to 48 kbps and
    - if the voicing parameter v is set or the correlation parameter c exceeds 0.85, then
      - the frame loss concealment method specified in subclause 5.4.3.6 is applied;
      - otherwise,
        - the frame loss concealment method specified in subclause 5.4.3.5 is applied.
      - otherwise (in case the bit rate r is larger than 48 kbps), then:
        - the frame loss concealment method specified in subclause 5.4.3.6 is applied under the same condition as above (for bit rates less or equal to 48 kbps) except for the case that the spectral envelope stability based speech/music classification S_{plc} indicates music, in which case this frame loss concealment method is only applied if the correlation parameter c is below 0.6 or if the voicing parameter v is set;
        - otherwise, if the above condition is not satisfied the frame loss concealment method specified in subclause 5.4.3.5 is applied.
- However, in addition to the conditions specified above, the frame loss concealment method specified in subclause 5.4.3.6 is only applied under the provision that the current frame is the first bad frame following a good frame and that the transient condition vector does not indicate a transient in the previous or it indicates a transient in the frame before the previous frame. If this provision is not satisfied, the frame loss concealment method specified in subclause 5.4.3.5 is applied.

The decoding of HQ MDCT for NB includes the following modules:

- a frequency domain packet loss concealment (PLC) block,
- a spectrum decoding block,
- a memory update block,

- an IMDCT block,
- and a time-domain PLC block.

If it is determined that there is an erased frame, the erased frame is concealed using a PLC method. The bad frame indicator (BFI) set to 1 indicates that a current frame is erased, or that no useful information exists for that frame. Similarly, the Prev_BFI flag set to 1 indicates that a previous frame has been erased.

Figure 2 shows the block diagram for packet loss concealment of NB signals for the MDCT mode. A frequency-domain approach operates on the frequency domain signal such as the input to the IMDCT block in the figure. A time-domain approach operates on the time domain signal after the IMDCT block. When a frame erasure occurs, the spectral coefficients of the current frame are estimated. To accomplish this using the frequency-domain approach, the synthesized spectral coefficients of the last good frame are repeated for the current frame with signal modification such as a gain scaling and a random sign changing. In the time-domain approach, an additional PLC operation is added to enhance the performance of the frequency-domain approach depending on the input signal characteristics. For this additional operation, the appropriate packet loss concealment tool, either the phase matching tool or the repetition and smoothing tool is selected.

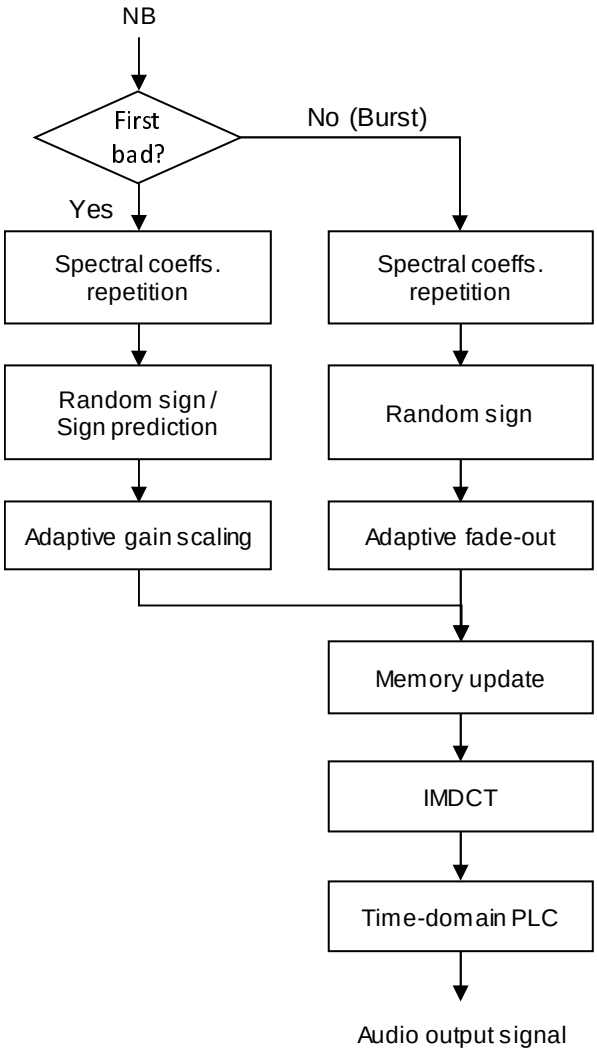

Figure 2: Block diagram for NB PLC for MDCT mode

5.4.3.3 MDCT frame repetition with random sign and gain scaling

When a first frame erasure occurs, packet loss concealment is performed as follows. In order to conceal the erasure, the signal characteristics of a decoded signal are used, which results in a classification of the characteristics of the decoded

signal into a stationary and normal frame. A current frame is determined to be transient using the frame type (is_transient) which is transmitted from the encoder. The energy difference (energy_diff) is used to determine if the current frame is stationary, and is represented by the following equation. The energy difference indicates the absolute value of a normalized energy difference between energy E_{curr} of the current frame and a moving average E_{MA} of per-frame energy. E_{MA} will be updated to E_{MA_old} in the next frame.

$$E_d = \left| \frac{(E_{curr} - E_{MA})}{E_{MA}} \right| \tag{169}$$

Where,

$$E_{MA} = 0.8 * E_{MA\_old} + 0.2 * E_{curr} \tag{170}$$

$$E_{curr} = \frac{\left(\sum_{k=0}^{n-1} norm(k) \right)}{n} \tag{171}$$

Depending on the frame type and characteristics, scaling and a random sign are used when the spectral coefficients are repeated for the current erased frame.

```
   if ( is_transient == 0 ) {
        if(energy_diff < ED_THRES) {
 /* Stationary frame */
           Repeating the spectral coefficients of the last good frame without scaling;
}
else{
           /* Non-stationary frame */
           Repeating the spectral coefficients of the last good frame with 3dB scale-down;
}
   else {
  if( st->old_is_transient[1] == 1 )     {
    Repeating the spectral coefficients of the last good frame with 3dB scale-down;
         }
         else  {
    Repeating the spectral coefficients of the last good frame with 3dB scale-down;
    Use random sign from the 2nd band (8th spectral coefficient)
          }
    }
```

When multiple erasures have occurred, an adaptive fade-out by regression method is used. In this adaptive fade-out by regression, a grouped average norm value of an erased frame is predicted using K grouped average norm values of the previous good frame through regression analysis.

Figure 3 illustrates the structure of grouped sub-bands when the regression analysis is applied to a narrowband (supported up to 4.0 KHz) signal. Grouped average norm values obtained from grouped sub-bands form a vector, which

is referred to as an average vector of grouped norms. K grouped average norm values of each grouped sub-band (GSb) are used for the regression analysis.

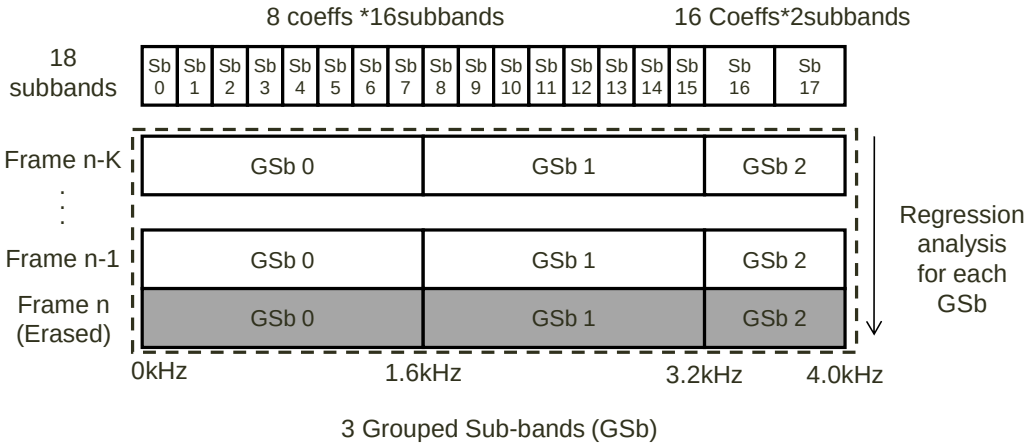

Figure 3: Structure of grouped sub-bands

Figure 4 illustrates the concept of a linear regression analysis and a non-linear regression analysis. Between the two methods the linear regression analysis is applied to the adaptive fade-out, wherein the 'average of norms' indicates an average norm value obtained by grouping several bands and is the target the regression analysis is applied to. A linear regression analysis is performed when the quantized value of the norm is used for an average norm value of a previous frame. 'Number of PGF', which is used for a regression analysis, indicates the number of the previous good frames and is used for a regression analysis is a variable. The linear regression analysis is represented by equations (172) and (173).

$$y = ax + b \tag{172}$$

$$\begin{bmatrix} m & \sum x_k \\ \sum x_k & \sum x_k^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_k \\ \sum x_k y_k \end{bmatrix} \tag{173}$$

As in equation (172), when a linear equation is used, the upcoming transition(y) is predicted by obtaining a and b. In this equation, x can be a frame index. In equation (173), a and b are obtained by an inverse matrix. Gauss-Jordan Elimination is a simple method of obtaining an inverse matrix.

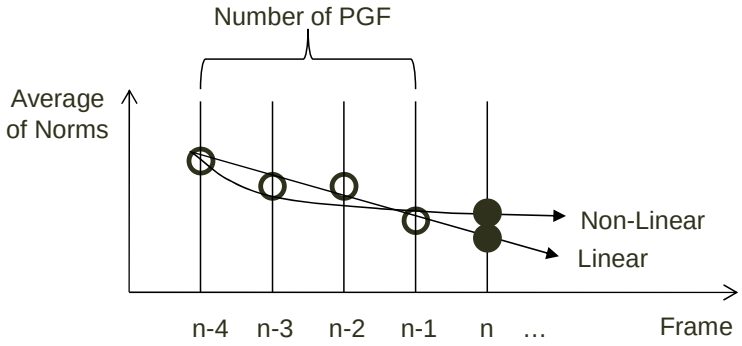

Figure 4: The concept of a linear regression analysis and a non-linear regression analysis

Figure 5 is a block diagram of a packet loss concealment block with adaptive fade-out. Referring to Figure 5, the signal characteristic determiner determines the characteristics of a signal by using a decoded signal and classifies the characteristics of the decoded signal into transient and normal frames. A method of determining a transient frame is

now described. The current frame classification of transient is determined by two parameters: the frame type (is_transient) which is transmitted from the encoder, and the energy difference (energy_diff), which is represented by Equation (169).

```
if(energy_diff < ED_THRES && is_transient == 0 ) {
        /* Not Transient */
          num_pgf = 4;
      }
       else{
          num_pgf = 2;
      }
```

In the above context, ED_THRES denotes a threshold and is set to 1.0. According to the result of the transient determination, the number of PGFs (num_pgf), referred to in the subclause on regression analysis, can be controlled for packet loss concealment.

Another parameter for packet loss concealment is a scaling method of burst erasure duration. The same energy difference value is used for the duration of a single burst.

```
if((energy_diff<ED_THRES) && (is_transient==0)) {
      /* Not Transient */
       mute_start = 5;
       random_start = 2;
   }
   else {
       mute_start = 2;
       random_start = 2;
   }
```

If it is determined that the current frame is an erasure and is not transient, then when a burst erasure occurs frames starting from the fifth frame of the burst are forcibly scaled to a fixed value of 3 dB regardless of the regression analysis of the decoded spectral coefficient of the previous frame. Otherwise, if it is determined that the current frame that is erased is transient, when a burst erasure occurs, frames starting from the second frame are forcibly scaled to a fixed value of 3 dB regardless of the regression analysis of the decoded spectral coefficient of the previous frame.

Because regression analysis is performed only when a burst erasure has occurred, when nbLostCmpt indicates the number of contiguous erased frames is two, that is, from the second contiguous erased frame, the regression analysis is performed.

```
if (nbLostCmpt==2){
  regression_anaysis();
 }
```

Even though an nth frame is a good frame, if the $(n-1)$ th and $(n+1)$ th frames are erased frames, a totally different signal is generated in an overlapping process. Thus, when erasures occur in a non-consecutive order (an erasure frame, a good frame, and an erasure frame), although nbLostCmpt of the third frame (the second erasure) is 1, nbLostCmpt is forcibly increased by 1. As a result, nbLostCmpt is 2, and it is determined that a burst erasure has occurred, and thus the regression analysis will be used.

```
if( prev_old_bfi == 1 && nbLostCmpt == 1 && output_frame_org == L_FRAME8k )
   {
       nbLostCmpt++;
   }
```

In the above context, prev_old_bfi denotes frame erasure information of the second previous frame. This process is applicable when the current frame is an erased frame. To reduce complexity, the regression analyzer block forms each group by grouping 8 or 2 bands, and applying the regression analysis to the mean vector of grouped norms. The number of previous good frames for the regression analysis is set to either 2 or 4, and is controlled by the result of the signal characteristic determiner block. In addition, the number of rows of the matrix for the regression analysis is set to 2. As a result of the regression analysis by the regression analyzer block, an average norm value of each group is predicted for an erased frame and is done by calculating the values a and b from a linear regression analysis equation (173). In this block the calculated value a can be adjusted to the predetermined range as follows. In EVS the range is always limited to a negative value. In the following pseudo-code, norm_values is an average norm value of each group in the previous good frame and norm_p is a predicted average norm value of each group.

```
if( a > 0 ){
       a = 0;
       norm_p[i] = norm_values[0];
   }
   else   {
       norm_p[i] = (b+a*(nbLostCmpt-1+num_pgf);
   }
```

With this modified value of a , the average norm value of each group is predicted. When the predicted norm is larger than zero and the norm of the previous frame is non-zero, the gain calculator block calculates a gain between the average norm value of each group that is predicted for the erased frame and an average norm value of each group in the previous good frame. Otherwise, the gain is scaled down by 3 dB from the initial value of 1.0.

The calculated gain is also adjusted to a predetermined range. In EVS, the maximum value of the gain is 1.0.

The scaler block applies gain scaling to the previous good frame to predict spectral coefficients of the erased frame. The scaler block also applies adaptive muting to the erased frame and a random sign to predicted spectral coefficients according to characteristics of an input signal, which is also controlled by the results of the signal characteristic determiner block.

The number indicated by mute_start indicates that muting forcibly starts when bfi_cnt is equal to or greater than mute_start and when continuous frame erasures occur. In addition, random_start, related to the random sign, is analysed in the same way.

According to a method of applying adaptive muting, spectral coefficients are forcibly down-scaled by 3dB. In addition, the sign of each of the spectral coefficients is randomly modified to reduce modulation noise generated due to repetition of spectral coefficients in each frame.

In addition, the random sign is applied to frequency bands equal to or higher than the second frequency band, as it should be better to use the sign of a spectral coefficient that is identical to that of the previous frame in a very low frequency band (0~200Hz for the first band). Accordingly, a sharp change in the signal can be smoothed, and an erased frame accurately restored to be adaptive to the characteristics of the signal, in particular, a transient characteristic.

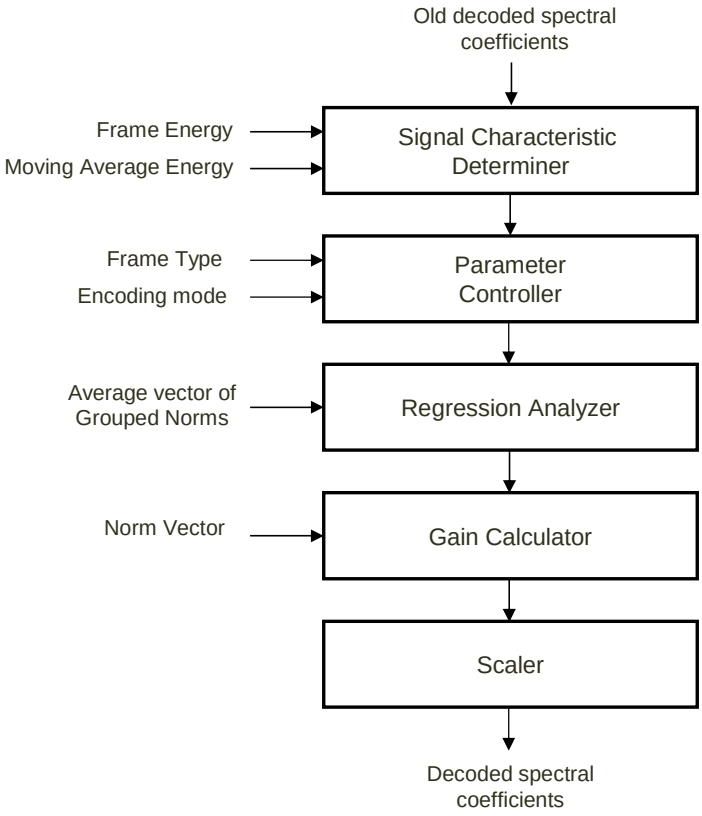

Figure 5: Block diagram of a packet loss concealment block with adaptive fade-out

5.4.3.4 MDCT frame repetition with sign prediction

An analysis of the sign change of the MDCT coefficients in the received frames is continuously performed. The analysis of $X_M'^{[m]}(k)$ and $X_M'^{[m-1]}(k)$ is performed on 4-dimensional bands up to 1.6 kHz ($k = 64$ MDCT coefficients divided into $B = 16$ bands).

Two 16-dimensional state variables, used to determine the sign of the reconstructed MDCT vector, $C^{[m]}(b)$ and $\Delta C^{[m]}(b)$ hold the number of sign switches between consecutive frames. The analysis takes also into account signal dynamics (measured by a transient detector), to decide on the reliability of past data. Updates for both state variables are done only for $isTransient == 0$, if $isTransient == 1$ the values are set to zero.

Within a sub- band $b = 0,\ldots,B-1$, first state variable is incremented whenever the sign of the corresponding MDCT coefficients switches:

$$\begin{aligned} &for \quad k \in b \\ &\quad if \quad X_M'^{[m]}(k) \times X_M'^{[m-1]}(k) < 0 \\ &\qquad C^{[m]}(b) = C^{[m]}(b) + 1 \end{aligned} \tag{174}$$

The second state variable accumulates number of sign switches over consecutive frames:

$$\Delta C^{[m]}(b) = C^{[m]}(b) + C^{[m-1]}(b), \quad b \in B \tag{175}$$

When frame m is lost, the missing MDCT vector is reconstructed by copying the last available coefficients. The sign of the reconstructed vector can be preserved or changed on a sub-band basis (every 4 coefficients). Inside a band b the decision whether to change the sign or not is based on comparing the second state variable to a pre-determined threshold as follows (wherein a sign flip or reversal is indicated by -1 and preservation of the sign is indicated by +1):

$$for \quad k \in b$$

$$sign\left(X_M^{'[m]}(k)\right) = \begin{cases} -1 & if \quad \Delta C^{[m-1]}(b) \geq T \\ +1 & if \quad \Delta C^{[m-1]}(b) < T \end{cases} \tag{176}$$

The threshold T is adjusted to the past decision of the transient detector. The sequential decision logic is illustrated in Table 10.

Table 10: Sign extrapolation decision logic

1. If any of frames $m-1$ or $m-2$ contains transient	Apply random sign to the copied coefficients
2. If frames $m-1$ or $m-2$ are good, but frame $m-3$ contains transient	Apply sign extrapolation with $T=3$
3. If frames $m-1, m-2$, and $m-3$ are good	Apply sign extrapolation with $T=6$

5.4.3.5 Phase ECU

Phase ECU is a frame loss concealment method especially suitable for general audio and music signals. It provides a smooth and faithful time evolution of the reconstructed signal for a lost frame, wherein the audible impact of a frame loss is minimized.

Phase ECU is a frame loss concealment technique that operates with a sinusoidal model under the assumption that the audio signal is composed of a limited number of individual sinusoidal components. The general principle of Phase ECU comprises sinusoidal analysis of a previously received good HQ MDCT coded frame of the audio signal (analysis frame), wherein the sinusoidal analysis involves identifying frequencies of sinusoidal components of the audio signal. Further, a sinusoidal model is applied on this previously synthesized frame, wherein it is used as a prototype frame in order to create a substitution frame for a lost audio frame. The creation of the substitution frame is done by time-evolving the identified sinusoidal components of the prototype frame, up to the time instance of the lost audio frame, in response to the corresponding identified sinusoidal frequencies.

In more detail Phase ECU operation comprises the steps of sinusoidal analysis, described in subclause 5.4.3.5.2, and application of the sinusoidal model based on a prototype frame of the earlier synthesized signal in order to generate a substitution frame for the lost audio frame, described in subclause 5.4.3.5.3. In addition and prior to this basic Phase ECU operation a transient analysis step is carried out, described in subclause 5.4.3.5.1 with the purpose to detect audio signal and burst frame loss conditions under which the basic Phase ECU operation is adapted in order to ensure maximum reconstruction signal quality.

5.4.3.5.1 Transient analysis

The purpose of the calculations in transient analysis is the detection of properties of the previously reconstructed good signal frame or the frame loss statistics that could lead to suboptimal signal reconstruction quality with the Phase ECU. Upon such detected conditions phase and magnitude of the substitution frame are selectively adjusted in order to mitigate potential quality degradations. Conditions under which such adjustments are carried out are detected transients or burst losses with several consecutive frame losses. The result of the transient analysis is phase and magnitude modification factors corresponding to such adjustments.

Transient analysis is performed on each lost frame, and the following steps are performed for the first lost frame or for the second lost frame in case the first lost frame was handled with the method according to subclause 5.4.3.6. For

subsequent lost frames transient analysis relies on previously calculated and stored parameters (that were calculated based on the synthesis of the last good HQ MDCT frame). For these losses transient analysis adjusts magnitude spectrum attenuation factors and phase dithering degrees in response to detected transient or burst loss condition.

The transient analysis is performed in the frequency domain. Two FFTs are performed on a left and a right part of the analysis frame buffer which contains the previous synthesis

$$\begin{aligned} X_F^l(k) &= FFT\left(w_h(n)\cdot x_{prev}(-L_{tran}-n)\right) \\ X_F^r(k) &= FFT\left(w_h(n)\cdot x_{prev}(-L_{tran}-n-L)\right) \end{aligned} \tag{177}$$

where L_{tran} is the length of the transient analysis, set to 64, 128, or 192 for WB, SWB, and FB, and $w_h(n)$ is a hamming window of corresponding length. The resulting FFT spectrum is split into bands according to Table 11 that are approximately following the size of the human auditory critical bands, and the energy in each band is calculated.

Table 11: Band start of Phase ECU

b	0	1	2	3	4	5	6	7	8
$k_{start}(b)$	1	3	6	10	16	32	64	128	192

$$\begin{aligned} E_l(b) &= \sum_{k=k_{start}(b)}^{k_{end}(b)} \left|X_F^l(k)\right|^2 \\ E_r(b) &= \sum_{k=k_{start}(b)}^{k_{end}(b)} \left|X_F^r(k)\right|^2 \end{aligned} \tag{178}$$

Next the ratio of these energies is calculated as

$$G_{tran}(b) = \frac{E_r(b)}{E_l(b)} \tag{179}$$

This means that the transient detection is made frequency selectively for each frequency band b. The gain $G_{tran}(b)$ is then compared with an upper and a lower threshold for onset or respectively offset detection. If $G_{tran}(b) > 10$ or $G_{tran}(b) < 0.1$ is fulfilled, then band b contains a transient and $T_{tran}(b)$ is set to 1. The gains $G_{mag}(b)$ are set to 1. If a band has a transient, then gain $G_{mag}(b)$ is updated to:

$$G_{mag}(b) = \min\left(1, \sqrt{G_{tran}(b)}\right) \tag{180}$$

The gains $G_{mag}(b)$ for the first lost frame are saved into $G'_{mag}(b)$.

The derivation of magnitude and phase modification factors in response to a detected burst loss condition is described in the following. The variable $\alpha(b)$ is set to 1, $\beta(b)$ is set to 0, and $\beta_{mute} = 0.5$. An average energy of each band is calculated:

$$\overline{G}_{tran}(b) = \sqrt{\frac{1}{2}\cdot\frac{E_r(b)+E_l(b)}{k_{end}(b)-k_{start}(b)}} \tag{181}$$

$\overline{G}_{tran}(b)$ corresponds to a low-resolution spectrum of the last good frame. This spectrum is used as part of the burst loss handling feature of Phase ECU. It is used for a spectrally shaped additive noise signal to which the substitution signal is pulled in case of burst frame losses.

For subsequent lost frames, the gain $G_{mag}(b)$ is updated according to:

$$G_{mag}(b) = G'_{mag}(b)\cdot 10^{-G_{att}/20} \tag{182}$$

where

$$\begin{cases} G_{att} = 0, & N_{lost} < T_{att} \\ G_{att} = K_{att}\left(N_{lost} - T_{att}\right), & T_{att} < N_{lost} \le 15 + T_{att} \\ G_{att} = 15K_{att} + \left(N_{lost} - T_{att} - 15\right) \cdot 6.0206, & N_{lost} > 15 + T_{att} \\ K_{att} = 4 - \text{round}(S) & \\ T_{att} = 2 + \text{round}(S) & \end{cases} \quad (183)$$

Here N_{lost} is the number of consecutive lost frames and $S \in [0,1]$ is envelope stability feature described in [5] subclause 6.2.3.2.1.3.3, where the range endpoints 0 and 1 represent speech and music respectively. If $N_{lost} > 10$ then $\beta_{mute} = \beta_{mute} \cdot 0.5$.

The attenuation factors $\alpha(b)$ and $\beta(b)$ are updated as:

$$\begin{aligned} \alpha(b) &= G_{mag}(b) \\ \beta(b) &= \beta_{mute} \cdot \sqrt{1 - \alpha^2(b)} \end{aligned} \quad (184)$$

Through variable $\alpha(b)$ the concealment method is modified by selectively adjusting the magnitude of the substitution frame spectrum, based on the frequency domain transient detector status $G_{tran}(b)$, see equation (189).

The scaling factor $\beta(b)$ is used to scale the spectrally shaped additive noise signal such that, except for the incorporated gradual muting behaviour through factor β_{mute}, it compensates for the energy loss caused by the attenuation with factor $\alpha(b)$. This is an aspect of the long-term muting behaviour which is outlined in subclause 5.4.6.2.2.

For $b = 5$, $\beta(b)$ is further adjusted as $\beta(b) = \beta(b) \cdot 0.5$ and for $b \ge 6$ $\beta(b) = \beta(b) \cdot 0.1$. This superimposes a low-pass characteristic on the additive noise signal, which avoids unpleasant high-frequency noise in the substitution signal.

The variable $phase\_dither$ is initialized to 0, and for $N_{lost} > 3 + \text{round}(S)$ the phase dither is calculated:

$$phase\_dither = 2\pi \cdot \min\left(1, \frac{N_{lost} - (3 - round(S))}{2}\right) \quad (185)$$

5.4.3.5.2 Spectrum analysis

The spectrum analysis is carried out in the frequency domain. It is only done once for the first lost frame after a good HQ MDCT frame. The buffer with the previous synthesis of the last good HQ MDCT frame (analysis frame) is windowed and passed through a FFT.

$$X_F(k) = FFT\left(w_{hr}(n) \cdot x_{prev}(L_{prot} - n)\right) \quad (186)$$

where $w_{hr}(n)$ is a hamming-rectangular window, and L_{prot} is the length of the FFT set to 512, 1024, or 1536 for WB, SWB, and FB signals,

$$w_{hr}(n) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{\pi n}{L_h}\right) & 0 \le n < L_h \\ 1 & L_h \le n < L - L_h \\ 0.54 + 0.46\cos\left(\dfrac{\pi(n - L + 2L_h)}{L_h}\right) & L - L_h \le n < L \end{cases} \quad (187)$$

and L_h is the length of the hamming part, and is 96, 192, or 288 for WB, SWB, and FB.

The spectrum $X_F(k)$ is saved and used for all consecutive frame losses. Then the magnitude spectrum $|X_F(k)|$ is calculated. Then the peaks of this magnitude spectrum are located by a peak picking method. The number of found peaks is N_{peaks}, and the peaks locations are $k_p(j),\ j = 0,\ldots,N_{peaks} - 1$. The frequency resolution of these peak locations is however still insufficient for good Phase ECU performance, since the true frequencies of the sinusoidal model

components are rather found in the vicinity of them. Thus, after the peaks in the magnitude FFT spectrum are found, their positions are further refined to make them available in highest possible resolution. The refinement is carried out by using parabolic interpolation, which yields the fractional peak locations $k'_p(j),\ j=0,...,N_{peaks}-1$.

This sinusoidal model is also used in reconstruction of the lost audio frame.

5.4.3.5.3 Frame reconstruction

The substitution frame for the lost frame is calculated by applying the sinusoidal model on a frame of the previously synthesized good frame signal, where this frame serves as a prototype frame. The previously calculated sinusoidal components of this prototype frame are time evolved to the time instant of the lost frame. For numerical simplicity, this prototype frame and its spectrum are chosen to be identical to the windowed analysis frame and, respectively, its already calculated and saved spectrum (see subclause 5.4.3.5.2). While the exact time evolution of the sinusoids of the windowed prototype frame would require a complex superposition of frequency-shifted, phase-evolved and sampled instances of the spectrum of the used window function, Phase ECU operates with an approximation of the window function spectrum such that it comprises only a region around its main lobe. With this approximation the substitution frame spectrum is composed of strictly non-overlapping portions of the approximated window function spectrum and hence the time-evolution of the sinusoids of the windowed prototype frame reduces to phase-shifting the sinusoidal components of the prototype spectrum in δ-regions around the corresponding spectral peaks j by an amount $\theta(j)$. Note that this amount $\theta(j)$ merely depends on the respective sinusoidal frequency (peak location) $k'_p(j),\ j=0,...,N_{peaks}-1$ and the time shift between the lost frame and the prototype frame. This is expressed in the following equation. The phase shift is calculated as:

$$\theta(j) = 2\pi \frac{k'_p(j)}{L_{prot}} \cdot \left(2L - (2L - L_{prot})/2 - L/2 + k_{offs} - K\right) \tag{188}$$

where k_{offs} is the offset in number of samples since the last good frame. k_{offs} is a variable incremented by L for each lost frame, and K equals 40, 80, or 120 for WB, SWB, or FB signals.

Note, that if either of the last two frames have the *isTransient* flag set, then the number of peaks is set to 0.

Next the spectrum around the spectral peaks is updated and random noise component related to burst loss handling is added

$$X'_F(k) = \alpha(b) \cdot X_F(k) \cdot e^{i\theta(j)} + \beta(b) \cdot \overline{G}_{tran}(b) \cdot e^{i \cdot 2\pi \,\mathrm{rand}} \tag{189}$$

where $k = j - \delta_1,..., j + \delta_2$, b is set according to Table 11, rand is a random number between -1 and 1, and

$$\begin{aligned} \delta_1 &= \min\left(6, \frac{k_p(j) - k_p(j-1) - 1}{2}\right) \\ \delta_2 &= \min\left(6, \frac{k_p(j+1) - k_p(j) - 1}{2}\right) \end{aligned} \tag{190}$$

If *phase_dither* is non-zero, the amplitude is adjusted, and a small random component is added to the phase

$$\theta(j) = \theta(j) + \mathrm{rand} \cdot \mathit{phase\_dither} \tag{191}$$

$$\alpha(j) = \alpha(j) \cdot \left(0.5 + 0.5 \cdot \left(1 - \mathit{phase\_dither}/2\pi\right)\right) \tag{192}$$

The spectral coefficients which have not been updated are also updated in a similar manner but with a randomized phase.

For clarity it is to be noted that the first additive term in equation (189) relates to phase shifting the sinusoidal components of the prototype spectrum. In addition, if *phase_dither* is non-zero, the phase is modified with a random component. This avoids quality degrading tonal sounds due to too strong periodicity and is useful both in case of transients and burst frame loss. In addition, for the same reason the magnitude of the prototype frame spectral coefficients is attenuated with the scaling factor $\alpha(b)$. The second additive term in equation (189) modifies the

substitution frame spectral coefficients by an additive noise component, where the magnitude of the additive noise component corresponds to the scaled coefficient of the low-resolution magnitude spectrum of the previous good frame, $\overline{G}_{tran}$, which derivation is described in subclause 5.4.3.5.1. The scaling factor $\beta(b)$ is chosen such that, except for the incorporated gradual muting behaviour, it compensates for the energy loss caused by the attenuation with factor $\alpha(b)$. This is an aspect of the long-term muting behaviour which is outlined in subclause 5.4.6.2.2.

The reconstructed substitution frame spectrum is passed through the IFFT to create a time domain substitution frame.

$$x_{ph}(n) = IFFT(X'_F(k)) \tag{193}$$

Where $n = 0,...,L_{prot}$.The signal $x_{ph}(n)$ is zero extended outside of this range. This signal $x_{ph}(n + L_{prot}/2 - n_{zero})$ is then windowed and time-domain aliased as described in [5], clause 5.3.2.2 (n_{zero} is the number of zero samples in the ALDO window). The resulting windowed and time-domain aliased signal is then overlap-added with the previous frame as described in [5], clause 6.2.4.1.

5.4.3.6 MDCT concealment based on sinusoidal synthesis and adaptive noise filling

The MDCT concealment based on sinusoidal synthesis and adaptive noise injection is illustrated in Figure 6. Note that the resampling to 8 kHz and pitch search are already described in clause 5.4.3.1.

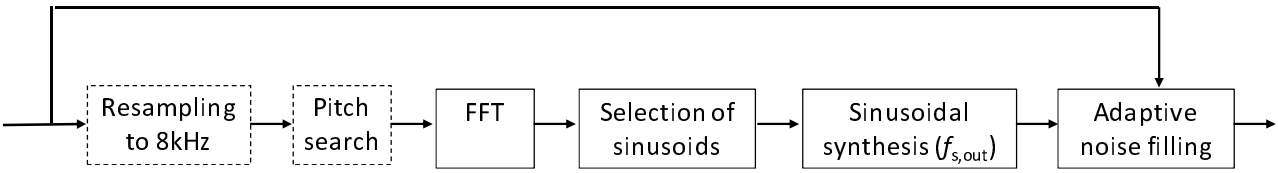

Figure 6: Block diagram of MDCT concealment based on sinusoidal synthesis and adaptive noise filling

5.4.3.6.1 FFT

The pitch cycle of length T_c is extracted from the resampled past synthesis $\hat{s}_8^{prev}(n)$ of length 320 as: $s_{target}(n) = \hat{s}_8^{prev}(320 - 24 - T_c + n), n = 0,...,T_c - 1$. This pitch cycle is analyzed in frequency domain using the following steps:

- The signal $s_{target}(n), n = 0,...,T_c - 1$ is linearly interpolated to a length corresponding to a power of 2 to obtain the segment $s_{target}'(n)$ of length T_c' such that $T_c' = 2^{\lceil \log_2(T_c) \rceil}$ where $\lceil . \rceil$ is the rounding upward to the nearest integer. A linear interpolation is applied as follows:

$$\begin{cases} s_{target}'(0) = s_{target}(0) \\ s_{target}'(T_c' - 1) = s_{target}(T_c - 1) \\ s_{target}'(n) = s_{target}(\lfloor n\tau \rfloor) + (n\tau - \lfloor n\tau \rfloor)(s_{target}(\lfloor n\tau \rfloor + 1) - s_{target}(\lfloor n\tau \rfloor)), n = 1,\ldots,T_c' - 2 \end{cases} \tag{194}$$

  where $\lfloor . \rfloor$ is the rounding downward to the nearest integer and $\tau = (T_c - 1)/(T_c' - 1)$.

- The segment $s_{target}'(n)$ is decomposed in frequency domain by FFT of length $N_{FFT} = T_c'$ to obtain the spectrum $T(k)$, $k = 0,\ldots,N_{FFT} - 1$

- The amplitude spectrum $|T(k)|$ is computed for $k = 0,\ldots,N_{FFT}/2 + 1$ and the overall amplitude is also computed as:

$$\sigma_T = \sum_{k=0}^{N_{FFT}/2+1} |T(k)| \tag{195}$$

5.4.3.6.2 Selection of sinusoidal components

Sinusoidal components are selected by first detecting the number N_{peak} of spectral peaks following condition: $|T(k)| > |T(k-1)|$ and $|T(k)| > |T(k+1)|$. When the binary voicing indication has the value v =1, the peak selection is extended to select not only the peak at index k meeting the preceding condition but also neighboring peaks at index $k-1$ and $k+1$; this allows capturing a larger portion of the overall spectral energy and lowering the noise to be re-injected for voiced signals.

The final number of peaks to be kept is $N_{peak,final} = \min(20, N_{peak})$ to reduce the computational load of the subsequent sinusoidal synthesis. This final selection of peaks is performed by iteratively selecting the peak maximizing $|T(k)|$ among peaks that are not yet selected as long as the conditions v =1 or $\sum_{k=0}^{Npeak,selected} |T(k)| < 0.7\sigma_T$ is still met, where the latter condition ensures that 70% of the amplitude spectrum is covered.

For each *i*-th peak that gets selected, the amplitude $A_i = |T(k_i)|$, phase $\varphi_i = \angle T(k_i)$ and normalized frequency $f_i = 2k_i / N_{FFT}$ are computed.

5.4.3.6.3 Sinusoidal synthesis

A segment of length $L_{\sin}$ corresponding to 2 frames of 20 ms (40 ms) and 8 kHz to resampling delay is generated from the $N_{peak,final}$ selected frequency bins as:

$$\hat{s}_{\sin}(n) = \sum_{i=0}^{N_{peak,final}} A(i)\sin(\pi f_i + \varphi_i), \quad n = 0,\ldots,L_{\sin}-1 \tag{196}$$

This sinusoidal synthesis is implemented using an autoregressive of order 2. The extra segment length (after the current frame) is used for crossfading with the next decoded frame and to compensate for resampling delay.

5.4.3.6.4 Adaptive noise filling

Frequency components that do not correspond to selected sinusoids below 4 kHz or that are above 4 kHz are re-injected by adaptive noise filling, in particular to compensate for energy loss.

The pitch computed according to clause 5.4.3.1.2 is mapped to the output sampling rate $f_{s,out}$ as δT_c , where δ is the decimation factor used in sub-clause 5.4.3.1.1 with δ =2, 4, 6 for $f_{s,out}$ =16, 32 or 48 kHz respectively. For a 20 ms frame length L_{frame} at $f_{s,out}$, a residual signal is computed as:

$$v_{noise}(n) = \hat{s}^{prev}(n) - \hat{s}_{\sin}(n), \quad n = 0,\ldots,L_{noise}-1 \tag{197}$$

where the residual length is $L_{noise} = \delta T_c$ if $\delta T_c < L_{frame}$ and $L_{noise} = L_{frame}$ otherwise. Then, if the binary voicing indicator has the value v =1, the residual is further scaled down by a factor of 0.25 as $v_{noise}(n) = 0.25 v_{noise}(n)$, $n = 0,\ldots,L_{noise}-1$.

This residual signal is repeated iteratively by adding blocks of variable length until the length of 2 frames (40 ms) is reached. The start index for the residual repetition is initialized to $N_{rand} = 0$. In the m -th iteration:

- A block length $L_{rand}{}^{[m]}$ is pseudo-randomly computed by alternating between $L_{rand}{}^{[m]} = (0.5 + 0.2.rand)L_{noise}$ and $L_{rand}{}^{[m]} = (0.7 + 0.3.rand)L_{noise}$, where $rand$ is a random number between 0 and 1.
- A sine window $w_{rand}{}^{[m]}(n)$ of length L_{rand} is computed as:

$$w_{rand}{}^{[m]}(n) = \sin\left(\frac{n+0.5}{L_{rand}{}^{[m]}}\frac{\pi}{2}\right), \quad n = 0,\ldots,L_{rand}{}^{[m]}-1 \tag{198}$$

This calculation is performed by running an autoregressive filter of order 2.

- Two blocks are extracted from the residual signal: $b_1(n) = v_{noise}(L_{noise} - L_{rand}{}^{[m]} + n)$, $n = 0,\ldots,L_{rand}{}^{[m]}-1$ and $b_2(n) = v_{noise}(L_{noise} + n)$, $n = 0,\ldots,L_{rand}{}^{[m]}-1$. Note that the blocks overlap with each other and the length of the overlap depends on the value of $L_{rand}{}^{[m]}$ in the current iteration.
- These two blocks are overlap-added to update the noise vector $\hat{s}_{noise}(n)$ from the current start index $n = N_{rand}$:

$$\hat{s}_{noise}(n) = w_{rand}{}^{[m]}(n)b_1(n) + w_{rand}{}^{[m]}(L_{rand}{}^{[m]} - n - 1)b_2(n), \quad n = N_{rand},\ldots,N_{rand} + L_{rand}{}^{[m]} - 1 \tag{199}$$

  Note that the upper limit $N_{rand} + L_{rand}{}^{[m]} - 1$ of the time interval is actually saturated to $\max\left(N_{rand} + L_{rand}{}^{[m]} - 1, L_{frame}\right)$.

- The start index is updated: $N_{rand} = N_{rand} + \sum_{l=0}^{m} L_{rand}{}^{[l]}$

The iterations stop as soon as $N_{rand} \geq 2L_{frame}$.

5.4.3.6.5 Synthesis

The signal is synthesized as:

$$\hat{s}(n) = \hat{s}_{\sin}(n) + \hat{s}_{noise}(n), \quad n = 0,\ldots,L_{\sin}-1 \tag{200}$$

Note that when the binary voicing indicator has the value v=1, the noise vector $\hat{s}_{noise}(n)$ has been scaled down by a factor of 0.25, to avoid artefacts for voiced signals.

This signal is overlap-added with the previously decoded synthesis to ensure signal continuity between frames.

5.4.3.7 Time-domain PLC and OLA

5.4.3.7.1 PLC mode selection

The frequency domain PLC block includes a frequency domain erasure concealment algorithm and operates when the BFI flag is set to 1 and the decoding mode of the previous frame is the frequency domain mode. The frequency domain PLC block generates spectral coefficients of the erased frame by repeating the synthesized spectral coefficients of the previous good frame stored in memory. With these coefficients the IMDCT block generates the time domain signal by performing a time-frequency inverse transform. The conventional OLA block performs a general OLA processing by using the time domain signal of the previous frame, and generates a final time domain signal of the current frame as a result of the general OLA processing.

To achieve an additional quality enhancement taking into account the input signal characteristics, the time-domain PLC introduces two concealment tools, consisting of a phase matching tool and a repetition and smoothing tool. With these tools, an appropriate concealment method is selected by checking the stationarity of the input signal.

Figure 7 shows the two concealment tools and the conventional OLA for the time-domain PLC.

The phase matching block in the figure will be introduced in subclause 5.4.3.7.2 and the repetition and smoothing block in the figure will be introduced in subclause 5.4.3.7.3.

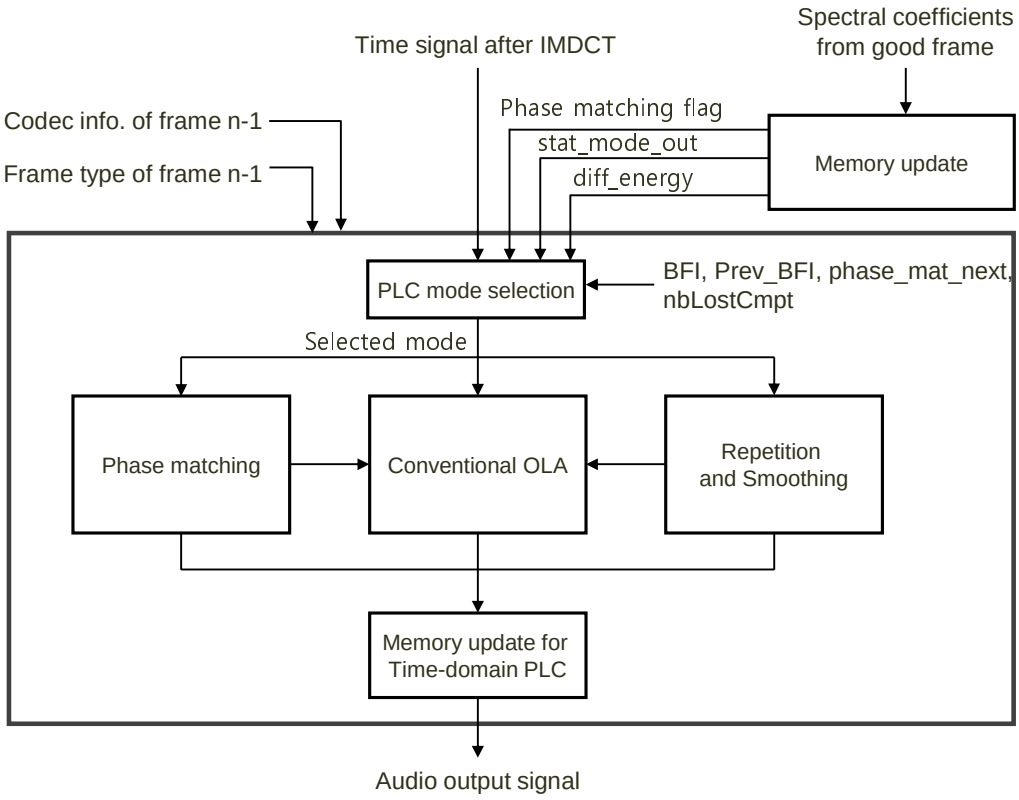

Figure 7: Block diagram of a Time-domain PLC module

Table 12 summarizes the PLC modes that are used for time-domain PLC. There are two tools for the time-domain PLC. Each of these tools has several modes representing the erased frame types. The erased frame types are classified as single erasure frame, burst erasure frame, next good frame after erasure frame, and next good frame after burst erasure.

Table 12: Used PLC modes for Time-domain PLC

Name of tools	Single erasure frame	Burst erasure frame	Next good frame	Next good frame after burst erasures
Phase matching	Phase matching for erased frame	Phase matching for burst erasures	Phase matching for next good frame	Phase matching for next good frame
Repetition & Smoothing	Repetition &smoothing for erased frame	Repetition &smoothing for erased frame	Repetition &smoothing for next good frame	Next good frame after burst erasures

Table 13 summarizes the PLC mode selection method for the PLC mode selection block in Figure.7.

Table 13: PLC mode selection

Parameters	Status of Parameters						Definitions
BFI	1	0	1	1	0	0	Bad frame indicator for the current frame
Prev_BFI	-	1	1	-	1	1	BFI for the previous frame
nbLostCmpt	1	-	-	-	-	>1	The number of contiguous erased frames
Phase_mat_flag	1	-	-	0	0	0	The flag for the Phase matching process (1: used, 0: not used)
Phase_mat_next	0	1	1	0	0	0	The flag for the Phase matching process for

							burst erasures or next good frame (1: used, 0: not used)
stat_mode_out	-	-	-	(1)*	(1) *	0	The flag for Repetition &smoothing process (1: used, 0: not used)
diff_energy	-	-	-	(<0.159063) *	(<0.159063) *	≥0.159063	Energy difference
Selected PLC mode	Phase matching for erased frame	Phase matching for next good frame	Phase matching for burst erasures	Repetition &smoothing for erased frame	Repetition &smoothing for next good frame	Next good frame after burst erasures	
Name of tools	Phase matching			Repetition and Smoothing			
NOTE: *The () means "OR" connections.							

The pseudo code to select a PLC mode for the phase matching tool is as follows.

```
if( (nbLostCmpt==1)&&(phase_mat_flag==1)&&(phase_mat_next==0) ) {

   Phase matching for erased frame ();

  }

 else if((prev_bfi == 1)&&(bfi == 0) &&(phase_mat_next == 1)) {

     Phase matching for next good frame ();

  }

 else if((prev_bfi == 1)&&(bfi == 1) &&(phase_mat_next == 1))  {

   Phase matching for burst erasures ();

  }
```

Using this selection method, the phase matching flag (phase_mat_flag) determines at the point of the memory update block in the previous good frame whether phase matching erasure concealment processing is used for every good frame when an erasure occurs in a next frame. To this end, energy and spectral coefficients of each sub-band are used. The energy is obtained from the norm value. More specifically, when a sub-band having the maximum energy in a current frame belongs to a predetermined low frequency band, and the inter-frame energy change is not large, the phase matching flag is set to 1.

The detailed method is as follows. When a sub-band having the maximum energy in the current frame is within the range of 75 Hz to 1000 Hz, a difference between the index of the current frame and the index of a previous frame with respect to a corresponding sub-band is 1 or less, and the current frame is a stationary frame of which an energy change is less than the threshold (ED_THRES_90P), and three past frames stored in the buffer are not transient frames, then phase matching erasure concealment processing will be applied to a next frame to which an erasure has occurred.

```
    if ((Min_ind<5) && ( abs(Min_ind - old_Min_ind)< 2) && (diff_energy<ED_THRES_90P) && (!bfi) &&
    (!prev_bfi) && (!prev_old_bfi) && (!is_transient) && (!old_is_transient[1]))        {

            if((Min_ind==0) && (Max_ind<3))   {

                phase_mat_flag = 0;

            }

            else  {

                phase_mat_flag = 1;

            }

        }

        else {

            phase_mat_flag = 0;

        }
```

The PLC mode selection method for the repetition and smoothing tool and the conventional OLA is as follows.

The result of the stationarity detection of an erased frame is performed by a memory update block. In this detection we introduce a hysteresis in order to prevent a frequent change of the detected result. The stationarity detection of the erased frame determines whether the current erased frame is stationary by receiving information including a stationary mode stat_mode_old of the previous frame, an energy difference diff_energy, and the like. Specifically, the stationary mode flag stat_mode_curr of the current frame is set to 1 when the energy difference diff_energy is less than 0.032209. The energy difference(E_d) is given by the Equation (169).

If it is determined that the current frame is stationary, the hysteresis application generates a final stationarity parameter, stat_mode_out from the current frame by applying the stationarity mode parameter stat_mode_old of the previous frame to prevent a frequent change in stationarity information of the current frame. The pseudo code for the hysteresis application is as follows.

```
/* Apply Hysteresis to prevent frequent mode changing */
      if(stat_mode_old == stat_mode_curr)
      {
          stat_mode_out = stat_mode_curr;
      }
      stat_mode_old = stat_mode_curr;
```

First, the operation of the PLC mode selection depends on whether the current frame is an erased frame or the next good frame after an erased frame. Referring to Table 13, for an erased frame, a determination is made whether the input signal is stationary by using various parameters. More specifically, when the previous good frame is stationary and the energy difference is less than the threshold, it is concluded that the input signal is stationary. In this case, the repetition and smoothing processing is performed. If it is determined that the input signal is not stationary, then the general OLA processing is be performed.

Referring to Table 13, a determination whether the input signal is stationary is made by using the same parameters and same method. If the input signal is not stationary, then for the next good frame after an erased frame a determination is made whether the previous frame is a burst erasure frame by checking whether the number of consecutive erased frames is greater than one. If this is the case, then erasure concealment processing on the next good frame is performed in response to the previous frame that is a burst erasure frame. If it is determined that the input signal is not stationary and the previous frame is a random erasure, then the conventional OLA processing is performed.

If the input signal is stationary, then the erasure concealment processing, i.e. repetition and smoothing processing, on the next good frame is performed in response to the previous frame that is erased. This repetition and smoothing for next good frame has two types of concealment methods. One is repetition and smoothing method for the next good frame after an erased frame, and the other is repetition and smoothing method for the next good frame after burst erasures.

The pseudo code to select a PLC mode for the Repetition and Smoothing tool and the conventional OLA is as follows.

```
if(BFI == 0 && st->prev_ BFI == 1)    {
      if((stat_mode_out==1) || (diff_energy<0.032209) ) {
    Repetition &smoothing for next good frame ();
      }
      else if(nbLostCmpt > 1)  {
    Next good frame after burst erasures ();
      }
      else {
           Conventional OLA ();
```

```
        }
    }
    else {  /* if(BFI == 1) */
        if( (stat_mode_out==1) || (diff_energy<0.032209) )  {
            if(Repetition &smoothing for erased frame () )  {
                Conventional OLA ();
            }
        }
        else {
               Conventional OLA ();
        }
    }
```

5.4.3.7.2 Phase matching

Figure 8 is a block diagram of the phase matching PLC module. The phase matching tool includes a PLC mode selection block and three phase matching packet loss concealment blocks.

Basically the phase matching error concealment performs phase matching packet loss concealment processing on a current erased frame when the previous good frame has the maximum energy in a predetermined low frequency band and the change in energy is less than a predetermined threshold.

The phase matching tool does not use the conventional OLA block but generates the time domain signal for the current erased frame by copying the phase-matched time domain signal obtained from the previous good frames. Once the phase matching tool is used for an erased frame, the tool shall also be used for the next good frame or subsequent burst erasures. For the next good frame, the phase matching for next good frame tool is used. For subsequent burst erasures, the phase matching tool for burst erasures is used.

The phase matching tool for the next good frame performs phase matching packet loss concealment processing on the current frame when the previous frame is an erasure and when phase matching error concealment processing on the previous frame has been performed.

The phase matching function for burst erasures performs phase matching packet loss concealment processing on the current frame that is part of a burst erasure when the previous frame is an erasure and phase matching error concealment processing on the previous frame has been performed.

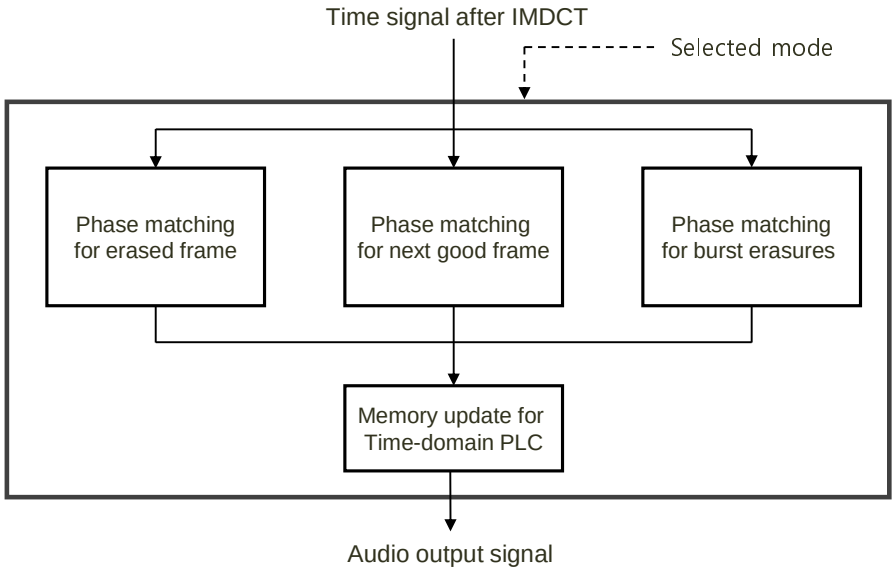

Figure 8: Block diagram of a phase matching PLC module

Figure 9 shows a block diagram of the phase matching for erased frame block in Figure 8. In order to use the phase matching tool, the phase_mat_flag shall be set to 1. Even though this condition is satisfied, a second condition shall be satisfied. As a second condition, a correlation scale accA is obtained, and either phase matching erasure concealment processing or general OLA processing is selected. The selection depends on whether the correlation scale accA is within a predetermined range. That is, phase matching packet loss concealment processing is conditionally performed depending on whether a correlation between segments exists in a search range and a cross-correlation between a search segment and the segments exists in the search range. The correlation scale $accA$ is given by Equation (201).

$$accA = \min\left(\frac{R_{xy}[d]}{R_{yy}[d]}\right),\ d = 0,...,D \tag{201}$$

In Equation (201), d denotes the number of segments existing in the search range, R_{xy} denotes a cross-correlation used to search for the matching segment having the same length as the search segment (x signal) with respect to the N past good frames (y signal) stored in the buffer, and R_{yy} denotes a correlation between segments existing in the N past good frames stored in the buffer. Next, it is be determined whether the correlation scale $accA$ is within the predetermined range. If this is the case phase matching erasure concealment processing takes place on the current erased frame. Otherwise, the conventional OLA processing on the current frame is performed. If the correlation scale $accA$ is less than 0.5 or greater than 1.5, the conventional OLA processing is performed. Otherwise, phase matching erasure concealment processing is performed.

The phase matching packet loss concealment processing includes a maximum correlation search block, a copying block, a smoothing block, and a memory update block. The maximum correlation search block searches for a matching segment, which has the maximum correlation to, i.e. is most similar to, a search segment adjacent to a current frame, from a decoded signal in a previous good frame from among N past good frames stored in a buffer. A position index of the matching segment obtained as a result of the search is provided to the copying block.

The copying block copies a predetermined duration starting from an end of the matching segment to the current frame that is an erasure frame by referring to the location index of the matching segment. At this time, a duration corresponding to a window length is copied to the current frame. When the copy starting from the end of the matching segment is shorter than the window length, the copy, starting from the end of the matching segment will be repeatedly copied into the current frame.

The smoothing block generates a time domain signal on the -concealed current frame by performing smoothing processing through OLA to minimize the discontinuity between the current frame and adjacent frames. After smoothing, the memory update for the phase matching will be performed in the memory update block.

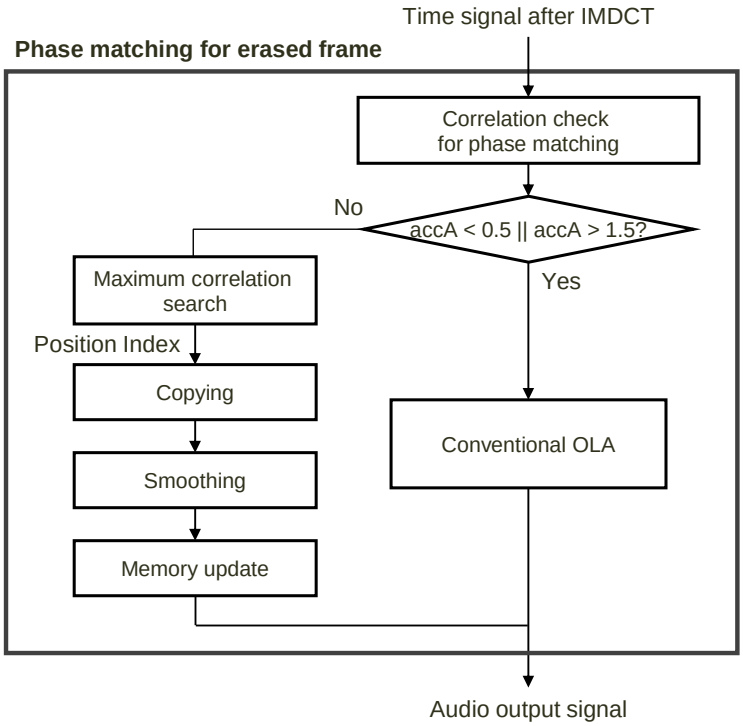

Figure 9: Block diagram of the phase matching for erased frame

Figure 10 illustrates the operation of phase matching erasure concealment described in Figure 9. Referring to Figure 10, the decoded signal from a previous frame from among the N past good frames stored in a buffer is searched for a matching segment. When the copy process is completed, the overlapping process on a copied signal and on an Oldauout signal stored in the previous frame n-1 for overlapping is performed at the beginning part of the current frame n by a first overlap duration. The length of the overlap duration is 2 ms. This results in the generation of the final repeated signal.

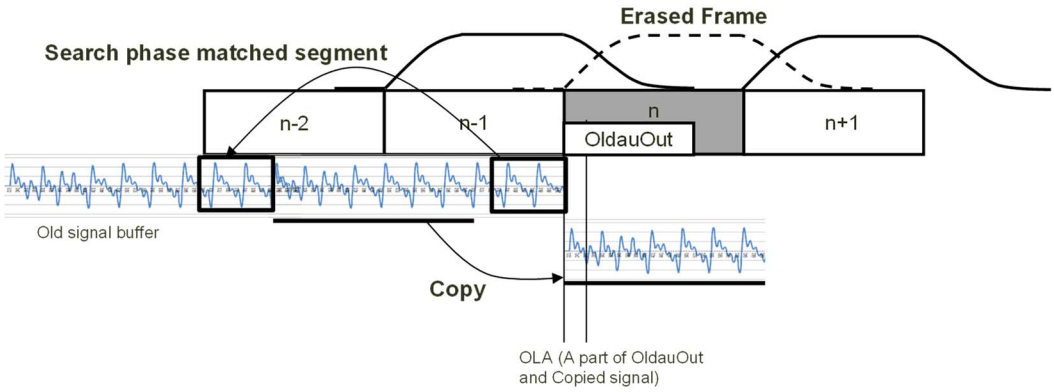

Figure 10: The operation of phase matching erasure concealment

Phase matching for burst erasures as shown in Figure 8 is described as follows. This method utilizes a smoothing process similar to that of phase matching for the first erased frame. Phase matching for burst erasures does not have maximum correlation search block nor the copying block, as all information needed for these blocks can be reused by phase matching for the erased frame. The only difference for the smoothing block is the smoothing that is done between the signal corresponding to the overlap duration of the copied signal and the Oldauout signal stored in the current frame n for overlapping purposes. The Oldauout is actually a copied signal by the phase matching process in the previous frame.

The phase matching for next good frame in Figure 8 is described as follows.

This method utilizes the mean_en_high parameter, denoting a mean energy of high bands and indicating the similarity of the last good frames. This parameter is calculated by following equation,

$$mean\_en\_high = \frac{\sum_{n=k}^{N_{sb}-1}\left(\frac{0.5norm_{k-1}(n)+0.5norm_{k-2}(n)}{norm_k(n)}\right)}{N_{sb}-k} \quad (202)$$

where k is start band index of the determined high bands.

If $mean\_en\_high$ is larger than 2.0 or smaller than 0.5, oldout_pha_idx is set to 1. oldout_pha_idx is used as a switch using the Oldauout memory. The two sets of Oldauout were saved at the both the phase matching for erased frame block and the phase matching for burst erasures block. The 1st Oldauout is generated from a copied signal by a phase matching process, and the 2nd Oldauout is generated by the time domain signal resulting from the IMDCT. If the oldout_pha_idx is set to 1, it indicates that the high band signal is unstable and the 2nd Oldauout will be used for the OLA process in the next good frame. If the oldout_pha_idx is set to 0, it indicates that the high band signal is stable and the 1st Oldauout will be used for OLA process in the next good frame.

```
if((mean_en_high>2.0)||(mean_en_high<0.5))  {
    oldout_pha_idx = 1;
}
else  {
    oldout_pha_idx = 0;
}
```

5.4.3.7.3 Repetition and smoothing

Figure 11 depicts the repetition and smoothing tool (OLA modes for time-domain PLC).

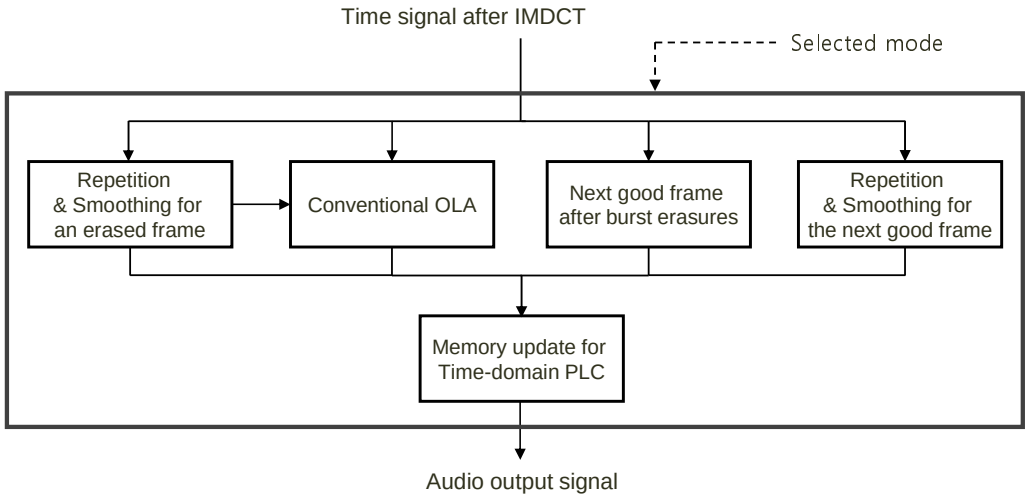

Figure 11: Repetition and smoothing

Each tool in the block diagram is described as follows. Figure 12 is a block diagram of conventional OLA method. The conventional OLA method includes a windowing block and an OLA block. Referring to Figure 12, the windowing block performs a windowing process on an IMDCT signal of the current frame to remove time domain aliasing. The case of a window having an overlap duration less than 50% will be described below with reference to Figure 13. The OLA block performs OLA processing on the windowed IMDCT signal.

Figure 13 illustrates the general OLA method with the window format for concealing an erased frame. When an erasure occurs in frequency domain encoding, past spectral coefficients are usually repeated, and thus, it may be impossible to remove time domain aliasing in the erased frame.

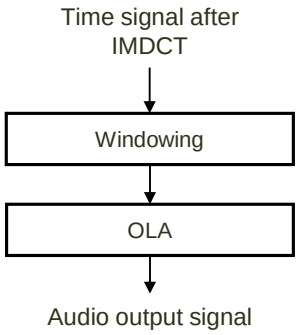

Figure 12: Block diagram of conventional OLA

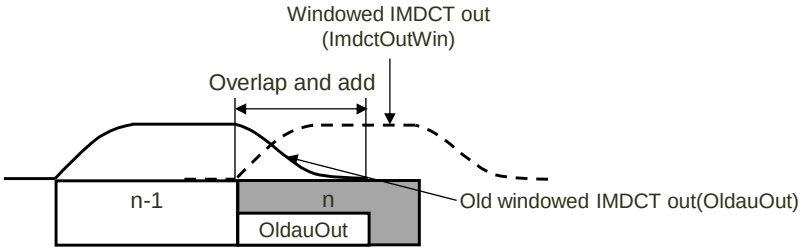

Figure 13: Diagram for describing a windowing of conventional OLA

Figure 14 is a block diagram of the repetition and smoothing method for an erased frame. When the current frame is an erasure, and if a method of repeating past spectral coefficients obtained in the frequency domain is used, and if OLA processing is performed after IMDCT and windowing, a time domain aliasing component in the beginning part of the current frame is modified. Thus perfect reconstruction is not possible, thereby resulting in unexpected noise. The repetition and smoothing method is used to minimize the occurrence of noise even though the original repetition method is used.

The repetition and smoothing method includes a windowing block, a repetition block, and a smoothing block. Referring to Figure 14, the windowing block performs the same operation as that of the windowing block of Figure 12. The repetition block applies an IMDCT signal of a frame that is two frames previous to the current frame (referred to as "previous old" in figure 15) to a beginning part of the current erased frame. The smoothing block consists of the OLA unit and the smoothing unit. The OLA unit performs OLA processing on the signal repeated by the repetition block and the IMDCT signal of the current frame. As a result, the audio output signal of the current frame is generated, and the occurrence of noise in a beginning part of the audio output signal is reduced. When scaling is applied together with the repetition of the spectrum of the previous frame in the frequency domain, the likelihood of noise occurring in the beginning part of the current frame is greatly reduced. The smoothing unit applies a smoothing window between the signal of the previous frame (old audio output) and the signal of the current frame (referred to as "current audio output") and performs OLA processing. The smoothing window is formed such that the sum of overlap durations between adjacent windows is equal to one. In the EVS codec, the sine wave window is used, and in this case, the window function $w(k)$ is represented by Equation (203).

$$w(k) = \sin^2\left(\frac{\pi k}{2 * OV\_SIZE}\right), n = 0,...,OV\_SIZE - 1 \quad (203)$$

In Equation (203), OV_SIZE denotes the duration of the overlap to be used in the smoothing processing. By performing smoothing processing as described above, when the current frame is an erasure, the discontinuity between the previous frame and the current frame, which may occur by using an IMDCT signal copied from the frame that is two frames previous to the current frame instead of an IMDCT signal stored in the previous frame, is prevented.

After completion of the repetition and smoothing, the energy $Pow1$ of an overlapping region is compared with the energy $Pow2$ of a non-overlapping region. When the energy of the overlapping region decreases after the packet loss concealment processing, conventional OLA processing is performed. The comparison is made by the operation depicted in Figure 14.

If the energy difference between the overlapping region ($Pow1$) and the non-overlapping region ($Pow2$) is large as a result of the comparison in the block, conventional OLA processing is performed.

Figure 15 illustrates the repetition and smoothing method with an example window for concealing an erased frame.

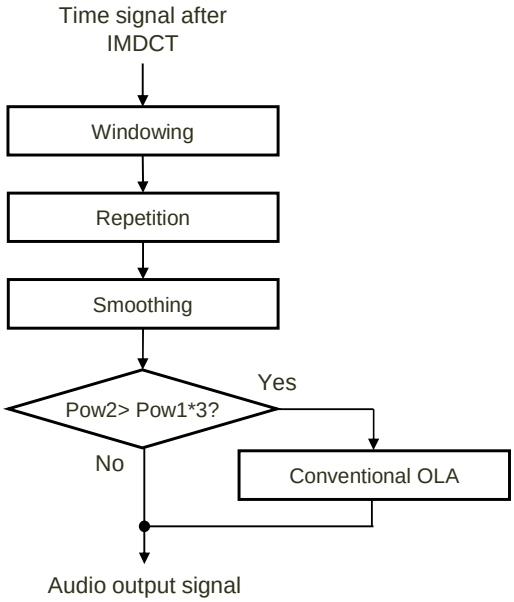

Figure14: Block diagram of repetition & smoothing method for erased frame

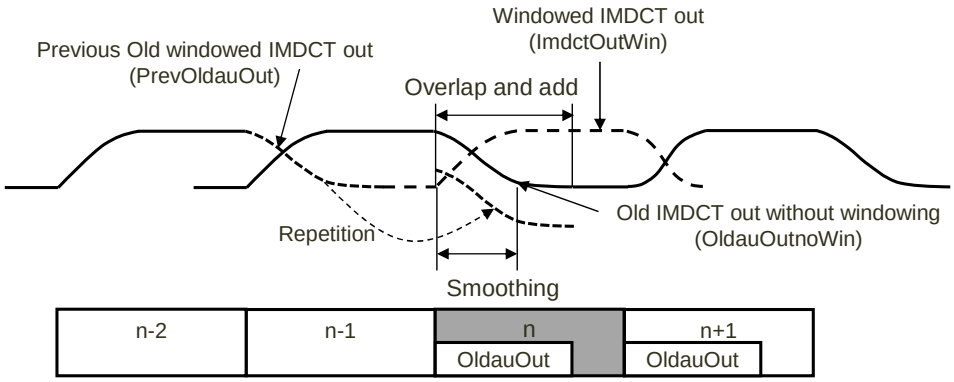

Figure 15: Diagram for describing a windowing of repetition & smoothing method for erased frame

Figure 16 is a block diagram of the repetition and smoothing method for the next good frame after an erased frame. This method only includes the smoothing block. The smoothing block applies the smoothing window to the old IMDCT signal and to a current IMDCT signal and performs OLA processing. Likewise, the smoothing window is formed such that a sum of overlap durations between adjacent windows is equal to one. That is, when the previous frame is a first erased frame and a current frame is a good frame, it is difficult to remove time domain aliasing in the overlap duration between an IMDCT signal of the previous frame and an IMDCT signal of the current frame. Thus, noise can be minimized by performing the smoothing processing based on the smoothing window instead of the conventional OLA processing. Figure 17 illustrates the repetition and smoothing method with an example of a window for smoothing the next good frame after an erased frame.

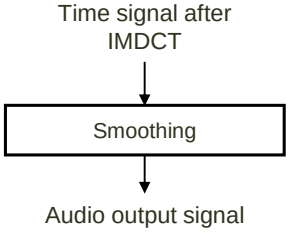

Figure 16: Block diagram of repetition and smoothing method for the next good frame after an erased frame

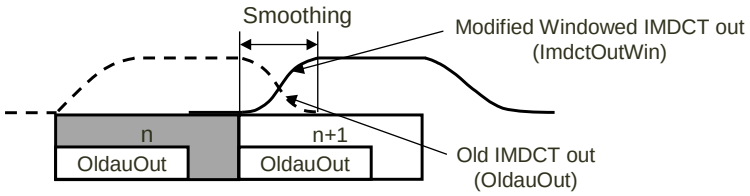

Figure 17: Diagram for describing a windowing of repetition and smoothing method for the next good frame after an erased frame

If the input signal is stationary and the previous frame is a burst erasure frame, then the repetition and smoothing method for the next good frame after the multiple erased frames as depicted in Figure 18 is used.

This method includes a repetition block, a scaling block, a first smoothing block, and a second smoothing block. Referring to Figure18, the repetition block copies, to a beginning part of the current frame, a part used for the next frame of the IMDCT signal of the current frame. The scaling block adjusts the scale of the current frame to prevent a sudden signal increase. In the EVS codec, the scaling block performs down-scaling by 3 dB. The first smoothing block applies a smoothing window to the IMDCT signal of the previous frame and the copied IMDCT signal from a future frame and performs OLA processing. Likewise, the smoothing window is formed such that a sum of overlap durations between adjacent windows is equal to one. That is, when the copied signal is used, windowing is necessary to remove the discontinuity which may occur between the previous frame and the current frame, and an old IMDCT signal may be replaced with a signal obtained by OLA processing of the first smoothing block. The second smoothing block performs the OLA processing while removing the discontinuity by applying a smoothing window between the old IMDCT signal that is a replaced signal and a current IMDCT signal that is the current frame signal. Likewise, the smoothing window is formed such that the sum of overlap durations between adjacent windows is equal to one. That is, when the previous frame is a burst erasure and the current frame is a good frame, time domain aliasing in the overlap duration between the IMDCT signal of the previous frame and the IMDCT signal of the current frame cannot be removed. In the burst erasure frame, since noise may occur due to a decrease in energy or continuous repetitions, the method of copying a signal from the future frame for overlapping with the current frame is applied. In this case, smoothing processing is performed twice to remove the noise which may occur in the current frame and simultaneously remove the discontinuity which occurs between the previous frame and the current frame. Figure 19 illustrates the repetition and smoothing method with an example window for smoothing the next good frame after burst erasures.

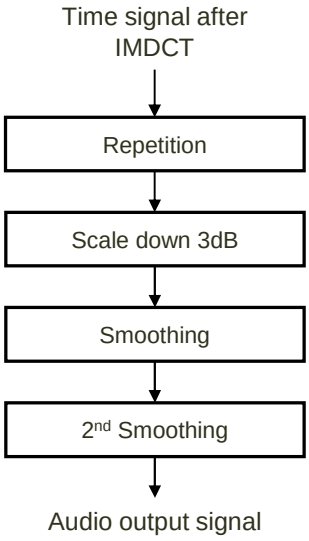

Figure 18: Block diagram of repetition and smoothing method for the next good frame after burst erasures

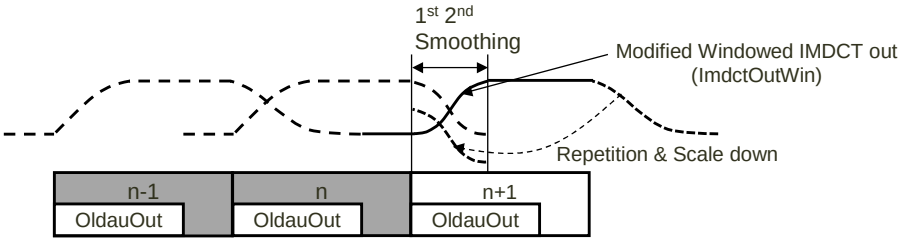

Figure 19: Diagram for describing a windowing of repetition and smoothing method for the next good frame after burst erasures

Figure 20 is the block diagram of the next good frame after burst erasures shown in Figure 11. Regarding the usage of the future signal the main operation is same as that of the repetition and smoothing method for the next good frame after burst erasures shown in Figure 18.

This method includes a repetition block, a scaling block, a smoothing block, and an OLA block. Referring to Figure 20, the repetition block, scaling block, and smoothing block are exactly the same as that of Figure 18. Instead of the second smoothing block, the next good frame after burst erasures uses the OLA between the replaced OldauOut signal and the current IMDCT signal. Figure 21 illustrates the next good frame after burst erasures.

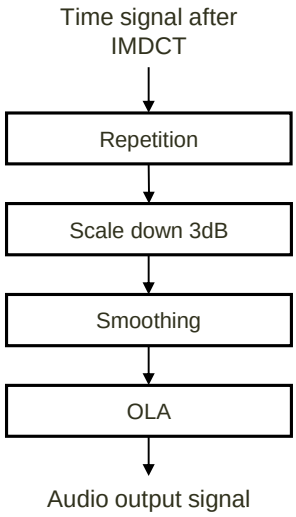

Figure 20: Block diagram of the next good frame after burst erasures

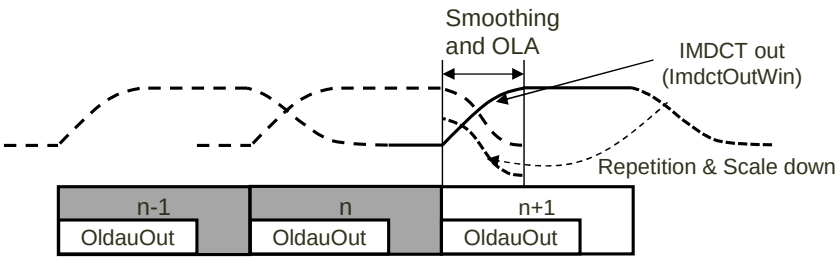

Figure 21: Diagram for describing a windowing of the next good frame after burst erasures

5.4.4 Void

5.4.5 Guided concealment and recovery

5.4.5.1 Transmission of the synthesis class

Instead of performing the signal classification in the decoder (see clause 5.1.2 Signal classification), the synthesis class is transmitted in the bitstream for the rates 48, 96 and 128 kbps.

5.4.5.2 Transmission of the LTP pitch lag

Despite of the fact that no LTP post processing is performed for the rates 96 and 128 kbps, the LTP pitch lag is transmitted in the bitstream to allow reliable functioning of the decoder concealment modules which depend on this LTP lag.

5.4.5.3 Transmission of a voicing indicator

A flag v is used in the decoder for the concealment method described in clause 5.4.3.6, to adapt several parameters (pitch search range, sinusoid selection, noise level to be re-injected).

At the encoder, the flag v is set to 1 when the current frame is classified as GENERIC or VOICED, otherwise the flag v is set to 0 (for all other signal classes: UNVOICED, TRANSIENT, INACTIVE, AUDIO).

5.4.5.3a Transmission of a tonality flag

A flag indicating the frame as tonal type (1) or non-tonal type (0) is transmitted in the bitsteam for the rates 48, 96 and 128 kbps. It is used in the decision criterion to select the concealment method of non-tonal concealment with waveform adjustment.

5.4.5.4 ACELP to MDCT mode recovery

For the ACELP concealment at 9.6, 16.4 and 24.4 kbps, as well as for the TCX time domain concealment, an additional segment (half frame) of signal is generated by predictive decoding from the previous frame and stored in a temporary buffer. This additional segment is used to recover from the loss of a transition frame between ACELP to MDCT (HQ MDCT or non-transition TCX 20) and it is generated in advance without prior knowledge that the transition frame will be lost, before receiving the next frame. The extra complexity associated with generating this additional segment has been found to be outside the critical path of complexity of the EVS decoder, therefore this extra processing does not impact worst-case decoding complexity.

To create this additional segment, the parameters used to generate the half frame of signal are predicted based on the parameters in the previous frame. The bit indicating the sampling frequency of the ACELP core is implicitly repeated; the excitation decoding done in the previous ACELP frame is extended in the additional segment.

When the current frame is MDCT, the additional segment (half frame) is then overlap and added to the MDCT frame decoded in the current frame, being HQ MDCT or non-transition TCX 20, using a symmetric sine window of length 8.75 milliseconds.

For the recovery in the TCX20 transition frame (TCX20 after lost ACELP frame):

- If the ACELP PLC was used, the same ACELP to TCX transition as if the previous ACELP frame would have been received is used, but with the samples of the past frame replaced with the concealed frame.
- If the TD TCX PLC was used, the additional half frame constructed in TD TCX PLC is overlap and added to the transition TCX frame, using HALF_OVERLAP (the symmetric sine window of length 3.75 milliseconds).

5.4.5.5 Recovery after TCX MDCT concealment

During recovery after TCX MDCT concealment fading the background level as described in 5.4.6.1.3.1, the overlap-add buffer is rescaled by multiplying each element of it with the latest target background noise level g^{cng} (see equation 109).

5.4.6 Handling of multiple frame losses and muting

5.4.6.1 TCX MDCT

5.4.6.1.1 Background level tracing for rates 48, 96 and 128 kbps

A background noise level is traced in the time domain using a simplified version of the minimum statistics algorithm [7]. The tracing depends on the class being transmitted in the bitstream: It is performed for UC only.

In contrast to the FD-CNG - which also makes use of the minimum statistics approach (see [5], subclause 4.4.3) - the noise level estimation is not carried for each spectral band separately, but directly in the time domain. The background level tracing delivers therefore an estimate of the total noise level. Furthermore, the bias compensation is disregarded in this application. Tracing of the noise level is hence achieved by computing a smoothed version of the decoder output frame amplitude and by searching for the minimum smoothed amplitude over a sliding temporal window.

If L_{TCX} denotes the frame size in samples, k denotes the sample index, λ denotes the frame index, and $s_{\mathrm{TCX}}(\lambda,k)$ is the output frame of the core decoder at the TCX sampling rate, the current total frame level is computed as follows:

$$A_{\mathrm{s}}(\lambda) = \sqrt{\frac{\sum_{k=0}^{L_{\mathrm{TCX}}-1}\left(s_{\mathrm{TCX}}(\lambda,k)\right)^2}{L_{\mathrm{TCX}}}}\,. \tag{204}$$

It is first lower-limited by 0.01 and smoothed with a first-order recursive low-pass process, i.e.

$$\overline{A}_{\mathrm{s}}(\lambda) = a_{\mathrm{opt}}(\lambda)\cdot\overline{A}_{\mathrm{s}}(\lambda-1) + (1-a_{\mathrm{opt}}(\lambda))\cdot\max(0.01, A_{\mathrm{s}}(\lambda))\,, \tag{205}$$

where

$$a_{\mathrm{opt}}(\lambda) = \frac{\min\left(A_{\mathrm{s}}(\lambda-1), A_{\mathrm{n}}(\lambda-1)\right)}{\max\left(A_{\mathrm{s}}(\lambda-1), A_{\mathrm{n}}(\lambda-1)\right)} \tag{206}$$

is an optimal smoothing parameter which depends on the signal level $A_{\mathrm{s}}(\lambda-1)$ and the background tracing level $A_{\mathrm{n}}(\lambda-1)$ in the previous frame. The tracing level $A_{\mathrm{n}}(\lambda)$ for the current frame λ is obtained by searching for the minimum in a buffer containing the last 50 values of the smoothed level $\overline{A}_{\mathrm{s}}(\lambda)$:

$$A_{\mathrm{n}}(\lambda) = \min\left(\overline{A}_{\mathrm{s}}(\lambda), \overline{A}_{\mathrm{s}}(\lambda-1),..., \overline{A}_{\mathrm{s}}(\lambda-49)\right). \tag{207}$$

At initialization, the buffer is filled with the value 0.01 and the smooth signal level is initialized as $\overline{A}_{\mathrm{s}}(0) = 0.01$.

5.4.6.1.2 TCX time domain concealment

In the case of TCX time domain concealment as stated in subclause 5.4.2.2, the following applies.

5.4.6.1.2.1 Fading to background level

At rates 9.6, 16.4 and 24.4kbps the fading is identical to what is described in subclause 5.3.4.2.1.

At rates 48, 96 and 128kbps the fading is identical to what is described in subclause 5.3.4.2.1 with the exception, that the target level in the time domain is not derived from the FFT provided by CNG, but that it is gained from the background level tracing as described in subclause 5.4.6.1.

5.4.6.1.2.2 Fading to background spectral shape

No fading of the LPC is applied.

5.4.6.1.3 MDCT frame repetition with sign scrambling

In the case of TCX frequency domain concealment, i.e. frame repetition with sign scrambling as stated in subclause 5.4.2.3 and/or tonal concealment using phase prediction as stated in subclause 5.4.2.4, the following applies.

5.4.6.1.3.1 Fading to background level

The time domain signal is faded towards a target background noise level as described in equation (107) and (107a) . The initial gain is 1. The derivation of *alpha* is outlined in subclause 5.4.6.1.4.

At rates 9.6, 16.4 and 24.4kbps the target level $\hat{g}^{cng}$ is derived during the first lost frame based on the background noise spectrum derived by CNG during clean channel decoding (section 4.3 of [5]) as stated in subclause 5.3.4.2.1 under a).

At rates 48, 96 and 128kbps the target level $\hat{g}^{cng}$ is gained from the background level tracing as described in subclause 5.4.6.4.

The gain compensation for the LPC synthesis / de-emphasis as given in equation (109) is applied, see also subsection 5.2.5.

5.4.6.1.3.2 Fading to background spectral shape

The fading to background spectral shape is achieved by the following fading procedures, taking place in parallel:

a) The excitation itself is faded towards white noise in the frequency domain prior to the FDNS, on which a tilt is applied.

b) The excitation is shaped by FDNS towards a previously measured background shape.

c) The LTP is faded out.

5.4.6.1.3.2.1 Fading the excitation to noise

For 9.6, 16.4 and 24.4kbps, the sign scrambled excitation (input to FDNS, see subclause 5.4.2.3) is faded towards a white noise, on which a tilt is applied prior to the fading procedure. The method is based on the following parameters: the last received excitation spectrum $C_{exc\_lastGood}(k)$, a noise tilt compensation factor *tiltCompFactor* (derived similar to the clean channel operation) and a damping factor *dampingFac* .

The tilt factor is given by

$$tiltFactor = \max(0.375, tiltCompFactor)^{\frac{1}{nSamples_{m-1}}} \tag{208}$$

Subsequently a tilt vector is derived as

$$\begin{aligned} &tilt(0) = 1 \\ &tilt(k) = tilt(k-1) \cdot tiltFactor, k = [1,...,igfStartLine-1] \end{aligned} \tag{209}$$

The *randomVector* given by equation (123) then gets multiplied with the tilt to achieve a target noise vector with the desired tilt:

$$C_{noise\_tilt}(k) = tilt(k) \cdot randomVector(k), \quad \text{k} = [0,\ldots,\text{igfStartLine} - 1] \tag{210}$$

The energy of this target noise vector is derived

$$E_{noise} = \sum_{k=0}^{igfStartLine-1} \left(C_{noise\_tilt}(k)\right)^2 \tag{211}$$

and the energy of the last excitation is derived

$$E_C = \sum_{k=0}^{igfStartLine-1} \left(C_{exc\_lastGood}(k)\right)^2 . \tag{212}$$

The excitation is then derived as follows:

$$C_{exc}^{[m]}(k) = gain \cdot C_{noise\_tilt}(k) + dampingFac \cdot \hat{C}_{exc}^{[m]}(k), \text{ for k = [0,…, igfStartLine -1]} \tag{213}$$

with $gain = \sqrt{\frac{E_C}{E_{noise}}} \cdot (1 - dampingFac)$ and $\hat{C}_{exc}^{[m]}$ is given by equation (122). The fading speed controlled by $dampingFac$ as described in subclause 5.4.6.1.4.

5.4.6.1.3.2.2 Shaping the excitation towards the background shape

The excitation is shaped towards a target spectral shape by altering the LPC coefficients. The fading from the last good LPC coefficients to the target LPC coefficients is performed in the LSF domain as follows:

$$f^{[m]} = alpha \cdot f^{[m-1]} + (1 - alpha) \cdot f^{\text{target}} \tag{214}$$

where: $f^{[m]}$ are LPC coefficients in the LSF domain of the current frame;

$f^{[m-1]}$ are LPC coefficients in the LSF domain of the previous frame;

f^{target} are the target LPC coefficients, derived according to formula 111

$alpha$ is the fading factor as described in subclause 5.3.4.2.3, but limited to the minimum value of 0.8.

void (215)

For 9.6, 16.4 and 24.4kbps, the target spectral shape of the excitation is derived during the first lost frame based on the background noise spectrum derived by CNG during clean channel decoding (see section 4.3 of [5]). Its derivation is performed as described in subclause 5.3.4.2.2 for the harmonic excitation.

For 48, 96 and 128kbps, the target spectral shape of the excitation is the short term mean of the last three LPC coefficient sets. Its derivation is performed as described in subclause 5.3.4.2.2 for the innovative excitation.

The achieved LPC is converted into FDNS parameters as follows:

$$re_k = a_k \cdot \cos\left(\frac{k\pi}{128}\right), \qquad for\ k = 0,\ldots,16 \tag{216}$$

$$im_k = -a_k \cdot \sin\left(\frac{k\pi}{128}\right), \qquad for\ k = 0,\ldots,16 \tag{217}$$

where a_k are the LPC coefficients. The two signals re and im get zero filled to the length of 128 before a complex Fourier transform of length 128 will be applied on them to receive the real part Re and the imaginary part Im (see [5], subsection 5.1.4). The FDNS parameters will finally be obtained as:

$$FDNS_k = \sqrt{\mathrm{Re}_k{}^2 + \mathrm{Im}_k{}^2}, \qquad for\ k = 0,\ldots,63 \tag{218}$$

5.4.6.1.3.2.3 LTP fade-out

The LTP continues to run during concealment. The LTP lag is kept constant. The LTP gain is faded towards zero as follows:

$$g_{ltp}^{[m]} = g_{ltp}^{[m-1]} \cdot dampingFac \tag{219}$$

where: $g_{ltp}^{[m]}$ is the LTP gain of the current frame;

$g_{ltp}^{[m-1]}$ is the LTP gain of the previous frame;

$dampingFac$ is the damping factor, its derivation is outlined in subclause 5.4.6.1.4.

5.4.6.1.4 Fading speed

Several algorithms use a time-varying damping factor for fade-out, cross-fade etc. Depending on the application, either the damping factor or the cumulative damping factor is needed.

The damping factor, here described as $dampingFac$, depends on the number of lost frames and the ISF stability factor. The ISF stability factor is already computed in the clean channel. With the lost frame having the index 0, it is derived as follows

$$dampingFac^{[0,1]} = 0.7 + 0.3 \cdot stabFac \tag{220}$$

$$dampingFac^{[2]} = 0.45 + 0.4 \cdot stabFac \tag{221}$$

$$dampingFac^{[3,...]} = 0.35 + 0.4 \cdot stabFac \tag{222}$$

The cumulative damping factor, here described as $dampingFac\ \ Cum$, is initialized with 1 during clean-channel decoding and derived as follows during concealment:

$$dampingFac\ \ Cum^{[m]} = dampingFac\ \ Cum^{[m-1]} \cdot dampingFac^{[m]} \tag{223}$$

5.4.6.1.5 Waveform adjustment

The fade out is performed as described in section 5.4.6.1.3, just that no lpc gain compensation (see section 5.2.5) takes place.

5.4.6.2 HQ MDCT

5.4.6.2.1 Burst loss handling for 8 kHz audio output sampling rate

The burst loss handling for 8 kHz audio sampling rate is described as part of the HQ MDCT PLC method description for 8 kHz signals, see clauses 5.4.3.3 and 5.4.3.4.

5.4.6.2.2 Burst loss handling audio output sampling rates larger or equal to 16 kHz

In case the audio output signal frequency exceeds 8 kHz and the current frame loss is the first loss after a good HQ MDCT frame the PLC method is selected according to the method described in subclause 5.4.3.2. If however the current frame loss is at least the second consecutive loss after a preceding good HQ MDCT frame, then the procedure described in this clause applies.

In case the current frame loss is the second loss in a row and the PLC method according to subclause 5.4.3.6 was applied for the first bad frame, Phase ECU according to subclause 5.4.3.5 is applied with the following adaptations: Transient analysis and spectrum analysis are carried out with the previous synthesis signal of the last good HQ MDCT frame. The offset k_{offs} in number of samples since the last good frame is accordingly incremented by L.

Otherwise, in case the current frame loss is the second loss in a row and if Phase ECU was applied for the first frame loss, Phase ECU according to subclause 5.4.3.5 is applied with the adaptation that no spectral analysis is carried out and that transient analysis relies on previously calculated and stored parameters. Details are described in subclause 5.4.3.5.1.

In case the current frame loss is the third or more in a row Phase ECU is applied according to subclause 5.4.3.5 with the adaptation that no spectral analysis is carried out and that transient analysis relies on previously calculated and stored parameters (that were calculated based on the synthesis of the last good HQ MDCT frame). The operation of the Phase ECU is modified in response to the frame loss burst condition. Specifically, magnitude and phase of the substitution frame spectrum are adjusted in order to mitigate potential quality losses that might otherwise arise from too periodic or tonal sounds. With increasing loss burst length, the magnitude spectrum is adjusted by gradually increasing attenuation. At the same time the phase spectrum is dithered with an increasing degree. Further details are described in subclause 5.4.3.5.1.

A special feature is the long-term muting behaviour in case of long loss bursts with many consecutive lost frames. In that case, the quality of the audio signal that is reconstructed by Phase ECU might still suffer from tonal artefacts, despite the performed phase randomization. Too strong magnitude attenuation could at the same time lead to quality impairments, as this could be perceived as signal drop-outs. The feature avoids such impairments to a large degree by gradually superposing the substitution signal of the Phase ECU with a noise signal, where the frequency characteristic of the noise signal is a low-resolution spectral representation of a previously received good frame. With increasing number of frame losses in a row, the substitution signal of the Phase ECU is gradually attenuated. At the same time, the frame energy loss is compensated for through the addition of a noise signal with similar spectral characteristics like the last received good frame but with a certain degree of low-pass behaviour. For very long frame loss bursts ($N_{lost} > 10$) the additional noise contribution faded out in order to enforce a muting characteristic of the decoder. Further details of the long-term muting feature are described in subclauses 5.4.3.5.1 and 5.4.3.5.3.

5.5 SID frame concealment operation

In the case of the loss of an SID frame, the comfort noise will be generated based on the last received SID frame.

Annex A (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2014-09	SA#65	SP-140462				Presented at TSG SA#65 for approval	1.0.0
2014-09	SA#65					Approved at TSG SA#65	12.0.0
2014-12	SA#66	SP-140727	0001	4		Corrections to the description of the packet loss concealment algorithm	12.1.0
2015-03	SA#67	SP-150087	0002	-		Corrections to the description of the packet loss concealment algorithm	12.2.0
2015-06	SA#68	SP-150204	0003	-		Corrections to the description of the packet loss concealment algorithm	12.3.0
2015-09	SA#69	SP-150434	0004	1		Corrections to the Algorithmic Description	12.4.0
2015-12	SA#70	SP-150639	0005	2		Corrections to the Algorithmic Description	12.5.0
2015-12	SA#70					Version for Release 13	13.0.0
2016-06	SA#72	SP-160257	0007	-	A	Corrections to the Algorithmic Description	13.1.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-03	SA#75					Version for Release 14	14.0.0
2017-06	SA#76	SP-170316	0010	-	A	Corrections to the Algorithmic Description	14.1.0
2018-06	SA#80					Version for Release 15	15.0.0
2019-03	SA#83	SP-190036	0011	1	B	Correction and addition of reference to Alt_FX_EVS implementation	16.0.0
2020-06	SA#88-e	SP-200386	0017	0	A	Corrections to the Algorithmic Description	16.1.0

History

Document history		
V16.1.0	September 2020	Publication