



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.729

Annex E
(09/98)

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

Digital transmission systems – Terminal equipments –
Coding of analogue signals by methods other than PCM

Coding of speech at 8 kbit/s using Conjugate-
Structure Algebraic-Code-Excited Linear-Prediction
(CS-ACELP)

**Annex E: 11.8 kbit/s CS-ACELP speech coding
algorithm**

ITU-T Recommendation G.729 – Annex E

(Previously CCITT Recommendation)

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
INTERNATIONAL ANALOGUE CARRIER SYSTEM	
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
TESTING EQUIPMENTS	
TRANSMISSION MEDIA CHARACTERISTICS	
General	G.600–G.609
Symmetric cable pairs	G.610–G.619
Land coaxial cable pairs	G.620–G.629
Submarine cables	G.630–G.649
Optical fibre cables	G.650–G.659
Characteristics of optical components and sub-systems	G.660–G.699
DIGITAL TRANSMISSION SYSTEMS	
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

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

ITU-T RECOMMENDATION G.729

CODING OF SPEECH AT 8 kbit/s USING CONJUGATE-STRUCTURE ALGEBRAIC-CODE-EXCITED LINEAR-PREDICTION (CS-ACELP)

ANNEX E

11.8 kbit/s CS-ACELP speech coding algorithm

Summary

This Annex provides the high level description of the higher bit-rate extensions of Recommendation G.729 designed to accommodate wide range of input signals, such as speech, with background noise and even music.

Source

Annex E to ITU-T Recommendation G.729 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 25th of September 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The 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 Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 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.

INFORMATIVE NOTE

The use of this Recommendation could require licensing from the copyright holders in line with the provisions of the ITU Copyright Policy, which is currently in preparation and which may, once completed, require the revision of Recommendations referring to copyright matters.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The 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, the ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 1999

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
Annex E – 11.8 kbit/s CS-ACELP speech coding algorithm	1
E.1 Introduction	1
E.2 General description of the speech codec	1
E.2.1 Encoder	2
E.2.2 Decoder	2
E.2.3 Delay	3
E.2.4 Speech coder description.....	3
E.3 Functional description of the encoder	3
E.3.1 Pre-processing	3
E.3.2 Linear prediction analysis and quantization.....	3
E.3.3 Perceptual weighting	7
E.3.4 Open-loop pitch analysis.....	8
E.3.5 Computation of the impulse response	8
E.3.6 Computation of the target signals	8
E.3.7 Adaptive-codebook search	8
E.3.8 Fixed codebook structure and search	8
E.3.9 Quantization of the gains	12
E.3.10 Memory update	12
E.4 Functional description of the decoder	12
E.4.1 Parameter decoding procedure	14
E.4.2 Post-processing	15
E.4.3 Encoder and decoder initialization.....	17
E.4.4 Concealment of frame erasures.....	17
E.5 Bit-exact description of the CS-ACELP coder.....	20
E.5.1 Use of the simulation software.....	20
E.5.2 Organization of the simulation software	20
E.6 Normative references.....	22

Recommendation G.729

CODING OF SPEECH AT 8 kbit/s USING CONJUGATE-STRUCTURE ALGEBRAIC-CODE-EXCITED LINEAR-PREDICTION (CS-ACELP)

ANNEX E

11.8 kbit/s CS-ACELP speech coding algorithm

(Geneva, 1998)

E.1 Introduction

This Annex provides the high level description of the higher bit-rate extension of Recommendation G.729 designed to accommodate wide range of input signals, such as speech, with background noise and even music.

E.2 General description of the speech codec

The extension algorithm has been designed to limit as much as possible the modifications and additions brought to the original G.729 algorithm. The only actual additions to G.729 concern the LP part with the introduction of a backward LP analysis suited for music signals and stationary background noises and the design of two new algebraic excitation codebooks to extend the bit rate up to 11.8 kbit/s: one codebook is used in forward mode, the other one, larger, in backward mode. All the remaining procedures are strictly the same as in G.729 except some minor modifications to the postfiltering and perceptual weighting procedures. Error concealment has also been modified to be adapted to the backward/forward LP structure.

Two LP analyses are performed at the frame rate: one backward on the synthesis signal and one forward on the input signal. An adaptive decision procedure chooses the best filter and performs the switch if needed. The LP forward part of the algorithm is the same as the G.729 one with the same LSP quantization scheme. The backward LP analysis has an order of 30 and is performed both in the coder and in the decoder. Since the LP coefficients are not transmitted, the spare bit rate is used to increase the size of the algebraic excitation codebooks. One information bit is needed to indicate the LP mode and is protected by a parity bit. In the proposed extension, all the additional bit rate from 8 kbit/s to 11.8 kbit/s, except two bits (LP indication mode + parity bit), is used to increase the size of the algebraic codebooks. The bit allocation of the coder parameters is shown in Table E.1.

The backward/forward decision criterion enables to operate a real discrimination between speech (mainly coded in forward mode) and music (mainly coded in backward mode). The backward/forward procedure has been also designed to reduce the number of switches and to perform, when necessary, smooth switching between filters with no artefacts. The LP mode and the related information is used to better adapt postfiltering and perceptual weighting to either music or speech. This is also used for the error concealment.

In the following subclauses, a high level description of the 11.8 kbit/s extension of G.729 is provided. Only the modifications or additions to the G.729 algorithm will be described.

**Table E.1/G.729 – Bit allocation of the 11.8 kbit/s
CS-ACELP algorithm (10 ms frame)**

	EXTENSION AT 11.8 kbit/s	
LP mode indication bit	1 + 1 (parity)	
	FORWARD	BACKWARD
LP filter	18	0
LTP Delay ($1^{\text{st}}/2^{\text{nd}}$ sub-fr.)	8 + 1 (parity)/5	8 + 1 (parity)/5
EXC Codes ($1^{\text{st}}/2^{\text{nd}}$ sub-fr.)	35/35	44/44
Gains (LTP + EXC) ($1^{\text{st}}/2^{\text{nd}}$ sub-fr.)	7/7	7/7
Total	118	118
NOTE – The numbers of bits corresponding to modified parts of the structure (compared to G.729) are typed in bold.		

E.2.1 Encoder

In order to obtain this high quality with music while keeping a good robustness to transmission errors and avoiding degradation of less stationary signals and especially speech (compared with a pure forward structure used in G.729), a new technique called mixed backward/forward LP structure has been introduced. A criterion enables to choose the most suitable LP analysis given the stationarity of the input signal and the backward and forward filters prediction gains.

For music signals, generally very stationary, the LP backward mode is mainly used: the LP analysis is performed on the synthesis signal with no transmission of the coefficients with two benefits:

- The LP order is increased up to 30 coefficients which is far more suited for the complex spectrum of music signals (the 10 coefficients LP filter of LP forward codecs like G.729 is not sufficient for music).
- The bit rate is better allocated: no bit rate is wasted on successive very similar LP filters. All the spare bit rates are used to extend the size of the excitation codebook. An algebraic codebook with 44 bits is used for the fixed codebook excitation.

The weak points of pure backward LP analysis mainly concern the non-stationary signals with sharp spectrum transitions and the sensitivity to transmission errors. With the mixed LP backward/forward structure, if a spectrum transition occurs, the forward mode is selected and the 10 LP coefficients are coded and transmitted. Besides, even if backward mode is dominant, the transmission of forward LP filters clearly improves the robustness when compared with a pure backward structure.

In forward mode, the encoder is almost identical to G.729 with more bits allocated to the excitation codebooks. An algebraic codebook with 35 bits is used for the fixed codebook excitation.

E.2.2 Decoder

First, the parameter's indices are extracted from the received bit stream. These indices are decoded to obtain the coder parameters corresponding to a 10 ms speech frame. The first parameter decoded is the LP mode information and its parity bit. According to this information, the frame is classified either as forward, backward or erased. In forward mode, the parameters are the LSP coefficients, the two fractional pitch delays, the two forward fixed-codebook vectors, and the two sets of adaptive-and fixed-codebook gains. In backward mode, the parameters are the two fractional pitch delays, the two backward fixed-codebook vectors, and the two sets of adaptive-and fixed-codebook gains. First, the LP backward analysis is performed. Then, if the frame is in forward mode, the LSP coefficients are interpolated and converted to LP filter coefficients for each subframe. Except for the construction of fixed-codebook excitation, the decoding procedure is very similar to the G.729 decoding procedure.

Then, for each 5 ms subframe the following steps are done:

- the excitation is constructed by adding the adaptive-and fixed-codebook vectors scaled by their respective gains;
- the speech is reconstructed by filtering the excitation through the LP synthesis filter (either forward or backward);
- the reconstructed speech signal is passed through a post-processing stage, which includes an adaptive postfilter based on the long-term and short-term synthesis filters, followed by a high-pass filter and scaling operation. Compared with G.729, the weighting factors of the postfilter have been made adaptive.

E.2.3 Delay

The same as 2.3/G.729.

E.2.4 Speech coder description

The description of the speech coding algorithm of this Recommendation is made in terms of bit-exact, fixed-point mathematical operations. The ANSI C code indicated in E.5, which constitutes an integral part of this Recommendation, reflects this bit-exact, fixed-point descriptive approach. The mathematical descriptions of the encoder (subclause E.3), and decoder (subclause E.4), can be implemented in several other fashions, possibly leading to a codec implementation not complying with this Recommendation. Therefore, the algorithm description of the ANSI C code of E.5 shall take precedence over the mathematical descriptions of E.3 and E.4, whenever discrepancies are found. A non-exhaustive set of test signals, which can be used with ANSI C code, is available from the ITU.

E.3 Functional description of the encoder

In this subclause, the different functions of the encoder are described. The main body of this Recommendation is referred to in most of this subclause, except the parts where algorithmic modifications or additions have been carried out.

E.3.1 Pre-processing

The same as 3.1/G.729.

E.3.2 Linear prediction analysis and quantization

Two LP analyses are performed simultaneously at the 10 ms frame rate: one forward analysis on the input signal which is strictly the same as G.729 with also the same quantization scheme, and one backward analysis performed on the past synthesized signal.

E.3.2.1 Windowing and autocorrelation computation

- *Forward LP analysis*

The same as 3.2.1/G.729.

- *Backward LP analysis*

A hybrid recursive windowing scheme the same as in G.728 is used.

Let sample 1 be the more recent sample of the more recent synthesized frame,

and let the indices i represent past samples ordered so that oldest samples have highest indices. Samples $i = 1$ to 35 are windowed with the non-recursive part of the window:

$$w_{lpbwd}(i) = \sin(i * c_{lpbwd}), \quad i = 1, \dots, 35 \quad \text{where } c_{lpbwd} = 0.047783$$

The recursive part of the window is given by the function (samples > 35):

$$w_{lpbwd}(i) = b_{lpbwd} * a_{lpbwd}^{(i-36)}, \quad i > 35$$

$$\text{with } a_{lpbwd} = 0.9928337491 \text{ and } b_{lpbwd} = \sin(36 * c_{lpbwd})$$

The recursive calculation of the autocorrelation coefficients is performed as described in Recommendation G.728.

The same white noise correction factor as for forward LP analysis is applied to the first autocorrelation coefficient (1.0001), but the bandwidth expansion applied to the coefficients is reduced from 60 Hz (in G.729) to 5 Hz. A small additional spectral flattening is applied by a weighting function with $\gamma_{lpbwd} = 0.98$ on the LP coefficients (calculated in E.3.2.2).

E.3.2.2 Levinson Durbin algorithm

The algorithm used is the same for forward and backward analysis. Compared to the G.729 algorithm, the size of some arrays has been extended to cope with the higher LP order.

E.3.2.3 LP to LSP conversion

For forward LP filter, the same as 3.2.3/G.729. For backward LP filter, no LSP calculation is needed.

E.3.2.4 Quantization of LSP coefficients

The same as 3.2.4/G.729 for forward LSP coefficients. For backward LP filter, no LSP quantization is needed.

E.3.2.5 Interpolation of LP coefficients

- *For the forward LP analysis*

As in 3.2.5/G.729, the quantized (and unquantized) LP coefficients are used for the second subframe. For the first subframe, the forward quantized (and unquantized) LSP coefficients are interpolated as in 3.2.5/G.729 when the previous frame is in forward mode. When the previous frame is in backward mode, no interpolation is performed, the second subframe quantized (and unquantized) LP filter is also used for the first subframe.

- *For the backward LP analysis*

For the second subframe, either the current backward LP filter A_{bwd} computed in E.3.2.2 or a transition filter, as will be described in E.3.2.7.1, is used.

For the first subframe, the LP filter coefficients are directly interpolated with the same interpolation factors than G.729 (0.5,0.5) between the second subframe backward LP filter and the previous frame filter.

E.3.2.6 LSP to LP conversion

For forward LP filter, the same as 3.2.6/G.729. For backward LP filter, no conversion is needed.

E.3.2.7 Backward/forward decision and switch procedure

E.3.2.7.1 Switching procedure

This subclause describes how the switch is performed from a previous frame using a forward (respectively backward) filter to the current one where the backward (respectively forward) filter is chosen in order to avoid artefacts in the synthesized signal.

– *From forward LP filter to backward LP filter*

This generally occurs when the signal is stationary. It is consequently important to avoid any filter transition which would bring an audible artificial spectrum transition in the synthesized signal. To achieve this, the following interpolation is performed both at the encoder and at the decoder:

If switch is decided at frame n :

Let $A_{fwd}(n-1)$ be the forward LP filter at frame $n-1$.

Let $A_{bwd}(n)$ be the backward LP filter at current frame n computed in E.3.2.2.

The LP filter A used at frame $n+i$ is given by:

$$A(n+i) = 0.1 * i * A_{bwd}(n+i) + (1-0.1 * i) * A(n+i-1), \quad 0 \leq i \leq 9$$

$$A(n+i) = A_{bwd}(n+i), \quad i \geq 10$$

$$\text{with } A(n-1) = A_{fwd}(n-1)$$

After 10 transition frames, the filter used is exactly the backward filter.

– *From backward filter to forward filter*

This occurs when a spectrum transition exists in the input signal. No smoothing is then performed:

If switch is decided at frame n : $A(n) = A_{fwd}(n)$

E.3.2.7.2 The global stationarity indicator and high stationarity indicator

The global stationarity indicator at frame n (called $\text{Stat}(n)$) characterizes the global stationarity of the input signal. Calculated at frame n after the backward/forward decision has been taken, it will be used for the next frame ($n+1$) backward/forward decision calculated frame-by-frame to reduce the number of switches between filters. The principle is to progressively favour one mode according to the stationarity of the input signal and to reduce the number of switches to the other mode.

The computation of this indicator is based on the history of the backward/forward decisions and on the backward and forward filters prediction gains. It varies from a value representing a high stationarity of the input signal (value 32 000) to a value representing a low stationarity (value 0).

This indicator has slow frame-by-frame variations (with the given numerical values, it takes at least 80 frames to vary from min. to max.).

The adaptation depicted below is only performed for frames the energy of which is greater than 40 dB. For other frames that are considered as silence frames, $\text{Stat}(n)$ is equal to $\text{Stat}(n-1)$ bounded by 13 000.

– The first step of the adaptation is based on the preceding switch decisions:

Let n be the index of the current frame.

Let $N_{bwd}(n)$ be the number of consecutive backward frames measured at frame n .

If frame n is a forward frame, then $N_{bwd}(n)$ is equal to 0.

Let the value $\text{Stat}^1(n)$ represent the output of the first step stationarity evaluation.

If frame n is a backward to forward transition frame (i.e. frame $n-1$ is backward and n is forward) and if less than 20 consecutive backward frames have occurred:

$$\text{Stat}^1(n) = \text{Stat}(n-1) - (5000 - 250 * N_{bwd}(n-1))$$

else:

$$\text{if } (N_{bwd}(n) > 20) \text{Stat}^1(n) = \text{Stat}(n-1) + 500$$

else:

if ($N_{\text{bwd}}(n) = 20$) $\text{Stat}^1(n) = \text{Stat}(n - 1) + 2500$

else: $\text{Stat}^1(n) = \text{Stat}(n - 1)$

– The second step of the adaptation is based on the prediction gains:

Let x be the difference between the backward LP filter prediction gain and the forward LP filter prediction gain: $x = G_{\text{pred}_b} - G_{\text{pred}_f}$ (in dB).

$\text{Stat}(n) = \text{Stat}^1(n) + \Delta(x)$ with:

If $\text{Stat}^1(n) < 13\,000$,

$$\Delta(x) = \begin{cases} 3200 & \text{if } x > 4 \\ 2400 & \text{if } x \in]3, 4] \\ 1600 & \text{if } x \in]2, 3] \\ 800 & \text{if } x \in]1, 2] \\ 400 & \text{if } x \in]0, 1] \\ 0 & \text{if } x \in [-1, 0] \\ -400 & \text{if } x \in [-2, -1[\\ -800 & \text{if } x \in [-3, -2[\\ -1600 & \text{if } x \in [-4, -3[\\ -3200 & \text{if } x \in [-4.7, -4[\\ -6400 & \text{if } x < -4.7 \end{cases}$$

else:

$$\Delta(x) = \begin{cases} 0 & \text{if } x \geq -1 \\ -400 & \text{if } x \in [-2, -1[\\ -800 & \text{if } x \in [-3, -2[\\ -1600 & \text{if } x \in [-4, -3[\\ -3200 & \text{if } x \in [-4.7, -4[\\ -6400 & \text{if } x < -4.7 \end{cases}$$

A high stationarity state is also determined with the parameter value `High_Stat` set to 1. This high stationarity state is detected when the percentage of backward frames becomes significantly higher than the percentage of forward frames:

Let N_{bwd} (respectively N_{fwd}) represent the number of backward (respectively forward) frames in the previous $N_{f/b}$ frames ($N_{f/b} = N_{\text{bwd}} + N_{\text{fwd}}$). For the first 100 frames, N_{bwd} (respectively N_{fwd}) is the actual number backward (respectively forward) frames in the previous $N_{f/b}$ frames. Then whenever $N_{f/b}$ reaches the value 100, $N_{f/b}$, N_{bwd} , N_{fwd} are divided by 2.

If $N_{f/b} < 10$, `High_Stat` = 0

else:

If $N_{\text{bwd}} > 4 * N_{\text{fwd}}$ then `High_Stat` = 1

else: `High_Stat` = 0

This procedure is only performed for frames the energy of which is greater than 40 dB. For silence frames, $N_{f/b}$, N_{bwd} , N_{fwd} and High_Stat are not updated.

E.3.2.7.3 Backward/forward decision procedure

At current frame n , the backward forward decision is taken according to 4 criteria which apply sequentially.

– *1st criterion on prediction gains*

The prediction gains (in dB) of the backward, backward interpolated (different from backward only during forward-to-backward transitions) and forward LP filters are computed (called respectively G_{pred_b} and $G_{pred_{int}}$ and G_{pred_f}).

Let Gap be an adaptive decision threshold (in dB).

The first stage decision is:

The backward LP filter is selected if the following condition is verified:

$(G_{pred_{int}} > G_{pred_f} - \text{Gap})$ and $(G_{pred_b} > G_{pred_f} - \text{Gap})$ and $(G_{pred_b} > 0)$ and $(G_{pred_{int}} > 0)$

Otherwise, the forward LP filter is selected.

The Gap parameter is adapted according to the stationarity indicator:

$\text{Gap}(n) = 0.0366 * (\text{Stat}(n - 1)/320) + 1.0$ with $\text{Stat}(n - 1) \in [0, 32\ 000]$ (see E.3.2.7.2).

– *2nd criterion using the global stationarity indicator*

While the value of the global stationarity indicator $\text{Stat}(n - 1) \in [0, 32\ 000]$ remains below 13 000, the forward LP mode is selected (second stage decision). This avoids unnecessary switches to backward mode with speech or other signals of low or medium stationarity.

– *3rd criterion on LSP*

In order to avoid any artificial transition when the short term spectrum is stationary, the following Euclidean distance is computed between the LSP vectors of two successive forward LP filters:

\mathbf{LSP}_n is the LSP vector of the forward LP filter calculated at current frame n .

\mathbf{LSP}_{n-1} is the LSP vector of the forward LP filter calculated at frame $n - 1$.

$d_{LSP}(n) = \|\mathbf{LSP}_n, \mathbf{LSP}_{n-1}\|^2$ is the Euclidean distance between both vectors.

If $d_{LSP}(n) < \text{Thresh}_{LSP}(n)$, if the previous frame is in backward mode and if the prediction gains G_{pred_b} and $G_{pred_{int}}$ are positive, switching from backward to forward is forbidden (selection of backward mode as a third stage decision in this case).

Thresh_{LSP} is adapted at each frame according to the value of $\text{Stat}(n - 1)$:

If $\text{Stat}(n - 1) = 32\ 000$ (max. value), $\text{Thresh}_{LSP}(n) = 0.03$

Else $\text{Thresh}_{LSP}(n) = 0$

– *4th criterion on the energy*

In order to increase the robustness of the algorithm to transmission errors, the forward LP filter is imposed for frames with energy below 40 dB.

E.3.3 Perceptual weighting

The perceptual weighting filter is given by: $W(z) = \frac{A(z / \gamma_1)}{A(z / \gamma_2)}$

- *In Forward mode*
The parameters γ_1 and γ_2 are computed as in 3.3/G.729. The LP filter $A(z)$ is the unquantized forward filter $A_{\text{fwd}}(z)$ when High_Stat is equal to 0 and the quantized forward filter otherwise.
- *In backward mode*
If no High_Stat state is detected, the LP filter $A(z)$ used is the unquantized forward filter, else it is the backward calculated filter.
The parameters γ_1 and γ_2 take fixed values depending on the high stationarity indicator (High_Stat).
In case of high stationarity of the input signal (High_Stat = 1), the noise masking effect is reinforced:

$$\gamma_{1\text{bwdh}} = 0.98$$

$$\gamma_{2\text{bwdh}} = 0.4$$
In case of normal stationarity (High_Stat = 0):

$$\gamma_{1\text{bwdl}} = 0.9$$

$$\gamma_{2\text{bwdl}} = 0.4$$
- The weighed speech is calculated as indicated in equation (33) of 3.3/G.729, the filtering order depending on the selected weighting filter chosen (10 or 30).

E.3.4 Open-loop pitch analysis

The same as 3.4/G.729.

E.3.5 Computation of the impulse response

Similar to 3.5/G.729. (The order of the LP filters could be 30 instead of 10.)

E.3.6 Computation of the target signals

Similar to 3.6/G.729. (The order of the LP filters could be 30 instead of 10.)

E.3.7 Adaptive-codebook search

The adaptive-codebook search, the generation of the adaptive-codebook vector, the codeword computation for the delay index P1 and P2 and the computation of the adaptive-codebook gain are identical to the procedure described in 3.7/G.729. The parity bit P0 is computed on the seven (instead of six in G.279) most significant bits of the delay index P1 of the first subframe.

E.3.8 Fixed codebook structure and search

E.3.8.1 Fixed codebook in forward LP mode

In the forward LP mode, an algebraic codebook with 35 bits is used as the fixed codebook. In this codebook, each excitation vector contains 10 non-zero pulses. The pulse amplitudes are either -1 or $+1$. The 40 positions in each subframe are divided into 5 tracks where each track contains two pulses. In the design, the two pulses for each track may overlap resulting in a single pulse with amplitude $+2$ or -2 . The allowed positions for pulses are shown in Table E.2.

Table E.2/G.729 – Structure of fixed codebook in forward mode \mathcal{C}_{fwd}

Track	Pulses	Signs	Positions
1	$p0, p1$	$s_0, s_1: \pm 1$	0, 5, 10, 15, 20, 25, 30, 35
2	$p2, p3$	$s_2, s_3: \pm 1$	1, 6, 11, 16, 21, 26, 31, 36
3	$p4, p5$	$s_4, s_5: \pm 1$	2, 7, 12, 17, 22, 27, 32, 37
4	$p6, p7$	$s_6, s_7: \pm 1$	3, 8, 13, 18, 23, 28, 33, 38
5	$p8, p9$	$s_8, s_9: \pm 1$	4, 9, 14, 19, 24, 29, 34, 39

Similar to G.729, the selected codebook vector is filtered through the pre-filter:

$$P(z) = 1 / (1 - \beta z^{-T})$$

to enhance the harmonic components. The way β is adapted is the same as in the main body of Recommendation G.729.

E.3.8.1.1 Search procedure of the 35-bit codebook

The fixed codebook is searched by minimizing the mean-squared error between the weighted input speech and the weighted reconstructed speech. If $c_k(n)$ is the algebraic codevector at index k , $h(n)$ is the impulse response of the weighted synthesis filter, and $d(n)$ is the correlation between the target vector and $h(n)$, then the algebraic codebook is searched by maximizing the criterion:

$$T_k = \frac{(C_k)^2}{E_k},$$

where C is the correlation between $c_k(n)$ and $d(n)$ and E is the energy of the filtered codevector ($c_k(n) * h(n)$). Since the algebraic codevector contains few non-zero pulses, the correlation can be written as:

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

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

$$E = \sum_{i=0}^{N_p-1} \phi(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} s_i s_j \phi(m_i, m_j),$$

where $\phi(i, j)$ contains the correlations between $h(n-i)$ and $h(n-j)$. The signal $d(n)$ and the correlations $\phi(i, j)$ are computed before the codebook search.

Similar to G.729, in order to speed up the search procedure, the pulse amplitudes are pre-set outside the closed-loop search using the so-called *signal-selected pulse amplitude* approach. In this approach, the most likely amplitude of a pulse occurring at a certain position is estimated using a certain side information signal. In G.729, the signal $d(n)$ is used for pre-selecting the pulse amplitudes. In this bit rate extension, a signal $b(n)$, which is a weighted sum of the normalized $d(n)$ vector and the normalized long-term prediction residual, is used.

The signal $b(n)$ is given by:

$$b(n) = d(n) / \sigma_d + e(n) / \sigma_e$$

where $e(n)$ is the long-term prediction residual and σ_d and σ_e are the r.m.s. values of $d(n)$ and $e(n)$, respectively. The sign of a pulse at a certain position is set a priori equal to the sign of $b(n)$ at that position. The sign information is incorporated into the signals $d(n)$ and $\phi(i,j)$ before starting the search for the best pulse positions, similar to G.729.

The optimal pulse positions are determined using a non-exhaustive analysis-by-synthesis search procedure. The used procedure is a special case of a general depth-first tree search method which is efficient for searching huge codebooks with a reasonable complexity. In this approach, the N_p excitation pulses are partitioned into M subsets of N_m pulses. The search begins with subset 1 and proceeds with subsequent subsets according to a tree structure whereby subset m is searched at the m th level of the tree. The search is repeated by changing the order in which the pulses are assigned to the position tracks. In this particular codebook structure, the pulses are partitioned into 5 subsets of 2 pulses (the tree has 5 levels).

The pulse positions are determined as follows:

For each of the five tracks, the pulse positions with maximum absolute values of $d(n)$ are found. From these, the two successive tracks, T_{k_0} and $T_{(k_0+1) \bmod 5}$ with the largest combined maxima are determined. This index k_0 is used for the initial assignment of pulses to tracks. Then the two successive tracks, T_{k_1} and $T_{(k_1+1) \bmod 5}$ with the second largest combined maxima and the two successive tracks, T_{k_2} and $T_{(k_2+1) \bmod 5}$ with the third largest combined maxima are also determined.

In the first iteration, the pulses are assigned to the tracks as follows: the pulses i_n , $n = 0, \dots, 9$, are assigned to tracks $T_{(k_0+n) \bmod 5}$, $n = 0, \dots, 9$, respectively.

The pulses are searched in subsets of two pulses. We start by setting pulse i_0 to the maximum of track T_{k_0} and pulse i_1 to the maximum of track $T_{(k_0+1) \bmod 5}$. We then proceed by searching the pulse pair (i_2, i_3) by testing all the 8×8 possible position combinations in tracks $T_{(k_0+2) \bmod 5}$ and $T_{(k_0+3) \bmod 5}$ (given pulses i_0 and i_1 are known). The same procedure is repeated for the rest of the pulse pairs (i_4, i_5) , (i_6, i_7) , and (i_8, i_9) , by testing the 8×8 possible position combinations in their respective tracks. At each level of the tree, the test criterion is computed based only on the available pulses at that level. This results in a total of $4 \times 8 \times 8$ positions tested (since the first pulse pairs are set to their track maxima).

Other two iterations are carried out by changing pulse assignment to tracks (replacing k_0 by k_1 for the second iteration and k_0 by k_2 for the third iteration). All 10 initial pulse positions are assigned to tracks $T_{(k_1+n) \bmod 5}$ in the second iteration and to tracks $T_{(k_2+n) \bmod 5}$ in the third iteration. The same search procedure described above is repeated for these other two iterations. For the three iterations, the total number of tested position combinations is $3 \times 4 \times 8 \times 8 = 768$.

E.3.8.1.2 Codeword computation of the 35-bit fixed codebook

The two pulse positions in each track are encoded with 6 bits and the sign of the first pulse in each track is encoded with one bit. The second pulse sign is implicitly determined based on the order of pulse positions.

The two pulses in each track (2 positions and 2 signs) are encoded in 7 bits. Each pulse position needs 3 bits (8 possible positions) and each sign needs 1 bit. That is a total of 8 bits for each pair of pulses. However, 1 bit can be reduced considering the fact that about half the position combinations are redundant. For example, placing pulse 1 at position a and pulse 2 at position b is equivalent to placing pulse 1 at position b and pulse 2 at position a (when the signs are not considered). A simple approach of implementing the pulse encoding is to use only 1 bit for the sign information and 6 bits for the two positions, while ordering the positions in a way such that the other sign information can be easily deduced.

To better explain this, assume that the two pulses in a track are located at positions $p1$ and $p2$ with sign indices $s1$ and $s2$, respectively ($s = 0$ if the sign is positive and $s = 1$ if the sign is negative). The index of the two pulses is given by:

$$I = (p1/5) + s1 \times 8 + (p2/5) \times 16$$

If $p1 \leq p2$ then $s2 = s1$; otherwise, $s2$ is different from $s1$. Thus, when constructing the codeword, if the two signs are equal, then the smaller position is assigned to $p1$ and the larger position to $p2$; otherwise, the larger position is assigned to $p1$ and the smaller position to $p2$.

This procedure is repeated for each track to obtain five 7-bit indices.

E.3.8.2 Fixed codebook in backward LP mode

In the backward LP mode, the 18 bits needed for LP model are not transmitted. Thus, 9 bits are saved every subframe, which are used to increase the size of the fixed codebook from 35 to 44 bits. In this 44-bit codebook, each codebook vector contains 12 pulses. The positions in a subframe are divided into the same track structure described in Table E.2. However, two more pulses are placed, such that two consecutive tracks can contain three pulses instead of two. The two consecutive tracks containing three pulses will be called triple-pulse tracks and the other three tracks containing two pulses will be called double-pulse tracks.

The pulses in each double-pulse track are encoded with 7 bits (as in the 35-bit codebook) and those in each triple-pulse track are encoded with 10 bits. The index of the first triple-pulse track can have 5 different values (5 tracks). This index needs extra 3 bits. This results in a total of 44 bits ($3 \times 7 + 2 \times 10 + 3$).

E.3.8.2.1 Search procedure of the 44-bit codebook

The codebook search is very similar to that of the 35-bit codebook, with the exception that the tree has now 6 levels of pulse pairs. The same search procedure described in E.3.8.1.1 is followed.

The same procedure is used for pre-setting the pulse signs.

The initial tracks T_k and T_{k+1} are determined in the same manner.

The 12 pulses i_n , $n = 0, \dots, 11$ are assigned to tracks $T_{(k+n) \bmod 5}$, $n = 0, \dots, 11$ respectively.

The pulses are searched in subsets of two pulses, by initially setting pulse i_0 to the maximum of track T_k and pulse i_1 to the maximum of track $T_{(k+1) \bmod 5}$. Then it is proceeded by searching the pulse pair (i_2, i_3) by testing all the 8×8 possible position combinations in tracks $T_{(k+2) \bmod 5}$ and $T_{(k+3) \bmod 5}$ and repeating the procedure for the rest of the pulse pairs (i_4, i_5) , (i_6, i_7) , (i_8, i_9) , and (i_{10}, i_{11}) . This results now in a total of $5 \times 8 \times 8$ positions tested.

Two more iterations are carried out similar to the 35-bit codebook resulting in a total of $3 \times 5 \times 8 \times 8 = 960$ tested positions.

Similar to G.729 and to the 35-bit forward codebook, the selected codebook vector is filtered through the pre-filter $P(z) = 1/(1 - \beta z^{-T})$ to enhance the harmonic components.

E.3.8.2.2 Codeword computation of the 44-bit fixed codebook

The two pulses in each of the three double-pulse tracks are encoded using the same approach described in E.3.8.1.2.

The three pulses in a triple-pulse track are encoded using the same philosophy by adding three bits for the position of the third pulse. The three positions are encoded with 3 bits each and the sign of the first pulse is encoded with 1 bit. The signs of the other two pulses are deduced from the pulse orders, similar to the double-pulse tracks. Again, we will explain this with an example. Assume that the

three pulses in a triple-pulse track are located at positions $p1$, $p2$, and $p3$ with sign indices $s1$, $s2$, and $s3$, respectively. The index of the three pulses is given by:

$$I = (p1/5) + s1 \times 8 + (p2/5) \times 16 + (p3/5) \times 128$$

If $p1 \leq p2$ then $s2 = s1$; otherwise, $s2$ is different from $s1$. Similarly, if $p2 \leq p3$ then $s3 = s2$; otherwise, $s3$ is different from $s2$. When constructing the codeword, the pulse positions in a track are assigned to $p1$, $p2$, and $p3$ taking this sign relationship into consideration.

In total, 5 indices are returned, one for each track. The first index is that of the first triple-pulse track. This index is encoded with 13 bits; 10 for the positions and signs, as explained above, and 3 for the track index (0 to 4). The second index is that of the second triple-pulse track and is encoded with 10 bits. The last three indices are those of the three double-pulse tracks and are encoded with 7 bits each.

E.3.9 Quantization of the gains

The same as 3.9/G.729.

E.3.10 Memory update

The same as 3.10/G.729.

E.4 Functional description of the decoder

First the parameters are decoded. The transmitted parameters are listed in Table E.3. The first parameter decoded is the LP mode information and its parity bit. According to this information, the frame is classified either as forward, backward or erased. In forward mode, the decoder parameters are the LSP coefficients, the two fractional pitch delays, the two forward fixed-codebook vectors, and the two sets of adaptive- and fixed-codebook gains. In backward mode, the decoded parameters are the two fractional pitch delays, the two backward fixed-codebook vectors, and the two sets of adaptive- and fixed-codebook gains. Then, the LP backward analysis is performed on the past synthesized signal and the decoded parameters are used to compute the reconstructed speech signal as will be described in E.4.1. This reconstructed signal is enhanced by a post-processing operation consisting of a postfilter, a high-pass filter and an upscaling (see E.4.2). Subclause E.4.4 describes the error concealment procedure used when either a parity error has occurred, or when the frame erasure flag has been set.

Table E.3/G.729 – Description of transmitted parameters indices – The bit stream ordering is reflected by the order in the table – For each parameter the Most Significant Bit (MSB) is transmitted first

Table E.3a/G.729 – Description of transmitted parameters indices in forward mode

Symbol	Description	Bits
<i>M0</i>	Switch LP mode	1
<i>M1</i>	Parity bit for LP mode	1
<i>L0</i>	Switched MA predictor of LSP quantizer	1
<i>L1</i>	First stage vector of quantizer	7
<i>L2</i>	Second stage lower vector of LSP quantizer	5
<i>L3</i>	Second stage higher vector of LSP quantizer	5
<i>P1</i>	Pitch delay first subframe	8
<i>P0</i>	Parity bit for pitch delay	1
<i>C0_1</i>	Track 0 fixed codebook first subframe	7
<i>C1_1</i>	Track 1 fixed codebook first subframe	7
<i>C2_1</i>	Track 2 fixed codebook first subframe	7
<i>C3_1</i>	Track 3 fixed codebook first subframe	7
<i>C4_1</i>	Track 4 fixed codebook first subframe	7
<i>GA1</i>	Gain codebook (stage 1) 1 st subframe	3
<i>GB1</i>	Gain codebook (stage 2) 1 st subframe	4
<i>P2</i>	Pitch delay second subframe	5
<i>C0_2</i>	Track 0 fixed codebook second subframe	7
<i>C1_2</i>	Track 1 fixed codebook second subframe	7
<i>C2_2</i>	Track 2 fixed codebook second subframe	7
<i>C3_2</i>	Track 3 fixed codebook second subframe	7
<i>C4_2</i>	Track 4 fixed codebook second subframe	7
<i>GA2</i>	Gain codebook (stage 1) 2 nd subframe	3
<i>GB2</i>	Gain codebook (stage 2) 2 nd subframe	4

Table E.3b/G.729 – Description of transmitted parameters indices in backward mode

Symbol	Description	Bits
<i>M0</i>	Switch LP mode	1
<i>M1</i>	Parity bit for LP mode	1
<i>P1</i>	Pitch delay first subframe	8
<i>P0</i>	Parity bit for pitch delay	1
<i>C0_1</i>	Fixed codebook track index + pulses 0, 5 and 10 first subframe	13
<i>C1_1</i>	Fixed codebook pulses 1, 6 and 11 first subframe	10
<i>C2_1</i>	Fixed codebook pulses 2 and 7 first subframe	7
<i>C3_1</i>	Fixed codebook pulses 3 and 8 first subframe	7
<i>C4_1</i>	Fixed codebook pulses 4 and 9 first subframe	7
<i>GA1</i>	Gain codebook (stage 1) 1 st subframe	3
<i>GB1</i>	Gain codebook (stage 2) 1 st subframe	4
<i>P2</i>	Pitch delay second subframe	5
<i>C0_2</i>	Fixed codebook track index + pulses 0, 5 and 10 second subframe	13
<i>C1_2</i>	Fixed codebook pulses 1, 6 and 11 second subframe	10
<i>C2_2</i>	Fixed codebook pulses 2 and 7 second subframe	7
<i>C3_2</i>	Fixed codebook pulses 3 and 8 second subframe	7
<i>C4_2</i>	Fixed codebook pulses 4 and 9 second subframe	7
<i>GA2</i>	Gain codebook (stage 1) 2 nd subframe	3
<i>GB2</i>	Gain codebook (stage 2) 2 nd subframe	4

E.4.1 Parameter decoding procedure

Similar to G.729. The number of parameters is greater (more excitation codebooks parameters and one LP mode indication parameter). The decoding process is done in the following order.

E.4.1.1 Backward/forward decoding procedure

One bit is used to indicate to the decoder the LP mode: backward or forward. Then, the parity bit mode is compared with this LP mode bit. If these bits are not identical, the frame is considered as erased and the procedure described in E.4.4 is applied. Otherwise, according to this LP mode indication, the same switching procedure as described in E.3.2.7 is performed at the decoder to obtain the LP filter that will be used for the synthesis.

The high stationarity indicator High_Stat(n) is computed once per frame as described in E.3.2.7.2.

Another high stationarity indicator High_Stat2 that will be used by the gain attenuation procedure in case of erased frame is computed each subframe (see E.4.4.3). If the current subframe is at least the 30th of consecutive backward subframes, High_Stat2 is set to 1, else it is set to zero.

E.4.1.2 Decoding of LP parameters

E.4.1.2.1 Computing the LP backward filter

In any LP mode (backward or forward) and even if the frame is erased (see E.4.4), one backward LP analysis per frame is performed, using the same procedures as those performed in the encoder in

E.3.2 to obtain the encoder LP backward filter (windowing and autocorrelation computation, Levinson Durbin algorithm).

E.4.1.2.2 Forward mode

In forward mode, the same decoding procedure of the LP parameters is applied as in G.729. The interpolation procedure of the LP coefficients is the same as described in E.3.2.5.

E.4.1.2.3 Backward mode

In case that one of the previous frames has been erased, the current backward filter computed in E.4.1.2.1 $A_{bwd}^{(current)}$ is not directly used but linearly interpolated with the last "correct" backward filter (see E.4.4.) prior to the interpolation procedure of the LP coefficients described in E.3.2.5.

E.4.1.3 Computation of the parity bit of the adaptive-codebook delay

Before the excitation is reconstructed, the parity bit is recomputed from the adaptive-codebook delay index $P1$ (see E.3.7). If this bit is not identical to the transmitted parity bit $P0$, it is likely that bit errors occurred during transmission. If a parity error occurs on $P1$, the delay value T_1 is replaced by the delay value calculated in the previous subframe (see E.4.4.5).

E.4.1.4 Decoding of the adaptive-codebook vector

The same as 4.1.3/G.729.

E.4.1.5 Decoding of the fixed-codebook vector

The received codebook indices are used to extract the positions and signs of the pulses. This is done by reversing the process described in E.3.8.1.2 and E.3.8.2.2 for the 35-bit and 44-bit codebooks, respectively. Once the pulse positions and signs are decoded, the fixed codebook vector $c(n)$ is constructed by:

$$c(n) = \sum_{i=0}^{N_p-1} s_i \delta(n - p_i)$$

where s_i are pulse signs, p_i are the pulse positions, and N_p is the number of pulses (10 or 12). If the integer part of the pitch delay is less than the subframe size 40, $c(n)$ is modified similar to equation (48) in G.729.

E.4.1.6 Decoding of the adaptive- and fixed-codebook gains

The same as 4.1.5/G.729.

E.4.1.7 Computing the reconstructed speech

Similar to 4.1.6/G.729. (The order of the LP filter could be 30 instead of 10.)

E.4.2 Post-processing

As in G.729. The post-processing consists of three functions: adaptive postfiltering, high-pass filtering and signal upscaling. The adaptive postfiltering is similar to G.729 postfiltering except for the parameters γ_p , γ_n , and γ_d that have been made adaptive according to the high stationarity indicator High_Stat and the current frame LP mode. After 20 consecutive high stationarity backward frames, there is no more postfiltering.

E.4.2.1 Long-term postfilter

The long-term postfiltering procedure is the same as 4.2.1/G.729:

Adaptive filter:

$$H_p(z) = \frac{1}{1 + \gamma_p g_l} (1 + \gamma_p g_l z^{-T})$$

except for the value of the parameter γ_p that has been made adaptive according to the high stationarity indicator High_Stat and the current frame LP mode.

If the high stationarity state is detected on the input signal (High_Stat = 1), the long-term perceptual filter is progressively flattened.

At frame n, if (High_Stat = 1) and if the frame is in backward then:

$$\begin{aligned} \gamma_p(n) &= \gamma_p(n-1) - (\gamma_{pmax}/20) \\ \text{if } (\gamma_p(n) < 0) &\text{ then } \gamma_p(n) = 0 \end{aligned}$$

Else, the filter recovers progressively the initial value γ_{pmax} :

$$\begin{aligned} \gamma_p(n) &= \gamma_p(n-1) + (\gamma_{pmax}/20) \\ \text{if } (\gamma_p(n) > \gamma_{pmax}) &\text{ then } \gamma_p(n) = \gamma_{pmax} \end{aligned}$$

The value of γ_{pmax} is set to 0.25. When $\gamma_p(n)$ is equal to 0, there is no adaptive (neither harmonic, neither short-term) postfiltering.

E.4.2.2 Short-term postfilter

The only modifications brought to the G.729 algorithm concern:

- The LP filter used to calculate the short-term perceptual weighting filter $H_f(z)$ is the LP filter computed in E.4.1.2: either the 10 coefficients forward LP filter (computed in E.4.1.2.2) if the frame is in forward mode or the 30 coefficients backward LP filter (computed in E.4.1.2.3) if the frame is in backward mode.

$$H_f(z) = \frac{A(z/\gamma_n)}{A(z/\gamma_d)} = \frac{1 + \sum_{i=1}^{i=m_{b/f}} \gamma_n^i a_i z^{-i}}{1 + \sum_{i=1}^{i=m_{b/f}} \gamma_d^i a_i z^{-i}}$$

- The values of the parameters γ_n and γ_d that are adapted according to the high stationarity indicator High_Stat (see E.4.1.1) and the LP mode of the current frame (backward or forward).

If a high stationarity state is detected on the input signal (High_Stat = 1) and if the current frame is in backward, the short-term LP postfilter is progressively flattened down to no postfiltering at all ($\gamma_n(n) = \gamma_d(n) = 0$).

At frame n, if (High_Stat = 1 and LP_mode = 1) then:

$$\begin{aligned} \gamma_n(n) &= \gamma_n(n-1) - (\gamma_{nmax}/20) \\ \gamma_d(n) &= \gamma_d(n-1) - (\gamma_{dmax}/20) \\ \text{if } (\gamma_n(n) < 0) &\text{ then } \gamma_n(n) = 0 \\ \text{if } (\gamma_d(n) < 0) &\text{ then } \gamma_d(n) = 0 \end{aligned}$$

Else, the filter recovers progressively the initial values $\gamma_{n\max}$ and $\gamma_{d\max}$:

$$\begin{aligned}\gamma_n(n) &= \gamma_n(n-1) + (\gamma_{n\max}/20) \\ \gamma_d(n) &= \gamma_d(n-1) + (\gamma_{d\max}/20) \\ \text{if } (\gamma_n(n) > \gamma_{n\max}) &\text{ then } \gamma_n(n) = \gamma_{n\max} \\ \text{if } (\gamma_d(n) > \gamma_{d\max}) &\text{ then } \gamma_d(n) = \gamma_{d\max}\end{aligned}$$

With $\gamma_{n\max} = 0.7$ and $\gamma_{d\max} = 0.65$

E.4.2.3 Tilt compensation

The tilt compensation filtering is the same as 4.2.3/G.729, except for the computation of the first parcor where the length of the impulse response is 32 instead of 20.

E.4.2.4 Adaptive gain control

The same as 4.2.4/G.729.

E.4.2.5 High-pass filtering and up-scaling

The same as 4.2.5/G.729.

E.4.3 Encoder and decoder initialization

All static encoder and decoder variables should be initialized to 0, except the variables listed in Table 9/G.729 and in Table E.4.

Table E.4/G.729 – Description of parameters with non-zero initialization

Variable	Reference	Initial value
$Stat(-1)$	E.3.2.7.2	10 000
$\gamma_p(-1)$	E.4.2.2	0.25
$\gamma_n(-1)$	E.4.2.2	0.7
$\gamma_d(-1)$	E.4.2.2	0.65
$\alpha_g^{(-1)}$	E.4.4.3	1.
$T_{sav}^{(-1)}$	E.4.4.5	30

E.4.4 Concealment of frame erasures

Basically, the bad frame concealment procedure is similar to 4.4/G.729. The same voicing decision as in G.729 is used. But some refinements in the gain attenuation procedure have been brought taking into account the high stationarity indicator High_Stat2 to adapt the muting factor. A special procedure has also been added to improve the backward filter robustness to frame erasures.

The specific steps taken for an erased frame are:

- 1) repetition of the LP mode;
- 2) in forward mode, repetition of the synthesis filter parameter; in backward mode, use of the second step backward LP filter as described in E.4.4;
- 3) attenuation of adaptive- and fixed-codebook gains;
- 4) attenuation of the memory of the gain predictor;
- 5) generation of the replacement excitation.

E.4.4.1 Repetition of LP mode

When a frame is erased, the LP mode is set to the previous frame LP mode. The initial value is set to 0 (forward mode).

E.4.4.2 Computation of synthesis filter parameters

Note that the backward LP analysis described in E.4.1.2.1 is always performed, even if the frame is erased.

To improve the robustness of the backward filter, for each frame, a second step backward filter is computed. This filter is equal to the computed backward filter in error free conditions, but is different if an erasure has occurred at some frame before the current one: Let $A_{bwd}^*(n)$ denote the second step backward filter for frame n. The computed LP filter of the current frame n being denoted $A_{bwd}(n)$, $A_{bwd}^*(n)$ is obtained by linear interpolation of $A_{bwd}(n)$ and $A_{bwd}^*(n_e)$ where n_e represents the last erased frame (i.e. the last reliable second step backward filter).

$$A_{bwd}^*(n) = \alpha_{lpbwd} A_{bwd}^*(n_e) + (1 - \alpha_{lpbwd}) A_{bwd}(n)$$

The initial value of the interpolation factor α_{lpbwd} is 0. α_{lpbwd} is updated at the end of the current frame to be applied for the next frame. The adaptation procedure is the following:

Whenever an erased frame occurs, α_{lpbwd} is fixed to the maximum value 1.

For each valid frame n:

if frame n is in forward mode, $\alpha_{lpbwd} = 0$.

else (frame n is in backward mode) α_{lpbwd} is decreased of an amount depending on the value of the high stationarity indicator High_Stat: If High_Stat is equal to 1, then α_{lpbwd} is decreased by step of 0.1, else it is decreased by step of 0.5 (slow recovery for highly stationary signals, else fast recovery).

The second step backward filter $A_{bwd}^*(n)$ will then be used in the frame n processing.

If the erased frame is considered as forward, the same procedure as in 4.4.1/G.729 is applied.

E.4.4.3 Attenuation of adaptive- and fixed-codebook gains

The attenuation of the adaptive- and fixed-codebook gains depends on the number of consecutive erased subframes before the current subframe and the second stationary indicator High_Stat2 computed in E.4.1.1. Let N_{bf} be the number of consecutive erased subframes before the current subframe indexed m. The attenuation procedure is the following:

If less than 2 consecutive subframes have been erased ($N_{bf} < 2$),

$$g_c^{(m)} = g_c^{(m-1)}$$

If High_Stat2 is equal to 1 then $g_p^{(m)} = 1$, else $g_p^{(m)} = 0.95$

otherwise: ($N_{bf} \geq 2$)

$$g_p^{(m)} = g_p^{(m-1)} * \alpha_g^{(m-1)}$$

$$g_c^{(m)} = g_c^{(m-1)} * \alpha_g^{(m-1)}$$

The adaptation of the attenuation factor also depends on N_{bf} and on High_Stat2:

If less than 2 consecutive subframes have been erased ($N_{bf} < 2$),

$$\alpha_g^{(m)} = \alpha_g^{(m-1)} (= 1.)$$

otherwise: ($N_{bf} \geq 2$)

If High_Stat2 is equal to 1 then:

$$\text{if } (N_{bf} > 10) \text{ then } \alpha_g^{(m)} = \alpha_g^{(m-1)} * \alpha_g^h$$

$$\text{else } \alpha_g^{(m)} = \alpha_g^{(m-1)} * \alpha_g^l$$

with $\alpha_g^l = 0.98$ and $\alpha_g^h = 0.995$. When the subframe is not erased, $\alpha_g^{(m)}$ is reset to the initial value $\alpha_g^{(-1)}$ equal to 1.

E.4.4.4 Attenuation of the memory gain predictor

The same as 4.4.3/G.729.

E.4.4.5 Generation of the replacement excitation

As in 4.4.4/G.729, the excitation used depends on the periodicity classification. If the last reconstructed frame was classified as periodic, the current frame is considered to be periodic as well. In that case only the adaptive-codebook is used, and the fixed-codebook contribution is set to zero.

The adaptive-codevector index of an erroneous subframe m (either belonging to an erased frame or if the pitch delay parity bit has detected an error) uses a fractional pitch delay calculated and stored at the preceding subframe. Let $T^{(m)}$ be the fractional pitch delay of any valid or not subframe m and let $T_{sav}^{(m)}$ be the fractional pitch delay stored for the next subframe error concealment. If m is a valid subframe, $T^{(m)}$ takes the valid decoded value else $T^{(m)}$ is taken equal to $T_{sav}^{(m-1)}$. The computation of $T_{sav}^{(m)}$ is as follows:

Let us introduce the integer $stat_T^{(m)}$ with values in $[0,7]$ that indicates the stationary nature of the pitch delay. $stat_T^{(m)}$ is initialised to 0. Let $\text{int}(T^{(m)})$ denote the integer part of $T^{(m)}$.

If $\left| \text{int}(T^{(m)}) - \text{int}(T^{(m-1)}) \right| < 5$ then $stat_T^{(m)} = stat_T^{(m-1)} + 1$ and $T_{sav}^{(m)} = T^{(m)}$

else if there exists a multiple T_{mult} of $\min(\text{int}(T^{(m)}), \text{int}(T^{(m-1)}))$ such that $\left| T_{mult} - \max(\text{int}(T^{(m)}), \text{int}(T^{(m-1)})) \right| < 5$: if $stat_T^{(m-1)} > 0$ then $stat_T^{(m)} = stat_T^{(m-1)} - 1$ and $T_{sav}^{(m)} = T_{sav}^{(m-1)}$

otherwise $stat_T^{(m)} = 0$ and $T_{sav}^{(m)} = T^{(m)}$

Therefore multiples or submultiples of the pitch delay are replaced by the estimated pitch period during stationary voiced parts of the signal.

The adaptive-codebook gain is based on an attenuated value computed in E.4.4.3.

If the last reconstructed frame was classified as non-periodic, the current frame is considered to be non-periodic as well, and the adaptive-codebook contribution is set to zero. The fixed-codebook contribution is generated by randomly selecting the 5 codebook indices. The same random generator

as that of G.729 is used. The fixed-codebook gain is attenuated with the procedure described in E.4.4.3.

E.5 Bit-exact description of the CS-ACELP coder

ANSI C code specifying the 11.8 kbit/s CS-ACELP coder in 16-bit fixed-point is available from ITU-T. The current version of this ANSI C code is Version 1.2 of May 1998. The following subclauses summarize the use of this simulation code, and how the software is organized.

E.5.1 Use of the simulation software

The C code consists of two main programs **codere.c**, which simulates the encoder, and **decodere.c**, which simulates the decoder. The encoder is run as follows:

codere inputfile bitstreamfile rate_option

The decoder is run as follows:

decodere bitstreamfile outputfile

The input file and output file are sampled data files containing 16-bit PCM signals. The mapping table of the encoded bit stream is contained in the simulation software. The rate_option is either 1 to select the high level extension (11.8 kbit/s) or 0 to select the main Recommendation G.729 (8 kbit/s) or a file_rate_name: a binary file of 16-bit word containing either 0 or 1 to select the rate on a frame-by-frame basis; the default is 0 (8 kbit/s).

E.5.2 Organization of the simulation software

In the fixed-point ANSI C simulation, the types of fixed-point data and the set of basic operators used are the same as in the G.729 software. Some additional tables have been added that are found in tab_ld8e.h (see Table E.5).

Table E.5/G.729 – Summary of tables found in tab_ld8e.h

Table name	Size	Description
lag_h_bwd	30	Lag window for backward LP bandwidth expansion (high part)
lag_l_bwd	30	Lag window for backward LP bandwidth expansion (low part)
bitsno_E_fwd	18	Bit allocation in forward mode
bitsno_E_bwd	16	Bit allocation in backward mode
hw	145	Backward LP analysis window
bitrates	2	Table of available bit rates
tab_log	17	Lookup table in base 2 logarithm Q.11

The files can be classified into four groups:

- 1) Files identical to G.729 software files, part of ITU-T G.729 Recommendation listed in Table E.6.
- 2) Files similar to G.729 software files, some minor modifications have been introduced to cope with Annex E listed in Table E.7.
- 3) Files adapted from G.729 software files, some source code lines have been introduced to existing G.729 files to deal with Annex E listed in Table E.8.
- 4) Files specific to G.729 Annex E (new files) listed in Table E.9.

Table E.6/G.729 – List of software files identical to G.729 software

File name	Description
basic_op.c	Basic operators
oper_32b.c	Extended basic operators
dspfunc.c	Mathematical functions
gainpred.c	Gain predictor
lpcfunc.c	Miscellaneous routines related to LP filter
pred_lt3.c	Generation of adaptive codebook
pre_proc.c	Pre-processing (HP filtering and scaling)
p_parity.c	Compute pitch parity
qua_gain.c	Gain quantizer
pwf.c	Computation of perceptual weighting coefficients (8 kbit/s)
pitch.c	Pitch search
util.c	Utility functions
acelp_co.c	Search fixed codebook (8 kbit/s)
post_pro.c	Post processing (HP filtering and scaling)
de_acelp.c	Decode algebraic codebook (8 kbit/s)
dec_lag3.c	Decode adaptive-codebook index
basic_op.h	Basic operators prototypes
ld8k.h	Function prototypes
oper_32b.h	Extended basic operators prototypes
tab_ld8k.c	ROM tables
tab_ld8k.h	Extern ROM table declarations
typedef.h	Data type definition (machine dependent)

Table E.7/G.729 – List of software files similar to G.729 software

File name	Description	Size (Bytes)
qua_lspe.c	LSP quantizer	9999
filtere.c	Filter functions	3917

Table E.8/G.729 – List of software files adapted from G.729 software

File name	Description	Size (Bytes)
codere.c	Main encoder routine	6 679
cod_ld8e.c	Encoder routine	38 119
decodere.c	Main decoder routine	8 592
dec_ld8e.c	Decoder routine	20 664
decgain.c	Decode gains	5 633
pste.c	Postfilter routines	34 020
bitse.c	Bit manipulation routines	6 629
lspgetqe.c	LSP quantizer	5 804
lpce.c	LP analysis	23 394
lspdece.c	LSP decoding routing	3 453

Table E.9/G.729 – List of software files specific to G.729E software

File name	Description	Size (Bytes)
bffw.c	Backward/forward switch selection	11 598
bffwfunc.c	Miscellaneous routines related to backward/forward switch selection	7 416
ld8e.h	Function prototypes for G.729, Annex E	15 540
pwfe.c	Computation of perceptual weighting coefficients (11.8 kbit/s)	1 148
acelp_e.c	Search fixed codebook (11.8 kbit/s)	38 780
deacelp_e.c	Decode algebraic codebook (11.8 kbit/s)	3 542
tab_ld8e.c	ROM tables for G.729, Annex E	3 313
tab_ld8e.h	Extern ROM declarations for G.729, Annex E	358
track_pi.c	Pitch tracking	3 004

E.6 Normative references

- [1] CCITT Recommendation G.728 (1992), *Coding of speech at 16 kbit/s using low-delay code excited linear-prediction*.
- [2] ITU-T Recommendation G.729 (1996), *Coding of speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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	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	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
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 and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages