

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.722

Appendix III
(11/2006)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,
DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – Coding of analogue signals
by methods other than PCM

7 kHz audio-coding within 64 kbit/s

**Appendix III: A high-quality packet loss
concealment algorithm for G.722**

ITU-T Recommendation G.722 – Appendix III



ITU-T G-SERIES RECOMMENDATIONS
TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
General	G.700–G.709
Coding of analogue signals by pulse code modulation	G.710–G.719
Coding of analogue signals by methods other than PCM	G.720–G.729
Principal characteristics of primary multiplex equipment	G.730–G.739
Principal characteristics of second order multiplex equipment	G.740–G.749
Principal characteristics of higher order multiplex equipment	G.750–G.759
Principal characteristics of transcoder and digital multiplication equipment	G.760–G.769
Operations, administration and maintenance features of transmission equipment	G.770–G.779
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.780–G.789
Other terminal equipment	G.790–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation G.722

7 kHz audio-coding within 64 kbit/s

Appendix III

A high-quality packet loss concealment algorithm for G.722

Summary

Appendix III to ITU-T Recommendation G.722 specifies a high-quality packet loss concealment (PLC) algorithm for G.722. The algorithm performs the packet loss concealment in the 16-kHz output domain of the G.722 decoder. Periodic waveform extrapolation is used to fill in the waveform of lost packets, mixing with filtered noise according to signal characteristics prior to the loss. The extrapolated 16-kHz signal is passed through the QMF analysis filter bank, and the subband signals are passed to partial subband ADPCM encoders to update the states of the subband ADPCM decoders. Additional processing takes place for each packet loss in order to provide a smooth transition from the extrapolated waveform to the waveform decoded from the received packets. Among other things, the states of the subband ADPCM decoders are phase aligned with the first received packet after a packet loss, and the decoded waveform is time-warped in order to align with the extrapolated waveform before the two are overlap-added to smooth the transition. For protracted packet loss, the algorithm gradually mutes the output.

The algorithm operates on an intrinsic 10-ms frame size. It can operate on any packet or frame size that is a multiple of 10 ms. The longer input frame becomes a super frame, for which the packet loss concealment is called an appropriate number of times at its intrinsic frame size of 10 ms. It results in no additional delay when compared with regular G.722 decoding using the same frame size.

The PLC algorithm described in this appendix meets the same complexity requirements as the PLC in G.722 Appendix IV. At an additional complexity of 2.8 WMOPS worst-case and 2 WMOPS average compared with the G.722 decoder without PLC, the G.722 PLC algorithm described in this appendix provides significantly better speech quality than the G.722 PLC specified in G.722 Appendix IV, which provides an alternative quality-complexity trade-off.

Source

Appendix III to ITU-T Recommendation G.722 was agreed on 24 November 2006 by ITU-T Study Group 16 (2005-2008).

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	Page
Appendix III – A high-quality packet loss concealment algorithm for G.722	1
III.1 Scope	1
III.2 References	1
III.3 Abbreviations	1
III.4 Conventions	2
III.5 General description of the PLC algorithm.....	4
III.6 WB PCM PLC – Waveform extrapolation of G.722 output	8
III.7 Re-encoding of PLC output.....	20
III.8 Monitoring signal characteristics and their use for PLC	24
III.9 Time lag computation.....	27
III.10 Re-phasing.....	30
III.11 Time-warping	31
III.12 Bit-exact description of the G.722 PLC algorithm.....	34

ITU-T Recommendation G.722

7 kHz audio-coding within 64 kbit/s

Appendix III

A high-quality packet loss concealment algorithm for G.722

III.1 Scope

This appendix describes a high-quality packet loss concealment algorithm for G.722. The statistical analysis of the G.722 PLC Selection Test results has demonstrated that the PLC algorithm in this appendix was clearly the best performing PLC among the solutions examined (including the PLC in G.722 Appendix IV) in terms of speech quality for applications of G.722 in the presence of packet loss. This appendix meets the same complexity requirements as Appendix IV, but with a higher complexity. Due to its high quality, this appendix is suitable for general applications of G.722 that may encounter frame erasures or packet loss. As examples, these applications include VoIP, Voice over WiFi, and DECT Next Generation. The algorithm of this appendix adds a complexity of 2.8 WMOPS worst case and 2 WMOPS on average to the G.722 decoder. It is easy to accommodate, except for applications where there is practically no complexity headroom left after implementing the basic G.722 decoder without packet loss concealment.

III.2 References

- [ITU-T G.191 An.A] ITU-T Recommendation G.191 Annex A (2005), *Software tools for speech and audio coding standardization, Annex A: List of software tools available.*
- [ITU-T G.192] ITU-T Recommendation G.192 (1996), *A common digital parallel interface for speech standardization activities.*
- [ITU-T G.722] ITU-T Recommendation G.722 (1988), *7 kHz audio-coding within 64 kbit/s.*

III.3 Abbreviations

This appendix uses the following abbreviations:

ADPCM	Adaptive Differential PCM
DECT	Digital Enhanced Cordless Telecommunications
FIR	Finite Impulse Response
LPC	Linear Predictive Coding
OLA	Overlap-Add
PCM	Pulse Code Modulation
PLC	Packet Loss Concealment
PWE	Periodic Waveform Extrapolation
QMF	Quadrature Mirror Filter
STL2005	Software Tool Library 2005
VoIP	Voice over Internet Protocol
WB	Wideband
WiFi	Wireless Fidelity

III.4 Conventions

For the purposes of this appendix, the terms "frame loss" and "packet loss" are used interchangeably.

This appendix uses the following conventions:

- The PLC operates at an intrinsic frame size of 10 ms, and hence, the algorithm is described for 10-ms frames only. For packets of larger size (multiples of 10 ms) the received packet is decoded in 10-ms sections.
- The discrete time index of signals at the 16-kHz sampling rate level is generally referred to with either "j" or "i".
- The discrete time of signals at the 8-kHz sampling level is typically referred to with "n".
- Low-band signals (0-4 kHz) are identified with a subscript "L".
- High-band signals (4-8 kHz) are identified with a subscript "H".
- This appendix follows the conventions of [ITU-T G.722] where possible.

The following is a list of the most frequently used symbols and their descriptions:

$x_{out}(j)$	16-kHz G.722 decoder output
$x_{PLC}(i)$	16-kHz G.722 PLC output
$x_{out, FGF}(j)$	16-kHz G.722 decoder output for the 'first good frame' received after packet loss
$w(j)$	LPC window
$x_w(j)$	Windowed speech
$r(i)$	Autocorrelation
$\hat{r}(i)$	Autocorrelation after spectral smoothing and white noise correction
\hat{a}_i	Intermediate LPC predictor coefficients
a_i	LPC predictor coefficients
$d(j)$	16-kHz short-term prediction error signal
avm	Average magnitude of the short-term prediction residual signal
a'_i	Weighted short-term synthesis filter coefficients
$xw(j)$	16-kHz weighted speech
$xwd(n)$	Down-sampled weighted speech (2 kHz)
b_i	60th-order low-pass filter for down-sampling
$c(k)$	Correlation for coarse pitch analysis (2 kHz)
$E(k)$	Energy for coarse pitch analysis (2 kHz)
$c2(k)$	Signed squared correlation for coarse pitch analysis (2 kHz)
cpp	Coarse pitch period
$cpplast$	Coarse pitch period of last frame
$Ei(j)$	Interpolated E(k) (to 16 kHz)
$c2i(j)$	Interpolated c2(k) (to 16 kHz)
$\tilde{E}(k)$	Energy for pitch refinement (16 kHz)
$\tilde{c}(k)$	Correlation for pitch refinement (16 kHz)

$ppfe$	Pitch period for frame erasure
$ptfe$	Pitch tap for frame erasure
ppt	Pitch predictor tap
$merit$	Figure of merit of periodicity
G_r	Scaling factor for random component
G_p	Scaling factor for periodic component
$ltring(j)$	Long-term (pitch) ringing
$ring(j)$	Final ringing (including short-term)
$wi(j)$	Fade-in window
$wo(j)$	Fade-out window
$wn(j)$	Output of noise generator
$wgn(j)$	Scaled output of noise generator
$fn(j)$	Filtered and scaled noise
$cfecount$	Counter of consecutive 10-ms frame erasures
$w_i(j)$	Window for overlap-add
$w_o(j)$	Window for overlap-add
h_i	QMF filter coefficients
$x_L(n)$	Low-band subband signal (8 kHz)
$x_H(n)$	High-band subband signal (8 kHz)
$I_L(n)$	Index for low-band ADPCM coder (8 kHz)
$I_H(n)$	Index for high-band ADPCM coder (8 kHz)
$s_{Lz}(n)$	Low-band predicted signal, zero section contribution
$s_{Lp}(n)$	Low-band predicted signal, pole section contribution
$s_L(n)$	Low-band predicted signal
$e_L(n)$	Low-band prediction error signal
$r_L(n)$	Low-band reconstructed signal
$p_{Ll}(n)$	Low-band partially reconstructed truncated signal
$\nabla_L(n)$	Low-band log scale factor
$\Delta_L(n)$	Low-band scale factor
$\nabla_{L,m1}(n)$	Low-band log scale factor, 1st mean
$\nabla_{L,m2}(n)$	Low-band log scale factor, 2nd mean
$\nabla_{L,track}(n)$	Low-band log scale factor, tracking
$\nabla_{L,chg}(n)$	Low-band log scale factor, degree of change
$MPTH$	Multiple pitch threshold
$MPDTH$	Multiple pitch deviation threshold
$\beta_L(n)$	Stability margin of low-band pole section

$\beta_{L,MA}(n)$	Moving average of stability margin of low-band pole section
$\beta_{L,min}$	Minimum stability margin of low-band pole section
$s_{Hz}(n)$	High-band predicted signal, zero section contribution
$s_{Hp}(n)$	High-band predicted signal, pole section contribution
$s_H(n)$	High-band predicted signal
$e_H(n)$	High-band prediction error signal
$r_H(n)$	High-band reconstructed signal
$r_{H,HP}(n)$	High-band high-pass filtered reconstructed signal
$p_H(n)$	High-band partially reconstructed signal
$p_{H,HP}(n)$	High-band high-pass filtered partially reconstructed signal
$\nabla_H(n)$	High-band log scale factor
$\nabla_{H,m}(n)$	High-band log scale factor, mean
$\nabla_{H,track}(n)$	High-band log scale factor, tracking
$\nabla_{H,chg}(n)$	High-band log scale factor, degree of change
$\alpha_{LP}(n)$	Coefficient for low-pass filtering of high-band log scale factor
$\nabla_{H,LP}(n)$	Low-pass filtered high-band log scale factor
$r_{Le}(n)$	Estimated low-band reconstructed error signal
$es(n)$	Extrapolated signal for time lag calculation of re-phasing
$R_{SUB}(k)$	Sub-sampled normalized cross-correlation
$R(k)$	Normalized cross-correlation
T_{LSUB}	Sub-sampled time lag
T_L	Time lag for re-phasing
$es_{tw}(n)$	Extrapolated signal for time lag refinement for time-warping
T_{Lwarp}	Time lag for time-warping
$X_{warp}(j)$	Time-warped signal (16 kHz)
$es_{ola}(j)$	Extrapolated signal for overlap-add (16 kHz)

III.5 General description of the PLC algorithm

For ease of understanding, six types of frames are defined and referred to in the text:

- Type 1: Received frame, with no lost packets in the preceding eight frames
- Type 2: Lost frames 1 and 2 of packet loss
- Type 3: Lost frames 3 through 6 of packet loss
- Type 4: Lost frames beyond frame 6 of packet loss
- Type 5: First frame received immediately following packet loss
- Type 6: Received frames 2 through 8 following packet loss

This is illustrated with the time-line example in Figure III.1. The PLC algorithm operates on an intrinsic frame size of 10 ms in duration.

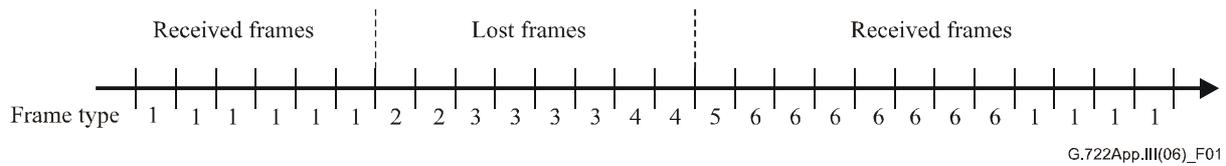


Figure III.1 – Time-line of frame types

Type 1

Type 1 frames are decoded according to [ITU-T G.722] with the addition of maintaining some state memory and processing to facilitate the PLC and associated processing. This is shown in Figure III.2.

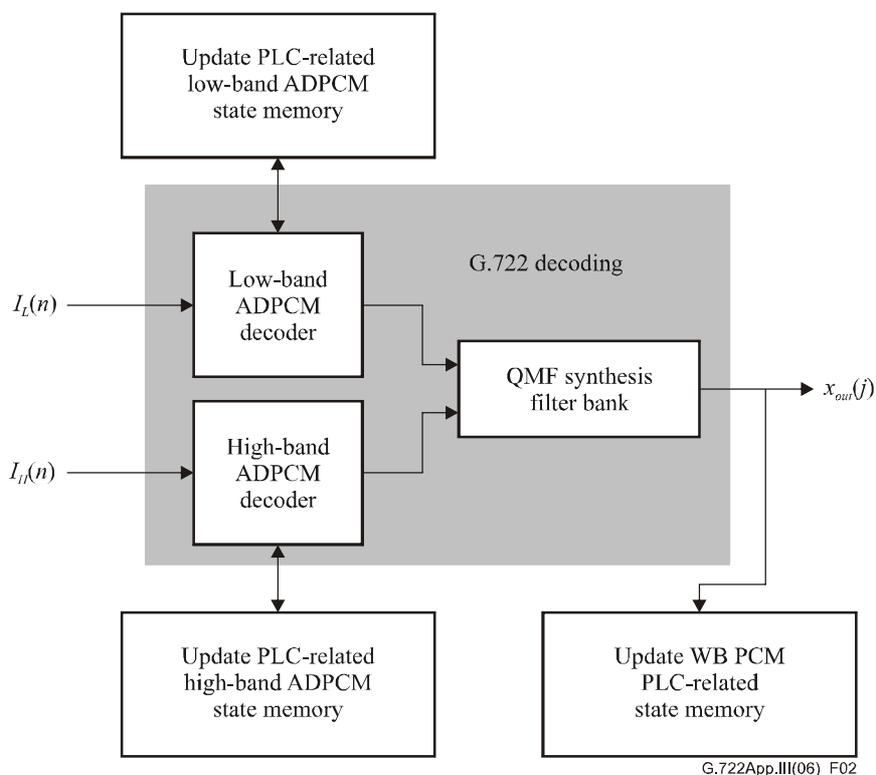


Figure III.2 – Regular G.722 decoding (Type 1)

Types 2, 3 and 4

The algorithm performs wideband (WB) PCM PLC in the 16-kHz output speech domain for frames of Types 2, 3 and 4. A block diagram of the WB PCM PLC is shown in Figure III.3. Past output speech of G.722 is buffered and passed to the WB PCM PLC. The WB PCM PLC is based on periodic waveform extrapolation (PWE), and pitch estimation is an important component of the WB PCM PLC. Initially, a coarse pitch is estimated based on a down-sampled (to 2 kHz) signal in the weighted speech domain. Subsequently, this estimate is refined at full resolution using the original 16-kHz sampling. The WB PCM PLC output is a linear combination of the periodically extrapolated waveform and noise shaped by LPC. For extended erasures, the output waveform is gradually muted. The muting starts after 20 ms of frame loss and becomes complete after 60 ms of loss.

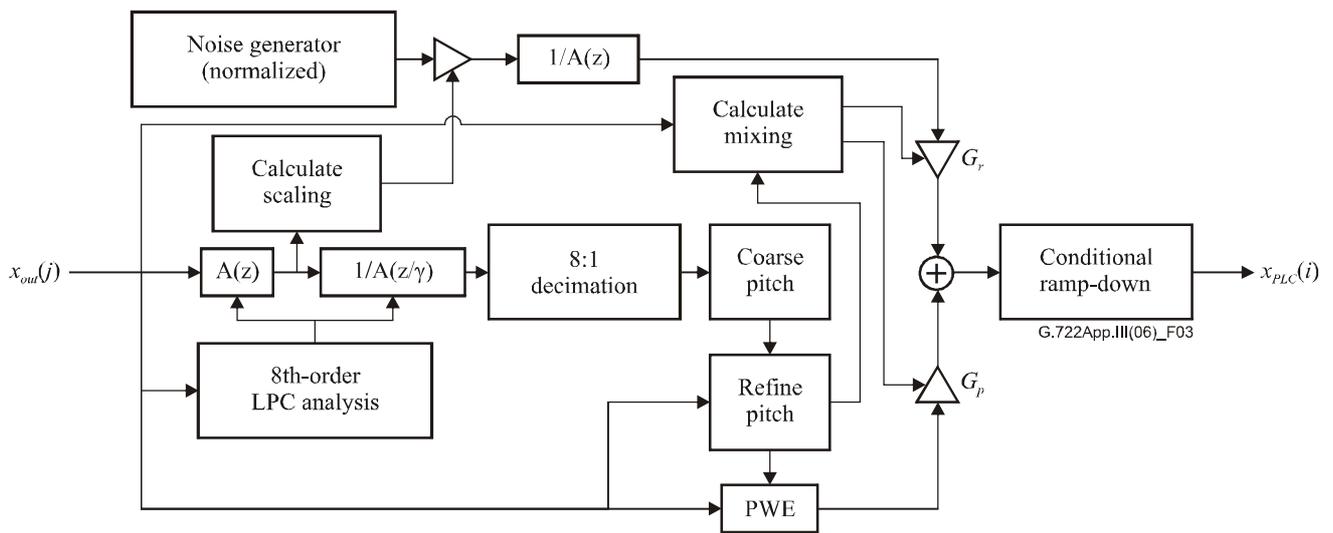


Figure III.3 – Block diagram of WB PCM PLC (for frames of Types 2, 3 and 4)

As shown in Figure III.4, for frames of Types 2, 3 and 4, the output of the WB PCM PLC is passed through the G.722 QMF analysis filter bank to obtain corresponding subband signals that are subsequently passed to modified low-band and high-band ADPCM encoders, respectively, in order to update the states and memory of the decoder. Only partial simplified subband ADPCM encoders are used for this update.

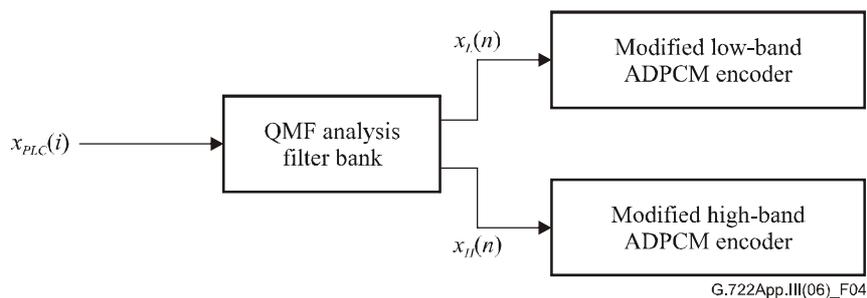


Figure III.4 – Re-encoding of PLC output to update subband ADPCM states (for frames of Types 2, 3 and 4)

The processing in Figures III.3 and III.4 takes place during lost frames. The modified low-band and high-band ADPCM encoders of Figure III.4 are simplified to reduce complexity. They are described in detail in clause III.7. One feature that is different from the regular encoders is an adaptive reset of the decoders based on signal property and the duration of packet loss.

Type 5

The most complex processing takes place for frame Type 5. This is the first received frame, where the transition from extrapolated to decoded waveform takes place. Primary techniques are re-phasing and time-warping. A high-level block diagram of re-phasing and time-warping is shown in Figure III.5.

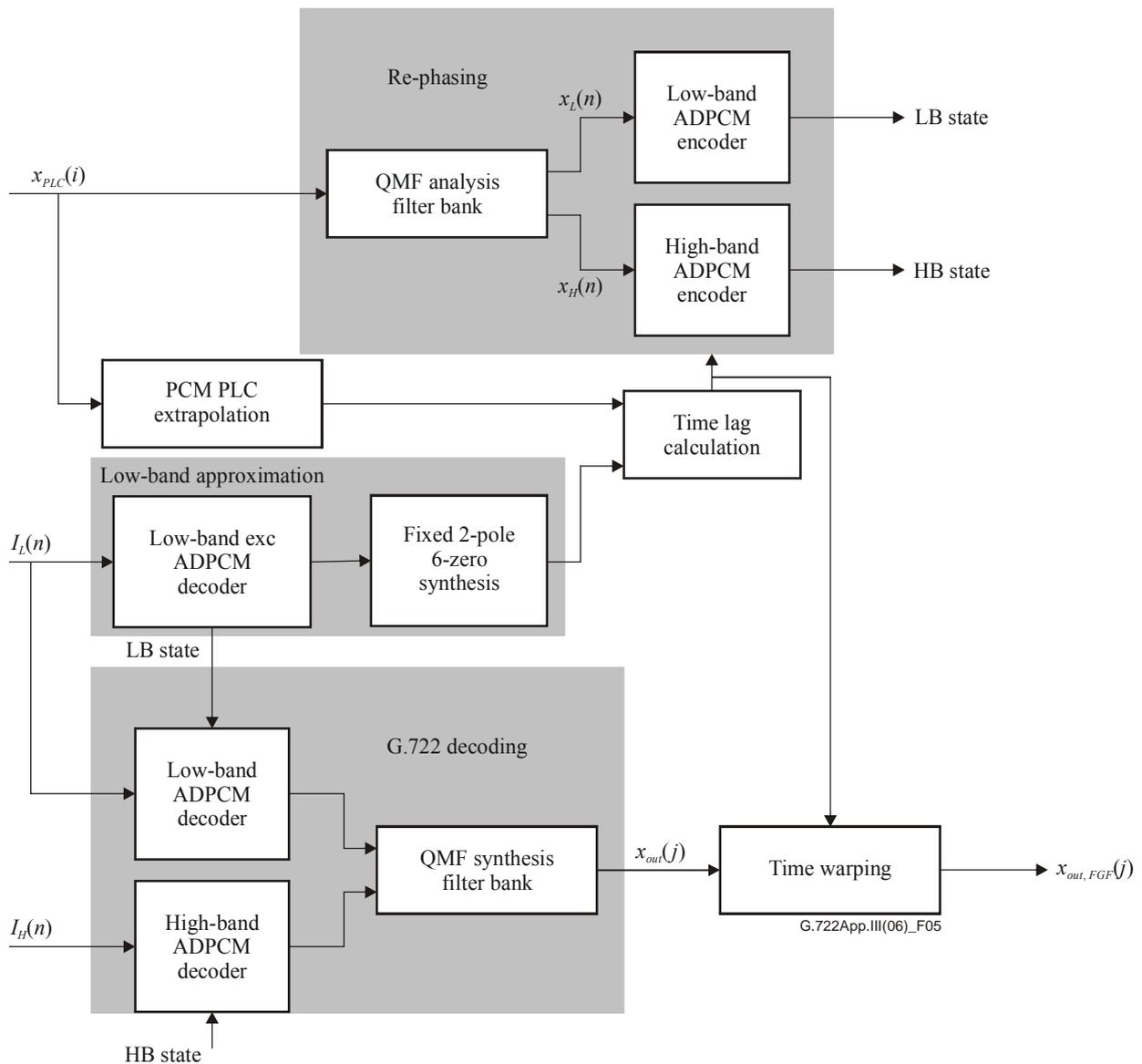


Figure III.5 – Re-phasing and time-warping (Type 5)

Additionally, at the first received frame following packet loss, the memory of the QMF synthesis filter bank at the decoder needs to be updated. See clause III.9 for details. Other important processing takes place in frames of Type 5 as well. This includes techniques such as adaptive setting of low-band and high-band log-scale factors at the beginning of the first received frame (clauses III.8.1 and III.8.2). Furthermore, the state memory update indicated for Type 1 frames applies to Type 5 frames as well. The techniques of clauses III.8.2.3, III.8.3 and III.8.4 apply to Type 5 frames and extend into Type 6 frames.

Types 5 and 6

Frames of Type 6 are decoded with modified and constrained subband ADPCM decoders. The block diagram for decoding of Type 6 frames is depicted in Figure III.6.

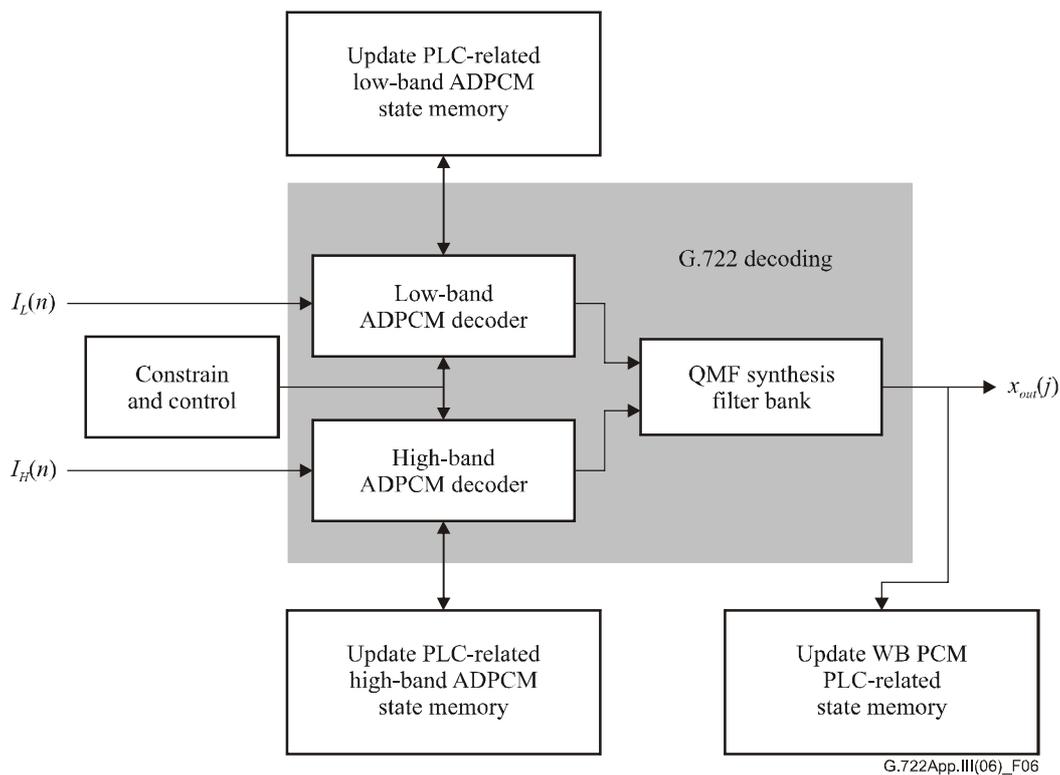


Figure III.6 – Constrained and controlled decoding in first received frames (Types 5 and 6)

The constrain and control of the subband ADPCM decoders are imposed for both Types 5 and 6 frames, i.e., for the first 80 ms after packet loss. Some do not extend beyond 40 ms, while others are adaptive in duration and/or degree. See clause III.8 for details.

In error-free channel conditions, the algorithm is bit-exact with G.722. Furthermore, in error conditions, the algorithm is identical to G.722 beyond the eighth frame after the end of packet loss, and without bit-errors convergence towards the G.722 error-free output should be expected.

The PLC algorithm supports any packet size that is a multiple of 10 ms. For packet sizes greater than 10 ms the PLC algorithm is simply called multiple times per packet, at 10-ms intervals. Accordingly, in the following the PLC algorithm is described in this context in terms of the intrinsic frame size of 10 ms.

III.6 WB PCM PLC – Waveform extrapolation of G.722 output

For lost frames corresponding to packet loss (Types 2, 3 and 4 frames), the WB PCM PLC scheme depicted in Figure III.3 extrapolates the G.722 output waveform $x_{out}(j)$ of the previous frames to fill up the current frame. Such extrapolated wideband signal waveform $x_{PLC}(i)$ is then used as the output waveform of the G.722 PLC during Types 2, 3 and 4 frames. For convenience in describing various blocks in Figure III.3, once the signal $x_{PLC}(i)$ has been calculated by the WB PCM PLC for lost frames, this signal $x_{PLC}(i)$ is considered to be written to the buffer for $x_{out}(j)$, which is the final output signal of the entire G.722 decoder/PLC system.

Each block of Figure III.3 will now be described in more detail in the following subclauses.

III.6.1 Eighth-order LPC analysis

The eighth-order LPC analysis in Figure III.3 is performed near the end of the frame processing loop, after the current frame of signal $x_{out}(j)$ has been calculated and stored in the buffer. This eighth-order LPC analysis is the common type of autocorrelation LPC analysis, with a 10-ms asymmetric analysis window applied to the $x_{out}(j)$ signal of the current received frame. This asymmetric window is given by:

$$w(j) = \begin{cases} \frac{1}{2} \left[1 - \cos\left(\frac{(j+1)\pi}{121}\right) \right], & \text{for } j = 0, 1, 2, \dots, 119 \\ \cos\left(\frac{(j-120)\pi}{80}\right), & \text{for } j = 120, 121, \dots, 159 \end{cases} \quad (\text{III-1})$$

Let $x_{out}(0), x_{out}(1), \dots, x_{out}(159)$ represent the G.722 decoder/PLC output wideband signal samples in the current received frame. The windowing operation is performed as follows.

$$x_w(j) = x_{out}(j)w(j), \quad j = 0, 1, 2, \dots, 159 \quad (\text{III-2})$$

Next, the autocorrelation coefficients are calculated as follows.

$$r(i) = \sum_{j=i}^{159} x_w(j)x_w(j-i), \quad i = 0, 1, 2, \dots, 8 \quad (\text{III-3})$$

The spectral smoothing and white noise correction operations are then applied to the autocorrelation coefficients as follows.

$$\hat{r}(i) = \begin{cases} 1.0001 \times r(0), & i = 0 \\ r(i) e^{\frac{-(2\pi i \sigma / f_s)^2}{2}}, & i = 1, 2, \dots, 8 \end{cases} \quad (\text{III-4})$$

where $f_s = 16000$ is the sampling rate of the input signal and $\sigma = 40$.

Next, the Levinson-Durbin recursion is used to convert the autocorrelation coefficients $\hat{r}(i)$ to the LPC predictor coefficients \hat{a}_i , $i = 0, 1, \dots, 8$. If the Levinson-Durbin recursion exits pre-maturely before the recursion is completed (for example, because the prediction residual energy $E(i)$ is less than zero), then the short-term predictor coefficients of the last frame are also used in the current frame. To do the exception handling this way, there needs to be an initial value of the \hat{a}_i array. The initial value of the \hat{a}_i array is set to $\hat{a}_0 = 1$ and $\hat{a}_i = 0$ for $i = 1, 2, \dots, 8$. The Levinson-Durbin recursion algorithm is specified below.

- 1) If $\hat{r}(0) \leq 0$, use the \hat{a}_i array of the last frame, and exit the Levinson-Durbin recursion.
- 2) $E(0) = \hat{r}(0)$
- 3) $k_1 = -\hat{r}(1) / \hat{r}(0)$
- 4) $\hat{a}_1^{(1)} = k_1$
- 5) $E(1) = (1 - k_1^2)E(0)$
- 6) If $E(1) \leq 0$, use the \hat{a}_i array of the last frame, and exit the Levinson-Durbin recursion.

7) For $i = 2, 3, 4, \dots, 8$, do the following:

$$k_i = \frac{-\hat{r}(i) - \sum_{j=1}^{i-1} \hat{a}_j^{i-1} \hat{r}(i-j)}{E(i-1)}$$

$$\hat{a}_i^{(i)} = k_i$$

$$\hat{a}_j^{(i)} = \hat{a}_j^{i-1} + k_i \hat{a}_{i-j}^{i-1}, \quad \text{for } j = 1, 2, \dots, i-1$$

$$E(i) = (1 - k_i^2)E(i-1)$$

If $E(i) \leq 0$, use the \hat{a}_i array of the last frame, and exit the Levinson-Durbin recursion.

If the recursion exited pre-maturely, the \hat{a}_i array of the last frame is used. If the recursion is completed successfully (which is normally the case), the LPC predictor coefficients are taken as:

$$\hat{a}_0 = 1 \quad (\text{III-5})$$

$$\hat{a}_i = \hat{a}_i^{(8)}, \quad \text{for } i = 1, 2, \dots, 8 \quad (\text{III-6})$$

By applying the bandwidth expansion operation to the coefficients derived above, the final set of LPC predictor coefficients is obtained as:

$$a_i = (0.96852)^i \hat{a}_i^{(8)}, \quad \text{for } i = 0, 1, \dots, 8 \quad (\text{III-7})$$

III.6.2 Calculation of short-term prediction residual signal

The block in Figure III.3 with a label of "A(z)" represents a short-term linear prediction error filter, with the filter coefficients of a_i for $i = 0, 1, \dots, 8$ as calculated above. This block is performed after the eighth-order LPC analysis is performed. This block calculates the short-term prediction residual signal $d(j)$ as follows:

$$d(j) = x_{out}(j) + \sum_{i=1}^8 a_i \cdot x_{out}(j-i), \quad \text{for } j = 0, 1, 2, \dots, 159 \quad (\text{III-8})$$

As is conventional, the time index n of the current frame continues from the time index of the last frame. In other words, if the time index range of $0, 1, 2, \dots, 159$ represents the current frame, then the time index range of $-160, -159, \dots, -1$ represents the last frame. Thus, in the equation above, if the index $(j-i)$ is negative, it just points to the signal sample near the end of the last frame.

III.6.3 Calculation of scaling factor

The block in Figure III.3 with a label of "Calculate scaling" calculates the average magnitude of the short-term prediction residual signal in the current frame. This block is performed after the short-term prediction residual signal $d(j)$ is calculated as described in clause III.6.2. This average magnitude avm is calculated as follows:

$$avm = \frac{1}{160} \sum_{j=0}^{159} |d(j)| \quad (\text{III-9})$$

If the next frame is a lost frame (i.e., corresponds to a packet loss), this average magnitude avm may be used as a scaling factor to scale a white Gaussian noise sequence if the current frame is sufficiently unvoiced.

III.6.4 Calculation of weighted speech signal

The block in Figure III.3 with a label of "1/A(z/γ)" represents a weighted short-term synthesis filter. This block is performed after the short-term prediction residual signal $d(j)$ is calculated for the current frame. The coefficients of this filter, a'_i for $i = 0, 1, \dots, 8$, are calculated as follows with $\gamma_1 = 0.75$.

$$a'_i = \gamma_1^i a_i, \quad \text{for } i = 1, 2, \dots, 8 \quad (\text{III-10})$$

The short-term prediction residual signal $d(j)$ passes through this weighted short-term synthesis filter. The corresponding output weighted speech signal $xw(j)$ is calculated as:

$$xw(j) = d(j) - \sum_{i=1}^8 a'_i \cdot xw(j-i), \quad \text{for } j = 0, 1, 2, \dots, 159 \quad (\text{III-11})$$

III.6.5 Eight-to-one decimation

Table III.1 – Coefficients for 60th order FIR filter

Lag, i	b_i in Q15	Lag, i	b_i in Q15	Lag, i	b_i in Q15
0	1209	20	-618	40	313
1	728	21	-941	41	143
2	1120	22	-1168	42	-6
3	1460	23	-1289	43	-126
4	1845	24	-1298	44	-211
5	2202	25	-1199	45	-259
6	2533	26	-995	46	-273
7	2809	27	-701	47	-254
8	3030	28	-348	48	-210
9	3169	29	20	49	-152
10	3207	30	165	50	-89
11	3124	31	365	51	-30
12	2927	32	607	52	21
13	2631	33	782	53	58
14	2257	34	885	54	81
15	1814	35	916	55	89
16	1317	36	881	56	84
17	789	37	790	57	66
18	267	38	654	58	41
19	-211	39	490	59	17

The weighted speech signal is passed through a 60th-order minimum-phase FIR low-pass filter, and then 8:1 decimation is performed to down-sample the resulting 16-kHz low-pass filtered weighted speech signal to 2-kHz down-sampled weighted speech signal $xwd(n)$. This decimation operation is performed after the weighted speech signal $xw(j)$ is calculated. To reduce complexity, the FIR low-pass filtering operation is carried out only when a new sample of $xwd(n)$ is needed. Thus, the down-sampled weighted speech signal $xwd(n)$ is calculated as:

$$xwd(n) = \sum_{i=0}^{59} b_i \cdot xw(8n + 7 - i), \quad \text{for } n = 0, 1, 2, \dots, 19 \quad (\text{III-12})$$

where $b_i, i = 0, 1, 2, \dots, 59$ are the filter coefficients for the 60th-order FIR low-pass filter as given in Table III.1.

III.6.6 Coarse pitch period extraction

To reduce the computational complexity, in WB PCM PLC the pitch extraction is performed in two stages:

- 1) determination of the coarse pitch period with a time resolution of the 2-kHz decimated signal;
- 2) pitch period refinement with a time resolution of the 16-kHz undecimated signal. Such pitch extraction is performed only during the received frames after the down-sampled weighted speech signal $xwd(n)$ is calculated.

This clause describes the first-stage coarse pitch period extraction algorithm, which is represented in Figure III.3 by the block labelled "Coarse Pitch". This algorithm is based on maximizing the normalized cross-correlation with some additional decision logic.

A pitch analysis window of 15 ms is used in the coarse pitch period extraction. The end of the pitch analysis window is lined up with the end of the current frame. At a sampling rate of 2 kHz, 15 ms correspond to 30 samples. Without loss of generality, let the index range of $n = 0$ to $n = 29$ corresponds to the pitch analysis window for $xwd(n)$. The coarse pitch period extraction algorithm starts by calculating the following values:

$$c(k) = \sum_{n=0}^{29} xwd(n)xwd(n-k) \quad (\text{III-13})$$

$$E(k) = \sum_{n=0}^{29} [xwd(n-k)]^2 \quad (\text{III-14})$$

$$c2(k) = \begin{cases} c^2(k), & \text{if } c(k) \geq 0 \\ -c^2(k), & \text{if } c(k) < 0 \end{cases} \quad (\text{III-15})$$

for all integers from $k = MINPPD - 1$ to $k = MAXPPD + 1$, where $MINPPD = 5$ and $MAXPPD = 33$ are the minimum and maximum pitch period in the decimated domain, respectively. The coarse pitch period extraction algorithm then searches through the range of $k = MINPPD, MINPPD + 1, MINPPD + 2, \dots, MAXPPD$ to find all local peaks of the array $\{c2(k)/E(k)\}$ for which $c(k) > 0$. (A value is characterized as a local peak if both of its adjacent values are smaller.) Let N_p denote the number of such positive local peaks. Let $k_p(j), j = 1, 2, \dots, N_p$ be the indices where $c2(k_p(j))/E(k_p(j))$ is a local peak and $c(k_p(j)) > 0$, and let $k_p(1) < k_p(2) < \dots < k_p(N_p)$. For convenience, the term $c2(k)/E(k)$ will be referred to as the "normalized correlation square".

If $N_p = 0$, that is, if there is no positive local peak for the function $c2(k)/E(k)$, then the algorithm searches for the largest negative local peak with the largest magnitude of $|c2(k)/E(k)|$. If such a largest negative local peak is found, the corresponding index k is used as the output coarse pitch period cpp , and the processing of this block is terminated. If the normalized correlation square function $c2(k)/E(k)$ has neither a positive nor a negative local peak, then the output coarse pitch period is set to $cpp = MINPPD$, and the processing of this block is terminated. If $N_p = 1$, the output coarse pitch period is set to $cpp = k_p(1)$, and the processing of this block is terminated.

If there are two or more local peaks ($N_p \geq 2$), then this block uses *Algorithms 1, 2, 3 and 4* (to be described below), in that order, to determine the output coarse pitch period cpp . Variables calculated in one of the four algorithms will be carried over and used in later algorithms.

Algorithm 1 below is used to identify the largest quadratically interpolated peak around local peaks of the normalized correlation square $c2(k_p)/E(k_p)$. Quadratic interpolation is performed for $c(k_p)$, while linear interpolation is performed for $E(k_p)$. Such interpolation is performed with the time resolution of the undecimated 16-kHz speech signal. In the algorithm below, D denotes the decimation factor used when decimating $xw(n)$ to $xwd(n)$. Thus, $D = 8$ here.

Algorithm 1 – Find the largest quadratically interpolated peak around $c2(k_p)/E(k_p)$:

- i) Set $c2max = -1$, $Emax = 1$, and $jmax = 0$.
- ii) For $j=1, 2, \dots, N_p$, do the following 12 steps:
 - 1) Set $a = 0.5 [c(k_p(j) + 1) + c(k_p(j) - 1)] - c(k_p(j))$
 - 2) Set $b = 0.5 [c(k_p(j) + 1) - c(k_p(j) - 1)]$
 - 3) Set $ji = 0$
 - 4) Set $ei = E(k_p(j))$
 - 5) Set $c2m = c2(k_p(j))$
 - 6) Set $Em = E(k_p(j))$
 - 7) If $c2(k_p(j) + 1)E(k_p(j) - 1) > c2(k_p(j) - 1)E(k_p(j) + 1)$, do the remaining part of step 7:

$$\Delta = [E(k_p(j) + 1) - ei]/D$$
 For $k = 1, 2, \dots, D/2$, do the following indented part of step 7:

$$ci = a (k/D)^2 + b(k/D) + c(k_p(j))$$

$$ei \leftarrow ei + \Delta$$
 If $(ci)^2 Em > (c2m) ei$, do the next three indented lines:

$$ji = k$$

$$c2m = (ci)^2$$

$$Em = ei$$
 - 8) If $c2(k_p(j) + 1)E(k_p(j) - 1) \leq c2(k_p(j) - 1)E(k_p(j) + 1)$, do the remaining part of step 8:

$$\Delta = [E(k_p(j) - 1) - ei]/D$$
 For $k = -1, -2, \dots, -D/2$, do the following indented part of step 8:

$$ci = a (k/D)^2 + b(k/D) + c(k_p(j))$$

$$ei \leftarrow ei + \Delta$$
 If $(ci)^2 Em > (c2m) ei$, do the next three indented lines:

$$ji = k$$

$$c2m = (ci)^2$$

$$Em = ei$$
 - 9) Set $lag(j) = k_p(j) + ji/D$
 - 10) Set $c2i(j) = c2m$
 - 11) Set $Ei(j) = Em$
 - 12) If $c2m \times Emax > c2max \times Em$, do the following three indented lines:

$$jmax = j$$

$$c2max = c2m$$

$$Emax = Em$$

iii) Set the first candidate for coarse pitch period as $cpp = lag(jmax)$.

The symbol \leftarrow indicates that the parameter on the left-hand side is being updated with the value on the right-hand side.

To avoid picking a coarse pitch period that is around an integer multiple of the true coarse pitch period, a search through the time lags corresponding to the local peaks of $c2(k_p)/E(k_p)$ is performed to see if any of such time lags is close enough to the output coarse pitch period of the last frame, denoted as $cpplast$. (For the very first frame, $cpplast$ is initialized to 12.) If a time lag is within 25% of $cpplast$, it is considered close enough. For all such time lags within 25% of $cpplast$, the corresponding quadratically interpolated peak values of the normalized correlation square $c2(k_p)/E(k_p)$ are compared, and the interpolated time lag corresponding to the maximum normalized correlation square is selected for further consideration. *Algorithm 2* below performs the task described above. The interpolated arrays $c2i(j)$ and $Ei(j)$ calculated in *Algorithm 1* above are used in this algorithm.

Algorithm 2 – Find the time lag maximizing interpolated $c2(k_p)/E(k_p)$ among all time lags close to the output coarse pitch period of the last frame:

- i) Set index $im = -1$
- ii) Set $c2m = -1$
- iii) Set $Em = 1$
- iv) For $j=1, 2, \dots, N_p$, do the following:
If $|k_p(j) - cpplast| \leq 0.25 \times cpplast$, do the following:
If $c2i(j) \times Em > c2m \times Ei(j)$, do the following three lines:
 $im = j$
 $c2m = c2i(j)$
 $Em = Ei(j)$

Note that if there is no time lag $k_p(j)$ within 25% of $cpplast$, then the value of the index im will remain at -1 after *Algorithm 2* is performed. If there are one or more time lags within 25% of $cpplast$, the index im corresponds to the largest normalized correlation square among such time lags.

Next, *Algorithm 3* determines whether an alternative time lag in the first half of the pitch range should be chosen as the output coarse pitch period. Basically, it searches through all interpolated time lags $lag(j)$ that are less than 16, and checks whether any of them has a large enough local peak of normalized correlation square near every integer multiple of it (including itself) up to 32. If there are one or more such time lags satisfying this condition, the smallest of such qualified time lags is chosen as the output coarse pitch period.

Again, variables calculated in *Algorithms 1* and *2* above carry their final values over to *Algorithm 3* below. In the following, the parameter $MPDTH$ is 0.06, and the threshold array $MPTH(k)$ is given as $MPTH(2) = 0.7$, $MPTH(3) = 0.55$, $MPTH(4) = 0.48$, $MPTH(5) = 0.37$, and $MPTH(k) = 0.30$, for $k > 5$.

Algorithm 3 – Check whether an alternative time lag in the first half of the range of the coarse pitch period should be chosen as the output coarse pitch period:

For $j = 1, 2, 3, \dots, N_p$, in that order, do the following while $lag(j) < 16$:

- i) If $j \neq im$, set $threshold = 0.73$; otherwise, set $threshold = 0.4$.
- ii) If $c2i(j) \times Emax \leq threshold \times c2max \times Ei(j)$, disqualify this j , skip step iii for this j , increment j by 1 and go back to step i.

- iii) If $c2i(j) \times Emax > threshold \times c2max \times Ei(j)$, do the following:
 - a) For $k = 2, 3, 4, \dots$, do the following while $k \times lag(j) < 32$:
 - 1) $s = k \times lag(j)$
 - 2) $a = (1 - MPDTH) s$
 - 3) $b = (1 + MPDTH) s$
 - 4) Go through $m = j + 1, j + 2, j + 3, \dots, N_p$, in that order, and see if any of the time lags $lag(m)$ is between a and b . If none of them is between a and b , disqualify this j , stop step iii, increment j by 1 and go back to step i. If there is at least one such m that satisfies $a < lag(m) \leq b$ and $c2i(m) \times Emax > MPTH(k) \times c2max \times Ei(m)$, then it is considered that a large-enough peak of the normalized correlation square is found in the neighborhood of the k -th integer multiple of $lag(j)$; in this case, stop step iii a 4, increment k by 1, and go back to step iii a 1.
 - b) If step iii a is completed without stopping prematurely, that is, if there is a large enough interpolated peak of the normalized correlation square within $\pm 100 \times MPDTH$ % of every integer multiple of $lag(j)$ that is less than 32, then stop this algorithm, skip *Algorithm 4* and set $cpp = lag(j)$ as the final output coarse pitch period.

If *Algorithm 3* above is completed without finding a qualified output coarse pitch period cpp , then *Algorithm 4* examines the largest local peak of the normalized correlation square around the coarse pitch period of the last frame, found in *Algorithm 2* above, and makes a final decision on the output coarse pitch period cpp . Again, variables calculated in *Algorithms 1* and *2* above carry their final values over to *Algorithm 4* below. In the following, the parameters are $SMDTH = 0.095$ and $LPTH1 = 0.78$.

Algorithm 4 – Final decision of the output coarse pitch period

- i) If $im = -1$, that is, if there is no large enough local peak of the normalized correlation square around the coarse pitch period of the last frame, then use the cpp calculated at the end of *Algorithm 1* as the final output coarse pitch period, and exit this algorithm.
- ii) If $im = jmax$, that is, if the largest local peak of the normalized correlation square around the coarse pitch period of the last frame is also the global maximum of all interpolated peaks of the normalized correlation square within this frame, then use the cpp calculated at the end of *Algorithm 1* as the final output coarse pitch period, and exit this algorithm.
- iii) If $im < jmax$, do the following indented part:
 - If $c2m \times Emax > 0.43 \times c2max \times Em$, do the following indented part of step iii:
 - a) If $lag(im) > MAXPPD/2$, set output $cpp = lag(im)$ and exit this algorithm.
 - b) Otherwise, for $k = 2, 3, 4, 5$, do the following indented part:
 - 1) $s = lag(jmax)/k$
 - 2) $a = (1 - SMDTH) s$
 - 3) $b = (1 + SMDTH) s$
 - 4) If $lag(im) > a$ and $lag(im) < b$, set output $cpp = lag(im)$ and exit this algorithm.
- iv) If $im > jmax$, do the following indented part:
 - If $c2m \times Emax > LPTH1 \times c2max \times Em$, set output $cpp = lag(im)$ and exit this algorithm.
- v) If algorithm execution proceeds to here, none of the steps above have selected a final output coarse pitch period. In this case, just accept the cpp calculated at the end of *Algorithm 1* as the final output coarse pitch period.

III.6.7 Pitch period refinement

The block in Figure III.3 with a label of "Refine pitch" performs the second-stage processing of the pitch period extraction algorithm by searching in the neighborhood of the coarse pitch period in full 16-kHz time resolution using the G.722 decoded output speech signal. This block first converts the coarse pitch period cpp to the undecimated signal domain by multiplying it by the decimation factor D , where $D = 8$. The pitch refinement analysis window size WSZ is chosen as the smaller of $cpp \times D$ samples and 160 samples (corresponding to 10 ms): $WSZ = \min(cpp \times D, 160)$.

Next, the lower bound of the search range is calculated as $lb = \max(MINPP, cpp \times D - 4)$, where $MINPP = 40$ samples is the minimum pitch period. The upper bound of the search range is calculated as $ub = \min(MAXPP, cpp \times D + 4)$, where $MAXPP = 265$ samples is the maximum pitch period.

This block maintains a buffer of 16-kHz G.722 decoded speech signal $x_{out}(j)$ with a total of $XQOFF = MAXPP + 1 + FRSZ$ samples, where $FRSZ = 160$ is the frame size. The last $FRSZ$ samples of this buffer contain the G.722 decoded speech signal of the current frame. The first $MAXPP + 1$ samples are populated with the G.722 decoder/PLC output signal in the previous frames immediately before the current frame. The last sample of the analysis window is aligned with the last sample of the current frame. Let the index range from $j = 0$ to $j = WSZ - 1$, corresponding to the analysis window, which is the last WSZ samples in the $x_{out}(j)$ buffer, and let negative indices denote the samples prior to the analysis window. The following correlation and energy terms in the undecimated signal domain are calculated for time lags k within the search range $[lb, ub]$.

$$\tilde{c}(k) = \sum_{j=0}^{WSZ-1} x_{out}(j)x_{out}(j-k) \quad (\text{III-16})$$

$$\tilde{E}(k) = \sum_{j=0}^{WSZ-1} x_{out}(j-k)^2 \quad (\text{III-17})$$

The time lag $k \in [lb, ub]$ that maximizes the ratio $\tilde{c}^2(k) / \tilde{E}(k)$ is chosen as the final refined pitch period for frame erasure, or $ppfe$. That is:

$$ppfe = \arg \max_{k \in [lb, ub]} \left[\frac{\tilde{c}^2(k)}{\tilde{E}(k)} \right] \quad (\text{III-18})$$

Next, the block labelled "Refine pitch" also calculates two more pitch-related scaling factors. The first is called $ptfe$, or pitch tap for frame erasure. It is the scaling factor used for periodic waveform extrapolation. It is calculated as the ratio of the average magnitude of the $x_{out}(j)$ signal in the analysis window and the average magnitude of the portion of the $x_{out}(j)$ signal that is $ppfe$ samples earlier, with the same sign as the correlation between these two signal portions.

$$ptfe = \text{sign}(\tilde{c}(ppfe)) \left[\frac{\sum_{j=0}^{WSZ-1} |x_{out}(j)|}{\sum_{j=0}^{WSZ-1} |x_{out}(j - ppfe)|} \right] \quad (\text{III-19})$$

In the degenerate case when $\sum_{j=0}^{WSZ-1} |x_{out}(j - ppfe)| = 0$, $ptfe$ is set to 0. After such calculation of $ptfe$, the value of $ptfe$ is range-bound to $[-1, 1]$.

The second pitch-related scaling factor is called ppt , or pitch predictor tap. It is used for calculating the long-term filter ringing signal (to be described later). It is calculated as $ppt = 0.75 \times ptfe$.

III.6.8 Calculate mixing ratio

The block labelled as "Calculate mixing" in Figure III.3 calculates a figure of merit to determine the mixing ratio between the periodically extrapolated waveform and the filtered noise waveform during lost frames. This calculation is performed only during the very first lost frame in each occurrence of packet loss, and the resulting mixing ratio is used throughout that particular packet loss. The figure of merit is a weighted sum of three signal features: logarithmic gain, first normalized autocorrelation, and pitch prediction gain. Each of them is calculated as follows.

Using the same indexing convention for $x_{out}(j)$ as in clause III.6.7, the energy of the $x_{out}(j)$ signal in the pitch refinement analysis window is:

$$sige = \sum_{j=0}^{WSZ-1} x_{out}^2(j) \quad (III-20)$$

and the base-2 logarithmic gain lg is calculated as:

$$lg = \begin{cases} \log_2(sige), & \text{if } sige \neq 0 \\ 0, & \text{if } sige = 0 \end{cases} \quad (III-21)$$

If $\tilde{E}(ppfe) \neq 0$, the pitch prediction residual energy is calculated as:

$$rese = sige - \tilde{c}^2(ppfe) / \tilde{E}(ppfe) \quad (III-22)$$

and the pitch prediction gain pg is calculated as:

$$pg = \begin{cases} 10 \log_{10} \left(\frac{sige}{rese} \right), & \text{if } rese \neq 0 \\ 20, & \text{if } rese = 0 \end{cases} \quad (III-23)$$

If $\tilde{E}(ppfe) = 0$, set $pg = 0$. If $sige = 0$, also set $pg = 0$.

The first normalized autocorrelation ρ_1 is calculated as:

$$\rho_1 = \begin{cases} \left[\frac{\sum_{j=0}^{WSZ-2} x_{out}(j)x_{out}(j+1)}{sige} \right], & \text{if } sige \neq 0 \\ 0, & \text{if } sige = 0 \end{cases} \quad (III-24)$$

After these three signal features are obtained, the figure of merit is calculated as:

$$merit = lg + pg + 12\rho_1 \quad (III-25)$$

The *merit* calculated above determines the two scaling factors G_p and G_r , which effectively determine the mixing ratio between the periodically extrapolated waveform and the filtered noise waveform. There are two thresholds used for *merit*: merit high threshold *MHI* and merit low threshold *MLO*. These thresholds are set as $MHI = 28$ and $MLO = 20$. The scaling factor G_r for the random (filtered noise) component is calculated as:

$$G_r = \frac{MHI - merit}{MHI - MLO} \quad (III-26)$$

and the scaling factor G_p for the periodic component is calculated as:

$$G_p = 1 - G_r \quad (III-27)$$

III.6.9 Periodic waveform extrapolation

The block labelled PWE ("Periodic waveform extrapolation") in Figure III.3 periodically extrapolates the previous output speech waveform during the lost frames if $merit > MLO$.

At the very first lost frame of each packet loss, the average pitch period increment per frame is calculated. A pitch period history buffer $pph(m)$, $m = 1, 2, \dots, 5$ holds the pitch period $ppfe$ for the previous five frames. The average pitch period increment is obtained as follows. Start with the immediate last frame, and calculate the pitch period increment from its preceding frame to that frame (negative value means pitch period decrement). If the pitch period increment is zero, the algorithm checks the pitch period increment at the preceding frame. This process continues until the first frame with a non-zero pitch period increment or until the fourth previous frame has been examined. If all five previous frames have identical pitch periods, the average pitch period increment is set to zero. Otherwise, if the first non-zero pitch period increment is found at the m -th previous frame, and if the magnitude of the pitch period increment is less than 5% of the pitch period at that frame, then the average pitch period increment $ppinc$ is obtained as the pitch period increment at that frame divided by m , and then the resulting value is limited to the range of $[-1, 2]$.

In the second consecutive lost frame in a packet loss, the average pitch period increment $ppinc$ is added to the pitch period $ppfe$, and the resulting number is rounded to the nearest integer and then limited to the range of $[MINPP, MAXPP]$.

If the current frame is the very first lost frame of a packet loss, a so-called "ringing signal" is calculated for use in overlap-add to ensure smooth waveform transition at the beginning of the frame. The overlap-add length for the ringing signal and the periodically extrapolated waveform is 20 samples for the first lost frame. Let the index range of $j = 0, 1, 2, \dots, 19$ correspond to the first 20 samples of the current first lost frame, which is the overlap-add period, and let the negative indices correspond to previous frames. The long-term ringing signal is obtained as a scaled version of the short-term prediction residual signal that is one pitch period earlier than the overlap-add period.

$$ltrring(j) = x_{out}(j - ppfe) + \sum_{i=1}^8 a_i \cdot x_{out}(j - ppfe - i), \quad \text{for } j = 0, 1, 2, \dots, 19 \quad (\text{III-28})$$

After these 20 samples of $ltrring(j)$ are calculated, they are further scaled by the scaling factor ppt calculated in clause III.6.7.

$$ltrring(j) \leftarrow ppt \cdot ltrring(j), \quad \text{for } j = 0, 1, 2, \dots, 19 \quad (\text{III-29})$$

With the filter memory $ring(j)$, $j = -8, -7, \dots, -1$ initialized to the last 8 samples of the $x_{out}(j)$ signal in the last frame, the final ringing signal is obtained as:

$$ring(j) = ltrring(j) - \sum_{i=1}^8 a_i \cdot ring(j - i), \quad \text{for } j = 0, 1, 2, \dots, 19 \quad (\text{III-30})$$

Let the index range of $j = 0, 1, 2, \dots, 159$ correspond to the current first lost frame, and the index range of $j = 160, 161, 162, \dots, 209$ correspond to the first 50 samples of the next frame. Furthermore, let $wi(j)$ and $wo(j)$, $j = 0, 1, \dots, 19$, be the triangular fade-in and fade-out windows, respectively, so that $wi(j) + wo(j) = 1$. Then, the periodic waveform extrapolation is performed in two steps as follows.

Step 1:

$$x_{out}(j) = wi(j) \cdot ptfe \cdot x_{out}(n - ppfe) + wo(j) \cdot ring(j), \quad \text{for } j = 0, 1, 2, \dots, 19 \quad (\text{III-31})$$

Step 2:

$$x_{out}(j) = ptfe \cdot x_{out}(j - ppfe), \quad \text{for } j = 20, 21, 22, \dots, 209 \quad (\text{III-32})$$

III.6.10 Normalized noise generator

If $merit < MHI$, the block labelled "Noise generator (normalized)" in Figure III.3 generates a sequence of white Gaussian random noise with an average magnitude of unity. To save computational complexity, the white Gaussian random noise is pre-calculated and stored in a table. To avoid using a very long table, but without repeating the same noise pattern due to a short table, a special indexing scheme is used. In this scheme, the white Gaussian noise table $wn(j)$ has 127 entries, and the scaled version of the output of this noise generator block is:

$$wgn(j) = avm \times wn(\text{mod}(cfecount \times j, 127)), \quad \text{for } j = 0, 1, 2, \dots, 209 \quad (\text{III-33})$$

where $cfecount$ is the frame counter with $cfecount = k$ for the k -th consecutive lost frame into the current packet loss, and $\text{mod}(m, 127) = m - 127 \times \lfloor m/127 \rfloor$ is the modulo operation.

III.6.11 Filtering of noise sequence

The block in Figure III.3 with a label of " $1/A(z)$ " represents a short-term synthesis filter. If $merit < MHI$, it filters the scaled white Gaussian noise to give it the same spectral envelope as that of the $x_{out}(j)$ signal in the last frame. The filtered noise $fn(j)$ is obtained as:

$$fn(j) = wgn(j) - \sum_{i=1}^8 a_i \cdot fn(j-i), \quad \text{for } j = 0, 1, 2, \dots, 209 \quad (\text{III-34})$$

III.6.12 Mixing of periodic and random components

If $merit > MHI$, only the periodically extrapolated waveform $x_{out}(j)$ calculated in clause III.6.9 is used as the output of the WB PCM PLC. If $merit < MLO$, only the filtered noise signal $fn(j)$ is used as the output of the WB PCM PLC. If $MLO \leq merit \leq MHI$, then the two components are mixed as:

$$x_{out}(j) \leftarrow G_p \cdot x_{out}(j) + G_r \cdot fn(j) \quad \text{for } j = 0, 1, 2, \dots, 209 \quad (\text{III-35})$$

The first 40 extra samples of extrapolated $x_{out}(j)$ signal for $j = 160, 161, 162, \dots, 199$ will become the ringing signal $ring(j), j = 0, 1, 2, \dots, 39$ of the next frame. If the next frame is again a lost frame, only the first 20 samples of this ringing signal will be used for overlap-add. If the next frame is a received frame, then all 40 samples of this ringing signal will be used for overlap-add.

III.6.13 Conditional ramp down

If the packet loss lasts 20 ms or less, the $x_{out}(j)$ signal calculated in clause III.6.12 is used as the WB PCM PLC output signal. If the packet loss lasts longer than 60 ms, the WB PCM PLC output signal is completely muted. If the packet loss lasts longer than 20 ms but no more than 60 ms, the $x_{out}(j)$ signal calculated in clause III.6.12 is linearly ramped down (attenuate toward zero in a linear fashion). This conditional ramp down is performed as specified in the following algorithm during the lost frames when $cfecount > 2$. The array $gawd()$ is given by $\{-52, -69, -104, -207\}$ in Q15 format. Again, the index range of $j = 0, 1, 2, \dots, 159$ corresponds to the current frame of $x_{out}(j)$.

If $cfecount \leq 6$, do the next nine indented lines:

$$\delta = gawd(cfecount - 3)$$

$$gaw = 1$$

For $j = 0, 1, 2, \dots, 159$, do the next two lines:

$$x_{out}(j) = gaw \cdot x_{out}(j)$$

$$gaw = gaw + \delta$$

If $cfecount < 6$, do the next three lines:

For $j = 160, 161, 162, \dots, 209$, do the next two lines:

$$x_{out}(j) = gaw \cdot x_{out}(j)$$

$$gaw = gaw + delta$$

Otherwise (if $cfecount > 6$), set $x_{out}(j) = 0$ for $j = 0, 1, 2, \dots, 209$.

III.6.14 Overlap-add in the first received frame

In Type 5 frames, the output from the G.722 decoder $x_{out}(j)$ is overlap-added with the ringing signal from the last lost frame, $ring(j)$ (see clause III.6.12):

$$x_{out}(j) = w_i(j) \cdot x_{out}(j) + w_o(j) \cdot ring(j) \quad j = 0 \dots L_{OLA} - 1 \quad (\text{III-36})$$

where:

$$L_{OLA} = \begin{cases} 8 & \text{if } G_p = 0 \\ 40 & \text{otherwise} \end{cases} \quad (\text{III-37})$$

III.7 Re-encoding of PLC output

In order to update the memory and parameters of the G.722 ADPCM decoders during lost frames (frame Types 2, 3 and 4), conceptually, the PLC output is passed through the G.722 encoder. This involves:

- 1) passing the PLC output through the QMF analysis filter bank;
- 2) encoding the low-band subband signal with the low-band ADPCM encoder; and
- 3) encoding the high-band subband signal with the high-band ADPCM encoder.

To save complexity, simplified ADPCM subband encoders are designed. Figure III.4 shows the block diagram of steps 1 through 3.

III.7.1 Passing the PLC output through the QMF analysis filter bank

The memory of the QMF analysis filter bank is initialized to provide subband signals that are continuous with the decoded subband signals. The first 22 samples of the WB PCM PLC output constitutes the filter memory, and the subband signals are calculated according to:

$$x_L(n) = \sum_{i=0}^{11} h_{2i} \cdot x_{PLC}(23 + j - 2i) + \sum_{i=0}^{11} h_{2i+1} \cdot x_{PLC}(22 + j - 2i) \quad (\text{III-38})$$

and:

$$x_H(n) = \sum_{i=0}^{11} h_{2i} \cdot x_{PLC}(23 + j - 2i) - \sum_{i=0}^{11} h_{2i+1} \cdot x_{PLC}(22 + j - 2i) \quad (\text{III-39})$$

where $x_{PLC}(0)$ corresponds to the first sample of the 16-kHz WB PCM PLC output of the current frame, $x_L(n=0)$ and $x_H(n=0)$ correspond to the first samples of the 8-kHz low-band and high-band subband signals, respectively, of the current frame. The filtering is identical to the transmit QMF of the G.722 encoder except for the extra 22 samples of offset, and that the WB PCM PLC output (as opposed to the input) is passed to the filter bank. Furthermore, in order to generate a full frame (80 samples \sim 10 ms) of subband signals, the WB PCM PLC needs to extend beyond the current frame by 22 samples and generate (182 samples \sim 11.375 ms). Subband signals $x_L(n)$, $n = 0, 1, \dots, 79$, and $x_H(n)$, $n = 0, 1, \dots, 79$ are generated according to Equations III-38 and III-39, respectively.

III.7.2 Re-encoding of low-band signal

The low-band signal, $x_L(n)$, is encoded with a simplified low-band ADPCM encoder shown in Figure III.7. The inverse quantizer has been eliminated, and the unquantized prediction error replaces the quantized prediction error. Furthermore, since the update of the adaptive quantizer is only based on an 8-member subset of the 64-member set represented by the 6-bit low-band encoder index, $I_L(n)$, the prediction error is only quantized to the 8-member set. This provides identical update of the adaptive quantizer, yet simplifies the quantization. Table III.2 lists the decision levels, output code, and multipliers for the 8-level simplified quantizer based on the absolute value of $e_L(n)$.

Table III.2 – Decision levels, output code, and multipliers for the 8-level simplified quantizer

m_L	Lower threshold	Upper threshold	I_L	Multiplier, W_L
1	0.00000	0.14103	3c	-0.02930
2	0.14103	0.45482	38	-0.01465
3	0.45482	0.82335	34	0.02832
4	0.82335	1.26989	30	0.08398
5	1.26989	1.83683	2c	0.16309
6	1.83683	2.61482	28	0.26270
7	2.61482	3.86796	24	0.58496
8	3.86796	∞	20	1.48535

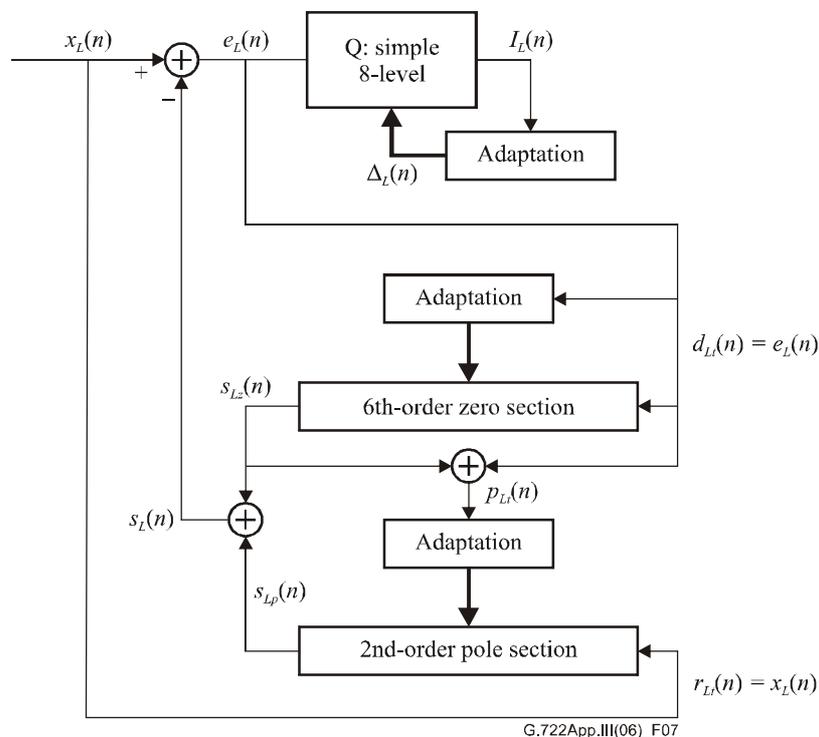


Figure III.7 – Low-band ADPCM sub-band re-encoding

The entities of Figure III.7 are calculated like their equivalents in the G.722 low-band ADPCM subband encoder:

$$s_{Lz}(n) = \sum_{i=1}^6 b_{L,i}(n-1) \cdot e_L(n-i) \quad (\text{III-40})$$

$$s_{Lp}(n) = \sum_{i=1}^2 a_{L,i}(n-1) \cdot x_L(n-i) \quad (\text{III-41})$$

$$s_L(n) = s_{Lp}(n) + s_{Lz}(n) \quad (\text{III-42})$$

$$e_L(n) = x_L(n) - s_L(n) \quad (\text{III-43})$$

$$p_{Lt}(n) = s_{Lz}(n) + e_L(n) \quad (\text{III-44})$$

The adaptive quantizer is updated exactly as in the G.722 encoder, see clause 3.5 of [ITU-T G.722]. The adaptation of the zero and pole sections take place as in the G.722 encoder, described in clauses 3.6.3 and 3.6.4 of [ITU-T G.722].

The low-band decoder is automatically reset after 60 ms of frame loss, but it may reset adaptively as early as 30 ms into frame loss. During re-encoding of the low-band signal, the properties of the partially reconstructed signal, $p_{Lt}(n)$, are monitored and control the adaptive reset of the low-band ADPCM decoder. The sign of $p_{Lt}(n)$ is monitored over the entire loss, and hence is reset to zero at the first lost frame:

$$\text{sgn}[p_{Lt}(n)] = \begin{cases} \text{sgn}[p_{Lt}(n-1)]+1 & p_{Lt}(n) > 0 \\ \text{sgn}[p_{Lt}(n-1)] & p_{Lt}(n) = 0 \\ \text{sgn}[p_{Lt}(n-1)]-1 & p_{Lt}(n) < 0 \end{cases} \quad (\text{III-45})$$

The property of $p_{Lt}(n)$ as a constant signal is monitored on a frame basis for lost frames, and hence is reset at the beginning of every lost frame. It is updated as:

$$\text{cnst}[p_{Lt}(n)] = \begin{cases} \text{cnst}[p_{Lt}(n-1)]+1 & p_{Lt}(n) = p_{Lt}(n-1) \\ \text{cnst}[p_{Lt}(n-1)] & p_{Lt}(n) \neq p_{Lt}(n-1) \end{cases} \quad (\text{III-46})$$

At the end of lost frames 3 through 5, the low-band decoder is reset if the following condition is satisfied:

$$\left| \frac{\text{sgn}[p_{Lt}(n)]}{N_{lost}} \right| > 36 \quad \text{OR} \quad \text{cnst}[p_{Lt}(n)] > 40 \quad (\text{III-47})$$

where N_{lost} is the number of lost frames, i.e., 3, 4, or 5.

III.7.3 Re-encoding of high-band signal

The high-band signal, $x_H(n)$, is encoded with a simplified high-band ADPCM encoder as shown in Figure III.8. The adaptive quantizer has been eliminated as the algorithm overwrites the log scale factor at the first received frame with a moving average prior to the loss, and hence, does not need the high-band re-encoded log scale factor. The quantized prediction error of the high-band ADPCM encoder is substituted with the unquantized prediction error.

The property of $p_H(n)$ as a constant signal is monitored on a frame basis for lost frames, and hence is reset at the beginning of every lost frame. It is updated as:

$$\text{cnst}[p_H(n)] = \begin{cases} \text{cnst}[p_H(n-1)]+1 & p_H(n) = p_H(n-1) \\ \text{cnst}[p_H(n-1)] & p_H(n) \neq p_H(n-1) \end{cases} \quad (\text{III-54})$$

At the end of lost frames 3 through 5, the high-band decoder is reset if the following condition is satisfied:

$$\left| \frac{\text{sgn}[p_H(n)]}{N_{lost}} \right| > 36 \quad \text{OR} \quad \text{cnst}[p_H(n)] > 40 \quad (\text{III-55})$$

III.8 Monitoring signal characteristics and their use for PLC

III.8.1 Low-band log scale factor

Characteristics of the low-band log scale factor, $\nabla_L(n)$, are updated during received frames and used at the first received frame after frame loss to adaptively set the state of the adaptive quantizer for the scale factor. A measure of the stationarity of the low-band log scale factor is derived and used to determine proper resetting of the state.

III.8.1.1 Stationarity of low-band log scale factor

The stationarity of the low-band log scale factor, $\nabla_L(n)$, is calculated and updated during received frames. It is based on a first-order moving average, $\nabla_{L,m1}(n)$, of $\nabla_L(n)$ with constant leakage:

$$\nabla_{L,m1}(n) = 7/8 \cdot \nabla_{L,m1}(n-1) + 1/8 \cdot \nabla_L(n) \quad (\text{III-56})$$

A measure of the tracking, $\nabla_{L,trck}(n)$, of the first-order moving average is calculated as:

$$\nabla_{L,trck}(n) = 127/128 \cdot \nabla_{L,trck}(n-1) + 1/128 \cdot \left| \nabla_{L,m1}(n) - \nabla_{L,m1}(n-1) \right| \quad (\text{III-57})$$

A second-order moving average, $\nabla_{L,m2}(n)$, with adaptive leakage is calculated according to Equation III-58.

$$\nabla_{L,m2}(n) = \begin{cases} 7/8 \cdot \nabla_{L,m2}(n-1) + 1/8 \cdot \nabla_{L,m1}(n) & \nabla_{L,trck}(n) < 3277 \\ 3/4 \cdot \nabla_{L,m2}(n-1) + 1/4 \cdot \nabla_{L,m1}(n) & 3277 \leq \nabla_{L,trck}(n) < 6554 \\ 1/2 \cdot \nabla_{L,m2}(n-1) + 1/2 \cdot \nabla_{L,m1}(n) & 6554 \leq \nabla_{L,trck}(n) < 9830 \\ \nabla_{L,m2}(n) = \nabla_{L,m1}(n) & 9830 \leq \nabla_{L,trck}(n) \end{cases} \quad (\text{III-58})$$

The stationarity of the low-band log scale factor is measured as a degree of change according to:

$$\nabla_{L,chg}(n) = 127/128 \cdot \nabla_{L,chg}(n-1) + 1/128 \cdot 256 \cdot \left| \nabla_{L,m2}(n) - \nabla_{L,m2}(n-1) \right| \quad (\text{III-59})$$

During lost frames, there is no update, i.e.:

$$\begin{aligned} \nabla_{L,m1}(n) &= \nabla_{L,m1}(n-1) \\ \nabla_{L,trck}(n) &= \nabla_{L,trck}(n-1) \\ \nabla_{L,m2}(n) &= \nabla_{L,m2}(n-1) \\ \nabla_{L,chg}(n) &= \nabla_{L,chg}(n-1) \end{aligned} \quad (\text{III-60})$$

III.8.1.2 Resetting of log scale factor of the low-band adaptive quantizer

At the first received frame after recovery from frame loss, the low-band log scale factor is reset (overwritten) adaptively depending on the stationarity prior to the frame loss:

$$\nabla_L(n-1) \leftarrow \begin{cases} \nabla_{L,m2}(n-1) & \nabla_{L,chg}(n-1) < 6554 \\ \frac{\nabla_L(n-1)}{3276} [\nabla_{L,chg}(n-1) - 6554] + \frac{\nabla_{L,m2}(n-1)}{3276} [9830 - \nabla_{L,chg}(n-1)] & 6554 \leq \nabla_{L,chg}(n-1) \leq 9830 \\ \nabla_L(n-1) & 9830 < \nabla_{L,chg}(n-1) \end{cases} \quad (\text{III-61})$$

III.8.2 High-band log scale factor

Characteristics of the high-band log scale factor, $\nabla_H(n)$, are updated during received frames and used at the first received frame following frame loss to set the state of the adaptive quantization scale factor. Furthermore, the characteristics adaptively control the convergence of the high-band log scale factor after frame loss.

III.8.2.1 Moving average and stationarity of high-band log scale factor

The tracking of $\nabla_H(n)$ is calculated according to:

$$\nabla_{H,trck}(n) = 0.97 \cdot \nabla_{H,trck}(n-1) + 0.03 \cdot [\nabla_{H,m}(n-1) - \nabla_H(n)] \quad (\text{III-62})$$

Based on the tracking, the moving average is calculated with adaptive leakage as:

$$\nabla_{H,m}(n) = \begin{cases} 255/256 \cdot \nabla_{H,m}(n-1) + 1/256 \cdot \nabla_H(n) & |\nabla_{H,trck}(n)| < 1638 \\ 127/128 \cdot \nabla_{H,m}(n-1) + 1/128 \cdot \nabla_H(n) & 1638 \leq |\nabla_{H,trck}(n)| < 3277 \\ 63/64 \cdot \nabla_{H,m}(n-1) + 1/64 \cdot \nabla_H(n) & 3277 \leq |\nabla_{H,trck}(n)| < 4915 \\ 31/32 \cdot \nabla_{H,m}(n-1) + 1/32 \cdot \nabla_H(n) & 4915 \leq |\nabla_{H,trck}(n)| \end{cases} \quad (\text{III-63})$$

The moving average is used for resetting the high-band log scale factor at the first received frame as described in clause III.8.2.2.

A measure of the stationarity of the high-band log scale factor is calculated from the moving average above according to:

$$\nabla_{H,chg}(n) = 127/128 \cdot \nabla_{H,chg}(n-1) + 1/128 \cdot 256 \cdot |\nabla_{H,m}(n) - \nabla_{H,m}(n-1)| \quad (\text{III-64})$$

The measure of stationarity is used to control re-convergence of $\nabla_H(n)$ after frame loss: see clause III.8.2.3.

During lost frames there is no update, i.e.:

$$\begin{aligned} \nabla_{H,trck}(n) &= \nabla_{H,trck}(n-1) \\ \nabla_{H,m}(n) &= \nabla_{H,m}(n-1) \\ \nabla_{H,chg}(n) &= \nabla_{H,chg}(n-1) \end{aligned} \quad (\text{III-65})$$

III.8.2.2 Resetting of log scale factor of the high-band adaptive quantizer

At the first received frame, the high-band log scale factor is reset to the running mean of received frames prior to the loss:

$$\nabla_H(n-1) \leftarrow \nabla_{H,m}(n-1) \quad (\text{III-66})$$

III.8.2.3 Convergence of log scale factor of the high-band adaptive quantizer

The convergence of the high-band log-scale factor after frame loss is controlled by the measure of stationarity, $\nabla_{H,chg}(n)$, prior to the frame loss. For stationary cases, an adaptive low-pass filter is applied to $\nabla_H(n)$ after packet loss. The low-pass filter is applied over either 0 ms, 40 ms, or 80 ms, during which the degree of low-pass filtering is gradually reduced. The duration in samples, N_{LP,∇_H} , is determined according to:

$$N_{LP,\nabla_H} = \begin{cases} 640 & \nabla_{H,chg} < 819 \\ 320 & \nabla_{H,chg} < 1311 \\ 0 & \nabla_{H,chg} \geq 1311 \end{cases} \quad (\text{III-67})$$

The low-pass filtering is given by:

$$\nabla_{H,LP}(n) = \alpha_{LP}(n)\nabla_{H,LP}(n-1) + (1-\alpha_{LP}(n))\nabla_H(n) \quad (\text{III-68})$$

where the coefficient is given by:

$$\alpha_{LP}(n) = 1 - \left(\frac{n+1}{N_{LP,\nabla_H} + 1} \right)^2 \quad n = 0, 1, \dots, N_{LP,\nabla_H} - 1 \quad (\text{III-69})$$

Hence, the low-pass filtering reduces sample by sample with the time n . The low-pass filtered log scale factor simply replaces the regular log scale factor during the N_{LP,∇_H} samples.

III.8.3 Low-band pole section

An entity referred to as the stability margin (of the pole section) is updated during received frames for the low-band ADPCM decoder and used to constrain the pole section following frame loss.

III.8.3.1 Stability margin of low-band pole section

The stability margin of the low-band pole section is defined as:

$$\beta_L(n) = 1 - |a_{L,1}(n)| - a_{L,2}(n) \quad (\text{III-70})$$

where $a_{L,1}(n)$ and $a_{L,2}(n)$ are the two pole coefficients. A moving average of the stability margin is updated according to:

$$\beta_{L,MA}(n) = 15/16 \cdot \beta_{L,MA}(n-1) + 1/16 \cdot \beta_L(n) \quad (\text{III-71})$$

during received frames.

During lost frames, the moving average is not updated:

$$\beta_{L,MA}(n) = \beta_{L,MA}(n-1) \quad (\text{III-72})$$

III.8.3.2 Constraint on low-band pole section

During regular G.722 low-band (and high-band) ADPCM encoding and decoding, a minimum stability margin of $\beta_{L,min} = 1/16$ is maintained. During the first 40 ms after a frame loss, an increased minimum stability margin is maintained for the low-band ADPCM decoder. It is a function of both the time since the frame loss and the moving average of the stability margin.

For the first three 10-ms frames, a minimum stability margin of:

$$\beta_{L,\min} = \min\{3/16, \beta_{L,MA}(n-1)\} \quad (\text{III-73})$$

is set at the frame boundary and imposed throughout the frame. At the frame boundary into the fourth 10-ms frame, a minimum stability margin of:

$$\beta_{L,\min} = \min\left\{2/16, \frac{1/16 + \beta_{L,MA}(n-1)}{2}\right\} \quad (\text{III-74})$$

is imposed, while the regular minimum stability margin of $\beta_{L,\min} = 1/16$ is imposed for all other frames.

III.8.4 High-band partially reconstructed and reconstructed signals

During all frames, both lost and received, high-pass filtered versions of the high-band partially reconstructed signal, $p_H(n)$, and reconstructed signal, $r_H(n)$, are maintained:

$$p_{H,HP}(n) = 0.97[p_H(n) - p_H(n-1) + p_{H,HP}(n-1)] \quad (\text{III-75})$$

$$r_{H,HP}(n) = 0.97[r_H(n) - r_H(n-1) + r_{H,HP}(n-1)] \quad (\text{III-76})$$

This corresponds to a 3-dB cut-off at about 40 Hz, basically DC removal.

During the first 40 ms after frame loss, the regular partially reconstructed signal and regular reconstructed signal are substituted with their respective high-pass filtered versions for the purpose of high-band pole section adaptation and high-band reconstructed output, respectively.

III.9 Time lag computation

The re-phasing (clause III.10) and time-warping (clause III.11) techniques require that the number of samples by which the concealment waveform $x_{PLC}(j)$ and the signal in the first received frame are misaligned be known.

III.9.1 Low-complexity estimate of the lower sub-band reconstructed signal

The signal used to calculate the time lag in the first received frame is obtained by filtering the lower sub-band truncated difference signal, $d_{Lt}(n)$ (see Equation 3-11 of [ITU-T G.722]) with the pole-zero filter coefficients ($a_{Lpwe,i}(159)$, $b_{Lpwe,i}(159)$) and other required state information obtained from $STATE_{159}$ (see clause III.10.1):

$$r_{Le}(n) = \sum_{i=1}^2 a_{Lpwe,i}(159) \cdot r_{Le}(n-i) + \sum_{i=1}^6 b_{Lpwe,i}(159) \cdot d_{Lt}(n-i) + d_{Lt}(n) \quad , n = 0, 1, \dots, 79 \quad (\text{III-77})$$

III.9.2 Determination of re-phasing and time-warping requirement

If the last received frame is unvoiced, as indicated by the value of merit, the time lag T_L is set to zero:

$$IF \text{ merit} \leq MLO, T_L = 0 \quad (\text{III-78})$$

Likewise, if the first received frame is unvoiced, as indicated by the normalized first autocorrelation coefficient:

$$r(1) = \frac{\sum_{n=0}^{78} r_{Le}(n) \cdot r_{Le}(n)}{\sum_{n=0}^{78} r_{Le}(n) \cdot r_{Le}(n+1)} \quad (\text{III-79})$$

the time lag is set to zero:

$$IF\ r(1) < 0.125, T_L = 0 \quad (\text{III-80})$$

Otherwise, the time lag is computed as explained in the following clause.

III.9.3 Computation of the time lag

Computation of the time lag involves the following steps:

- 1) generation of the extrapolated signal;
- 2) coarse time lag search;
- 3) refined time lag search.

These steps are described in the following subclauses.

III.9.3.1 Generation of the extrapolated signal

The time lag represents the misalignment between $x_{PLC}(j)$ and $r_{Le}(n)$. To compute the misalignment, $x_{PLC}(j)$ is extended into the first received frame and a normalized cross-correlation function is maximized. This clause describes how $x_{PLC}(j)$ is extrapolated and specifies the length of signal that is needed. As in clause III.6, it is assumed that $x_{PLC}(j)$ is copied into the $x_{out}(j)$ buffer. Since this is a Type-5 frame (first received frame), the assumed correspondence is:

$$x_{out}(j-160) = x_{PLC}(j), \quad j = 0, 1, \dots, 159 \quad (\text{III-81})$$

The range over which the correlation is searched is given by:

$$\Delta_{TL} = \min(\lfloor ppfe \cdot 0.5 + 0.5 \rfloor + 3, \Delta_{TLMAX}) \quad (\text{III-82})$$

where $\Delta_{TLMAX} = 28$ and $ppfe$ is the pitch period for periodic waveform extrapolation used in the generation of $x_{PLC}(j)$.

The window size (at 16-kHz sampling) for the lag search is given by:

$$LSW_{16k} = \begin{cases} 80 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor < 80 \\ 160 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor > 160 \\ \lfloor ppfe \cdot 1.5 + 0.5 \rfloor & otherwise \end{cases} \quad (\text{III-83})$$

It is useful to specify the lag search window, LSW , at 8-kHz sampling as:

$$LSW = \lfloor LSW_{16k} \cdot 0.5 \rfloor \quad (\text{III-84})$$

Given the above, the total length of the extrapolated signal that needs to be derived from $x_{PLC}(j)$ is given by:

$$L = 2 \cdot (LSW + \Delta_{TL}) \quad (\text{III-85})$$

The starting position of the extrapolated signal in relation to the first sample in the received frame is:

$$D = 12 - \Delta_{TL} \quad (\text{III-86})$$

The extrapolated signal $es(j)$ is constructed according to the following:

If $D < 0$

$$es(j) = x_{out}(D + j) \quad j = 0, 1, \dots, -D - 1$$

If $(L + D \leq ppfe)$

$$es(j) = x_{out}(-ppfe + D + j) \quad j = -D, -D + 1, \dots, L - 1$$

Else

$$es(j) = x_{out}(-ppfe + D + j) \quad j = -D, -D + 1, \dots, ppfe - D - 1$$

$$es(j) = es(j - ppfe) \quad j = ppfe - D, ppfe - D + 1, \dots, L - 1$$

Else

$$ovs = ppfe \cdot \lceil D / ppfe \rceil - D$$

If $(ovs \geq L)$

$$es(j) = x_{out}(-ovs + j) \quad j = 0, 1, \dots, L - 1$$

Else

If $(ovs > 0)$

$$es(j) = x_{out}(-ovs + j) \quad j = 0, 1, \dots, ovs - 1$$

If $(L - ovs \leq ppfe)$

$$es(j) = x_{out}(-ovs - ppfe + j) \quad j = ovs, ovs + 1, \dots, L - 1$$

Else

$$es(j) = x_{out}(-ovs - ppfe + j) \quad j = ovs, ovs + 1, \dots, ovs + ppfe - 1$$

$$es(j) = es(j - ppfe) \quad j = ovs + ppfe, ovs + ppfe + 1, \dots, L - 1$$

III.9.3.2 Coarse time lag search

A coarsely estimated time lag, T_{LSUB} , is first computed by searching for the peak of the sub-sampled normalized cross-correlation function $R_{SUB}(k)$:

$$R_{SUB}(k) = \frac{\sum_{i=0}^{LSW/2-1} es(4i - k + \Delta_{TL}) \cdot r_{Le}(2i)}{\sqrt{\sum_{i=0}^{LSW/2-1} es^2(4i - k + \Delta_{TL}) \sum_{i=0}^{LSW-1} r_{Le}^2(2i)}}, \quad k = -\Delta_{TL}, -\Delta_{TL} + 4, -\Delta_{TL} + 8, \dots, \Delta_{TL} \quad (\text{III-87})$$

To avoid searching out of bounds during refinement, T_{LSUB} may be adjusted as follows:

$$\text{If } (T_{LSUB} > \Delta_{TLMAX} - 4) \quad T_{LSUB} = \Delta_{TLMAX} - 4 \quad (\text{III-88})$$

$$\text{If } (T_{LSUB} < -\Delta_{TLMAX} + 4) \quad T_{LSUB} = -\Delta_{TLMAX} + 4 \quad (\text{III-89})$$

III.9.3.3 Refined time lag search

The search is then refined to give the time lag, T_L , by searching for the peak of $R(k)$ given by:

$$R(k) = \frac{\sum_{i=0}^{LSW-1} es(2i - k + \Delta_{TL}) \cdot r_{Le}(i)}{\sqrt{\sum_{i=0}^{LSW-1} es^2(2i - k + \Delta_{TL}) \sum_{i=0}^{LSW-1} r_{Le}^2(i)}}, \quad k = -4 + T_{LSUB}, -2 + T_{LSUB}, \dots, 4 + T_{LSUB} \quad (\text{III-90})$$

Finally, the following conditions are checked:

If:

$$\sum_{i=0}^{LSW-1} r_{Le}^2(i) = 0 \quad (\text{III-91})$$

or:

$$\sum_{i=0}^{LSW-1} es(2i - T_L + \Delta_{TL}) \cdot r_{Le}(i) \leq 0.25 \cdot \sqrt{\sum_{i=0}^{LSW-1} r_{Le}^2(i)} \quad (\text{III-92})$$

or:

$$(T_L > \Delta_{TLMAX} - 2) \parallel (T_L < -\Delta_{TLMAX} + 2) \quad (\text{III-93})$$

then:

$$T_L = 0$$

III.10 Re-phasing

Re-phasing is the process of setting the internal states to a point in time where the lost frame concealment waveform $x_{PLC}(j)$ is in phase with the last input signal sample immediately before the first received frame. The re-phasing can be broken down into the following steps:

- 1) Store intermediate G.722 states during re-encoding of lost frames.
- 2) Adjust re-encoding according to the time lag.
- 3) Update QMF synthesis filter memory.

The following subclauses describe these steps in more detail.

III.10.1 Storage of intermediate G.722 states during re-encoding

As described in clause III.7, the reconstructed signal $x_{PLC}(j)$ is re-encoded during lost frames to update the G.722 decoder state memory. Let $STATE_j$ be the G.722 state and PLC state after re-encoding the j th sample of $x_{PLC}(j)$. Then, in addition to the G.722 state at the frame boundary that would normally be maintained (i.e., $STATE_{159}$), the $STATE_{159-\Delta_{TLMAX}}$ is also stored. To facilitate re-phasing, the sub-band signals $x_L(n)$, $x_H(n)$, $n = 69 - \Delta_{TLMAX}/2 \dots 79 + \Delta_{TLMAX}/2$ are also stored.

III.10.2 Adjustment of the re-encoding according to the time lag

Depending on the sign of the time lag, the procedure for adjustment of the re-encoding is as follows:

If $\Delta_{TL} > 0$:

- 1) Restore the G.722 state and PLC state to $STATE_{159-\Delta_{TLMAX}}$.
- 2) Re-encode $x_L(n)$, $x_H(n)$, $n = 80 - \Delta_{TLMAX}/2 \dots 79 - \Delta_{TL}/2$, according to clauses III.7.2 and III.7.3.

If $\Delta_{TL} < 0$:

- 1) Restore the G.722 state and PLC state to $STATE_{159}$.
- 2) Re-encode $x_L(n)$, $x_H(n)$, $n = 80 \dots 79 + |\Delta_{TL}/2|$, according to clauses III.7.2 and III.7.3.

Note that to facilitate re-encoding of $x_L(n)$ and $x_H(n)$ up to $n = 79 + |\Delta_{TL}/2|$, samples of $x_{PLC}(j)$ are required up to $\Delta_{TLMAX} + 182$.

III.10.3 Update of QMF synthesis filter memory

At the first received frame, the QMF synthesis filter memory needs to be calculated since the QMF synthesis filterbank is inactive during lost frames due to the PLC taking place in the 16-kHz output speech domain. Time-wise, the memory would generally correspond to the last samples of the last lost frame. However, the re-phasing needs to be taken into account. According to [ITU-T G.722], the QMF synthesis filter memory is given by:

$$x_d(i) = r_L(n-i) - r_H(n-i), \quad i = 1, 2, \dots, 11 \quad (\text{III-94})$$

$$x_s(i) = r_L(n-i) + r_H(n-i), \quad i = 1, 2, \dots, 11 \quad (\text{III-95})$$

as the first two output samples of the first received frame is calculated as:

$$x_{out}(j) = 2 \sum_{i=0}^{11} h_{2i} \cdot x_d(i) \quad (\text{III-96})$$

$$x_{out}(j+1) = 2 \sum_{i=0}^{11} h_{2i+1} \cdot x_s(i) \quad (\text{III-97})$$

The filter memory, i.e., $x_d(i)$ and $x_s(i)$, $i = 1, 2, \dots, 11$, is calculated from the last 11 samples of the re-phased input to the simplified subband ADPCM encoders during re-encoding, $x_L(n)$ and $x_H(n)$, $n = 69 - \Delta_{TL}/2, 69 - \Delta_{TL}/2 + 1, \dots, 79 - \Delta_{TL}/2$, i.e., the last samples up till the re-phasing point:

$$x_d(i) = x_L(80 - \Delta_{TL}/2 - i) - x_H(80 - \Delta_{TL}/2 - i), \quad i = 1, 2, \dots, 11 \quad (\text{III-98})$$

$$x_s(i) = x_L(80 - \Delta_{TL}/2 - i) + x_H(80 - \Delta_{TL}/2 - i), \quad i = 1, 2, \dots, 11 \quad (\text{III-99})$$

where $x_L(n)$ and $x_H(n)$ have been stored in state memory during the lost frame.

III.11 Time-warping

Time-warping is the process of stretching or shrinking a signal along the time axis. This clause describes how $x_{out}(j)$ is time-warped to improve alignment with the periodic waveform of the extrapolated signal $x_{PLC}(j)$. The algorithm described in the following subclauses is only executed if $T_L \neq 0$.

III.11.1 Time lag refinement

The time lag, T_L , is refined for time-warping by maximizing the cross-correlation in the overlap-add window. The estimated starting position of the overlap-add window within the first received frame based on T_L is given by:

$$SP_{OLA} = \max(0, MIN_UNSTBL - T_L) \quad (\text{III-100})$$

where $MIN_UNSTBL = 16$.

The starting position of the extrapolated signal in relation to SP_{OLA} is given by:

$$D_{ref} = SP_{OLA} - T_L - RSR \quad (\text{III-101})$$

where $RSR = 4$ is the refinement search range.

The required length of the extrapolated signal is given by:

$$L_{ref} = OLALG + RSR \quad (\text{III-102})$$

An extrapolated signal, $es_{tw}(j)$, is obtained using the same procedures as clause III.9.3.1, except $LSW = OLALG$, $L = L_{ref}$, and $D = D_{ref}$.

A refinement lag, T_{ref} , is computed by searching for the peak of the following:

$$R(k) = \frac{\sum_{i=0}^{OLALG-1} es_{tw}(i-k+RSR) \cdot x_{out}(i+SP_{OLA})}{\sqrt{\left(\sum_{i=0}^{OLALG-1} es_{tw}^2(i-k+RSR)\right) \cdot \left(\sum_{i=0}^{OLALG-1} x_{out}^2(i+SP_{OLA})\right)}}, k = -RSR, -RSR+1, \dots, RSR \quad (\text{III-103})$$

The final time lag used for time-warping is then obtained by:

$$T_{Lwarp} = T_L + T_{ref} \quad (\text{III-104})$$

III.11.2 Computation of time-warped $x_{out}(j)$ signal

The signal $x_{out}(j)$ is time-warped by T_{Lwarp} samples to form the signal $x_{warp}(j)$ which is later overlap-added with the waveform extrapolated signal $es_{ola}(j)$. Three cases, depending on the value of T_{Lwarp} , are illustrated in Figure III.9. In case a, $T_{Lwarp} < 0$ and $x_{out}(j)$ undergoes shrinking or compression. The first MIN_UNSTBL samples of $x_{out}(j)$ are not used in the warping to create $x_{warp}(j)$ and $xstart=MIN_UNSTBL$. In case b, $0 \leq T_{Lwarp} < MIN_UNSTBL$, and $x_{out}(j)$ is stretched by T_{Lwarp} samples. Again, the first MIN_UNSTBL samples of $x_{out}(j)$ are not used and $xstart=MIN_UNSTBL$. In case c, $T_{Lwarp} \geq MIN_UNSTBL$, and $x_{out}(j)$ is once more stretched by T_{Lwarp} samples. However, the first T_{Lwarp} samples of $x_{out}(j)$ are not needed in this case since an extra T_{Lwarp} sample will be created during warping; therefore, $xstart= T_{Lwarp}$.

In each case, the number of samples per add/drop is given by:

$$spad = \frac{(160 - xstart)}{|T_{Lwarp}|} \quad (\text{III-105})$$

The warping is implemented via a piece-wise single sample shift and triangular overlap-add, starting from $x_{out}[xstart]$. To perform shrinking, a sample is periodically dropped. From the point of sample drop, the original signal and the signal shifted left (due to the drop) are overlap-added. To perform stretching, a sample is periodically repeated. From the point of sample repeat, the original signal and the signal shifted to the right (due to the sample repeat) are overlap-added. The length of the overlap-add window, $L_{olawarp}$, (note that this is different from the OLA region depicted in Figure III.9) depends on the periodicity of the sample add/drop and is given by:

$$\begin{aligned} \text{If } T_{Lwarp} < 0, L_{olawarp} &= \frac{(160 - xstart - |T_{Lwarp}|)}{|T_{Lwarp}|} \\ \text{Else } L_{olawarp} &= \lceil spad \rceil \\ L_{olawarp} &= \min(8, L_{olawarp}) \end{aligned} \quad (\text{III-106})$$

The length of the warped input signal, x_{warp} , is given by:

$$L_{xwarp} = \min(160, 160 - MIN_UNSTBL + T_{Lwarp}) \quad (\text{III-107})$$

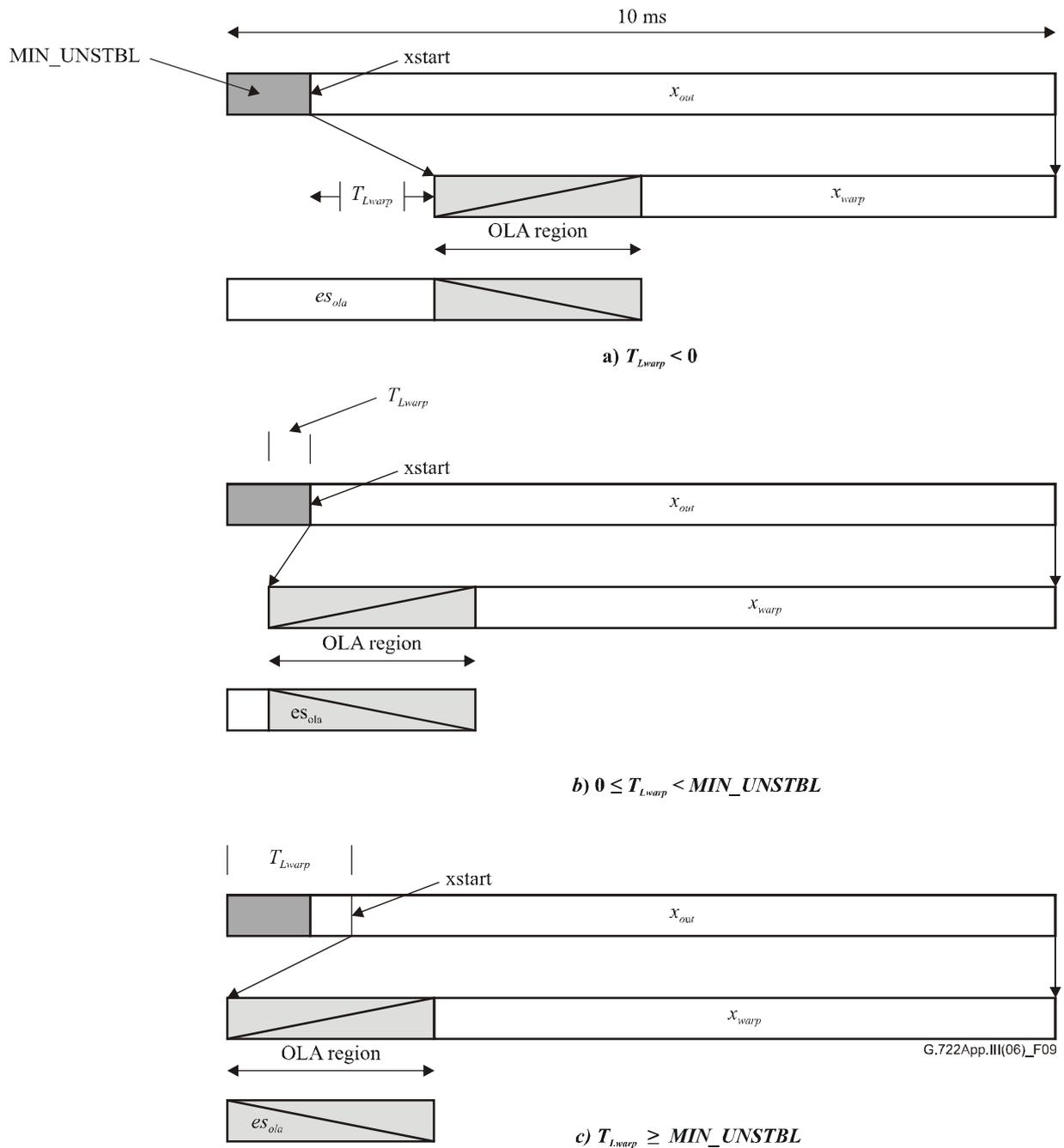


Figure III.9 – Three cases for warping of x_{out}

III.11.3 Computation of the waveform extrapolated signal

The warped signal $x_{warped}(j)$ and the extrapolated signal $es_{ola}(j)$ are overlap-added in the first received frame as shown in Figure III.9. The extrapolated signal $es_{ola}(j)$ is generated directly within the $x_{out}(j)$ signal buffer in a two-step process according to:

Step 1:

$$es_{ola}(j) = x_{out}(j) = ptf_e \cdot x_{out}(j - ppfe), \quad j = 0, 1, \dots, 160 - L_{xwarped} + 39 \quad (\text{III-108})$$

Step 2:

$$x_{out}(j) = x_{out}(j) \cdot w_i(j) + ring(j) \cdot w_o(j), \quad j = 0, 1, \dots, 39 \quad (\text{III-109})$$

where $w_i(j)$ and $w_o(j)$ (respectively) are triangular upward and downward ramping overlap-add windows of length 40 and $ring(j)$ is the ringing signal computed in clause III.6.12.

III.11.4 Overlap-add of time-warped signal with the waveform extrapolated signal

The extrapolated signal computed in clause III.11.3 is overlap-added with the warped signal $x_{warp}(j)$ according to:

$$x_{out}(160 - L_{xwarp} + j) = x_{out}(160 - L_{xwarp} + j) \cdot w_o(j) + x_{warp}(j) \cdot w_i(j), \quad j = 0, 1, \dots, 39 \quad (\text{III-110})$$

The remaining part of $x_{warp}(j)$ is then simply copied into the signal buffer:

$$x_{out}(160 - L_{xwarp} + j) = x_{warp}(j), \quad j = 40, 41, \dots, L_{xwarp} - 1 \quad (\text{III-111})$$

III.12 Bit-exact description of the G.722 PLC algorithm

The G.722 PLC is specified in a bit-exact manner by the fixed-point ANSI C-code. The C-code specification takes precedence over the text specification in this Recommendation in case of discrepancy. The fixed-point ANSI C-code is implemented using the G.722 C-code of the ITU-T G.191 Software Tool Library (STL2005) and version 2.3 of the 16-bit fixed-point operator library, also of STL2005.

III.12.1 Use of the simulation software

The decg722 executable (including PLC) is called by:

```
decg722 [-fsize N] g192_input_file speech_output_file
```

where N, the frame size, is a multiple of 160, corresponding to multiples of 10 ms.

Note that the Mode and frame size is indirectly embedded in the G.192 bit-stream; conflict with a command line-specified frame size will cause the decoder to use a frame size consistent with the G.192 bit-stream. For frame sizes of 10 ms and 20 ms, the decoder can uniquely determine both mode and frame size from the G.192 bit-stream. However, for frame sizes of 30 ms and longer, the frame size must be specified in a command line so as to allow the decoder to correctly determine both mode and frame size.

III.12.2 Organization of the simulation software

The source code is contained in the directory named "src". A Microsoft Visual C 6.0 workspace file is located in "workspace/VC6.0/". Opening g722_plc_g192.dsw will open the G.722 PLC C source code and project.

After compilation a simple test for bit-exact operation can be performed in the directory named "testplc". Executing the Pearl script named "testplc.pl" in the directory will execute the test.

Note that passing this simple test for bit-exactness only provides a simple check and is far from exhaustive in verifying an implementation for proper operation.

Table III.3 lists the new G.722 decoder state memory to support the PLC, and Table III.4 provides an overview of new tables.

Table III.3 – New G.722 decoder state memory (structure WB_PLC_state)

Member	Words (16-bit)	Description
energymax32	2	Energy
cormax	2	Correlation
wsz	1	Window size
scaled_flag	1	Scaling flag
xq	638	16-kHz speech output buffer
stsym1	8	Short-term synthesis filter memory
al	9	LPC filter coefficients
alast	9	Past LPC filter coefficients
ppt	1	Pitch predictor tap
stwpml	8	Short-term weighted all-pole filter memory
xwd	45	Down-sampled weighted speech buffer
xwd_exp	1	Exponent of down-sampled weighted speech buffer
dfm	60	Down-sampling filter memory
scaler	1	Scaling factor for random component
merit	1	Figure of merit for mixing ratio
ptfe	1	Pitch tap for frame erasure
ppf	1	Pitch period – "floating" point value
ppinc	1	Pitch period increment
pweflag	1	Periodic waveform extrapolation flag
cpplast	1	Coarse pitch period, last
pph	5	Pitch period history
pp	1	Pitch period
cfecount	1	Consecutive frame erasure counter
ngfae	1	Number of good frames after erasure
nfle	1	Number of frames in last erasure
avm	1	Average magnitude
lag	1	Time shift lag
psml_mean	1	Pole section margin, low-band mean
nbpl_mean1	1	nbpl first mean (low-band)
nbpl_mean2	1	nbpl second mean (low-band)
nbpl_trck	1	nbpl tracking (low-band)
nbpl_chng	1	nbpl change (low-band)
pl_postn	1	pl signal positive-negative measure (low-band)
lb_reset	1	Low-band decoder reset flag
nbph_mean	1	nbph mean (high-band)
nbph_trck	1	nbph tracking (high-band)
nbph_chng	1	nbph change (high-band)
nbh_mode	1	nbh mode for convergence (high-band)
hp_flag	1	Flag for hp filter on rh and ph signals (high-band)

Table III.3 – New G.722 decoder state memory (structure WB_PLC_state)

Member	Words (16-bit)	Description
nbph_lp	1	Low-pass filtered nbph (high-band)
ph_postn	1	ph signal positive-negative measure (high-band)
hb_reset	1	High-band decoder reset flag
rhhp_m1	1	Past sample of high-pass filtered rh signal (high-band)
rh_m1	1	Past sample of rh signal (high-band)
phhp_m1	1	Past sample of high-pass filtered ph signal (high-band)
ph_m1	1	Past sample of ph signal (high-band)
sb_sample	1	Sub-band sample number
cpl_postn	1	Copy of pl_postn
cph_postn	1	Copy of ph_postn
crhhp_m1	1	Copy of rhhp_m1
crh_m1	1	Copy of rh_m1
cphp_m1	1	Copy of phhp_m1
cph_m1	1	Copy of ph_m1
ds	104	Copy of regular G.722 decoder state
lb	39	Low-band signal
hb	39	High-band signal

Table III.4 – New G.722 decoder table ROM

Table	Words (16-bit)	Description
inv_frm_size	3	Inverse of frame size
wlil4rilil	9	Table for low-band scale factor update
q4	8	Table for low-band scale factor update
NGFAEOFFSET_P1	8	Sample offset into 10 ms frames
div_n	20	Table for division
gawd	4	Table for gradual muting
olaup	16	Table for overlap-add
oladown	16	Table for overlap-add
wn	127	Table of normalized noise samples
bdf	60	Filter for 8:1 decimation
x	4	For coarse pitch
x2	4	For coarse pitch
invk	4	For coarse pitch
MPTH	4	For coarse pitch
sstwin_h	8	Upper 16-bit for spectral smoothing
sstwin_l	8	Lower 16-bit for spectral smoothing
bwel	9	Bandwidth expansion
STWAL	9	For short-term weighting filter
win	160	Window for LPC analysis
tablog	33	Table for log2 function
olaug	40	Window for overlap-add
oladg	40	Window for overlap-add
nbphtab	8	Table for nbph
nbpltab	6	Table for nbpl
ola3	3	Table for overlap-add
ola4	4	Table for overlap-add
ola5	5	Table for overlap-add
ola6	6	Table for overlap-add
ola7	7	Table for overlap-add
ola8	8	Table for overlap-add

Table III.5 lists C-code source files of [ITU-T G.722] from [ITU-T G.191] that remain identical, Table III.6 lists files of [ITU-T G.722] that are modified, and Table III.7 names the new files of the G.722 PLC.

Table III.5 – G.722 identical source files

File name	Description
decg722.c	G.722 main decoder function
softbit.c/h	G.722 softbit functions
g722_com.h	Common G.722 definitions
ugstdemo.h	Definitions for UGST demo programs

Table III.6 – G.722 modified source files

File name	Description
funcg722.c/h	G.722 functions
g722.c	G.722 frame encoding and decoding functions

Table III.7 – G.722 PLC new source files

File name	Description
apfilter.c	All-pole filter functions
autocor.c	Autocorrelation function
azfilter.c	All-zero filter function
coarptch.c	Coarse pitch analysis
decim.c	8:1 decimation function
dspfunc.c	DSP functions
g722plc.c/h	G.722 PLC functions
levinson.c	Levinson-Durbin recursion function
memutil.c/h	Memory utility functions allow automatic scratch memory usage
merit.c	Merit calculation function
ppchange.c/h	Re-phasing and time-warping related functions
prfn.c	Pitch refinement function
table.c/h	Table ROM
utility.c/h	Utility functions

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems