

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.723.1

(05/2006)

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

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

Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s

ITU-T Recommendation G.723.1

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 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
ETHERNET 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.723.1

Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s

Summary

This Recommendation specifies a coded representation that can be used for compressing the speech or other audio signal component of multimedia services at a very low bit rate as part of the overall H.324 family of standards. This coder has two bit rates associated with it: 5.3 and 6.3 kbit/s. The higher bit rate has greater quality. The lower bit rate gives good quality and provides system designers with additional flexibility. Both rates are a mandatory part of the encoder and decoder. It is possible to switch between the two rates at any frame boundary. An option for variable rate operation using discontinuous transmission and noise fill during non-speech intervals is also possible.

This coder was optimized to represent speech with a high quality at the above rates using a limited amount of complexity. It encodes speech or other audio signals in frames using linear predictive analysis-by-synthesis coding. The excitation signal for the high rate coder is Multipulse Maximum Likelihood Quantization (MP-MLQ) and for the low rate coder is Algebraic-Code-Excited Linear-Prediction (ACELP). The frame size is 30 ms and there is an additional look ahead of 7.5 ms, resulting in a total algorithmic delay of 37.5 ms. All additional delays in this coder are due to processing delays of the implementation, transmission delays in the communication link and buffering delays of the multiplexing protocol.

The description of this Recommendation is made in terms of bit-exact, fixed-point mathematical operations. The ANSI C code indicated in clause 5 constitutes an integral part of this Recommendation and shall take precedence over the mathematical descriptions in this text if discrepancies are found. A non-exhaustive set of test sequences which can be used in conjunction with the C code are available in the electronic attachment.

This revision integrates the correction of defects identified either in Implementors' Guides or in SG 16 meeting reports.

Source

ITU-T Recommendation G.723.1 was approved on 29 May 2006 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

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, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© 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

1	Introduction	1
1.1	Scope	1
1.2	Bit rates.....	1
1.3	Possible input signals	1
1.4	Delay.....	1
1.5	Speech coder description.....	1
2	Encoder principles	2
2.1	General description.....	2
2.2	Framer.....	3
2.3	High pass filter	4
2.4	LPC analysis	4
2.5	LSP quantizer	4
2.6	LSP decoder.....	6
2.7	LSP interpolation.....	6
2.8	Formant perceptual weighting filter	7
2.9	Pitch estimation	7
2.10	Subframe processing	7
2.11	Harmonic noise shaping	7
2.12	Impulse response calculator	8
2.13	Zero input response and ringing subtraction	8
2.14	Pitch predictor	9
2.15	High rate excitation (MP-MLQ).....	9
2.16	Low rate excitation (ACELP).....	10
2.17	Excitation decoder	13
2.18	Decoding of the pitch information	13
2.19	Memory update.....	14
2.20	Bit allocation	14
2.21	Coder initialization	15
3	Decoder principles	16
3.1	General description.....	16
3.2	LSP decoder.....	16
3.3	LSP interpolator.....	16
3.4	Decoding of the pitch information	16
3.5	Excitation decoder	17
3.6	Pitch postfilter	17
3.7	LPC synthesis filter	18
3.8	Formant postfilter	19
3.9	Gain scaling unit.....	19



Documents

ITU-T Rec. G.723.1 (05/2006)

Document Sharing Hub

iii

	Page
3.10 Frame interpolation handling	19
3.11 Decoder initialization	20
4 Bitstream packing	21
5 ANSI C code.....	22
6 Glossary	23
Annex A – Silence compression scheme	26
A.1 Introduction	26
A.2 Description of the VAD.....	27
A.3 General description of the CNG	29
A.4 Description of the CNG encoder part.....	31
A.5 Description of the decoder part	36
A.6 Bit stream packing.....	39
A.7 Glossary	39
A.8 Bit-exact, fixed-point C source code	40
Annex B – Alternative specification based on floating point arithmetic.....	41
B.1 Introduction	41
B.2 Algorithm description.....	41
B.3 ANSI C code.....	41
Annex C – Scalable channel coding scheme for wireless applications	42
C.1 Introduction	42
C.2 Channel encoder	43
C.3 Channel decoder	52
C.4 Fixed point C source code	56

Electronic attachment.

ITU-T Recommendation G.723.1

Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s

1 Introduction

1.1 Scope

This Recommendation specifies a coded representation that can be used for compressing the speech or other audio signal component of multimedia services at a very low bit rate. In the design of this coder, the principal application considered was very low bit rate visual telephony as part of the overall H.324 family of standards.

1.2 Bit rates

This coder has two bit rates associated with it. These are 5.3 and 6.3 kbit/s. The higher bit rate has greater quality. The lower bit rate gives good quality and provides system designers with additional flexibility. Both rates are a mandatory part of the encoder and decoder. It is possible to switch between the two rates at any 30-ms frame boundary. An option for variable rate operation using discontinuous transmission and noise fill during non-speech intervals is also possible.

1.3 Possible input signals

This coder was optimized to represent speech with a high quality at the above rates using a limited amount of complexity. Music and other audio signals are not represented as faithfully as speech, but can be compressed and decompressed using this coder.

1.4 Delay

This coder encodes speech or other audio signals in 30-ms frames. In addition, there is a look ahead of 7.5 ms, resulting in a total algorithmic delay of 37.5 ms. All additional delays in the implementation and operation of this coder are due to:

- i) actual time spent processing the data in the encoder and decoder;
- ii) transmission time on the communication link;
- iii) additional buffering delay for the multiplexing protocol.

1.5 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 clause 5, which constitutes an integral part of this Recommendation, reflects this bit-exact, fixed-point description approach. The mathematical descriptions of the encoder and decoder, given respectively in clauses 2 and 3, can be implemented in several other fashions, possibly leading to a codec implementation not complying with this Recommendation. Therefore, the algorithm description of the C code of clause 5 shall take precedence over the mathematical descriptions of clauses 2 and 3 whenever discrepancies are found. A non-exhaustive set of test sequences which can be used in conjunction with the C code are available from the ITU.

2 Encoder principles

2.1 General description

This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (ITU-T Rec. G.712) of the analogue input, then sampling at 8000 Hz and then converting to 16-bit linear PCM for the input to the encoder. The output of the decoder should be converted back to analogue by similar means. Other input/output characteristics, such as those specified by ITU-T Rec. G.711 for 64 kbit/s PCM data, should be converted to 16-bit linear PCM before encoding or from 16-bit linear PCM to the appropriate format after decoding. The bitstream from the encoder to the decoder is defined within this Recommendation.

The coder is based on the principles of linear prediction analysis-by-synthesis coding and attempts to minimize a perceptually weighted error signal. The encoder operates on blocks (frames) of 240 samples each. That is equal to 30 ms at an 8-kHz sampling rate. Each block is first high pass filtered to remove the DC component and then divided into four subframes of 60 samples each. For every subframe, a 10th order Linear Prediction Coder (LPC) filter is computed using the unprocessed input signal. The LPC filter for the last subframe is quantized using a Predictive Split Vector Quantizer (PSVQ). The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal.

For every two subframes (120 samples), the open-loop pitch period, L_{OL} , is computed using the weighted speech signal. This pitch estimation is performed on blocks of 120 samples. The pitch period is searched in the range from 18 to 142 samples.

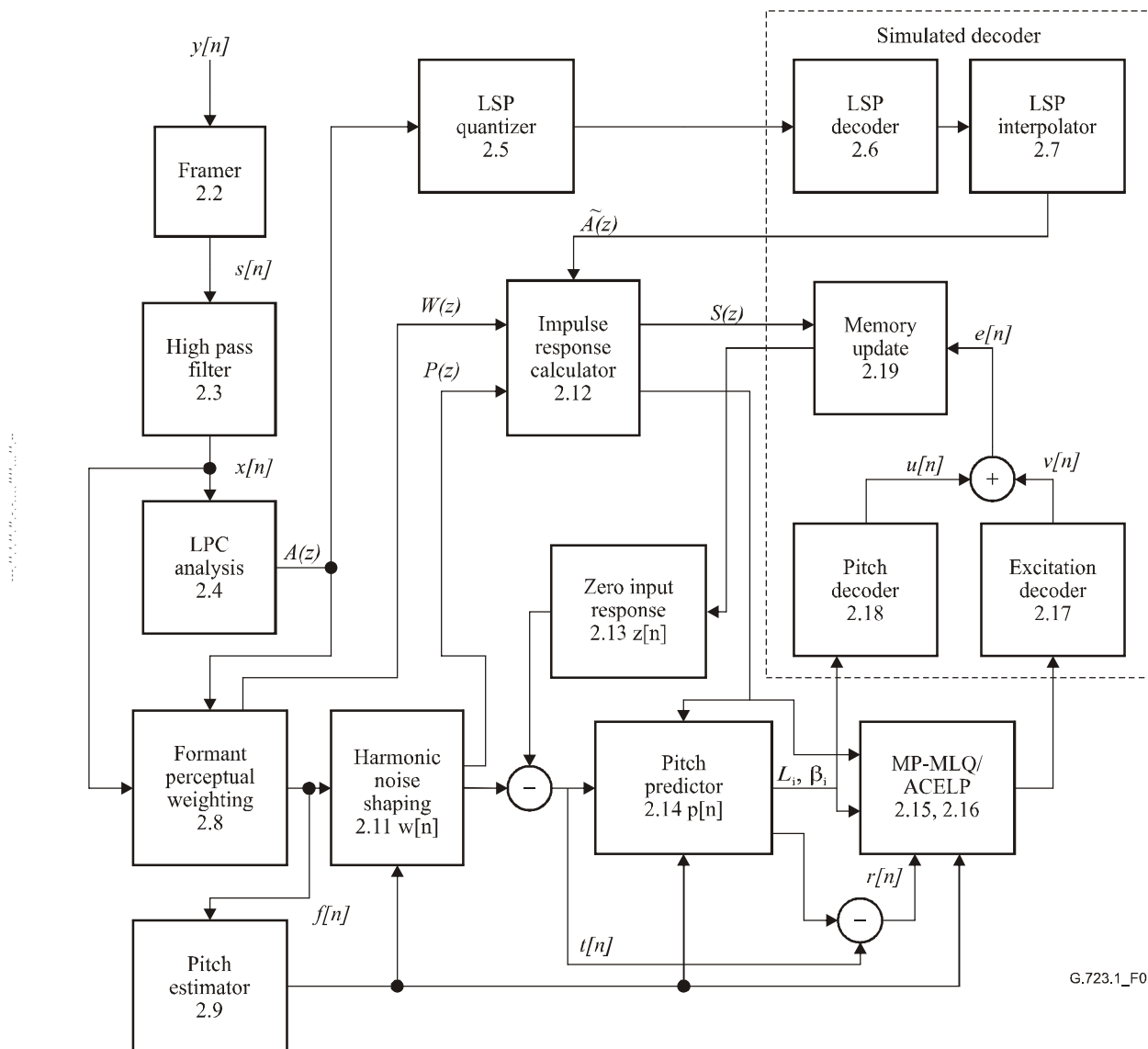
From this point the speech is processed on a 60 samples per subframe basis.

Using the estimated pitch period computed previously, a harmonic noise shaping filter is constructed. The combination of the LPC synthesis filter, the formant perceptual weighting filter, and the harmonic noise shaping filter is used to create an impulse response. The impulse response is then used for further computations.

Using the pitch period estimation, L_{OL} , and the impulse response, a closed-loop pitch predictor is computed. A fifth order pitch predictor is used. The pitch period is computed as a small differential value around the open-loop pitch estimate. The contribution of the pitch predictor is then subtracted from the initial target vector. Both the pitch period and the differential value are transmitted to the decoder.

Finally the non-periodic component of the excitation is approximated. For the high bit rate, Multipulse Maximum Likelihood Quantization (MP-MLQ) excitation is used, and for the low bit rate, an algebraic-code-excitation (ACELP) is used.

The block diagram of the encoder is shown in Figure 1.



G.723.1_F01

NOTE – For each block, the corresponding reference number is indicated.

Figure 1/G.723.1 – Block diagram of the speech coder

2.2 Framer

File : LBCCODEC.C	Procedure : main()	Reads 240 samples input frames
File : CODER.C	Procedure : Coder()	Performs subframe division

The coder processes the speech by buffering consecutive speech samples, $y[n]$, into frames of 240 samples, $s[n]$. Each frame is divided into two parts of 120 samples for pitch estimation computation. Each part is divided by two again, so that each frame is finally divided into four subframes of 60 samples each.

2.3 High pass filter

File : UTIL_LBC.C	Procedure : Rem_Dc()	Performs high pass filter
-------------------	----------------------	---------------------------

This block removes the DC element from the input speech, $s[n]$. The filter transfer function is:

$$H(z) = \frac{1 - z^{-1}}{1 - \frac{127}{128} z^{-1}} \quad (1)$$

The output of this filter is: $x[n]_n = 0..239$.

2.4 LPC analysis

File : LPC.C	Procedure : Comp_Lpc()	Performs LPC coefficients calculation
File : LPC.C	Procedure : Durbin()	Levinson-Durbin recursion

The LPC analysis is performed on signal $x[n]$ in the following way. Tenth order Linear Predictive (LP) analysis is used. For each subframe, a window of 180 samples is centred on the subframe. A Hamming window is applied to these samples. Eleven autocorrelation coefficients are computed from the windowed signal. A white noise correction factor of (1025/1024) is applied by using the formula $R[0] = R[0](1 + 1/1024)$. The other 10 autocorrelation coefficients are multiplied by the binomial window coefficients table. (The values for this table and all others are given in the C code.) The Linear Predictive Coefficients (LPC) are computed using the conventional Levinson-Durbin recursion. For every input frame, four LPC sets are computed, one for every subframe. These LPC sets are used to construct the short-term perceptual weighting filter. The LPC synthesis filter is defined as:

$$A_i(z) = \frac{1}{1 - \sum_{j=1}^{10} a_{ij} z^{-j}} \quad 0 \leq i \leq 3 \quad (2)$$

where i is the subframe index and is defined to be between 0 and 3.

2.5 LSP quantizer

File : LSP.C	Procedure : AtoLsp()	Converts LPC to LSP coefficients
File : LSP.C	Procedure : LspQnt()	LSP vector quantization
File : LSP.C	Procedure : Lsp_Svq()	LSP sub-vectors quantization

First, a small additional bandwidth expansion (7.5 Hz) is performed. Then, the resulting $A_3(z)$ LP filter is quantized using a predictive split vector quantizer. The quantization is performed in the following way:

- 1) The LP coefficients, $\{a_j\}_{j=1..10}$, are converted to LSP coefficients, $\{\mathbf{p}'_j\}_{j=1..10}$, by searching along the unit circle and interpolating for zero crossings.
- 2) The long-term DC component, $\mathbf{p}_{DC}$, is removed from the LSP coefficients, $\mathbf{p}'$, and a new DC removed LSP vector, $\mathbf{p}$, is obtained.

- 3) A first-order fixed predictor, $b = (12/32)$, is applied to the previously decoded LSP vector $\tilde{\mathbf{p}}_{n-1}$, to obtain the DC removed predicted LSP vector, $\bar{\mathbf{p}}_n$, and the residual LSP error vector, $\mathbf{e}_n$, at time (frame) n .

$$\mathbf{p}_n^T = [p_{1,n} \ p_{2,n} \ \cdots \ p_{10,n}] \quad (3.1)$$

$$\bar{\mathbf{p}}_n^T = [\bar{p}_{1,n} \ \bar{p}_{2,n} \ \cdots \ \bar{p}_{10,n}] \quad (3.2)$$

$$\bar{\mathbf{p}}_n = b[\tilde{\mathbf{p}}_{n-1} - \mathbf{p}_{DC}] \quad (3.3)$$

$$\mathbf{e}_n = \mathbf{p}_n - \bar{\mathbf{p}}_n \quad (3.4)$$

- 4) The unquantized LSP vector, $\mathbf{p}'_n$, the quantized LSP vector, $\tilde{\mathbf{p}}_n$, the residual LSP error vector, $\mathbf{e}_n$, are divided into 3 sub-vectors with dimension 3, 3 and 4 respectively. Each m th sub-vector is vector quantized using an 8-bit codebook. The index, l , of the appropriate sub-vector codebook entry that minimizes the error criterion $E_{l,m}$ is selected.

$$\mathbf{p}_m'^T = [p'_{1+3m} \ p'_{2+3m} \ \cdots \ p'_{K+3m}] \quad K_m = \begin{cases} 3, m=0 \\ 3, m=1 \\ 4, m=2 \end{cases} \quad (4.1)$$

$$\tilde{\mathbf{p}}_{l,m}^T = [\tilde{p}_{1,l,m} \ \tilde{p}_{2,l,m} \ \cdots \ \tilde{p}_{K_m,l,m}] \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.2)$$

$$\mathbf{p}' = \mathbf{p} + \mathbf{p}_{DC} \quad (4.3)$$

$$\tilde{\mathbf{p}}_{l,m} = \bar{\mathbf{p}}_m + \mathbf{p}_{DC_m} + \mathbf{e}_{l,m} \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.4)$$

$$E_{l,m} = (\mathbf{p}'_m - \tilde{\mathbf{p}}_{l,m})^T \mathbf{W}_m (\mathbf{p}'_m - \tilde{\mathbf{p}}_{l,m}) \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.5)$$

where:

$\mathbf{e}_{l,m}$ is the l th entry of the m th split residual LSP codebook, and

$\mathbf{W}_n$ is a diagonal weighting matrix, determined from the unquantized LSP coefficients vector $\mathbf{p}'$, with weights defined by:

$$\begin{aligned} w_{j,j} &= \frac{1}{\min\{p'_j - p'_{j-1}, p'_{j+1} - p'_j\}} \quad 2 \leq j \leq 9 \\ w_{1,1} &= \frac{1}{p'_2 - p'_1} \\ w_{10,10} &= \frac{1}{p'_{10} - p'_9} \end{aligned} \quad (5)$$

- 5) The selected indices are transmitted to the channel.

2.6 LSP decoder

File : LSP.C	Procedure: Lsp_Inq()	Inverse quantization of LSP
--------------	----------------------	-----------------------------

The decoding of the LSP coefficients is performed in the following way:

- 1) First, the three sub-vectors, $\{e_{m,n}\}_{m=0..2}$, are decoded to form a tenth order vector, $\tilde{\mathbf{e}}_n$.
- 2) The predicted vector, $\bar{\mathbf{p}}_n$, is added to the decoded vector, $\tilde{\mathbf{e}}_n$, and DC vector, $\mathbf{p}_{DC}$, to form the decoded LSP vector, $\tilde{\mathbf{p}}_n$.
- 3) A stability check is performed on the decoded LSP vector, $\tilde{\mathbf{p}}_n$, to ensure that the decoded LSP vector is ordered according to the following condition:

$$\tilde{\mathbf{p}}_{j+1,n} - \tilde{\mathbf{p}}_{j,n} \geq \Delta_{\min} \quad 1 \leq j \leq 9 \quad (6)$$

$\Delta_{\min}$ is equal to 31.25 Hz. If this stability check (6) fails for $\tilde{\mathbf{p}}_i$ and $\tilde{\mathbf{p}}_{i+1}$, then $\tilde{\mathbf{p}}_j$ and $\tilde{\mathbf{p}}_{j+1}$ are modified in the following way:

$$\tilde{\mathbf{p}}_{avg} = (\tilde{\mathbf{p}}_j + \tilde{\mathbf{p}}_{j+1}) / 2 \quad (7.1)$$

$$\tilde{\mathbf{p}}_j = \tilde{\mathbf{p}}_{avg} - \Delta_{\min} / 2 \quad (7.2)$$

$$\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{p}}_{avg} + \Delta_{\min} / 2 \quad (7.3)$$

The modification is performed until condition (6) is met. If after 10 iterations the condition of stability is not met, the previous LSP vector is used.

2.7 LSP interpolation

File : LSP.C	Procedure : Lsp_Int()	LSP interpolator
File : LSP.C	Procedure : LsptoA()	Converts LSP to LPC coefficients

Linear interpolation is performed between the decoded LSP vector, $\tilde{\mathbf{p}}_n$, and the previous LSP vector, $\tilde{\mathbf{p}}_{n-1}$, for each subframe. Four interpolated LSP vectors, $\{\tilde{\mathbf{p}}_i\}_{i=0..3}$, are converted to LPC vectors, $\{\tilde{\mathbf{a}}_i\}_{i=0..3}$.

$$\tilde{\mathbf{p}}_{ni} = \begin{cases} 0.75\tilde{\mathbf{p}}_{n-1} + 0.25\tilde{\mathbf{p}}_n & i=0 \\ 0.5\tilde{\mathbf{p}}_{n-1} + 0.5\tilde{\mathbf{p}}_n & i=1 \\ 0.25\tilde{\mathbf{p}}_{n-1} + 0.75\tilde{\mathbf{p}}_n & i=2 \\ \tilde{\mathbf{p}}_n & i=3 \end{cases} \quad (8)$$

$$\tilde{\mathbf{a}}_i^T = [\tilde{\mathbf{a}}_{i1} \tilde{\mathbf{a}}_{i2} \cdots \tilde{\mathbf{a}}_{i10}]^T \quad 0 \leq i \leq 3 \quad (9)$$

The quantized LPC synthesis filter, $\tilde{A}_i(z)$, is used for generating the decoded speech signal and is defined as:

$$\tilde{A}_i(z) = \frac{1}{1 - \sum_{j=1}^{10} \tilde{\mathbf{a}}_{ij} z^{-j}} \quad 0 \leq i \leq 3 \quad (10)$$

2.8 Formant perceptual weighting filter

File : LPC.C	Procedure : Wght_Lpc()	Computes perceptual filter coefficients
File : LPC.C	Procedure : Error_Wght()	Applies perceptual weighting filter

For each subframe a formant perceptual weighting filter is constructed, using the unquantized LPC coefficients $\{a_{ij}\}_{j=1,\dots,10}$. The filter has a transfer function:

$$W_i(z) = \frac{1 - \sum_{j=1}^{10} a_{ij} z^{-j\gamma_1^j}}{1 - \sum_{j=1}^{10} a_{ij} z^{-j\gamma_2^j}} \quad 0 \leq i \leq 3 \quad (11)$$

where $\gamma_1 = 0.9$ and $\gamma_2 = 0.5$. The input speech frame, $\{x[n]\}_{n=0..239}$, is then divided to four subframes and each subframe is filtered using the $W_i(z)$ filter, and the weighted output speech signal, $\{f[n]\}_{n=0..239}$ is obtained.

2.9 Pitch estimation

File : EXC_LBC.C	Procedure : Estim_Pitch()	Open-loop pitch estimation
------------------	---------------------------	----------------------------

Two pitch estimates are computed for every frame, one for the first two subframes and one for the last two. The open-loop pitch period estimate, L_{OL} , is computed using the perceptually weighted speech $f[n]$. A cross-correlation criterion, $C_{OL}(j)$, maximization method is used to determine the pitch period, using the following expression:

$$C_{OL}(j) = \frac{\left(\sum_{n=0}^{119} f[n] \cdot f[n-j] \right)^2}{\sum_{n=0}^{119} f[n-j] \cdot f[n-j]} \quad 18 \leq j \leq 142 \quad (12)$$

The index j which maximizes the cross-correlation, $C_{OL}(j)$, is selected as the open-loop pitch estimation for the appropriate two subframes. While searching for the best index, some preference is given to smaller pitch periods to avoid choosing pitch multiples. Maximums of $C_{OL}(j)$ are searched for beginning with $j = 18$. For every maximum $C_{OL}(j)$ found, its value is compared to the best previous maximum found, $C_{OL}(j')$. If the difference between indices j and j' is less than 18 and $C_{OL}(j) > C_{OL}(j')$, the new maximum is selected. If the difference between the indices is greater than or equal to 18, the new maximum is selected only if $C_{OL}(j)$ is greater than $C_{OL}(j')$ by 1.25 dB.

2.10 Subframe processing

From this point on, all the computational blocks are performed on a once per subframe basis.

2.11 Harmonic noise shaping

File : EXC_LBC.C	Procedure : Comp_Pw()	Computes harmonic noise filter coefficients
File : EXC_LBC.C	Procedure : Filt_Pw()	Applies harmonic noise filter

In order to improve the quality of the encoded speech, a harmonic noise shaping filter is constructed. The filter is:

$$P_i(z) = 1 - \beta z^{-L} \quad (13)$$

The optimal lag, L , for this filter is the lag which maximizes the criterion, $C_{PW}(j)$, while considering only positive correlation values for the numerator, $N(j)$, before squaring:

$$N(j) = \sum_{n=0}^{59} f[n] \cdot f[n-j] \quad (14.1)$$

$$C_{PW}(j) = \frac{(N(j))^2}{\sum_{n=0}^{59} f[n-j] \cdot f[n-j]} \quad L_1 \leq j \leq L_2 \quad (14.2)$$

where $L_1 = L_{OL} - 3$ and $L_2 = L_{OL} + 3$. The maximum value will be defined as C_L . The optimal filter gain, G_{opt} , is:

$$G_{opt} = \frac{\sum_{n=0}^{59} f[n] f[n-L]}{\sum_{n=0}^{59} f[n-L] f[n-L]} \quad (15)$$

G_{opt} is limited to the range $[0,1]$. The energy, E , of the weighted speech signal, $\{f[n]\}_{n=0..59}$ is given by:

$$E = \sum_{n=0}^{59} f^2[n] \quad (16)$$

Then the coefficient β of the harmonic noise shaping filter, $P(z)$, is given by:

$$\beta = \begin{cases} 0.3125G_{opt} & \text{if } -10 \log_{10} \left(1 - \frac{C_L}{E} \right) \geq 2.0 \\ 0.0 & \text{otherwise} \end{cases} \quad (17)$$

After computing the harmonic noise filter coefficients, the formant perceptually weighted speech, $f[n]$, is filtered using $P(z)$ to obtain the target vector, $w[n]$.

$$w[n] = f[n] - \beta f[n-L] \quad 0 \leq n \leq 59 \quad (18)$$

2.12 Impulse response calculator

File : LPC.C	Procedure : Comp_Ir()	Impulse response computation
--------------	-----------------------	------------------------------

For closed-loop analysis, the following combined filter, $S_i(z)$, is used:

$$S_i(z) = \tilde{A}_i(z) \cdot W_i(z) \cdot P_i(z) \quad 0 \leq i \leq 3 \quad (19)$$

where the components of $S_i(z)$ are defined in Formulas 10, 11 and 13. The impulse response of this filter is computed and will be referred as $\{h_i[n]\}_{n=0..59, i=0..3}$.

2.13 Zero input response and ringing subtraction

File : LPC.C	Procedure : Sub_Ring ()	Performs ringing subtraction
--------------	-------------------------	------------------------------

The zero input response of the combined filter, $S_i(z)$, is obtained by computing the output of that filter when the input signal is all zero-valued samples. The zero input response is denoted $\{z[n]\}_{n=0..59}$. The ringing subtraction is performed by subtracting the zero input response from the harmonic weighted speech vector, $\{w[n]\}_{n=0..59}$. The resulting vector is defined as $t[n] = w[n] - z[n]$.

2.14 Pitch predictor

File : EXC_LBC.C	Procedure : Find_Acbk()	Adaptive codebook contribution. Calls Get_Rez() and Decod_Acbk()
File : EXC_LBC.C	Procedure : Get_Rez()	Gets residual from the excitation buffer
File : EXC_LBC.C	Procedure : Decod_Acbk()	Decodes the adaptive codebook contribution

The pitch prediction contribution is treated as a conventional adaptive codebook contribution. The pitch predictor is a fifth order pitch predictor (see Equation 41.2). For subframes 0 and 2, the closed-loop pitch lag is selected from around the appropriate open-loop pitch lag in the range ± 1 and coded using 7 bits. (Note that the open-loop pitch lag is never transmitted.) For subframes 1 and 3 the closed-loop pitch lag is coded differentially using 2 bits and may differ from the previous subframe lag only by $-1, 0, +1$ or $+2$. The quantized and decoded pitch lag values will be referred to as L_i from this point on. The pitch predictor gains are vector quantized using two codebooks with 85 or 170 entries for the high bit rate and 170 entries for the low bit rate. The 170 entry codebook is the same for both rates. For the high rate if L_0 is less than 58 for subframes 0 and 1 or if L_2 is less than 58 for subframes 2 and 3, then the 85 entry codebook is used for the pitch gain quantization. Otherwise, the pitch gain is quantized using the 170 entry codebook. The contribution of the pitch predictor, $\{p[n]\}_{n=0..59}$, is subtracted from the target vector $\{t[n]\}_{n=0..59}$, to obtain the residual signal $\{r[n]\}_{n=0..59}$.

$$r[n] = t[n] - p[n] \quad (20)$$

2.15 High rate excitation (MP-MLQ)

File : EXC_LBC.C	Procedure : Find_Fcbk()	Fixed codebook contribution
File : EXC_LBC.C	Procedure : Find_Best()	Residual signal quantization
File : EXC_LBC.C	Procedure : Gen_Trn()	Generates a train of Dirac functions
File : EXC_LBC.C	Procedure : Fcbk_Pack()	Combinatorial coding of pulse positions

The residual signal $\{r[n]\}_{n=0..59}$, is transferred as a new target vector to the MP-MLQ block. This block performs the quantization of this vector. The quantization process is approximating the target vector $r[n]$ by $r'[n]$:

$$r'[n] = \sum_{j=0}^n h[j] \cdot v[n-j] \quad 0 \leq n \leq 59 \quad (21)$$

where $v[n]$ is the excitation to the combined filter $S(z)$ with impulse response $h[n]$ and defined as:

$$v[n] = G \sum_{k=0}^{M-1} \alpha_k d[n-m_k] \quad 0 \leq n \leq 59 \quad (22)$$

where G is the gain factor, $\delta[n]$ is a Dirac function, $\{\alpha_k\}_{k=0..M-1}$ and $\{m_k\}_{k=0..M-1}$ are the signs (± 1) and the positions of Dirac functions respectively and M is the number of pulses, which is 6 for even subframes and is 5 for odd subframes. There is a restriction on pulses positions. The positions can be either all odd or all even. This will be indicated by a grid bit. So, the problem is to estimate the unknown parameters, G , $\{\alpha_k\}_{k=0..M-1}$, and $\{m_k\}_{k=0..M-1}$, that minimize the mean square of the error signal $err[n]$:

$$err[n] = r[n] - r'[n] = r[n] - G \sum_{k=0}^{M-1} \alpha_k h[n-m_k] \quad (23)$$

The parameters estimation and quantization processes are based on an analysis-by-synthesis method. The $G_{\max}$ parameter is estimated and quantized as follows. First, the cross-correlation function $d[j]$ between the impulse response, $h[n]$, and the new target vector, $r[n]$, is computed:

$$d[j] = \sum_{n=j}^{59} r[n] \cdot h[n-j] \quad 0 \leq j \leq 59 \quad (24)$$

The estimated gain is given by:

$$G_{\max} = \frac{\max\{|d[j]|\}_{j=0..59}}{\sum_{n=0}^{59} h[n] \cdot h[n]} \quad (25)$$

Then the estimated gain $G_{\max}$ is quantized by a logarithmic quantizer. This scalar gain quantizer is common to both rates and consists of 24 steps, of 3.2 dB each. Around this quantized value, $\tilde{G}_{\max}$, additional gain values are selected within the range $[\tilde{G}_{\max} - 3.2, \tilde{G}_{\max} + 6.4]$. For each of these gain values, the signs and locations of the pulses are sequentially optimized. This procedure is repeated for both the odd and even grids. Finally, the combination of the quantized parameters that yields the minimum mean square of $err[n]$ is selected. The optimal combination of pulse locations and gain is transmitted. $\binom{30}{M}$ combinatorial coding is used to transmit the pulse locations. Furthermore, using the fact that the number of codewords in the fixed codebooks is not a power of 2, three additional bits are saved by combining the four most significant bits of the combinatorial codes for the four subframes to form a 13-bit index. The C code provides the details on how this information is packed.

To improve the quality of speech with a short pitch period, the following additional procedure is used. If L_0 is less than 58 for subframes 0 and 1 or if L_2 is less than 58 for subframes 2 and 3, a train of Dirac functions with the period of the pitch index, L_0 or L_2 , is used for each location m_k instead of a single Dirac function in the above quantization procedure. The choice between a train of Dirac functions or a single Dirac function to represent the residual signal is made based on the mean square error computation. The configuration which yields the lowest mean square error is selected and its parameter indices are transmitted.

2.16 Low rate excitation (ACELP)

File : EXC_LBC.C	Procedure : search_T0()	Pitch synchronous excitation
File : EXC_LBC.C	Procedure : ACELP_LBC_code()	Computes innovative vector
File : EXC_LBC.C	Procedure : Cor_h()	Correlations of impulse response
File : EXC_LBC.C	Procedure : Cor_h_X()	Correlation of target vector with impulse response
File : EXC_LBC.C	Procedure : D4i64_LBC()	Algebraic codebook search
File : EXC_LBC.C	Procedure : G_code()	Computes innovation vector gain

A 17-bit algebraic codebook is used for the fixed codebook excitation $v[n]$. Each fixed codevector contains, at most, four non-zero pulses. The four pulses can assume the signs and positions given in Table 1:

Table 1/G.723.1 – ACELP excitation codebook

Sign	Positions
± 1	0, 8, 16, 24, 32, 40, 48, 56
± 1	2, 10, 18, 26, 34, 42, 50, 58
± 1	4, 12, 20, 28, 36, 44, 52, (60)
± 1	6, 14, 22, 30, 38, 46, 54, (62)

The positions of all pulses can be simultaneously shifted by one (to occupy odd positions) which needs one extra bit. Note that the last position of each of the last two pulses falls outside the subframe boundary, which signifies that the pulse is not present.

Each pulse position is encoded with 3 bits and each pulse sign is encoded in 1 bit. This gives a total of 16 bits for the 4 pulses. Further, an extra bit is used to encode the shift resulting in a 17-bit codebook.

The codebook is searched by minimizing the mean square error between the weighted speech signal, $r[n]$, and the weighted synthesis speech given by:

$$E_{\xi} = \|r - G\mathbf{H}\mathbf{v}_{\xi}\|^2 \quad (26)$$

where $\mathbf{r}$ is the target vector consisting of the weighted speech after subtracting the zero-input response of the weighted synthesis filter and the pitch contribution, G is the codebook gain; $\mathbf{v}_{\xi}$ is the algebraic codeword at index ξ ; and $\mathbf{H}$ is a lower triangular Toeplitz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1), \dots, h(L-1)$, with $h(n)$ being the impulse response of the weighted synthesis filter $S_i(z)$.

It can be shown that the optimum codeword is the one which maximizes the term:

$$\tau_{\xi} = \frac{C_{\xi}^2}{\varepsilon_{\xi}} = \frac{(\mathbf{d}^T \mathbf{v}_{\xi})^2}{\mathbf{v}_{\xi}^T \Phi \mathbf{v}_{\xi}} \quad (27)$$

where $\mathbf{d} = \mathbf{H}^T \mathbf{r}$ is the correlation between the target vector signal, $r[n]$, and the impulse response, $h(n)$, and $\Phi = \mathbf{H}^T \mathbf{H}$ is the covariance matrix of the impulse response. The vector $\mathbf{d}$ and the matrix Φ are computed prior to the codebook search. The elements of the vector $\mathbf{d}$ are computed by:

$$d(j) = \sum_{n=j}^{59} r[n] \cdot h[n-j] \quad 0 \leq j \leq 59 \quad (28)$$

and the elements of the symmetric matrix $\Phi(i, j)$ are computed by:

$$\Phi(i, j) = \sum_{n=j}^{59} h[n-i] \cdot h[n-j] \quad \begin{matrix} j \geq i \\ 0 \leq i \leq 59 \end{matrix} \quad (29)$$

NOTE – Only the elements actually needed are computed and an efficient storage has been designed that speeds the search procedure up.

The algebraic structure of the codebook allows for very fast search procedures since the excitation vector $\mathbf{v}_{\xi}$ contains only 4 non-zero pulses. The search is performed in 4 nested loops, corresponding to each pulse positions, where in each loop the contribution of a new pulse is added. The correlation in Equation 27 is given by:

$$C = \alpha_0 d[m_0] + \alpha_1 d[m_1] + \alpha_2 d[m_2] + \alpha_3 d[m_3] \quad (30)$$

where m_k is the position of the k th pulse and α_k is its sign (± 1). The energy for even pulse position codevectors in Equation 27 is given by:

$$\begin{aligned} \varepsilon = & \Phi(m_0, m_0) \\ & + \Phi(m_1, m_1) + 2\alpha_0\alpha_1\Phi(m_0, m_1) \\ & + \Phi(m_2, m_2) + 2[\alpha_0\alpha_2\Phi(m_0, m_2) + \alpha_1\alpha_2\Phi(m_1, m_2)] \\ & + \Phi(m_3, m_3) + 2[\alpha_0\alpha_3\Phi(m_0, m_3) + \alpha_1\alpha_3\Phi(m_1, m_3) + \alpha_2\alpha_3\Phi(m_2, m_3)] \end{aligned} \quad (31)$$

For odd pulse position codevectors, the energy in Equation 27 is approximated by the energy of the equivalent even pulse position codevector obtained by shifting the odd position pulses to one sample earlier in time. To simplify the search procedure, the functions $d[j]$ and $\Phi(m_1, m_2)$ are modified. The simplification is performed as follows (prior to the codebook search). First, the signal $s[j]$ is defined and then the signal $d'[j]$ is constructed.

$$\begin{aligned} s[2j] = s[2j+1] = \text{sign}(d[2j]) & \quad \text{if } |d[2j]| > |d[2j+1]| \\ s[2j] = s[2j+1] = \text{sign}(d[2j+1]) & \quad \text{otherwise} \end{aligned} \quad (32)$$

and the signal d' is given by $d'[j] = d[j]s[j]$. Second, the matrix Φ is modified by including the signal information; that is, $\Phi'(i, j) = s[i]s[j]\Phi(i, j)$. The correlation in Equation 30 is now given by:

$$C = d'[m_0] + d'[m_1] + d'[m_2] + d'[m_3] \quad (33)$$

and the energy in Equation 31 is given by:

$$\begin{aligned} \varepsilon = & \Phi'(m_0, m_0) \\ & + \Phi'(m_1, m_1) + 2\Phi'(m_0, m_1) \\ & + \Phi'(m_2, m_2) + 2[\Phi'(m_0, m_2) + \Phi'(m_1, m_2)] \\ & + \Phi'(m_3, m_3) + 2[\Phi'(m_0, m_3) + \Phi'(m_1, m_3) + \Phi'(m_2, m_3)] \end{aligned} \quad (34)$$

A focused search approach is used to further simplify the search procedure. In this approach a precomputed threshold is tested before entering the last loop, and the loop is entered only if this threshold is exceeded. The maximum number of times the loop can be entered is fixed so that a low percentage of the codebook is searched. The threshold is computed based on the correlation C . The maximum absolute correlation and the average correlation due to the contribution of the first three pulses, max_3 and av_3 , are found prior to the codebook search. The threshold is given by:

$$thr_3 = av_3 + (max_3 - av_3) / 2 \quad (35)$$

The fourth loop is entered only if the absolute correlation (due to three pulses) exceeds thr_3 . Note that this results in a variable complexity search. To further control the search, the number of times the last loop is entered (for the 4 subframes) is not allowed to exceed 600. (The average worst case per subframe is 150 times. This can be viewed as searching only 150×8 entries of the codebook, ignoring the overhead of the first three loops.)

A special feature of the codebook is that, for pitch delays less than 60, a pitch contribution depending on the index $PGInd_i$ of the LTP pitch predictor gain vector is added to the code. That is, after the optimum algebraic code $v[n]$ is determined, it is modified by $v[n] \leftarrow v[n] + b(PGInd_i)v[n - L_i - e(PGInd_i)]$, the values $\beta(PGInd_i)$ and $\varepsilon(PGInd_i)$ are tabulated and L_i being the integer pitch period. Note that prior to the codebook search, the impulse response should be modified in a similar fashion if $L_i < 60$.

The last step, after getting the $v[n]$ sequence, is quantizing the gain G . The gain is quantized in the same way as in the high rate excitation. This is done by stepping through the gain quantization table and selecting the index $MGInd_i$ which minimizes the following expression: $|G - \tilde{G}_j|$, $0 \leq j \leq 23$.

2.17 Excitation decoder

File : EXC_LBC.C	Procedure : Fcbk_Unpk()	Decode fixed codebook excitation
------------------	-------------------------	----------------------------------

The decoding of pulses is performed as follows:

- 1) First, the maximum gain G_{max} index is derived using:

$$MGInd_i = GInd_i - PGInd_i \cdot GSize \quad (36)$$

where:

$GSize = 24$ is the size of the $\tilde{G}$ table; and

$PGInd_i$ is obtained in 2.18.

- 2) The positions of the pulses are decoded using $\binom{30}{M}$ combinatorial decoding where M is either 6 or 5, for the high rate. For the low rate, direct decoding of the position indices is performed.
- 3) The grid position (even/odd) is derived from the grid bit.
- 4) The pulse signs are derived from the sign bits.
- 5) For the high rate coder, the decoding of the pulse train bit is performed only if $L_i < 58$.
- 6) Then the $v[n]$ vector is reconstructed using the decoded parameters.
- 7) Finally, the pitch contribution, $u[n]$, and the pulse contributions, $v[n]$, are summed together to form the excitation vector $e[n]$.

2.18 Decoding of the pitch information

File : EXC_LBC.C	Procedure : Get_Rez()	Gets residual from the excitation buffer
File : EXC_LBC.C	Procedure : Decod_Acbk()	Decodes the adaptive codebook contribution

The decoding of pitch information is performed as described below:

- 1) First, the lag of pitch predictor for even subframes is decoded:

$$L_i = PInd_i + 18 \quad i = 0,2 \quad (37)$$

- 2) The lag of pitch predictor for odd subframes is decoded as follows:

$$L_i = L_{i-1} + \Delta_i \quad i = 1,3 \quad (38)$$

where $\Delta_i \in \{-1, 0, +1, +2\}$.

- 3) The gain vector of the pitch predictor in i th subframe is derived from the gain index $GInd_i$. For the low rate this index contains the information about the pitch predictor gain vector and the index of the gain of the pulse sequence. In this case pitch gain index $PGInd_i$ is derived as follows:

$$PGInd_i = \lfloor GInd_i / GSize \rfloor \quad i = 0..3 \quad (39)$$

where $\lfloor x \rfloor$ indicates the greatest integer $\leq x$. For the high rate, in the case that the condition $L_i \geq 58$ is met, this index is derived in the same manner as described in (39). In the above cases $PGInd_i$ is a pointer to 170 entries gain vector codebook. Otherwise, this index is a

pointer to 85 entries gain vector codebook and contains an additional information about the impulse train bit. In this case pitch gain index is derived as follows:

$$PGInd_i = \lfloor GInd_i \& 0x7FF / GSize \rfloor \quad i = 0..3 \quad (40)$$

The pitch predictor lag and gain vector are decoded from these indices and utilized for the pitch contribution $u[n]$ extraction as described below. First, a signal $e'[n]$ is defined by:

$$\begin{aligned} e'[0] &= e[-L_i - 2] \\ e'[1] &= e[-L_i - 1] \\ e'[n] &= e[(n \bmod L_i) - L_i] \quad 2 \leq n \leq 63 \end{aligned} \quad (41.1)$$

where mod stands for the modulus operation. Then,

$$u[n] = \sum_{j=0}^{j=4} \beta_{ij} e'[n+j] \quad 0 \leq n \leq 59 \quad (41.2)$$

2.19 Memory update

File : LPC.C	Procedure : Upd_Ring ()	Memory update
--------------	-------------------------	---------------

The last task of the i th subframe before proceeding to encode the next subframe is to update the memories of the synthesis filter $\tilde{A}_i(z)$, the formant perceptual weighting filter $W_i(z)$, and the harmonic noise shaping filter $P_i(z)$. To accomplish this, the complete response of combined filter $S_i(z)$, is computed by passing the reconstructed excitation sequence through this filter. At the end of the excitation filtering, the memory of the combined filter is saved and will be used to compute the zero input response during the encoding of the next speech vector.

2.20 Bit allocation

File : UTIL_LBC.C	Procedure : Line_Pack()	Bitstream packing
-------------------	-------------------------	-------------------

This clause presents the bit allocation tables for both high and low bit rates. The major differences between two rates are in the pulse positions and amplitudes coding. Also, at the lower rate 170 codebook entries are always used for the gain vector of the long term predictor. See Tables 2, 3 and 4.

Table 2/G.723.1 – Bit allocation of the 6.3 kbit/s coding algorithm

Parameters coded	Subframe 0	Subframe 1	Subframe 2	Subframe 3	Total
LPC indices					24
Adaptive codebook lags	7	2	7	2	18
All the gains combined	12	12	12	12	48
Pulse positions	20	18	20	18	73 (Note)
Pulse signs	6	5	6	5	22
Grid index	1	1	1	1	4
Total					189
NOTE – By using the fact that the number of codewords in the fixed codebook is not a power of 2, 3 additional bits are saved by combining the 4 MSBs of each pulse position index into a single 13-bit word.					