3RD GENERATION PARTNERSHIP PROJECT 2
"3GPP2"

Speech Service Option Standard for Wideband Spread Spectrum Systems

PREFACE

These technical requirements form a standard for Service Option 1, a variable rate, two-way speech service option. Service Option 1 conforms to the general requirements for service options specified in TIA/EIA/IS-95 “Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System,” and in ANSI J-STD-008 "Personal Station-Base Station Compatibility Requirements for 1.8 to 2.0 Ghz Code Division Multiple Access (CDMA) Personal Communication Systems". A mobile station operating in wideband spread spectrum (CDMA) mode conforming with TIA/EIA/IS-95 and this standard can obtain speech service in any system conforming with these standards.

This standard does not address the quality or reliability of Service Option 1, nor does it cover equipment performance or measurement procedures.

SECTION SUMMARY

1. General. This section defines the terms and numeric indicators used in this document.

2. Service Option 1: Variable Data Rate Two Way Voice. This section describes the requirements for Service Option 1. Included in these requirements is the description of a speech codec algorithm for variable rate, two-way voice.

NOTES

1. TIA/EIA/IS-125, “Recommended Minimum Performance Standard for Wideband Spread Spectrum Digital Cellular System Speech Service Options,” provides specifications and measurement methods for equipment designed to provide Service Option 1.
2. “Base station” refers to the functions performed on the land side, which are typically distributed among a cell, a sector of a cell, and a mobile or personal communications switching center.
3. Section 2 uses the following verbal forms: “Shall” and “shall not” identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. “Should” and “should not” indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others; that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. “May” and “need not” indicate a course of action permissible within the limits of the standard. “Can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.
4. Footnotes appear at various points in this specification to elaborate and further clarify items discussed in the body of the specification.

CONTENTS

NOTES

5. Unless indicated otherwise, this document presents numbers in decimal form.

   Binary numbers are distinguished in the text by the use of single quotation marks. In some tables, binary values may appear without single quotation marks if table notation clearly specifies that values are binary. The character ‘x’ is used to represent a binary bit of unspecified value. For example ‘xxx00010’ represents any 8-bit binary value such that the least significant five bits equal ‘00010’.

   Hexadecimal numbers (base 16) are distinguished in the text by use of the form 0xh…h where h…h represents a string of hexadecimal digits. For example, 0x2fa1 represents a number whose binary value is ‘10111110100001’ and whose decimal value is 913. Note that the exact number of bits in the binary representation of a hexadecimal number strictly depends on the implementation requirements for the variable being represented.

6. The following conventions apply to mathematical expressions in this standard:
   - $\lfloor x \rfloor$ indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor = 1$, $\lfloor 1.0 \rfloor = 1$.
   - $\lceil x \rceil$ indicates the smallest integer greater or equal to x: $\lceil 1.1 \rceil = 2$, $\lceil 2.0 \rceil = 2$.
   - $|x|$ indicates the absolute value of x: $|-17|=17$, $|17|=17$.
   - $\oplus$ indicates exclusive OR.
   - min (x, y) indicates the minimum of x and y.
   - max (x, y) indicates the maximum of x and y.
   - In figures, $\times$ indicates multiplication. In formulas within the text, multiplication is implicit. For example, if h(n) and $p_L(n)$ are functions, then $h(n)\, p_L(n) = h(n) \times p_L(n)$.
   - x mod y indicates the remainder after dividing x by y: $x \bmod y = x - (y \lfloor x/y \rfloor)$.
   - round(x) is traditional rounding: $\mathrm{round}(x) = \lfloor x + 0.5 \rfloor$.
   - $\mathrm{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$.
   - $\sum$ indicates summation. If the summation symbol specifies initial and terminal values, and the initial value is greater than the terminal value, then the value of the summation is 0. For example, if N=0, and if f(n) represents an arbitrary function, then

     $$\sum_{n=1}^{N} f(n) = 0.$$

   - The bracket operator, [ ], isolates individual bits of a binary value. VAR[n] refers to bit n of the binary representation of the value of the variable VAR, such that VAR[0] is the least significant bit of VAR. The value of VAR[n] is either 0 or 1.

NOTES

- This standard uses the two-sided z-transform as given below. See Oppenheim, A. V. and Schafer, R. W., *Digital Signal Processing*, pp. 45 - 86.

$$F(z) = \sum_{i=-\infty}^{\infty} x_i z^{-i} .$$

7. The term “mobile station” is equivalent to the term “personal station.”

8. All references to TIA/EIA/IS-95-A shall be inclusive of text adopted by TSB74.

9. References to ANSI J-STD-008, where applicable, are located by converting Sections 6 and 7 of TIA/EIA/IS-95-A to Sections 2 and 3 of ANSI J-STD-008.

CONTENTS

REFERENCES

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

—*American National Standards:*

1. ANSI/EIA/TIA-579, *Acoustic-to-Digital and Digital-to-Acoustic Transmission Requirements for ISDN Terminals*, March 1991.
2. ANSI J-STD-008, *Personal Station-Base Station Compatibility Requirements for 1.8 to 2.0 GHz Code Division Multiple Access (CDMA) Personal Communications Systems*, TBD.

—*Other Standards:*

3. CCITT Recommendation G.711, *Pulse Code Modulation (PCM) of Voice Frequencies*, Vol. III, Geneva 1972.
4. CCITT Recommendation G.714, *Separate Performance Characteristics for the Encoding and Decoding Sides of PCM Channels Applicable to 4-Wire Voice-Frequency Interfaces*, Blue Book, Vol. III, Melbourne 1988.
5. TIA/EIA/IS-95-A, *Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System*, May 1995.
6. IEEE Standard 269-1992, *IEEE Standard Methods for Measuring Transmission Performance of Analog and Digital Telephone Sets*, 1992.
7. IEEE Standard 661-1979, *Method for Determining Objective Loudness Ratings of Telephone Connections*, 1979.
8. TIA/EIA/IS-125, Recommended Minimum Performance Standard for Digital Cellular Wideband Spread Spectrum Speech Service Option 1, May 1995.

REVISIONS

The following table summarizes the revision history of this specification:

The specification was revised from TIA/EIA/IS-96 to TIA/EIA/IS-96-A to realize an algorithm enhancement that improves the codec performance under degraded channel conditions. The revised standard, TIA/EIA/IS-96-A, is not interoperable with the original standard TIA/EIA/IS-96. TIA/EIA/IS-96-A replaces TIA/EIA/IS-96 as Speech Service Option 1. The modified sections of this document include: 1. 2.4.3.2.7 Converting the LSP Frequencies to Transmission Codes 2. 2.4.3.3.1 Converting the LSP Transmission Codes to LSP Frequencies 3. 2.5 Summary of Service Option 1 Notation
The specification was revised from TIA/EIA/IS-96-A to TIA/EIA/IS-96-B to incorporate changes for support of service negotiation.
This specification was converted from TIA/EIA/IS-96-B to TIA/EIA-96-B. During the conversion, the Preface, Notes, References, Revisions, and Bibliography were updated. No changes were made to the text section of the document.

CONTENTS

No text.

CONTENTS

1 GENERAL.. 1-1
1.1 Terms and Numeric Information 1-1
2 SERVICE OPTION 1: VARIABLE DATA RATE TWO-WAY VOICE.................................. 2-1
2.1 General Description 2-1
2.2 Service Option Number 2-1
2.3 Multiplex Option 2-1
2.3.1 Required Multiplex Option Support 2-1
2.3.2 Interface to Multiplex Option 1 2-1
2.3.2.1 Transmitted Packets 2-1
2.3.2.2 Received Packets.. 2-2
2.3.3 Negotiation for Service Option 1 2-3
2.3.3.1 Procedures Using Service Option Negotiation 2-3
2.3.3.1.1 Initialization and Connection.. 2-3
2.3.3.1.1.1 Initialization and Connection in the Mobile Station......................... 2-3
2.3.3.1.1.2 Initialization and Connection in the Base Station 2-4
2.3.3.1.2 Service Option Control Orders 2-5
2.3.3.2 Procedures Using Service Negotiation 2-6
2.3.3.2.1 Initialization and Connection.. 2-7
2.3.3.2.1.1 Mobile Station Requirements 2-7
2.3.3.2.1.2 Base Station Requirements.. 2-7
2.3.3.2.2 Service Option Control Messages 2-8
2.3.3.2.2.1 Mobile Station Requirements 2-8
2.3.3.2.2.2 Base Station Requirements.. 2-9
2.4 Variable Rate Speech Coding Algorithm 2-10
2.4.1 Introduction 2-10
2.4.2 Input Audio Interface.. 2-14
2.4.2.1 Input Audio Interface in the Mobile Station... 2-14
2.4.2.1.1 Conversion and Scaling 2-14
2.4.2.1.2 Digital Audio Input.. 2-15
2.4.2.1.3 Analog Audio Input 2-15
2.4.2.1.3.1 Transmit Level Adjustment 2-15
2.4.2.1.3.2 Band Pass Filtering 2-15

CONTENTS

2.4.2.1.3.3 Echo Return Loss........ 2-15
2.4.2.2 Input Audio Interface in the Base Station........ 2-15
2.4.2.2.1 Sampling and Format Conversion........ 2-15
2.4.2.2.2 Transmit Level Adjust........ 2-16
2.4.2.2.3 Echo Canceling........ 2-16
2.4.3 Determining the Formant Prediction Parameters........ 2-16
2.4.3.1 Form of the Formant Synthesis Filter........ 2-16
2.4.3.2 Encoding........ 2-16
2.4.3.2.1 Removing the DC Component........ 2-17
2.4.3.2.2 Windowing the Samples........ 2-17
2.4.3.2.3 Computing the Autocorrelation Function........ 2-19
2.4.3.2.4 Determining the LPC Coefficients from the Autocorrelation Function... 2-19
2.4.3.2.5 Expanding the Bandwidth........ 2-20
2.4.3.2.6 Transforming the LPC Coefficients to Line Spectrum Pairs (LSPs)........ 2-20
2.4.3.2.7 Converting the LSP Frequencies to Transmission Codes........ 2-21
2.4.3.3 Decoding........ 2-24
2.4.3.3.1 Converting the LSP Transmission Codes to LSP Frequencies........ 2-24
2.4.3.3.2 Checking the Stability of the LSP Frequencies........ 2-24
2.4.3.3.3 Low-Pass Filtering the LSP Frequencies........ 2-25
2.4.3.3.4 Interpolating the LSP Frequencies........ 2-26
2.4.3.3.5 Converting the Interpolated LSP Frequencies to LPC Coefficients........ 2-27
2.4.4 Determining the Data Rate........ 2-28
2.4.4.1 Threshold Comparing........ 2-28
2.4.4.2 Constraints on Rate Selection........ 2-28
2.4.4.3 Updating Thresholds........ 2-28
2.4.5 Determining the Pitch Prediction Parameters........ 2-29
2.4.5.1 Encoding........ 2-29
2.4.5.1.1 Computing the Pitch Lag and Pitch Gain........ 2-31
2.4.5.1.2 Implementing the Pitch Search Convolutions........ 2-32
2.4.5.1.3 Converting the Pitch Gain and Pitch Lag to the Transmission Codes... 2-33
2.4.5.2 Decoding........ 2-33
2.4.6 Determining the Excitation Codebook Parameters........ 2-33

CONTENTS

2.4.6.1 Encoding 2-33
2.4.6.1.1 Computing the Codebook Index and Codebook Gain 2-36
2.4.6.1.2 Implementing the Codebook Search Convolutions 2-37
2.4.6.1.3 Converting Codebook Parameters into Transmission Codes 2-38
2.4.6.2 Decoding 2-44
2.4.6.2.1 Converting Codebook Transmission Codes for All Rates Except 1/8 2-44
2.4.6.2.2 Converting Codebook Transmission Codes for Rate 1/8 2-47
2.4.7 Data Packing 2-48
2.4.7.1 Rate 1 Parity Check Bits and Packing 2-48
2.4.7.1.1 Parity Check Bits 2-48
2.4.7.1.2 Rate 1 Packing 2-50
2.4.7.2 Rate 1/2 Packing 2-52
2.4.7.3 Rate 1/4 Packing 2-53
2.4.7.4 Rate 1/8 Packing 2-53
2.4.8 Decoding at the Transmitting Speech Codec and the Receiving Speech Codec 2-54
2.4.8.1 Generating the Scaled Codebook Vector 2-55
2.4.8.1.1 Generating the Scaled Codebook Vector for All Rates Except Rate 1/8 2-55
2.4.8.1.2 Generating the Scaled Codebook Vector for Rate 1/8 2-55
2.4.8.2 Generating the Pitch Synthesis Filter Output 2-56
2.4.8.3 Generating the Formant Synthesis Filter Output 2-56
2.4.8.4 Updating the Memories of W(z) in the Transmitting Speech Codec 2-56
2.4.8.5 The Adaptive Postfilter in the Receiving Speech Codec 2-57
2.4.8.6 Special Cases 2-58
2.4.8.6.1 Insufficient Frame Quality (Erased) Packets 2-58
2.4.8.6.2 Rate 1 Packets 2-59
2.4.8.6.3 Rate 1 Packets with Bit Errors 2-59
2.4.8.6.4 Blanked Packets 2-60
2.4.8.6.5 All Ones Rate 1/8 Packets 2-61
2.4.9 Speech Codec Initialization 2-61
2.4.10 Output Audio Interface 2-61
2.4.10.1 Output Audio Interface in the Mobile Station 2-61

CONTENTS

2.4.10.1.1 Band Pass Filtering 2-61
2.4.10.1.2 Receive Level Adjustment 2-61
2.4.10.2 Output Audio Interface in the Base Station 2-62
2.4.10.2.1 Receive Level Adjustment 2-62
2.4.11 Summary of Encoding and Decoding 2-62
2.4.11.1 Encoding Summary 2-62
2.4.11.2 Decoding Summary 2-64
2.4.12 Allowable Delays 2-68
2.4.12.1 Allowable Transmitting Speech Codec Encoding Delay 2-68
2.4.12.2 Allowable Receiving Speech Codec Decoding Delay 2-68
2.5 Summary of Service Option 1 Notation 2-68
ANNEX A BIBLIOGRAPHY A-1

FIGURES

2.4.1-1 Speech Synthesis Structure in the Receiving Speech Codec 2-11
2.4.1-2 Bit Allocation for a Rate 1 Packet 2-13
2.4.1-3 Bit Allocation for a Rate 1/2 Packet 2-13
2.4.1-4 Bit Allocation for a Rate 1/4 Packet 2-13
2.4.1-5 Bit Allocation for a Rate 1/8 Packet 2-13
2.4.3.2.7-1 Converting the LSP Frequencies to Transmission Codes 2-21
2.4.3.3.1-1 Converting the LSP Transmission Codes to LSP Frequencies 2-24
2.4.5.1-1 Analysis-by-Synthesis Procedure for the Pitch Parameter Search 2-30
2.4.6.1-1 Analysis-by-Synthesis Procedure for Codebook Parameter Search 2-35
2.4.6.1.3-1 Converting Codebook Parameters for All Rates Except Rate 1/8 2-38
2.4.6.1.3-2 Converting Codebook Parameters for Rate 1/8 2-43
2.4.6.2.1-1 Converting Codebook Transmission Codes for All Rates Except 1/8 2-45
2.4.6.2.2-1 Converting Codebook Transmission Codes for Rate 1/8 2-47
2.4.8-1 Decoding at the Transmitting Speech Codec 2-54
2.4.8-2 Decoding at the Receiving Speech Codec 2-54

TABLES

2.3.2.1-1 Packet Types Supplied by Service Option 1 to the Multiplex Sublayer 2-2
2.3.2.2-1 Packet Types Supplied by the Multiplex Sublayer to Service Option 1 2-2
2.3.3.1.2-1 Fraction of Packets at Rate 1 and Rate 1/2 with Rate Reduction........... 2-6
2.3.3.1.2-2 Sequence Parameters for Rate Reduction 2-6
2.3.3.2-1 Valid Service Configuration Attributes for Service Option 1................... 2-7
2.3.3.2.2.1-1 Sequence Parameters for Rate Reduction 2-9
2.3.3.2.2.2-1 Service Option Control Message Type-Specific Fields 2-9
2.3.3.2.2.2-2 Fraction of Packets at Rate 1 and Rate 1/2 with Rate Reduction.......... 2-10
2.4.1-1 Parameters Used for Each Rate.. 2-12
2.4.1-2 Transmission Codes and Bit Allocations.. 2-14
2.4.3.2.2-1 Hamming Window Values $W_H(n)$.. 2-18
2.4.3.2.7-1 Number of LSP Quantization Bits... 2-22
2.4.3.2.7-2 Maximum LSP Quantization Level... 2-23
2.4.3.2.7-3 Minimum LSP Quantization Level 2-23
2.4.3.3.4-1 LSP Subframe Interpolation for All Rates 2-26
2.4.5.1.1-1 Definition of Terms for Pitch Search... 2-32
2.4.6.1-1 Circular Codebook Values c(n) 2-34
2.4.6.1.1-1 Definition of Terms for Codebook Search... 2-36
2.4.6.1.3-1 Codebook Gain Prediction Filter Function $F_G(x)$ 2-40
2.4.6.1.3-2 Codebook Quantizer (Rate 1 and Rate 1/2) 2-41
2.4.6.1.3-3 Codebook Quantizer (Rate 1/4 and Rate 1/8).................................... 2-41
2.4.6.1.3-4 Conversion for CBGAIN (Rate 1 and Rate 1/2).................................. 2-42
2.4.6.1.3-5 Conversion for CBGAIN (Rate 1/4 and Rate 1/8) 2-42
2.4.6.2.1-1 Conversion for $\hat{G}_l$ to $\hat{G}_a$ 2-46
2.4.7.1.2-1 Rate 1 Packet Structure (Part 1 of 2) 2-50
2.4.7.1.2-1 Rate 1 Packet Structure (Part 2 of 2) 2-51
2.4.7.2-1 Rate 1/2 Packet Structure.. 2-52
2.4.7.3-1 Rate 1/4 Packet Structure.. 2-53
2.4.7.4-1 Rate 1/8 Packet Structure.. 2-53
2.5-1 Summary of Service Option 1 Notation (Part 1 of 5) 2-69
2.5-1 Summary of Service Option 1 Notation (Part 2 of 5) 2-70
2.5-1 Summary of Service Option 1 Notation (Part 3 of 5) 2-71
2.5-1 Summary of Service Option 1 Notation (Part 4 of 5) 2-72
2.5-1 Summary of Service Option 1 Notation (Part 5 of 5) 2-73

TABLES

1

1 GENERAL

1.1 Terms and Numeric Information

Autocorrelation Function. A function showing the relationship of a signal with a time-shifted version of itself.

Base Station. A station in the Domestic Public Cellular Radio Telecommunications Service, other than a mobile station, used for radio communications with mobile stations.

CELP. See Code Excited Linear Predictive Coding.

Codec. The combination of an encoder and decoder in series (encoder/decoder).

Code Excited Linear Predictive Coding (CELP). A speech coding algorithm. CELP coders use codebook excitation, a long-term pitch prediction filter, and a short-term formant prediction filter.

Codebook. A set of vectors used by the speech codec in Service Option 1. For each speech codec codebook subframe, one particular vector is chosen and used to excite the speech codec's filters. The codebook vector is chosen to minimize the weighted error between the original and synthesized speech after the pitch and formant synthesis filter coefficients have been determined.

Coder. Same as "encoder."

Decoder. Generally, a device for the translation of a signal from a digital representation into an analog format. For this standard, a device which converts speech encoded in the format specified in this standard to analog or an equivalent PCM representation.

Encoder. Generally, a device for the translation of a signal into a digital representation. For this standard, a device which converts speech from an analog or its equivalent PCM representation to the digital representation described in this standard.

Formant. A resonant frequency of the human vocal tract causing a peak in the short term spectrum of speech.

IIR Filter. An infinite-duration impulse response filter is a filter for which the output, in response to an impulse input, never totally dies away. This term is usually used in reference to digital filters.

Linear Predictive Coding (LPC). A method of predicting future samples of a sequence by a linear combination of the previous samples of the same sequence. Linear Predictive Coding is frequently used in reference to a class of speech codecs.

Line Spectral Pair (LSP). A representation of digital filter coefficients in a pseudo-frequency domain. This representation has good quantization and interpolation properties.

LPC. See Linear Predictive Coding.

LSB. Least significant bit.

LSP. See Line Spectral Pair.

Mobile Station. A station in the Domestic Public Cellular Radio Telecommunications Service intended to be used while in motion or during halts at unspecified points. It is assumed that mobile stations include portable units (e.g., hand-held personal units) and units installed in vehicles.

MSB. Most significant bit.

Packet. The unit of information exchanged between service option applications in the base station and the mobile station.

Parity Check Bits. Bits added to a sequence of information bits to provide redundancy. Depending upon the method used to process the parity check bits, the receiving decoder can detect, correct, or both detect and correct certain errors.

Pitch. The fundamental frequency in speech caused by the periodic vibration of the human vocal cords.

ROLR. See Receive Objective Loudness Rating.

Receive Objective Loudness Rating (ROLR). A measure of receive audio sensitivity. ROLR is a frequency-weighted ratio of the line voltage input signal to a reference encoder to the acoustic output of the receiver. IEEE 269 defines the measurement of sensitivity and IEEE 661 defines the calculation of objective loudness rating.

TOLR. See Transmit Objective Loudness Rating.

Transmit Objective Loudness Rating (TOLR). A measure of transmit audio sensitivity. TOLR is a frequency-weighted ratio of the acoustic input signal at the transmitter to the line voltage output of the reference decoder. IEEE 269 defines the measurement of sensitivity and IEEE 661 defines the calculation of objective loudness rating.

WAEPL. Weighted Acoustic Echo Path Loss. A measure of the echo performance under normal conversation. ANSI/EIA/TIA-579 defines the measurement of WAEPL.

Zero Input Response (ZIR). The filter output caused by the non-zero initial state of the filter when no input is present.

Zero State Response (ZSR). The filter output caused by an input when the initial state of the filter is zero.

ZIR. See Zero Input Response.

ZSR. See Zero State Response.

2 SERVICE OPTION 1: VARIABLE DATA RATE TWO-WAY VOICE

2.1 General Description

Service Option 1 provides two-way voice communications between the base station and the mobile station using the dynamically variable data rate speech codec algorithm described in this standard. The transmitting speech codec takes voice samples and generates an encoded speech packet for every Traffic Channel frame.[1] The receiving station generates a speech packet from every Traffic Channel frame and supplies it to the speech codec for decoding into voice samples.

The two speech codecs communicate at one of four rates corresponding to the 9600 bps, 4800 bps, 2400 bps, and 1200 bps frame rates.

2.2 Service Option Number

The variable data rate two-way voice service option using the speech codec algorithm described by this standard shall use service option number 1 and shall be called Service Option 1.

2.3 Multiplex Option

2.3.1 Required Multiplex Option Support

Service Option 1 shall support an interface with Multiplex Option 1. Speech packets for Service Option 1 shall only be transported as primary traffic.

2.3.2 Interface to Multiplex Option 1

2.3.2.1 Transmitted Packets

The speech codec shall generate and supply exactly one packet to the multiplex sublayer every 20 milliseconds. The packet contains the service option information bits which are transmitted as primary traffic. The packet shall be one of five types as shown in Table 2.3.2.1-1. The number of bits supplied to the multiplex sublayer for each type of packet shall also be as shown in Table 2.3.2.1-1. Unless otherwise commanded, the speech codec may supply a Rate 1, Rate 1/2, Rate 1/4 or Rate 1/8 packet. Upon command, the speech codec shall generate a Blank packet. Also upon command, the speech codec shall generate a non-blank packet with a maximum rate of Rate 1/2 (see 2.4.4.2).

A Blank packet contains no bits and is used for blank-and-burst transmission of signaling traffic (see 6.1.3.3.11 of TIA/EIA/IS-95-A).

---

[1]IS-95 uses the term frame to represent a 20 ms grouping of data on the Traffic Channel. Common speech codec terminology also uses the term frame to represent a quantum of processing. For Service Option 1, the speech codec frame corresponds to speech sampled over 20 ms. The speech samples are processed into a packet. This packet is transmitted in a Traffic Channel frame.

Table 2.3.2.1-1. Packet Types Supplied by Service Option 1 to the Multiplex Sublayer

Packet Type	Bits per Packet
Rate 1	171
Rate 1/2	80
Rate 1/4	40
Rate 1/8	16
Blank	0

2.3.2.2 Received Packets

The multiplex sublayer in the mobile station categorizes every received Traffic Channel frame (see 6.2.2.2 of TIA/EIA/IS-95-A), and supplies the packet type and accompanying bits, if any, to the speech codec as shown in Table 2.3.2.2-1. The speech codec processes the bits of the packet as described in 2.4. The first five received packet types shown in Table 2.3.2.2-1 correspond to the transmitted packet types shown in Table 2.3.2.1-1. The Blank packet type occurs when the receiving station determines that a blank-and-burst frame for signaling traffic or secondary traffic was transmitted. The Rate 1 with bit errors packet type occurs when the receiving station determines that the frame was transmitted at 9600 bps and the frame has one or more bit errors. The insufficient frame quality packet type occurs when the mobile station is unable to decide upon the data rate of the received frame or when the mobile station detects a frame in error which does not belong to the Rate 1 with bit errors packet type.

Table 2.3.2.2-1. Packet Types Supplied by the Multiplex Sublayer to Service Option 1

Packet Type	Bits per Packet
Rate 1	171
Rate 1/2	80
Rate 1/4	40
Rate 1/8	16
Blank	0
Rate 1 with bit errors	171
Insufficient frame quality (erasure)	0

2.3.3 Negotiation for Service Option 1

The mobile station and base station can negotiate for Service Option 1 using either service option negotiation, as described in TIA/EIA/IS-95, or service negotiation, as described in TIA/EIA/IS-95 and ANSI J-STD-008.

2.3.3.1 Procedures Using Service Option Negotiation

The mobile station shall perform service option negotiation for Service Option 1 as described in 6.6.4.1.2 of TIA/EIA/IS-95-A. The base station shall perform service option negotiation for Service Option 1 as described in 7.6.4.1.2 of TIA/EIA/IS-95-A.

2.3.3.1.1 Initialization and Connection

2.3.3.1.1.1 Initialization and Connection in the Mobile Station

If the mobile station sends a *Service Option Response Order* accepting Service Option 1 in response to receiving a *Service Option Request Order*, (see 6.6.4.1.2.2.1 of TIA/EIA/IS-95-A), the mobile station shall initialize and connect Service Option 1 according to the following:

- If the mobile station is in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the transmitting and receiving sides within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Request Order* (see 6.6.4.1.5 of TIA/EIA/IS-95-A), or
  - The time that the mobile station sends the *Service Option Response Order* accepting Service Option 1,

  whichever is later.
- If the mobile station is not in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the transmitting side within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Request Order*,
  - The time that the mobile station sends the *Service Option Response Order* accepting Service Option 1, or
  - The time that the mobile station enters the *Conversation Substate*,

  whichever is later.
- If the mobile station is not in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the receiving side within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Request Order*,
  - The time that the mobile station sends the *Service Option Response Order* accepting Service Option 1, or
  - If not in the *Conversation Substate*, the time that the mobile station enters the *Waiting for Answer Substate*,

  whichever is later.

If the mobile station receives a *Service Option Response Order* accepting its request for Service Option 1 (see 6.6.4.1.2.2.2 of TIA/EIA/IS-95-A), the mobile station shall initialize and connect Service Option 1 according to the following:

- If the mobile station is in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the transmitting and receiving sides within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Response Order* (see 6.6.4.1.5 of TIA/EIA/IS-95-A).
- If the mobile station is not in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the transmitting side within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Response Order*, or
  - The time that the mobile station enters the *Conversation Substate*,

  whichever is later.
- If the mobile station is not in the *Conversation Substate*, the mobile station shall complete the initialization and connection of the receiving side within 200 ms of:
  - The implicit or explicit action time associated with the *Service Option Response Order*, or
  - The time that the mobile station enters the *Waiting for Answer Substate*,

  whichever is later.

Service Option 1 initializations are described in 2.4.9.

When the transmitting side of Service Option 1 is connected, Service Option 1 shall generate and transfer packets to the multiplex sublayer. When the receiving side is connected, Service Option 1 shall transfer and process packets from the multiplex sublayer. Refer to 6.1.3.3.11.3 of TIA/EIA/IS-95-A when the transmitting side of a service option is not connected.

2.3.3.1.1.2 Initialization and Connection in the Base Station

The base station should wait until the action time associated with the most recently transmitted *Service Option Response Order* or *Service Option Request Order* before initializing and connecting Service Option 1.

If the base station accepts Service Option 1 (by sending a *Service Option Response Order* as described in 7.6.4.1.2.2.1 of TIA/EIA/IS-95-A), it should initialize and connect both the transmitting and receiving side of Service Option before the called party is connected, so that both the base station and mobile station speech codecs can stabilize. The base station may defer connecting the land party audio to the speech codec.

If the base station receives an acceptance of its request for Service Option 1 (by receiving a *Service Option Response Order* as described in 7.6.4.1.2.2.2 of TIA/EIA/IS-95-A), it should initialize and connect both the transmitting and receiving side of Service Option 1 before the called party is connected so that both the base station and mobile station speech codecs can stabilize. The base station may defer connecting the land party audio to the speech codec.

When the transmitting side of Service Option 1 is connected, Service Option 1 shall generate and transfer packets to the multiplex sublayer. When the receiving side is connected, Service Option 1 shall transfer and process packets from the multiplex sublayer. Refer to 7.1.3.5.11.3 of TIA/EIA/IS-95-A when the transmitting side of a service option is not connected.

2.3.3.1.2 Service Option Control Orders

The base station may send a *Service Option Control Order* to the mobile station on the Forward Traffic Channel (see 7.7.4 of TIA/EIA/IS-95-A). In addition to pending ACTION_TIMEs for messages or orders not related to the *Service Option Control Order* for Service Option 1, the mobile station shall support at least one pending ACTION_TIME for *Service Option Control Orders* for Service Option 1. The mobile station shall not send a *Service Option Control Order* for this service option.

If Service Option 1 is active, the mobile station shall treat the two least significant bits of the ORDQ field in the *Service Option Control Order* as follows:

- If ORDQ equals 'xxx000x1', then the mobile station shall initialize both the transmitting and receiving sides of the speech codec as described in 2.4.9. The initializations shall begin at the implicit or explicit action time (see 6.6.4.1.5 of TIA/EIA/IS-95-A) and shall be completed within 40 ms. In addition, if ORDQ equals 'xxx00011' then the mobile station should disable the audio output of the speech codec for 1 second after initialization, and shall process each received Blank packet as an insufficient frame quality (erasure) packet.[2] Under this condition, the speech codec may generate a Blank packet using the insufficient frame quality processing procedure (see 2.4.8.6.1) for decoding at the transmitting speech codec.

- If Service Option 1 is active and the mobile station receives a *Service Option Control Order* having an ORDQ field in which the 3 MSBs have values given in Table 2.3.3.1.2-1, then the mobile station shall generate the fraction of those packets normally generated as Rate 1 packets (see 2.4.4.1) at either Rate 1 or Rate 1/2 as specified by the corresponding line in the table. The mobile station shall continue to use these fractions until either of the following events occurs:
  - While Service Option 1 is active, the mobile station receives a *Service Option Control Order* that specifies different fractions, or
  - Service Option 1 is initialized.

[2]This capability is to support mobile station to mobile station calls which do not use tandem vocoding.

Table 2.3.3.1.2-1. Fraction of Packets at Rate 1 and Rate 1/2 with Rate Reduction

ORDQ (binary)	Fraction of Normally Rate 1 Packets to be Rate 1	Fraction of Normally Rate 1 Packets to be Rate 1/2
000XXXXX	1	0
001XXXXX	3/4	1/4
010XXXXX	1/2	1/2
011XXXXX	1/4	3/4
100XXXXX	0	1

The mobile station may use the following procedure to perform this rate reduction: Sequences of N packets are formed as shown in Table 2.3.3.1.2-2. The first L packets in this sequence are allowed to be at Rate 1. The next N-L packets are forced to be Rate 1/2. Whenever the rate determination process (see 2.4.4.1) selects a rate other than Rate 1, the sequence is reset. This ensures that the first packet in a talk spurt will be at Rate 1, unless ORDQ equals '100XXXXX' or the speech codec has been commanded by the multiplex sublayer to generate other than a Rate 1 packet (see 2.3.2.1).

Table 2.3.3.1.2-2. Sequence Parameters for Rate Reduction

ORDQ (binary)	Sequence Length, N	Maximum Number of Contiguous Rate 1 Packets in a Sequence, L	Number of Contiguous Rate 1/2 Packets in a Sequence, N-L
000XXXXX	1	1	0
001XXXXX	4	3	1
010XXXXX	2	1	1
011XXXXX	4	1	3
100XXXXX	1	0	1

Any other *Service Option Control Order* referring to Service Option 1 and having an ORDQ field other than those described in this section shall be rejected using the *Mobile Station Reject Order* with an ORDQ field equal to '00000100' (see Table 6.7.3-1 of TIA/EIA/IS-95-A).

2.3.3.2 Procedures Using Service Negotiation

The mobile station and base station shall perform service negotiation for Service Option 1 as described in TIA/EIA/IS-95 or ANSI J-STD-008, and the negotiated service configuration shall include only valid attributes for the service option as specified in Table 2.3.3.2-1.

Table 2.3.3.2-1. Valid Service Configuration Attributes for Service Option 1

Service Configuration Attribute	Valid Selections
Forward Multiplex Option	Multiplex Option 1
Reverse Multiplex Option	Multiplex Option 1
Forward Transmission Rates	Rate Set 1 with all four rates enabled
Reverse Transmission Rates	Rate Set 1 with all four rates enabled
Forward Traffic Type	Primary Traffic
Reverse Traffic Type	Primary Traffic

2.3.3.2.1 Initialization and Connection

2.3.3.2.1.1 Mobile Station Requirements

If the mobile station accepts a service configuration, as specified in a *Service Connect Message*, that includes a service option connection using Service Option 1, the mobile station shall perform the following:

- If the service option connection is new (that is, not part of the previous service configuration), the mobile station shall perform speech codec initialization (see 2.4.9) at the action time associated with the *Service Connect Message*. The mobile station shall complete the initialization within 40 ms.
- Commencing at the action time associated with the *Service Connect Message* and continuing for as long as the service configuration includes the service option connection, Service Option 1 shall process received packets and generate and supply packets for transmission as follows:
  - If the mobile station is in the *Conversation Substate*, Service Option 1 shall process the received packets and generate and supply packets for transmission in accordance with this standard.
  - If the mobile station is not in the *Conversation Substate*, Service Option 1 shall process the received packets in accordance with this standard, and shall generate and supply Rate 1/8 Packets with all bits set to ‘1’, except when commanded to generate a Blank packet.

2.3.3.2.1.2 Base Station Requirements

If the base station establishes a service configuration, as specified in a *Service Connect Message*, that includes a service option connection using Service Option 1, the base station shall perform the following:

- If the service option connection is new (that is, not part of the previous service configuration), the base station shall perform speech codec initialization (see 2.4.9) no later than the action time associated with the *Service Connect Message*.

- Commencing at the action time associated with the *Service Connect Message* and continuing for as long as the service configuration includes the service option connection, Service Option 1 shall process received packets and generate and supply packets for transmission in accordance with this standard. The base station may defer enabling the audio input and output.

2.3.3.2.2 Service Option Control Messages

2.3.3.2.2.1 Mobile Station Requirements

The mobile station shall support one pending *Service Option Control Message* for Service Option 1.

If the mobile station receives a *Service Option Control Message* for Service Option 1, then, at the action time associated with the message, the mobile station shall process the message as follows:

1. If the MOBILE_TO_MOBILE field is equal to ‘1’, the mobile station should disable the audio output of the speech codec for 1 second after initialization and shall process each received Blank packet as an insufficient frame quality (erasure) packet. Under this condition, the speech codec may generate a Blank packet using the insufficient frame quality processing procedure (see 2.4.8.6.1) for decoding at the transmitting speech codec.

   If the MOBILE_TO_MOBILE field is equal to ‘0’, the mobile station shall process each received packet as described in section 2.4.8.

2. If the INIT_CODEC field is equal to ‘1’, the mobile station shall perform speech codec initialization (see 2.4.9). The mobile station shall complete the initialization within 40 ms.

3. If the RATE_REDUC field is equal to a value defined in Table 2.3.3.2.2.2-2, Service Option 1 shall generate the fraction of those packets normally generated as Rate 1 packets (see 2.4.4.1) at either Rate 1 or Rate 1/2 as specified by the corresponding line in Table 2.3.3.2.2.2-2. Service Option 1 shall continue to use these fractions until either of the following events occur:

   - The mobile station receives a *Service Option Control Message* specifying a different RATE_REDUC, or

   - Service Option 1 is initialized.

   Service Option 1 may use the following procedure to perform this rate reduction: Sequences of N packets as are formed as shown in Table 2.3.3.2.2.1-1. The first L packets in this sequence are allowed to be at Rate 1, the next N-L packets are forced to be Rate 1/2. Whenever the rate determination process (see 2.4.4.1) selects a rate other than Rate 1, the sequence is reset. This ensures that the first packet in a talk spurt will be at Rate 1, unless RATE_REDUC equals ‘100’ or the speech codec has been commanded by the multiplex sublayer to generate other than a Rate 1 packet (see 2.3.2.1).

Table 2.3.3.2.2.1-1. Sequence Parameters for Rate Reduction

RATE_REDUC (binary)	Sequence Length, N	Maximum Number of Contiguous Rate 1 Packets in a Sequence, L	Number of Contiguous Rate 1/2 Packets in a Sequence, N-L
000	1	1	0
001	4	3	1
010	2	1	1
011	4	1	3
100	1	0	1

If the RATE_REDUC field is not equal to a value defined in Table 2.3.3.2.2.2-2, the mobile station shall reject the message by sending a *Mobile Station Reject Order* with the ORDQ field set equal to '00000100'.

2.3.3.2.2.2 Base Station Requirements

The base station may send a *Service Option Control Message* to the mobile station. If the base station sends a *Service Option Control Message*, the base station shall include the following type-specific fields for Service Option 1:

Table 2.3.3.2.2.2-1. Service Option Control Message Type-Specific Fields

Field	Length (bits)
RATE_REDUC	3
RESERVED	3
MOBILE_TO_MOBILE	1
INIT_CODEC	1

RATE_REDUC - Rate reduction.

The base station shall set this field to the RATE_REDUC value from Table 2.3.3.2.2.2-2 corresponding to the rate reduction that the mobile station is to perform.

RESERVED - Reserved bits.

The base station shall set this field to '000'.

MOBILE_TO_MOBILE - Mobile to mobile processing.

If the mobile station is to perform mobile to mobile processing (see 2.3.3.2.2.1), the base station shall set this field to ‘1’. In addition, if the mobile station is to disable the audio output of the speech codec for 1 second after initialization, the base station shall set the INIT_CODEC field and the MOBILE_TO_MOBILE field to ‘1’. If the mobile station is not to perform mobile to mobile processing, the base station shall set this field to ‘0’.

INIT_CODEC - Initialize speech codec.

If the mobile station is to initialize the speech codec (see 2.4.9), the base station shall set this field to ‘1’. Otherwise, the base station shall set this field to ‘0’.

Table 2.3.3.2.2.2-2. Fraction of Packets at Rate 1 and Rate 1/2 with Rate Reduction

RATE_REDUC (binary)	Fraction of Normally Rate 1 Packets to be Rate 1	Fraction of Normally Rate 1 Packets to be Rate 1/2
000	1	0
001	3/4	1/4
010	1/2	1/2
011	1/4	3/4
100	0	1
All other RATE_REDUC values are reserved.		

2.4 Variable Rate Speech Coding Algorithm

2.4.1 Introduction

The speech codec uses a code excited linear predictive (CELP) coding algorithm. This technique uses a codebook to vector quantize the residual signal using an analysis-by-synthesis method. The speech codec produces a variable output data rate based on speech activity. For typical two-way telephone conversations, the average data rate is reduced by a factor of two or more with respect to the maximum data rate.

The overall speech synthesis or decoder model is shown below in Figure 2.4.1-1. First a vector is taken from one of two sources depending on the rate. For Rate 1/8 a pseudorandom vector is generated. For all other rates, a vector specified by an index $\hat{I}$ is taken from the codebook, which is a table of vectors. This vector is multiplied by a gain term $\hat{G}$, and then is filtered by the long-term pitch synthesis filter whose characteristics are governed by the pitch parameters $\hat{L}$ and $\hat{b}$. This output is filtered by the formant synthesis filter, also called the linear predictive coding filter whose characteristics are governed by the filter coefficients $\hat{a}_1, \ldots, \hat{a}_{10}$, to reproduce a speech signal. The speech signal is filtered by the adaptive postfilter.

The speech codec encoding procedure involves determining the input parameters for the decoder which minimize the perceptual difference between the synthesized speech and the original speech. The selection processes for each set of parameters are described in the following subsections. The encoding procedure also includes quantizing the parameters and packing them into data packets for transmission.

The speech codec decoding procedure involves unpacking the data packets, unquantizing the received parameters, and reconstructing the speech signal from these parameters. The reconstruction consists of filtering the generated codebook vector as shown in Figure 2.4.1-1.

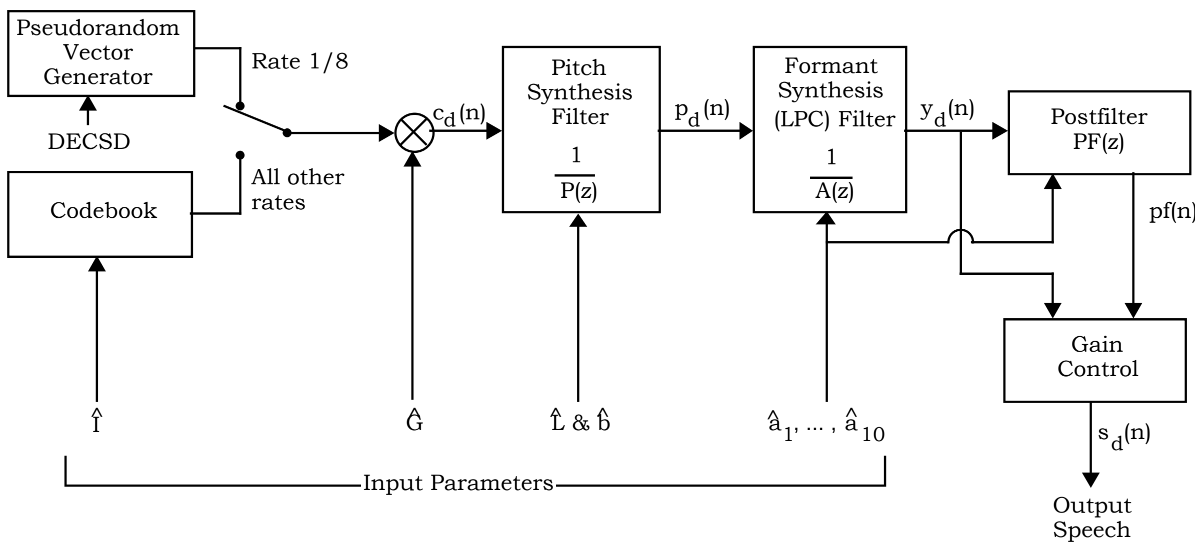


Figure 2.4.1-1. Speech Synthesis Structure in the Receiving Speech Codec

The input speech is sampled at 8 kHz. This speech is broken down into 20 ms speech codec frames consisting of 160 samples. The formant synthesis (LPC) filter coefficients are updated once per frame, regardless of the data rate selected. The number of bits used to encode the LPC parameters is a function of the selected data rate. Within each frame, the pitch parameters are updated a varying number of times, where the number of pitch parameter updates is also a function of the selected data rate. Similarly, the codebook parameters are updated a varying number of times, again where the number of updates is a function of the selected data rate. Table 2.4.1-1 describes the various parameters used for each rate.

Table 2.4.1-1. Parameters Used for Each Rate

Parameter	Rate 1	Rate 1/2	Rate 1/4	Rate 1/8
Linear predictive coding (LPC) updates per frame	1	1	1	1
Samples per LPC update, L_A	160 (20 ms)	160 (20 ms)	160 (20 ms)	160 (20 ms)
Bits per LPC update	40	20	10	10
Pitch updates (subframes) per frame	4	2	1	0
Samples per pitch subframe, L_P	40 (5 ms)	80 (10 ms)	160 (20 ms)	–
Bits per pitch update	10	10	10	–
Codebook updates (subframes) per frame	8	4	2	1
Samples per codebook subframe, L_C	20 (2.5 ms)	40 (5 ms)	80 (10 ms)	160 (20 ms)
Bits per codebook update	10	10	10	6*
*Note: Rate 1/8 uses six bits for pseudorandom excitation, rather than using the codebook.				

The components for each rate packet are shown in Figures 2.4.1-2 through 2.4.1-5. In these figures, each LPC frame corresponds to one 160-sample frame of speech.

The number in the LPC block of each figure is the number of bits used at that rate to encode the LPC coefficients. Each pitch block corresponds to a pitch update within each frame, and the number in each pitch block corresponds to the number of bits used to encode the updated pitch parameters. For example, at Rate 1, the pitch parameters are updated four times, once for each quarter of the speech frame, each time using ten bits to encode the new pitch parameters. This is done a varying number of times for the other rates as shown. Note that a pitch update is not done at Rate 1/8, as this rate is used to encode frames when little or no speech is present and pitch redundancies do not exist. Similarly, each codebook block corresponds to a codebook update within each frame, and the number in each codebook block corresponds to the number of bits used to encode the updated codebook parameters. For example, at Rate 1, the codebook parameters are updated eight times, once for each eighth of the speech frame, each time using ten bits to encode the parameters. The number of updates decreases as the rate decreases.

LPC Frame	40								Total = 160 bits
Pitch Subframe	10		10		10		10		(plus 11 parity
Codebook Subframe	10	10	10	10	10	10	10	10	check bits)

Figure 2.4.1-2. Bit Allocation for a Rate 1 Packet

LPC Frame	20				
Pitch Subframe	10		10		Total = 80 bits
Codebook Subframe	10	10	10	10	

Figure 2.4.1-3. Bit Allocation for a Rate 1/2 Packet

LPC Frame	10		
Pitch Subframe	10		Total = 40 bits
Codebook Subframe	10	10	

Figure 2.4.1-4. Bit Allocation for a Rate 1/4 Packet

LPC Frame	10	
Pitch Subframe	0	Total = 16 bits
Codebook Subframe	6	

Figure 2.4.1-5. Bit Allocation for a Rate 1/8 Packet

Table 2.4.1-2 lists all the parameter codes transmitted for each rate packet. The following list describes each parameter:

LSPi	Line Spectral Pair frequency i.
PLAGi	Pitch Lag for the ith pitch subframe.
PGAINi	Pitch Gain for the ith pitch subframe.
CBINDEXi	Codebook Index for the ith codebook subframe.
CBGAINi	Codebook Gain for the ith codebook subframe.
CBSEED	Random Seed for Rate 1/8 packets.
PCB	Parity Check Bits used to detect and correct errors in a Rate 1 packet.

This standard refers to the LSB of a particular code as CODE[0] and the more significant bits as CODE[1], CODE[2], etc. For example, if LSP1 = ‘1011’ in binary for a maximum rate frame, LSP1[0] = ‘1’, LSP1[1] = ‘1’, LSP1[2] = ‘0’, and LSP1[3] = ‘1’.

Table 2.4.1-2. Transmission Codes and Bit Allocations

	Rate					Rate			
Code	**1**	**1/2**	**1/4**	**1/8**	**Code**	**1**	**1/2**	**1/4**	**1/8**
LSP1	4	2	1	1	CBINDEX1	7	7	7	-
LSP2	4	2	1	1	CBINDEX2	7	7	7	-
LSP3	4	2	1	1	CBINDEX3	7	7	-	-
LSP4	4	2	1	1	CBINDEX4	7	7	-	-
LSP5	4	2	1	1	CBINDEX5	7	-	-	-
LSP6	4	2	1	1	CBINDEX6	7	-	-	-
LSP7	4	2	1	1	CBINDEX7	7	-	-	-
LSP8	4	2	1	1	CBINDEX8	7	-	-	-
LSP9	4	2	1	1	CBGAIN1	3	3	3	2
LSP10	4	2	1	1	CBGAIN2	3	3	3	-
PLAG1	7	7	7	-	CBGAIN3	3	3	-	-
PLAG2	7	7	-	-	CBGAIN4	3	3	-	-
PLAG3	7	-	-	-	CBGAIN5	3	-	-	-
PLAG4	7	-	-	-	CBGAIN6	3	-	-	-
PGAIN1	3	3	3	-	CBGAIN7	3	-	-	-
PGAIN2	3	3	-	-	CBGAIN8	3	-	-	-
PGAIN3	3	-	-	-	CBSEED	-	-	-	4
PGAIN4	3	-	-	-	PCB	11	-	-	-

2.4.2 Input Audio Interface

2.4.2.1 Input Audio Interface in the Mobile Station

The input audio may be either an analog or digital signal.

2.4.2.1.1 Conversion and Scaling

The speech shall be sampled at a rate of 8000 samples per second. The speech shall be quantized to a uniform PCM format with at least 13 bits of dynamic range.

The quantities in this standard assume a 14-bit integer input quantization with a range of ±8031. The following speech codec discussion assumes this 14-bit integer quantization. If the speech codec uses a different quantization, then appropriate scaling should be used.

2.4.2.1.2 Digital Audio Input

If the input audio is an 8-bit μlaw PCM signal, it shall be converted to a uniform PCM format according to Table 2 in CCITT Recommendation G.711 "Pulse Code Modulation (PCM) of Voice Frequencies."

2.4.2.1.3 Analog Audio Input

If the input is in analog form, the mobile station shall sample the analog speech and shall convert the samples to a digital format for speech codec processing. This shall be done by either the following or an equivalent method. First, the input gain audio level is adjusted. Then, the signal is bandpass filtered to prevent aliasing. Finally, the filtered signal is sampled and quantized (see 2.4.2.1.1).

2.4.2.1.3.1 Transmit Level Adjustment

The mobile station shall have a transmit objective loudness rating (TOLR) equal to -46 dB, when transmitting to a reference base station (see 2.4.10.2.1). The loudness ratings are described in IEEE Standard 661-1979 "IEEE Standard Method for Determining Objective Loudness Ratings of Telephone Connections." Measurement techniques and tolerances are described in TIA/EIA/IS-125.

2.4.2.1.3.2 Band Pass Filtering

Input anti-aliasing filtering shall conform to CCITT Recommendation G.714 "Separate Performance Characteristics for the Encoding and Decoding Sides of PCM Channels Applicable to 4-Wire Voice-Frequency Interfaces." Additional anti-aliasing filtering may be provided by the manufacturer.

2.4.2.1.3.3 Echo Return Loss

Provision shall be made to ensure adequate isolation between receive and transmit audio paths in all modes of operation. When no external transmit audio is present, the speech codec shall not generate packets at rates higher than Rate 1/8 (see 2.4.4) due to acoustic coupling of the receive audio into the transmit audio path (specifically with the receive audio at full volume). Target levels of 45 dB WAEPL should be met. See ANSI/EIA/TIA Standard 579 "Acoustic-to-Digital and Digital-to-Acoustic Transmission Requirements for ISDN Terminals." Refer to the requirements stated in TIA/EIA/IS-125.

2.4.2.2 Input Audio Interface in the Base Station

2.4.2.2.1 Sampling and Format Conversion

The base station converts the input speech (analog, μlaw companded Pulse Code Modulation, or other format) into a uniform quantized PCM format with at least 13 bits of dynamic range. The sampling rate is 8000 samples per second. The sampling and conversion process shall be as in 2.4.2.1.1.

2.4.2.2.2 Transmit Level Adjust

The base station shall set the transmit level so that a 1004 Hz tone at a level of 0 dBm0 at the network interface produces a level 3.17 dB below maximum amplitude at the output of the quantizer. Measurement techniques and tolerances are described in TIA/EIA/IS-125.

2.4.2.2.3 Echo Canceling

The base station shall provide a method to cancel echoes returned by the PSTN interface.[3] The echo canceling function should provide at least 30 dB of echo return loss enhancement. The echo canceling function should work over a range of PSTN echo return delays from 0 to 48 ms.

2.4.3 Determining the Formant Prediction Parameters

2.4.3.1 Form of the Formant Synthesis Filter

The formant synthesis filter is equivalent to the traditional LPC formant synthesis filter. The transfer function for the formant prediction error filter, which removes the short term redundancies in the speech, is

$$A(z) = 1 - \sum_{i=1}^{P} a_i z^{-i} \quad . \qquad (2.4.3.1\text{-}1)$$

The filter is a tenth-order filter (i.e. P equals 10). The formant synthesis filter, which reinserts the redundancies at the receiving end, is given by the inverse of A(z):

$$\frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}} \quad . \qquad (2.4.3.1\text{-}2)$$

The LPC coefficients, a_i, are computed from the input speech.

2.4.3.2 Encoding

The encoding process begins by determining the formant prediction parameters. This is performed by the following steps:

1. Remove the DC from the input samples.
2. Window the input samples using a Hamming window.
3. Compute the eleven values of the autocorrelation function corresponding to shifts from 0 to 10 samples.

---

[3]Because of the relatively long delays inherent in the speech coding and transmitting processes, echoes that are not sufficiently suppressed are noticeable to the mobile station user.

4. Determine the LPC coefficients from the autocorrelation values.
5. Scale the LPC coefficients to perform bandwidth expansion.
6. Transform the scaled coefficients to LSP frequencies.
7. Convert the LSP frequencies into LSP codes
(these codes are placed into the packet for transmission).

2.4.3.2.1 Removing the DC Component

A DC block is inserted to prevent a DC offset from artificially increasing R(0) (see 2.4.3.2.3) and thus disrupting the rate decision algorithm (see 2.4.4). One possible method for removing the DC offset is to take the average of the 160 samples in the current window of speech, low pass filter this average to prevent large discontinuities at the frame boundaries, and subtract this low passed filtered average from the 160 samples in the current window. One possible low pass filter for the DC offset would be to add 0.75 times the previous value to 0.25 times the current value.

2.4.3.2.2 Windowing the Samples

The coefficients are computed from a Hamming window of speech centered at the center of the fourth Rate 1 pitch subframe. The window is 160 samples long (i.e., L_A equals 160).

Let s(n) be the input speech signal with the DC removed, where s(0) denotes the first sample of the current frame. The windowed speech signal is defined as

$$s_W(n) = s(n + 60)\, W_H(n) \qquad \text{for } 0 \le n \le L_A - 1 \quad , \qquad (2.4.3.2.2\text{-}1)$$

where the Hamming window is defined in Table 2.4.3.2.2-1 in hexadecimal values with 14 fractional bits. The value for all n outside of the range specified in the table is equal to zero.

Note the offset of 60 samples, which results in the window of speech being centered between the 139th and 140th sample of the current 160 sample frame of speech.

Table 2.4.3.2.2-1. Hamming Window Values $W_H(n)$

n	$W_H(n)$	n	n	$W_H(n)$	n	n	$W_H(n)$	n
0	0x051f	159	27	0x1459	132	54	0x3247	105
1	0x0525	158	28	0x1560	131	55	0x333f	104
2	0x0536	157	29	0x166d	130	56	0x3431	103
3	0x0554	156	30	0x177f	129	57	0x351c	102
4	0x057d	155	31	0x1895	128	58	0x3600	101
5	0x05b1	154	32	0x19af	127	59	0x36db	100
6	0x05f2	153	33	0x1acd	126	60	0x37af	99
7	0x063d	152	34	0x1bee	125	61	0x387a	98
8	0x0694	151	35	0x1d11	124	62	0x393d	97
9	0x06f6	150	36	0x1e37	123	63	0x39f6	96
10	0x0764	149	37	0x1f5e	122	64	0x3aa6	95
11	0x07dc	148	38	0x2087	121	65	0x3b4c	94
12	0x085e	147	39	0x21b0	120	66	0x3be9	93
13	0x08ec	146	40	0x22da	119	67	0x3c7b	92
14	0x0983	145	41	0x2403	118	68	0x3d03	91
15	0x0a24	144	42	0x252d	117	69	0x3d80	90
16	0x0ad0	143	43	0x2655	116	70	0x3df3	89
17	0x0b84	142	44	0x277b	115	71	0x3e5b	88
18	0x0c42	141	45	0x28a0	114	72	0x3eb7	87
19	0x0d09	140	46	0x29c2	113	73	0x3f09	86
20	0x0dd9	139	47	0x2ae1	112	74	0x3f4f	85
21	0x0eb0	138	48	0x2bfd	111	75	0x3f89	84
22	0x0f90	137	49	0x2d15	110	76	0x3fb8	83
23	0x1077	136	50	0x2e29	109	77	0x3fdb	82
24	0x1166	135	51	0x2f39	108	78	0x3ff3	81
25	0x125b	134	52	0x3043	107	79	0x3fff	80
26	0x1357	133	53	0x3148	106			

2.4.3.2.3 Computing the Autocorrelation Function

Following the windowing operation, the kth value of the autocorrelation function is computed as

$$R(k) = \sum_{m=0}^{L_A - 1 - k} s_w(m)\ s_w(m + k)\ . \qquad (2.4.3.2.3\text{-}1)$$

Only the first 11 values of the autocorrelation function, R(0) through R(10), need to be computed from the windowed speech signal within the analysis window.

2.4.3.2.4 Determining the LPC Coefficients from the Autocorrelation Function

Next, the LPC coefficients are obtained from the autocorrelation function. A method is Durbin's recursion, described as follows.[4]

$$
\begin{array}{l}
\{ \\
\quad E^{(0)} = R(0) \\
\quad i = 1 \\
\quad \text{while } (i \le P) \\
\qquad \{ \\
\qquad\quad k_i = \left\{ R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right\} / E^{(i-1)} \\
\qquad\quad \alpha_i^{(i)} = k_i \\
\qquad\quad j = 1 \\
\qquad\quad \text{while } (j \le i-1) \\
\qquad\qquad \{ \\
\qquad\qquad\quad \alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \\
\qquad\qquad\quad j = j + 1 \\
\qquad\qquad \} \\
\qquad\quad E^{(i)} = (1 - k_i^2)\ E^{(i-1)} \\
\qquad\quad i = i + 1 \\
\qquad \} \\
\}
\end{array}
$$

The LPC coefficients before bandwidth expansion are $\alpha_j^{(P)}$, where $1 \le j \le P$.

---

[4]See Rabiner, L. R. and Schafer, R. W., *Digital Processing of Speech Signals*, (New Jersey: Prentice-Hall Inc, 1978), pp. 411-412. The superscripts in parentheses represent the stage of Durbin's recursion. For example $\alpha_j^{(i)}$ refers to α_j at the ith stage.

2.4.3.2.5 Expanding the Bandwidth

Next, the LPC coefficients are scaled to perform bandwidth expansion before they are transformed into LSP frequencies. Each LPC coefficient, $\alpha_i^{(P)}$, is scaled by ß^i (ß to the ith power) as follows:

$$a_i = ß^i \alpha_i^{(P)} \qquad 1 \le i \le P \quad , \qquad (2.4.3.2.5\text{-}1)$$

where ß is 0.9883.

2.4.3.2.6 Transforming the LPC Coefficients to Line Spectrum Pairs (LSPs)

Next, the LPC coefficients are transformed into line spectrum pair frequencies. The basic computation of the LSP frequencies follows.

As before, A(z) is given by

$$A(z) = 1 - a_1 z^{-1} - \ldots - a_{10} z^{-10} \quad , \qquad (2.4.3.2.6\text{-}1)$$

where a_i ($1 \le i \le 10$) are the LPC coefficients as described above.

Define $P_A(z)$ and $Q_A(z)$ as follows:

$$P_A(z) = A(z) + z^{-11}A(z^{-1}) = 1 + p_1 z^{-1} + \ldots + p_5 z^{-5} + p_5 z^{-6} + \ldots + p_1 z^{-10} + z^{-11} \qquad (2.4.3.2.6\text{-}2)$$

$$Q_A(z) = A(z) - z^{-11}A(z^{-1}) = 1 + q_1 z^{-1} + \ldots + q_5 z^{-5} - q_5 z^{-6} - \ldots - q_1 z^{-10} - z^{-11} \quad , \qquad (2.4.3.2.6\text{-}3)$$

where

$$p_i = - a_i - a_{11-i} \qquad 1 \le i \le 5 \qquad (2.4.3.2.6\text{-}4)$$

$$q_i = - a_i + a_{11-i} \qquad 1 \le i \le 5. \qquad (2.4.3.2.6\text{-}5)$$

The LSP frequencies are the ten roots which exist between w = 0 and w = 0.5 in the following two equations:

$$P'(w) = \cos 5(2\pi w) + p'_1 \cos 4(2\pi w) + \ldots + p'_4 \cos (2\pi w) + p'_5/2 \qquad (2.4.3.2.6\text{-}6)$$

$$Q'(w) = \cos 5(2\pi w) + q'_1 \cos 4(2\pi w) + \ldots + q'_4 \cos (2\pi w) + q'_5/2 \quad , \qquad (2.4.3.2.6\text{-}7)$$

where the p' and q' values are computed recursively as follows from the p and q values defined above.

$$p'_0 = q'_0 = 1 \qquad (2.4.3.2.6\text{-}8)$$

$$p'_i = p_i - p'_{i-1} \qquad 1 \le i \le 5 \qquad (2.4.3.2.6\text{-}9)$$

$$q'_i = q_i + q'_{i-1} \qquad 1 \le i \le 5. \qquad (2.4.3.2.6\text{-}10)$$

Since the formant synthesis (LPC) filter is stable, the roots of the two functions alternate; the smallest root, w_1, is the lowest root of P'(w), the next smallest root, w_2, is the lowest root of Q'(w), etc. Thus, w_1, w_3, w_5, w_7, and w_9 are the roots of P'(w), and w_2, w_4, w_6, w_8, and w_{10} are the roots of Q'(w).

2.4.3.2.7 Converting the LSP Frequencies to Transmission Codes

Once the LSP frequencies have been computed and the data rate has been selected (see 2.4.4), each LSP frequency is converted for transmission. The converter is shown in Figure 2.4.3.2.7-1.

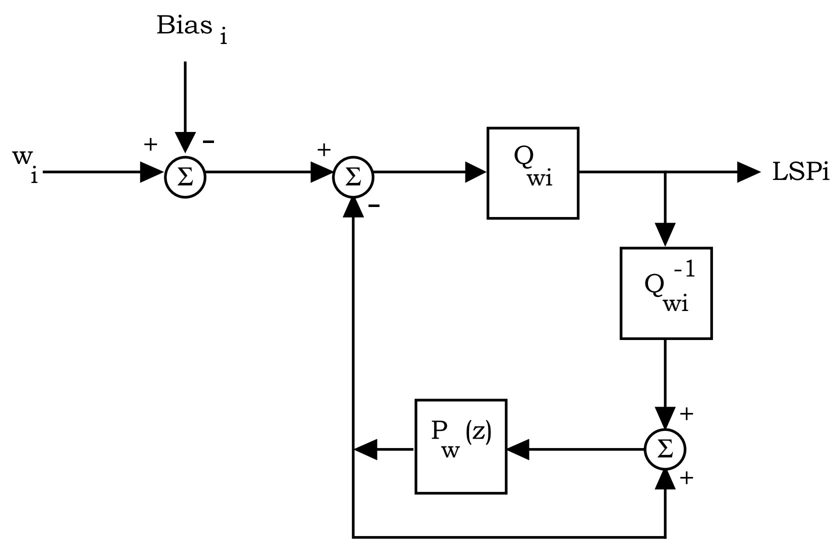


Figure 2.4.3.2.7-1. Converting the LSP Frequencies to Transmission Codes

Each of the ten LSP frequencies centers roughly around a bias value (the frequencies equal the bias values when the input speech has flat spectral characteristics and no formant prediction can be performed). The bias used for each LSP frequency is as follows:

$$\mathrm{Bias}_i = \frac{0.5i}{P+1} \qquad 1 \le i \le 10\ , \qquad (2.4.3.2.7\text{-}1)$$

where P is equal to 10.

The predictor P_w is a function of the rate:

$$P_w(z) = 0.90625\ z^{-1} \qquad \text{rate} = 1/2,\ 1/4,\ 1/8 \qquad (2.4.3.2.7\text{-}2)$$

$$P_w(z) = 0.0\ z^{-1} \qquad \text{rate} = 1\ .$$

The predictor is updated once per frame unless a Blank packet has been requested. There is one predictor for each LSP frequency.

The quantizer, Q_{wi}, for the ith LSP frequency is a linear quantizer which varies in dynamic range and step size with rate. Each LSP frequency is quantized as follows:

$$Q_{wi}(x) = \max\left(0, \min\left(2^N - 1, Q_{ti}(x)\right)\right) \quad , \tag{2.4.3.2.7-3}$$

where

$$Q_{ti}(x) = \mathrm{round}\left((2^N - 1)\frac{x - Q_{wi}^{min}}{Q_{wi}^{max} - Q_{wi}^{min}} \right) \quad . \tag{2.4.3.2.7-4}$$

N is the number of bits of quantization, Q_{wi}^{max} is the maximum quantization level given for the ith coefficient, Q_{wi}^{min} is the minimum quantization level given for the ith coefficient, and round (x) is the function rounding to the closest integer. The number of LSP quantization bits, N, is given in Table 2.4.3.2.7-1. The maximum quantization level, Q_{wi}^{max}, is given in Table 2.4.3.2.7-2. The minimum quantization level, Q_{wi}^{min}, is given in Table 2.4.3.2.7-3. Note that Equation 2.4.3.2.7-3 is a limiting function to maintain $Q_{wi}(x)$ between 0 and 2^N - 1.

Table 2.4.3.2.7-1. Number of LSP Quantization Bits

LSP Frequency	**Rate 1**	**Rate 1/2**	**Rate 1/4**	**Rate 1/8**
w_1	4	2	1	1
w_2	4	2	1	1
w_3	4	2	1	1
w_4	4	2	1	1
w_5	4	2	1	1
w_6	4	2	1	1
w_7	4	2	1	1
w_8	4	2	1	1
w_9	4	2	1	1
w_{10}	4	2	1	1
Total	**40**	**20**	**10**	**10**

Table 2.4.3.2.7-2. Maximum LSP Quantization Level

LSP Frequency	Rate 1	Rate 1/2	Rate 1/4	Rate 1/8
w_1	0.05455	0.015	0.01	0.01
w_2	0.03410	0.015	0.01	0.01
w_3	0.05364	0.03	0.01	0.01
w_4	0.06318	0.03	0.01	0.01
w_5	0.06273	0.03	0.01	0.01
w_6	0.05227	0.02	0.01	0.01
w_7	0.05182	0.02	0.01	0.01
w_8	0.03636	0.02	0.01	0.01
w_9	0.03091	0.02	0.01	0.01
w_{10}	0.01545	0.02	0.01	0.01

Table 2.4.3.2.7-3. Minimum LSP Quantization Level

LSP Frequency	Rate 1	Rate 1/2	Rate 1/4	Rate 1/8
w_1	-0.03045	-0.015	-0.01	-0.01
w_2	-0.06590	-0.015	-0.01	-0.01
w_3	-0.08636	-0.03	-0.01	-0.01
w_4	-0.09681	-0.03	-0.01	-0.01
w_5	-0.10727	-0.03	-0.01	-0.01
w_6	-0.10273	-0.02	-0.01	-0.01
w_7	-0.10318	-0.02	-0.01	-0.01
w_8	-0.09863	-0.02	-0.01	-0.01
w_9	-0.07409	-0.02	-0.01	-0.01
w_{10}	-0.07455	-0.02	-0.01	-0.01

2.4.3.3 Decoding

The decoding process consists of the following steps:

1. Convert the LSP transmission codes to LSP frequencies.
2. Check the stability of the LSP frequencies.
3. Low-pass filter the LSP frequencies.
4. Interpolate the LSP frequencies.
5. Convert the interpolated LSP frequencies to LPC coefficients.

The steps taken by the receiving decoder (see 2.4.11.2) are similar to the transmitting speech codec except for the possibility of receiving a packet type equal to insufficient frame quality (see 2.3.2.2).

2.4.3.3.1 Converting the LSP Transmission Codes to LSP Frequencies

The LSPs are decoded at both the transmitting encoder and the receiving decoder. First, the LSP codes are used to compute the regenerated LSP frequencies, $\tilde{w}_i$ (see Figure 2.4.3.3.1-1).

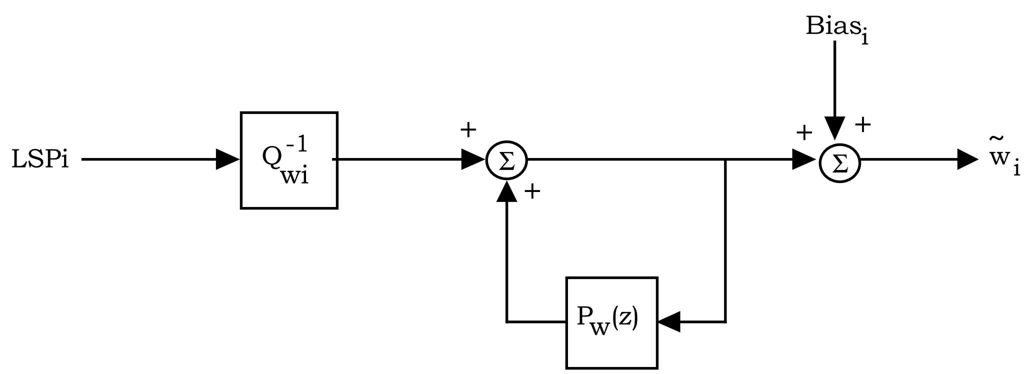


Figure 2.4.3.3.1-1. Converting the LSP Transmission Codes to LSP Frequencies

The rate dependent predictor $P_W(z)$ is the same as in Equation 2.4.3.2.7-2. The predictor is updated for every packet except for the Blank packet. The bias is given in Equation 2.4.3.2.7-1. The quantizer is the inverse of that given by Equation 2.4.3.2.7-3.

2.4.3.3.2 Checking the Stability of the LSP Frequencies

Before converting the LSP frequencies back to LPC coefficients, a check is done to ensure that the resulting filter has not been made unstable due to quantization noise or channel errors injecting noise into one or many LSP frequencies. Stability is guaranteed if the LSP frequencies remain ordered. In addition, the LSP frequencies are forced to be at least 80 Hz apart to prevent unusually large peaks in the formant synthesis filter response. This ordering and minimum spacing are enforced using the following algorithm:

```
{
  w̃ 0 = 0.0
  i = 0
  while (i < 10)
     {
       if  ( (w̃ i+1 - w̃ i) < Δw̃ min)
          w̃ i+1 = w̃ i + Δw̃ min
       i = i + 1
     }
  w̃ 11 = 0.5
  while (i > 0)
     {
       if  ( (w̃ i+1 - w̃ i) < Δw̃ min)
          w̃ i = w̃ i+1 - Δw̃ min
       i = i - 1
     }
}
```

A $\Delta\tilde{w}_{min}$ of 0.01 is used. This results in 80 Hz separation in the LSP domain.

2.4.3.3.3 Low-Pass Filtering the LSP Frequencies

Next, the LSP frequencies are low-pass filtered as follows to remove some of the quantization noise effects at lower rates:

$$\hat{w}_i(\text{current frame}) = SM\,\hat{w}_i(\text{previous frame}) + (1\text{-}SM)\,\tilde{w}_i(\text{current frame})\ . \qquad (2.4.3.3.3\text{-}1)$$

The value of SM depends upon the packet rate. For both the encoder and decoder, a counter is used to track the number of consecutive packets that are either Rate 1/4 or Rate 1/8. If the current packet is either Rate 1/4 or Rate 1/8, the counter is incremented. If the current packet is either Rate 1 or Rate 1/2, the counter is set to zero. Otherwise the counter is unchanged. The value of SM that is used in Equation 2.4.3.3.3-1 is given in Equation 2.4.3.3.3-2. A received packet categorized as Rate 1 with bit errors is treated as a Rate 1 packet if the packet is detected as having one or fewer errors; otherwise the packet is treated as an erasure (see 2.4.8.6.3).

$$SM = \begin{cases} 0 & \text{if packet is Rate 1} \\ 0.125 & \text{if packet is Rate } 1/2 \\ 0.125 & \text{if packet is Rate } 1/4 \text{ or } 1/8 \text{ and counter } < 10 \\ 0.9 & \text{if packet is Rate } 1/4 \text{ or } 1/8 \text{ and counter } \geq 10 \\ 0.875 & \text{if an insufficient frame quality packet (erasure)} \end{cases} \qquad (2.4.3.3.3\text{-}2)$$

2.4.3.3.4 Interpolating the LSP Frequencies

Next, the LSP frequencies are interpolated for each subframe of the pitch and codebook searches in the selected rate.

In calculating the original LPC coefficients, a speech window centered between the 139th and 140th sample of the frame was used. In performing the pitch and codebook searches for the smaller subframes, LPC coefficients which are accurate at the center of the particular pitch subframe should be used (except at Rate 1/8, where it is the center of the single codebook subframe). These LPC coefficients are approximated by interpolating between the previous frame's and the current frame's LSP frequencies, and then converting the resulting interpolated LSP frequencies back into LPC coefficients.

The exact interpolation used for each subframe of each rate is shown in Table 2.4.3.3.4-1. In all cases $\hat{w}_i$(previous) is the ith filtered LSP frequency from the previous frame and $\hat{w}_i$(current) is the ith filtered LSP frequency from the current frame.

Table 2.4.3.3.4-1. LSP Subframe Interpolation for All Rates

Rate 1	**For pitch subframe**	**For codebook subframes**
$\hat{w}'_i = 0.75\,\hat{w}_i$(previous) $+ 0.25\,\hat{w}_i$(current)	1	1 and 2
$\hat{w}'_i = 0.5\,\hat{w}_i$(previous) $+ 0.5\,\hat{w}_i$(current)	2	3 and 4
$\hat{w}'_i = 0.25\,\hat{w}_i$(previous) $+ 0.75\,\hat{w}_i$(current)	3	5 and 6
$\hat{w}'_i = \hat{w}_i$(current)	4	7 and 8

Rate 1/2	**For pitch subframe**	**For codebook subframe**
$\hat{w}'_i = 0.625\,\hat{w}_i$(previous) $+ 0.375\,\hat{w}_i$(current)	1	1 and 2
$\hat{w}'_i = 0.125\,\hat{w}_i$(previous) $+ 0.875\,\hat{w}_i$(current)	2	3 and 4

Rate 1/4	**For pitch subframe**	**For codebook subframe**
$\hat{w}'_i = 0.375\,\hat{w}_i$(previous) $+ 0.625\,\hat{w}_i$(current)	1	1 and 2

Rate 1/8	**For pitch subframe**	**For codebook subframe**
$\hat{w}'_i = 0.375\,\hat{w}_i$(previous) $+ 0.625\,\hat{w}_i$(current)	–	1

2.4.3.3.5 Converting the Interpolated LSP Frequencies to LPC Coefficients

The converted LSP frequencies are used by the receiving decoder for speech generation as described in 2.4.11.2. In addition, the interpolated LSP frequencies are converted back into LPC coefficients for use in the pitch and codebook searches. The conversion method is as follows:

First, $\hat{P}_A(z)$ and $\hat{Q}_A(z)$ are computed from the LSP frequencies using Equations 2.4.3.3.5-1 and 2.4.3.3.5-2:

$$\hat{P}_A(z) = (1 + z^{-1}) \prod_{j=1}^{5} (1 - 2z^{-1}\cos(2\pi\hat{w}'_{(2j-1)}) + z^{-2}) \qquad (2.4.3.3.5\text{-}1)$$

and

$$\hat{Q}_A(z) = (1 - z^{-1}) \prod_{j=1}^{5} (1 - 2z^{-1}\cos(2\pi\hat{w}'_{(2j)}) + z^{-2}) \ . \qquad (2.4.3.3.5\text{-}2)$$

Then the LPC coefficients are computed from the coefficients of $\hat{P}_A(z)$ and $\hat{Q}_A(z)$ as follows:

$$A(z) = \frac{\hat{P}_A(z) + \hat{Q}_A(z)}{2}$$

$$= 1 + \frac{(\hat{p}_1 + \hat{q}_1)}{2} z^{-1} + \ldots + \frac{(\hat{p}_5 + \hat{q}_5)}{2} z^{-5} + \frac{(\hat{p}_5 - \hat{q}_5)}{2} z^{-6} + \ldots + \frac{(\hat{p}_1 - \hat{q}_1)}{2} z^{-10}$$

$$= 1 - \hat{a}_1 z^{-1} - \ldots - \hat{a}_{10} z^{-10} \ , \qquad (2.4.3.3.5\text{-}3)$$

so

$$\hat{a}_i = \begin{cases} -\dfrac{\hat{p}_i + \hat{q}_i}{2} & 1 \le i \le 5 \\ -\dfrac{\hat{p}_{11-i} - \hat{q}_{11-i}}{2} & 6 \le i \le 10 \end{cases} \ . \qquad (2.4.3.3.5\text{-}4)$$

The LPC coefficients for the particular subframe are the $\hat{a}_i$ defined in Equation 2.4.3.3.5-4.

2.4.4 Determining the Data Rate

2.4.4.1 Threshold Comparing

The speech codec makes an initial rate selection based on the energy in the frame and a set of three thresholds. The energy in the frame is estimated by R(0), which is defined in Equation 2.4.3.2.3-1. The three thresholds are maintained as described in 2.4.4.3. They are based upon an estimate of the background noise level, B_i, computed for the current, or ith, frame.

R(0) is compared with the three thresholds: $T_1(B_i)$, $T_2(B_i)$, and $T_3(B_i)$. If R(0) is greater than all three thresholds, Rate 1 is selected. If R(0) is greater than only two thresholds, Rate 1/2 is selected. If R(0) is greater than only one threshold, Rate 1/4 is selected. If R(0) is below all three thresholds, Rate 1/8 is selected.

2.4.4.2 Constraints on Rate Selection

The rate selected by the procedure described in 2.4.4.1 is used for the current frame except where it is modified by the following constraints.

First, the data rate is only permitted to decrease by one rate per frame. If the previous frame was encoded at Rate 1 and the initial rate selection for the current frame is Rate 1/4 or Rate 1/8, then Rate 1/2 is chosen. Similarly, if the previous frame was encoded at Rate 1/2 and the initial selection for the current frame is Rate 1/8, then Rate 1/4 is chosen. There is no restriction on increases in data rate.

Second, if the speech codec has been commanded not to generate a Rate 1 packet and the rate determined by the threshold tests is Rate 1, it generates a Rate 1/2 packet. Third, if the speech codec has been told to generate a Blank packet, it generates a Blank packet regardless of the rate determined by the threshold tests.

2.4.4.3 Updating Thresholds

The three thresholds are updated every frame before the rate is determined. First, an estimate of the background noise level B_i is computed for the current, or ith, frame using the background noise level estimate B_{i-1} for the previous, or (i-1)st, frame and $R(0)_{prev}$, which is the value R(0) for the previous frame, as follows:

$$B_i = \min\left(R(0)_{prev},\ 5059644,\ \max\left(1.00547 B_{i-1},\ B_{i-1} + 1\right)\right) , \qquad (2.4.4.3\text{-}1)$$

where min (x,y,z) is the minimum of x, y, and z, and max (x,y) is the maximum of x and y.

At initialization, the background noise estimate for the first frame, B_1, is set to 5059644. If the audio input to the encoder is disabled, the background noise estimate is reinitialized whenever the audio is re-enabled.[5]

---

[5]This prevents the silence before the audio is connected from being mistaken as unusually low

(footnote continues on next page)

Then, for $B_i \le 160000$, the three thresholds are computed as a function of B_i as follows:

$$T_1(B_i) = -(5.544613 \times 10^{-6})\, B_i^2 + 4.047152\, B_i + 362$$

$$T_2(B_i) = -(1.529733 \times 10^{-5})\, B_i^2 + 8.750045\, B_i + 1136$$

$$T_3(B_i) = -(3.957050 \times 10^{-5})\, B_i^2 + 18.89962\, B_i + 3347\,. \qquad (2.4.4.3\text{-}2)$$

For Bi >160000, the three thresholds are computed as follows:

$$T_1(B_i) = (9.043945 \times 10^{-8})\, B_i^2 + 3.535748\, B_i - 62071$$

$$T_2(B_i) = -(1.986007 \times 10^{-7})\, B_i^2 + 4.941658\, B_i + 223951$$

$$T_3(B_i) = -(4.838477 \times 10^{-7})\, B_i^2 + 8.63002 B_i + 645864\,. \qquad (2.4.4.3\text{-}3)$$

2.4.5 Determining the Pitch Prediction Parameters

2.4.5.1 Encoding

All speech codec frames, except for frames being encoded into Rate 1/8 packets, are subdivided into equal length pitch subframes (see Table 2.4.1-1). There are four pitch subframes for a Rate 1 packet, two pitch subframes for a Rate 1/2 packet, and one pitch subframe for a Rate 1/4 packet. There are no pitch subframes for a Rate 1/8 packet. The pitch synthesis filter can be expressed as:

$$\frac{1}{P(z)} = \frac{1}{1 - bz^{-L}}\,. \qquad (2.4.5.1\text{-}1)$$

The pitch lag, L, is represented by seven bits and ranges between 17 and 143 inclusive.[6] The pitch gain, b, is represented by three bits and ranges from 0 to 2.0 (see 2.4.5.1.3). For each pitch subframe, the speech codec determines and encodes the pitch lag, L, and the pitch gain b.

The method used to select the pitch parameters is an analysis-by-synthesis method, where encoding is done by selecting parameters which minimize the weighted error between the input speech and the synthesized speech using those parameters. The synthesized speech

background noise.

[6] L = 16 is reserved to indicate b = 0 (see 2.4.5.1.3).

is the output of the pitch synthesis filter filtered by the formant synthesis (LPC) filter. The pitch lag, L, is selected from the set {17, 18, . . ., 143} and the pitch gain, b, is selected from the set {0, 0.25, 0.5, . . . , 2.0} (linearly quantized between 0 and 2.0 in steps of 0.25). The perceptual weighting filter is of the form:

$$W(z) = \frac{A(z)}{A(z/\zeta)} \quad , \tag{2.4.5.1-2}$$

where A(z) is the formant prediction error filter and ζ, which equals 0.8, is a perceptual weighting parameter. The LPC coefficients used in the perceptual weighting filter are those for the current pitch subframe (see 2.4.3.3.4 and 2.4.3.3.5).

Reduced processing can be obtained by the filter arrangement shown in Figure 2.4.5.1-1 (see Table 2.4.5.1.1-1 for definitions of the symbols).

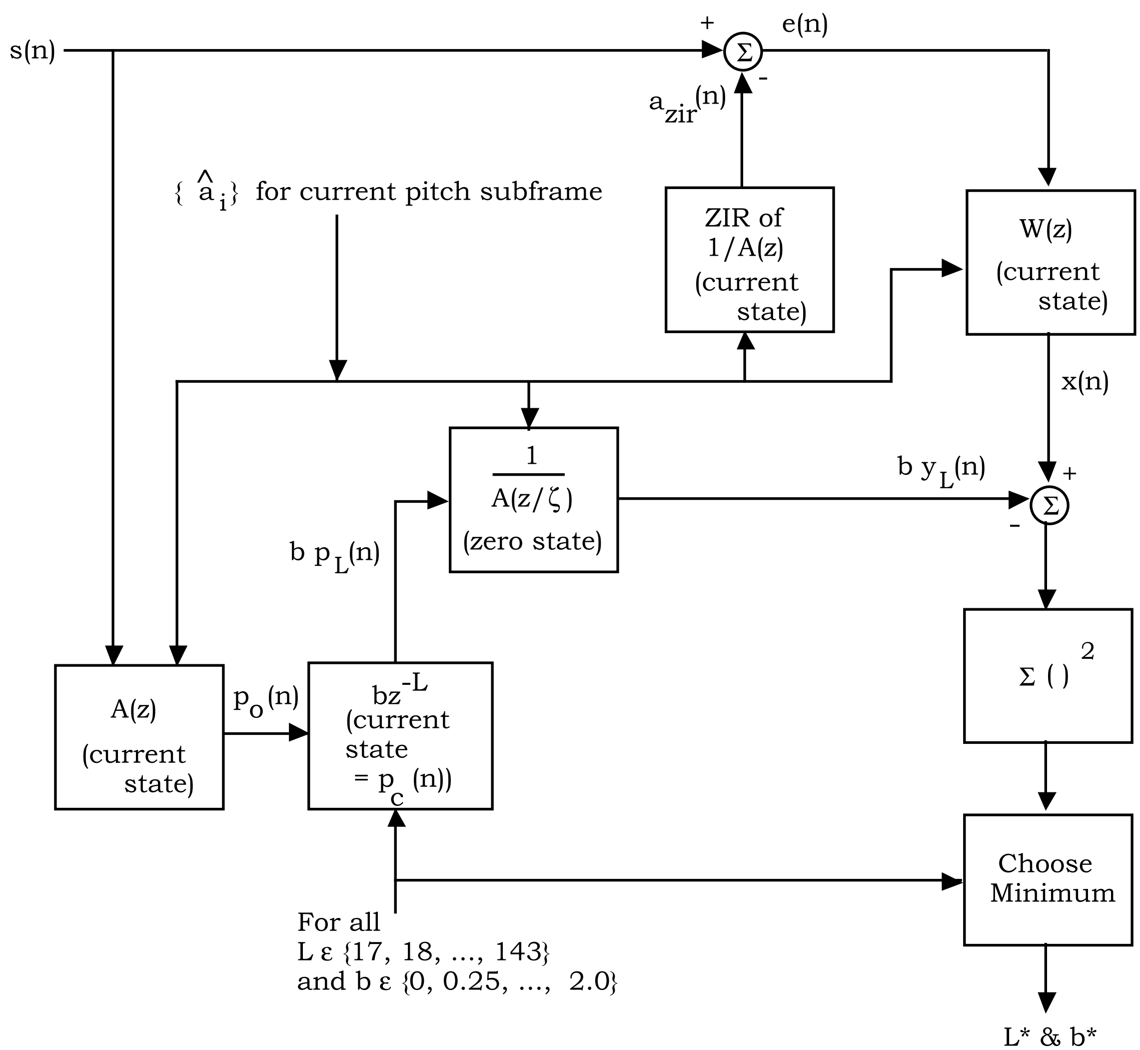


Figure 2.4.5.1-1. Analysis-by-Synthesis Procedure for the Pitch Parameter Search

In this form, the synthesis filter used in the speech encoder is called the weighted synthesis filter and is given by

$$H(z) = (1/A(z))W(z) = \frac{1}{A(z/\zeta)} \quad . \tag{2.4.5.1-3}$$

This is the formant synthesis filter followed by the perceptual weighting filter.

2.4.5.1.1 Computing the Pitch Lag and Pitch Gain

Table 2.4.5.1.1-1 lists the terms used to compute pitch lag and pitch gain.

Define

$$E_{xyL} = \sum_{n=0}^{L_P - 1} x(n) y_L(n) \tag{2.4.5.1.1-1}$$

$$E_{yyL} = \sum_{n=0}^{L_P - 1} y_L^2(n) \quad . \tag{2.4.5.1.1-2}$$

The optimal L, denoted by L*, and the optimal b, denoted by b*, are those values of L and b that result in the minimum value of:

$$\sum_{n=0}^{L_P - 1} \{x(n) - b y_L(n)\}^2 \quad . \tag{2.4.5.1.1-3}$$

This minimum is computed by searching for the minimum of

$$-2\, b\, E_{xyL} + b^2 E_{yyL} \tag{2.4.5.1.1-4}$$

over the allowable quantized values of L and b. The allowable quantized values are discussed in 2.4.5.1.

Table 2.4.5.1.1-1. Definition of Terms for Pitch Search

Term	Definition	Limits
L_P	Length, in samples, of the pitch subframe (see Table 2.4.1-1).	
$s(n)$	Input speech samples corresponding to the current pitch subframe with DC removed.	$0 \le n < L_P$
$\{\hat{a}_i\}$	LPC coefficients for the current pitch subframe.	
$a_{zir}(n)$	Zero input response, ZIR, of the formant synthesis filter, where $1/A(z)$ is initialized with the memories remaining in the decoder's $1/A(z)$ filter from the previous pitch subframe.	$0 \le n < L_P$
$e(n)$	$s(n) - a_{zir}(n)$	$0 \le n < L_P$
$x(n)$	$e(n)$ filtered by $W(z)$, where $W(z)$ is initialized with the memories remaining in the decoder's $W(z)$ filter after the last pitch subframe.	$0 \le n < L_P$
$p_c(n)$	Past outputs of the pitch synthesis filter. $p_c(-1)$ is the last output of the filter, $p_c(-2)$ is the second to last output, etc.	$-143 \le n < 0$
$p_o(n)$	An estimate of the future outputs of the pitch synthesis filter. This is $s(n)$ filtered by $A(z)$, using the appropriate LPC coefficients and memories (previous input speech samples) for the current pitch subframe. This estimate is only used in the pitch search.	$0 \le n < L_P$
$p(n)$	Combined past outputs and estimated future outputs of the pitch synthesis filter, where $$p(n) = \begin{cases} p_c(n) & -143 \le n < 0 \\ p_o(n) & 0 \le n < L_P \end{cases}$$	$-143 \le n < L_P$
$p_L(n)$	$p(n - L)$, the estimated output of the pitch synthesis filter for lag L, with b=1.	$0 \le n < L_P$
$h(n)$	Impulse response of $H(z)$ truncated to length of N_h elements (see 2.4.5.1.2).	$0 \le n < N_h$
$y_L(n)$	$p_L(n)$ convolved with $h(n)$.	$0 \le n < L_P$
L^*	Optimal pitch lag (see 2.4.5.1).	
b^*	Optimal pitch gain (see 2.4.5.1).	

2.4.5.1.2 Implementing the Pitch Search Convolutions

The zero state response of the weighted synthesis filter to $p_L(n)$, the estimated output of the pitch synthesis filter with lag L, can be calculated by convolving $p_L(n)$ with the impulse response of the weighted synthesis filter. The impulse response of the weighted synthesis filter can be truncated because it is typically small after 20 samples. Setting N_h equal to 20, the convolution is approximated by

$$y_L(n) = \sum_{i=0}^{\min(n, N_h-1)} h(i)\, p_L(n-i) \;, \qquad 16 < L \le 143, \qquad 0 \le n < L_P. \qquad (2.4.5.1.2\text{-}1)$$

Note also that

$$p_L(n) = p(n - L) = p_{L-1}(n - 1), \qquad 16 < L \le 143, \qquad 0 \le n < L_P. \qquad (2.4.5.1.2\text{-}2)$$

From Equation 2.4.5.1.2-1 and Equation 2.4.5.1.2-2,

$$y_L(n) = \begin{cases} h(0)\, p(-L) & n = 0 \qquad 17 < L \le 143 \\ y_{L-1}(n - 1) + h(n)\, p(-L) & 1 \le n < N_h \qquad 17 < L \le 143 \\ y_{L-1}(n - 1) & N_h \le n < L_P \qquad 17 < L \le 143\;. \end{cases} \qquad (2.4.5.1.2\text{-}3)$$

In this way, once the initial convolution for $y_{17}(n)$ is computed using Equation 2.4.5.1.2-1, the remaining convolutions can be done recursively by Equation 2.4.5.1.2-3.

2.4.5.1.3 Converting the Pitch Gain and Pitch Lag to the Transmission Codes

For each pitch subframe, the chosen parameters, b* and L*, are converted to transmission codes, PGAIN and PLAG. The chosen pitch gain, b*, which is a value from the set {0, 0.25, ..., 2.0}, is linearly quantized between 0 and 2.0 in steps of 0.25. The chosen lag, L*, is an integer from 17 to 143.

The value of PLAG depends upon both b* and L*. If b* = 0, then PLAG = 0. Otherwise, PLAG = L* - 16. Thus, PLAG is represented using seven bits. The value of PGAIN depends only upon b*. If b* = 0, then PGAIN = 0. Otherwise, PGAIN = b*/0.25 - 1. Thus, PGAIN is represented using three bits. Note that both b* = 0 and b* = 0.25 result in PGAIN = 0. These two cases are distinguished by the value of PLAG, which is zero in the first and non-zero in the second case.

2.4.5.2 Decoding

To convert the transmission codes to pitch gain and pitch lag, the pitch parameters are decoded by the reverse of the transformation described above; (i.e. $\hat{b} = 0$ when PLAG = 0, otherwise $\hat{b} = (\text{PGAIN} + 1)/4$ and $\hat{L} = \text{PLAG} + 16$).

2.4.6 Determining the Excitation Codebook Parameters

2.4.6.1 Encoding

Except for a Rate 1/8 packet, each pitch subframe consists of two codebook subframes (see Table 2.4.1-1). For each codebook subframe, the speech codec determines the codebook index, I, and the codebook gain, G. For a Rate 1/8 packet, only one codebook index and one codebook gain is determined for each frame and the index is discarded before transmission (see 2.4.6.1.3).

The excitation codebook consists of 2^M code vectors, where M = 7. The codebook is organized in a recursive fashion such that each code vector differs from the adjacent code vector by one sample. The samples in adjacent code vectors are shifted by one position such that a new sample is shifted in at one end and a sample is dropped at the other (see the definition of $c_I(n)$ in Table 2.4.6.1.1-1). Therefore a recursive codebook can be stored as a linear array that is $2^M + L_C - 1$ samples long. However, to simplify the implementation and to conserve memory space, a circular codebook 2^M samples long (128 samples) is used.

The circular codebook consists of the 128 values given in Table 2.4.6.1-1. The values are in signed decimal notation.

Table 2.4.6.1-1. Circular Codebook Values c(n)

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c(n)	0.0	-2.0	0.0	-1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

n	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
c(n)	0.0	-1.5	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.5

n	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
c(n)	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	1.5	1.0	0.0	1.5	2.0	0.0	0.0

n	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
c(n)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0

n	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
c(n)	-1.5	1.5	0.0	0.0	-1.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.5	0.0

n	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
c(n)	0.0	0.0	0.0	1.5	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0

n	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
c(n)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	3.0	-1.5	-2.0	0.0	-1.5	-1.5

n	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
c(n)	1.5	-1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The method used to select the codebook vector and gain is an analysis-by-synthesis method similar to that used for the pitch parameters search procedure. The chosen codebook index, I^*, and the chosen codebook gain, G^*, are the allowable values of I and G which minimize the weighted error between the synthesized speech and the input speech. The synthesized speech is the output of the codebook generator filtered by the pitch synthesis filter and the formant synthesis (LPC) filter. The perceptual weighting filter is of the form:

$$W(z) = \frac{A(z)}{A(z/\zeta)} , \qquad (2.4.6.1\text{-}1)$$

where A(z) is the formant prediction error filter and $\zeta = 0.8$ is a perceptual weighting parameter. The LPC coefficients of the perceptual weighting filter are those for the current codebook subframe (see 2.4.3.3.4 and 2.4.3.3.5).

Reduced processing can be obtained by the filter arrangement shown in Figure 2.4.6.1-1.

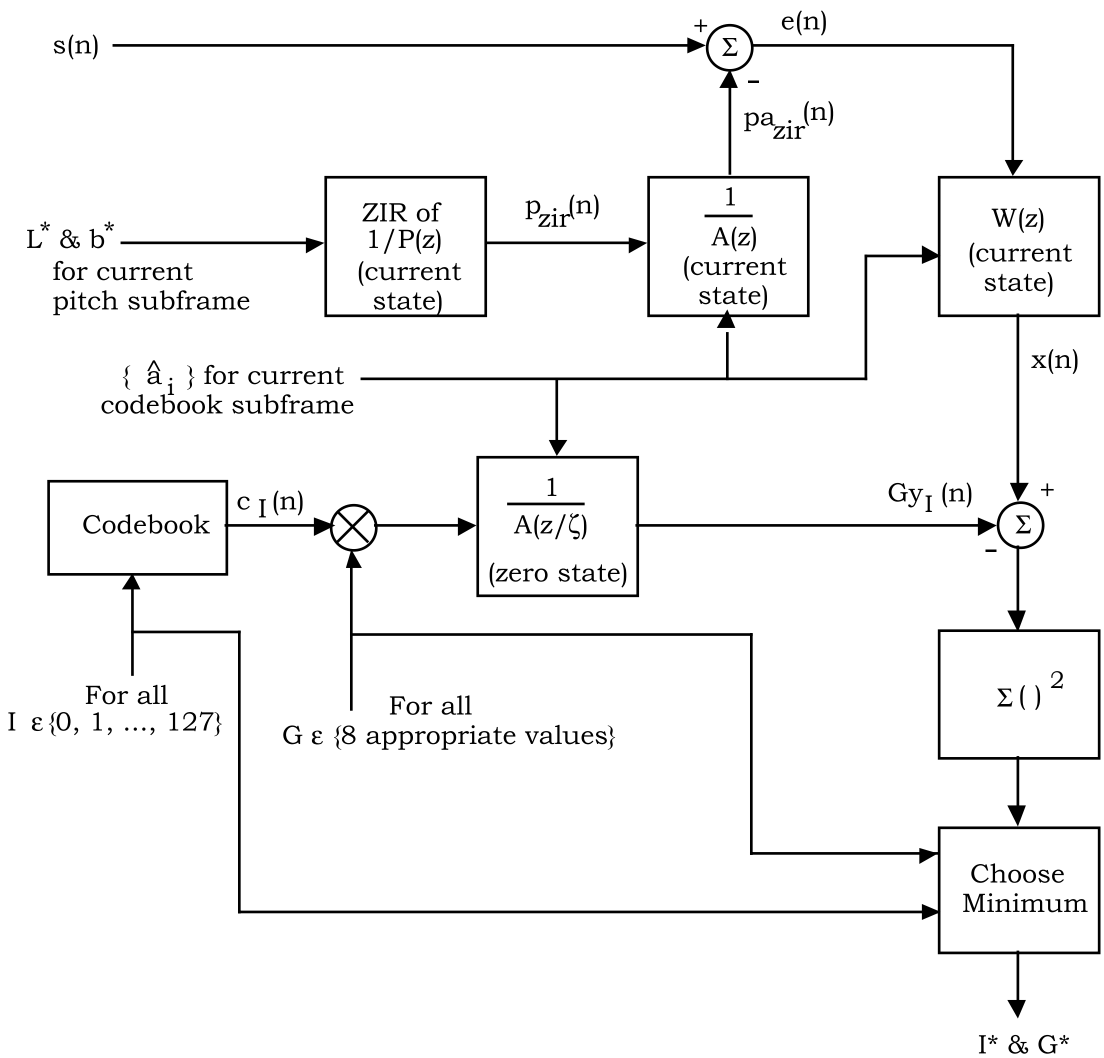


Figure 2.4.6.1-1. Analysis-by-Synthesis Procedure for Codebook Parameter Search

2.4.6.1.1 Computing the Codebook Index and Codebook Gain

The following terms are used to compute codebook index and codebook gain:

Table 2.4.6.1.1-1. Definition of Terms for Codebook Search

Term	Definition	Limits
L_C	Length, in samples, of the codebook subframe (see Table 2.4.1-1).	
s(n)	Input speech samples corresponding to the current codebook subframe with DC removed.	$0 \le n < L_C$
$\{\hat{a}_i\}$	LPC coefficients for the current codebook subframe.	
$p_{zir}(n)$	Zero input response, ZIR, of the pitch synthesis filter, with L^* and b^* for the corresponding pitch subframe and 1/P(z) initialized with the memories remaining in the decoder's 1/P(z) filter after the last codebook subframe.	$0 \le n < L_C$
$pa_{zir}(n)$	$p_{zir}(n)$, filtered by 1/A(z), where 1/A(z) is initialized with the memories remaining in the decoder's 1/A(z) filter after the last codebook subframe.	$0 \le n < L_C$
e(n)	s(n) - $pa_{zir}(n)$	$0 \le n < L_C$
x(n)	e(n) filtered by W(z), where W(z) is initialized with the memories remaining in the decoder's W(z) filter after the last codebook subframe.	$0 \le n < L_C$
c(n)	Circular codebook values.	$0 \le n < 128$
$c_I(n)$	The codebook vector for index I.	$0 \le n < L_C$
h(n)	Impulse response of H(z) truncated to N_h samples (see 2.4.5.1.2).	$0 \le n < N_h$
$y_I(n)$	$c_I(n)$ convolved with h(n). This assumes that the impulse response of 1/P(z) is either simply an impulse over the entire codebook subframe length L_C, or that the pitch gain b is small, so that the effect of the impulse response of 1/P(z) is negligible. The pitch gain is typically only large at full rate when the codebook subframe size is sufficiently small, so the above assumption holds for all cases.	$0 \le n < L_C$
I^*	Index of the optimal codebook vector (see 2.4.6.1.1).	
G^*	Optimal codebook gain (see 2.4.6.1.3).	

Now define:

$$E_{xyI} = \sum_{n=0}^{L_C - 1} x(n)\, y_I(n) \qquad (2.4.6.1.1\text{-}1)$$

$$E_{yyI} = \sum_{n=0}^{L_C - 1} y_I^{\,2}(n) \quad . \qquad (2.4.6.1.1\text{-}2)$$

The optimal I, denoted by I*, and the optimal G, denoted by G*, are those values of I and G that result in the minimum value of:

$$\sum_{n=0}^{L_C - 1} \{x(n) - G y_I(n)\}^2 \quad . \qquad (2.4.6.1.1\text{-}3)$$

This minimum is computed by searching for the minimum of

$$-2\, G\, E_{xyI} + G^2\, E_{yyI} \qquad (2.4.6.1.1\text{-}4)$$

over the allowable quantized values of I and G. I may take any value from 0 to 127. The eight allowable quantized values of G are discussed in 2.4.6.1.3.

2.4.6.1.2 Implementing the Codebook Search Convolutions

Due to the recursive nature of the codebook, the same recursive convolution procedure used in the pitch search can be used in the codebook search. The zero state response of the weighted synthesis filter to $c_I(n)$, the codebook vector for index I, can be calculated by convolving $c_I(n)$ with the impulse response of the weighted synthesis filter. The impulse response of the weighted synthesis filter can be truncated because it is typically small after 20 samples. Setting N_h equal to 20, the convolution is approximated by

$$y_I(n) = \sum_{i=0}^{\min(n,\, N_h - 1)} h(i)\, c_I(n - i) \;, \qquad 0 \le I < 128, \;\; 0 \le n < L_C. \qquad (2.4.6.1.2\text{-}1)$$

The codebook vector for index i, $c_I(n)$, is defined as[7]

$$c_I(n) = \begin{cases} c((n - I) \bmod 128) & n - I \ge 0 \\ c(128 + (n - I)) & n - I < 0 \end{cases} \qquad 0 \le I < 128, \quad 0 \le n < L_C \;. \qquad (2.4.6.1.2\text{-}2)$$

[7]For mod operations on negative operands, see note 7 in the front matter.

From Equation 2.4.6.1.2-1 and Equation 2.4.6.1.2-2,

$$y_I(n) = \begin{cases} h(0)\, c_I(0) & n = 0 \quad 1 \le I < 128 \\ y_{I-1}(n-1) + h(n)\, c_I(0) & 1 \le n < N_h \quad 1 \le I < 128 \\ y_{I-1}(n-1) & N_h \le n < L_C \quad 1 \le I < 128 \end{cases} . \qquad (2.4.6.1.2\text{-}3)$$

Once the initial convolution for $y_0(n)$ is completed using Equation 2.4.6.1.2-1, the remaining convolutions can be done recursively by Equation 2.4.6.1.2-3. Note also that when c((-I)mod 128) = 0,

$$y_I(n) = \begin{cases} 0 & n = 0 \quad 1 \le I < 128 \\ y_{I-1}(n-1) & 1 \le n < L_C \quad 1 \le I < 128 \end{cases} . \qquad (2.4.6.1.2\text{-}4)$$

2.4.6.1.3 Converting Codebook Parameters into Transmission Codes

Figure 2.4.6.1.3-1 shows the conversion scheme used for all rates except Rate 1/8.

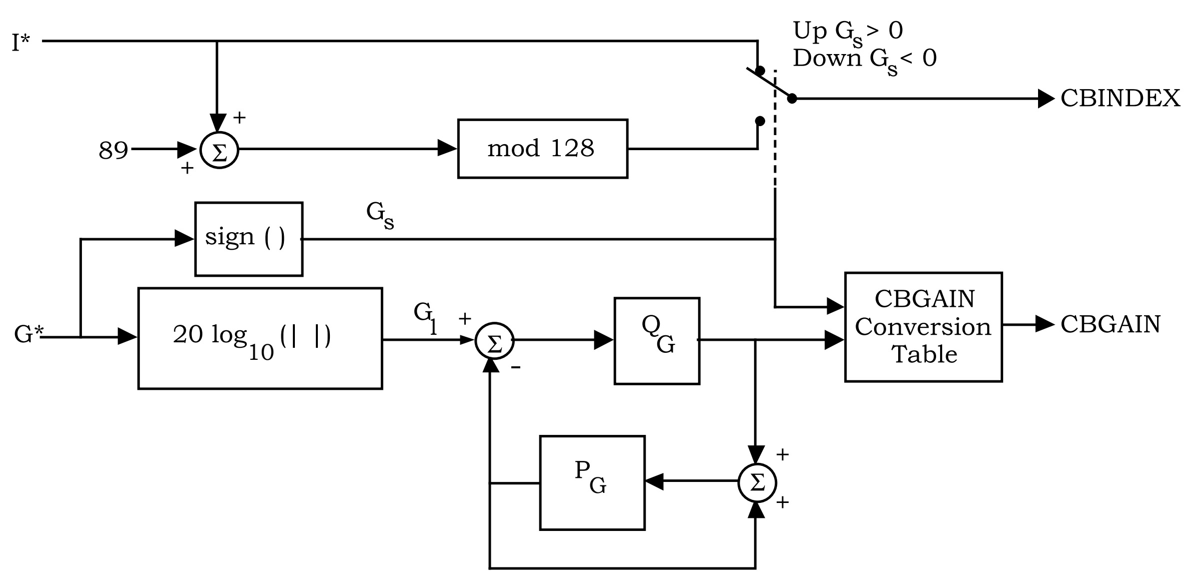


Figure 2.4.6.1.3-1. Converting Codebook Parameters for All Rates Except Rate 1/8

For all rates except Rate 1/8, one bit is used to represent the sign of the codebook gain, G_s, where G_s is defined as:

$$G_s = \text{sign}(G^*), \qquad (2.4.6.1.3\text{-}1)$$

where

$$\text{sign}(x) = \begin{cases} 1 & x \ge 0 \\ -1 & x < 0 \end{cases} . \qquad (2.4.6.1.3\text{-}2)$$

The magnitude of the codebook gain is coded using a single differential coder operating on the log of the magnitude of G, as follows:

$$G_l = 20 \log_{10}(|G^*|). \qquad (2.4.6.1.3\text{-}3)$$

The differential coder employs a 2-bit linear quantizer Q_G and a codebook gain predictor P_G. This differential coder operates on a codebook subframe basis regardless of the rate chosen for the frame. That is, the differential coder operates eight times during a Rate 1 frame, four times during a Rate 1/2 frame, two times during a Rate 1/4 frame, and only once during a Rate 1/8 frame. The output, $P_G(\mathbf{x},n)$, of the predictor P_G at time n for an input sequence $\mathbf{x} = \{\ldots,x(n-3),x(n-2),x(n-1),\ldots\}$, is defined as

$$P_G(\mathbf{x},n) = F_G\left(\left\lfloor \frac{x(n-1) + x(n-2)}{2} \right\rfloor\right) , \qquad (2.4.6.1.3\text{-}4)$$

where $\lfloor x$ is the largest integer less than or equal to x, and $F_G(x)$ is defined in Table 2.4.6.1.3-1.

Table 2.4.6.1.3-1. Codebook Gain Prediction Filter Function $F_G(x)$

x	$F_G(x)$	x	$F_G(x)$	x	$F_G(x)$	x	$F_G(x)$
-6	-2	14	13	34	30	54	48
-5	-2	15	14	35	31	55	49
-4	-2	16	15	36	32	56	50
-3	-2	17	16	37	33	57	51
-2	-1	18	17	38	34	58	52
-1	0	19	18	39	35	59	53
0	0	20	18	40	36	60	54
1	0	21	18	41	36	61	54
2	1	22	19	42	37	62	55
3	2	23	20	43	38	63	56
4	3	24	21	44	39	64	57
5	4	25	22	45	40	65	58
6	5	26	23	46	41	66	58
7	6	27	24	47	42		
8	7	28	25	48	43		
9	8	29	26	49	44		
10	9	30	27	50	45		
11	10	31	27	51	45		
12	11	32	28	52	46		
13	12	33	29	53	47		

The difference between the current G_l and the output of P_G is then linearly quantized by Q_G as shown in Table 2.4.6.1.3-2 and Table 2.4.6.1.3-3.

Table 2.4.6.1.3-2. Codebook Quantizer (Rate 1 and Rate 1/2)

Range of x	$Q_G(x)$
$x < -2$	-4
$-2 \le x < 2$	0
$2 \le x < 6$	4
$6 \le x$	8

Table 2.4.6.1.3-3. Codebook Quantizer (Rate 1/4 and Rate 1/8)

Range of x	$Q_G(x)$
$x < -3$	-4
$-3 \le x < -1$	-2
$-1 \le x < 1$	0
$1 \le x$	2

The eight allowable quantized values of G, the codebook gain, can be computed. Denote the output of P_G for the current codebook subframe by P_G. Then for Rate 1 and Rate 1/2, the allowable quantized values of G_l (see Equation 2.4.6.1.3-3) are $\{P_G - 4, P_G, P_G + 4, P_G + 8\}$. For Rate 1/4 and Rate 1/8, the allowable values are $\{P_G - 4, P_G - 2, P_G, P_G + 2\}$. Table 2.4.6.2.1-1 converts the allowable quantized values of G_l to the four allowable magnitudes of G. Finally, the eight allowable values of G may be either positive or negative.

The output of the quantizer, $Q_G(x)$, and the sign, G_s, is converted to CBGAIN as shown in Tables 2.4.6.1.3-4 and 2.4.6.1.3-5.

Table 2.4.6.1.3-4. Conversion Table for CBGAIN (Rate 1 and Rate 1/2)

G_S	$Q_G(x)$	CBGAIN
+1	-4	0
+1	0	1
+1	4	2
+1	8	3
-1	-4	4
-1	0	5
-1	4	6
-1	8	7

Table 2.4.6.1.3-5. Conversion Table for CBGAIN (Rate 1/4 and Rate 1/8)

G_S	$Q_G(x)$	CBGAIN
+1	-4	0
+1	-2	1
+1	0	2
+1	2	3
-1	-4	4
-1	-2	5
-1	0	6
-1	2	7

If G_S is negative, CBINDEX is set equal to (I+89) mod 128. If G_S is positive, CBINDEX is set equal to I. This is done to reduce the sensitivity of the reconstructed speech signal to errors in the codebook gain sign bit.

The conversion scheme shown in Figure 2.4.6.1.3-2 is used only for Rate 1/8.

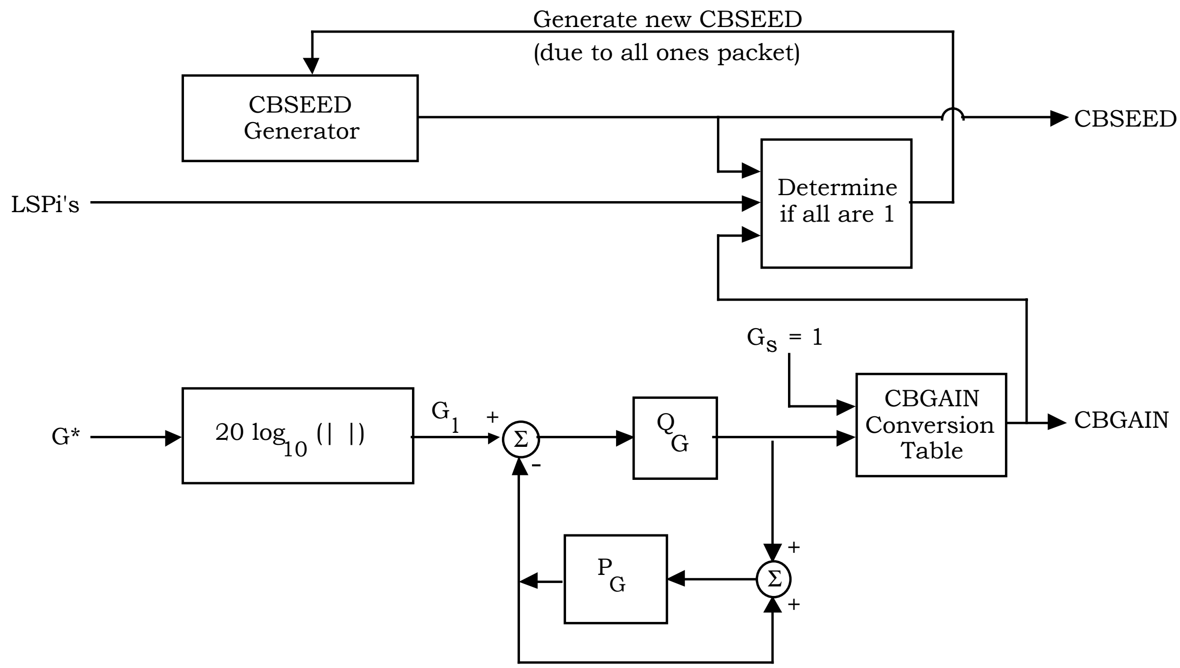


Figure 2.4.6.1.3-2. Converting Codebook Parameters for Rate 1/8

For Rate 1/8 frames, the excitation codebook vectors are replaced by a pseudorandom code vector in the decoding sections of the transmitting encoder and the receiving decoder. The codebook index and the sign of the codebook gain are not transmitted. The magnitude of the codebook gain is quantized for transmission in exactly the same way as described above, with the exception that G_S is always +1, resulting in a CBGAIN value between 0 and 3.

The pseudorandom code vector is generated by a pseudorandom number generator that is the same in the decoding sections of the transmitting encoder and the receiving decoder. This is accomplished by using the transmitted 16-bit data packet at Rate 1/8 as the seed for the pseudorandom number generator at both ends of transmission (see 2.4.8.1.2).

CBSEED, which consists of four bits, is used to ensure the randomness of the transmitted packet. These bits are generated by a pseudorandom number generator which generates relatively independent, uniformly distributed, pseudorandom numbers. A pseudorandom number generator using the integer SD which has been found to have satisfactory properties is

$$SD(new) = (521\ (SD(old)) + 259) \bmod 2^{16}\ . \qquad (2.4.6.1.3\text{-}5)$$

At transmitting encoder initialization, SD is set to 0.

For each new transmitted Rate 1/8 packet, a new SD is computed and the four bits of CBSEED are given by[8]

$$\text{CBSEED}[k] = \text{SD(new)}\,[\,4k + 3]\quad k = 0, 1, 2, 3\quad , \qquad (2.4.6.1.3\text{-}6)$$

where CBSEED[k] denotes bit k of CBSEED and SD(new) [4k + 3] denotes bit 4k + 3 of the binary representation of SD(new). SD(new) is then saved for use as SD(old) in the next Rate 1/8 packet.

As an example, if the previous value of SD = 40481 then

$$\begin{aligned} \text{SD(new)} &= (521(40481) + 259) \bmod 2^{16} \qquad (2.4.6.1.3\text{-}7) \\ &= 53804. \end{aligned}$$

In this case, CBSEED = ‘1001’, and SD = 53804 is saved for the next Rate 1/8 frame.

A 1200 bps frame consisting of all ones is null Traffic Channel data. The speech codec does not supply a Rate 1/8 packet with all ones bits to the multiplex sublayer. If an all ones Rate 1/8 packet occurs after packing (see 2.4.7.4), a new CBSEED is generated using the method above. The process is repeated until a CBSEED which is not all ones is generated. The packet is then repacked with the new CBSEED.

2.4.6.2 Decoding

2.4.6.2.1 Converting Codebook Transmission Codes for All Rates Except 1/8

Decoding of the codebook parameters is done by the reverse of the transformation described above. This is shown in Figure 2.4.6.2.1-1.

[8]See preface note 7 for an explanation of the bracket operator, [ ].

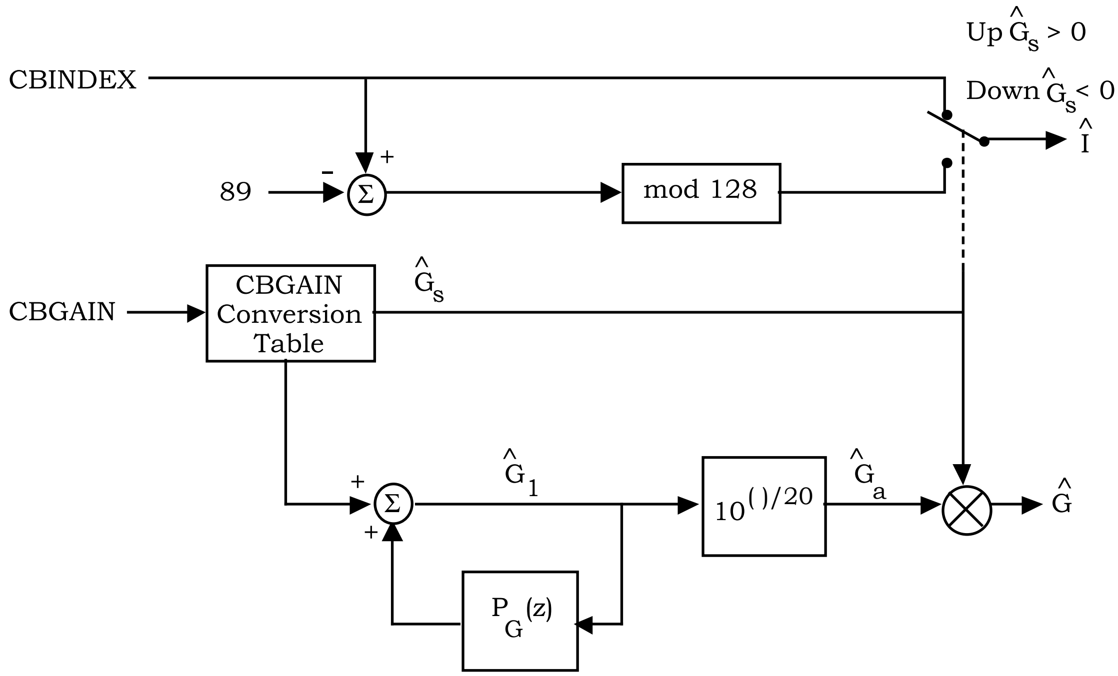


Figure 2.4.6.2.1-1. Converting Codebook Transmission Codes for All Rates Except 1/8

The sign of the codebook gain $\hat{G}_s$ is set to +1 if CBGAIN is less than 4 and -1 if CBGAIN is greater than or equal to 4. For Rate 1 and Rate 1/2, the least significant two bits of CBGAIN are converted back into -4, 0, 4, or 8 as shown in Table 2.4.6.1.3-4. For Rate 1/4, the least significant two bits of CBGAIN are converted back into -4, -2, 0, or 2 as shown in Table 2.4.6.1.3-5. This value is then added to the output of the predictor P_G to obtain the decoded value of $\hat{G}_1$.

The decoded $\hat{G}_1$ is then converted back into the linear domain via Table 2.4.6.2.1-1. The values in this table correspond to the linear values of $\hat{G}_a$ with three fractional bits. Finally, $\hat{G}$ is found by multiplying $\hat{G}_a$ by $\hat{G}_s$.

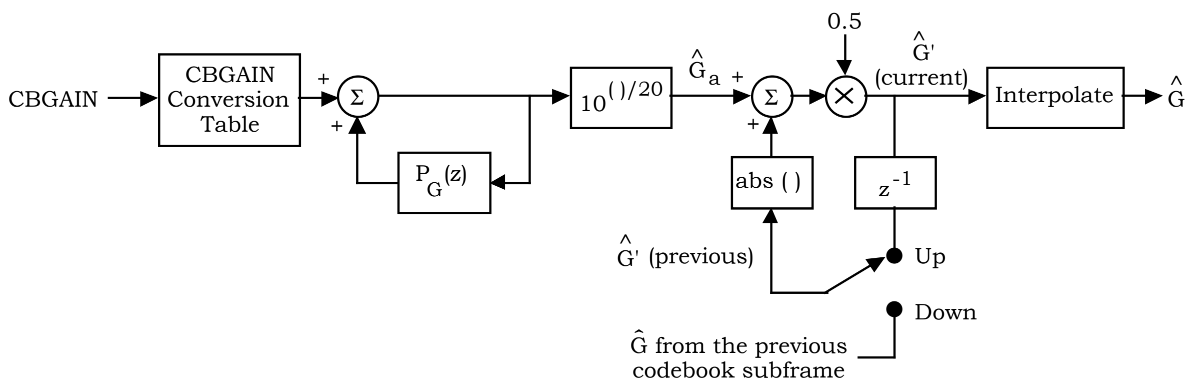
If the received sign of the codebook gain $\hat{G}_s$ equals -1, the codebook index $\hat{I}$ is set to (CBINDEX - 89) mod 128. If $\hat{G}_s$ equals +1, $\hat{I}$ is set to CBINDEX.

Table 2.4.6.2.1-1. Conversion Table for $\hat{G}_l$ to $\hat{G}_a$

$\hat{G}_l$	$\hat{G}_a$	$\hat{G}_l$	$\hat{G}_a$	$\hat{G}_l$	$\hat{G}_a$	$\hat{G}_l$	$\hat{G}_a$
-6	0.500	14	5.000	34	50.125	54	501.125
-5	0.500	15	5.625	35	56.250	55	562.375
-4	0.625	16	6.250	36	63.125	56	631.000
-3	0.750	17	7.125	37	70.750	57	708.000
-2	0.750	18	8.000	38	79.375	58	794.375
-1	0.875	19	8.875	39	89.125	59	891.250
0	1.000	20	10.000	40	100.000	60	1000.000
1	1.125	21	11.250	41	112.250	61	1122.000
2	1.250	22	12.625	42	125.875	62	1258.875
3	1.375	23	14.125	43	141.250	63	1412.500
4	1.625	24	15.875	44	158.500	64	1584.875
5	1.750	25	17.750	45	177.875	65	1778.250
6	2.000	26	20.000	46	199.500	66	1995.250
7	2.250	27	22.375	47	223.875		
8	2.500	28	25.125	48	251.250		
9	2.875	29	28.125	49	281.875		
10	3.125	30	31.625	50	316.250		
11	3.500	31	35.500	51	354.875		
12	4.000	32	39.750	52	398.125		
13	4.500	33	44.625	53	446.625		

2.4.6.2.2 Converting Codebook Transmission Codes for Rate 1/8

The procedure for determining the gain is shown in Figure 2.4.6.2.2-1. The least significant two bits of CBGAIN are converted back into -4, -2, 0, or 2 as shown in Table 2.4.6.1.3-5. The sign of the codebook gain, $\hat{G}_s$, is set to 1. The codebook index is not used in decoding Rate 1/8 packets (see 2.4.8.1.2).



Switch Up = Previous codebook subframe was from a Rate 1/8 packet
Switch Down = Previous codebook subframe was from other than a Rate 1/8 packet

Figure 2.4.6.2.2-1. Converting Codebook Transmission Codes for Rate 1/8

To prevent burstiness in the sound of the background noise, the current value of $\hat{G}_a$ is low-pass filtered as follows:

$$\hat{G}'(\text{current}) = 0.5\,|\hat{G}'(\text{previous})| + 0.5\,\hat{G}_a(\text{current}), \qquad (2.4.6.2.2\text{-}1)$$

where $\hat{G}_a$ (current) is the decoded linear codebook gain for the current codebook subframe, $\hat{G}'$ (previous) is the filtered linear codebook gain for the previous codebook subframe, and $|x|$ is the absolute value of x. If the previous frame were at other than Rate 1/8, then $\hat{G}'$ (previous) is the codebook gain from the previous codebook subframe (e.g., $\hat{G}$ for the codebook subframe). Since $\hat{G}_a$(current) is guaranteed to be positive, the absolute value of $\hat{G}_a$ (current) does not need to be taken.

The value of $\hat{G}'$ is then interpolated for each sample, indexed by n below, to produce a smoother-sounding background noise:

$$\hat{G} = \begin{cases} 0.875\ |\hat{G}'(\text{previous})| + 0.125\ \hat{G}'(\text{current}) & 0 \le n < 20 \\ 0.750\ |\hat{G}'(\text{previous})| + 0.250\ \hat{G}'(\text{current}) & 20 \le n < 40 \\ 0.625\ |\hat{G}'(\text{previous})| + 0.375\ \hat{G}'(\text{current}) & 40 \le n < 60 \\ 0.500\ |\hat{G}'(\text{previous})| + 0.500\ \hat{G}'(\text{current}) & 60 \le n < 80 \\ 0.375\ |\hat{G}'(\text{previous})| + 0.625\ \hat{G}'(\text{current}) & 80 \le n < 100 \\ 0.250\ |\hat{G}'(\text{previous})| + 0.750\ \hat{G}'(\text{current}) & 100 \le n < 120 \\ 0.125\ |\hat{G}'(\text{previous})| + 0.875\ \hat{G}'(\text{current}) & 120 \le n < 140 \\ \hat{G}'(\text{current}) & 140 \le n < 160 \end{cases} . \quad (2.4.6.2.2\text{-}2)$$

2.4.7 Data Packing

2.4.7.1 Rate 1 Parity Check Bits and Packing

2.4.7.1.1 Parity Check Bits

Eleven parity check bits are added to provide error correction and detection of the 18 most perceptually significant bits of the Rate 1 data. Ten parity check bits are generated from 18 information bits to form a BCH(28, 18) code word.[9] Then a single parity check bit is computed using the 28 bits of this code word. This forms the final BCH(29, 18) code word.

The 18 most perceptually significant bits are assembled into an information polynomial a(x) over GF(2) as follows:[10]

$$\begin{aligned} a(x) = {} & \text{LSP1[3]}\ x^{17} + \text{LSP2[3]}\ x^{16} + \text{LSP3[3]}\ x^{15} + \text{LSP4[3]}\ x^{14} \\ & + \text{LSP5[3]}\ x^{13} + \text{LSP6[3]}\ x^{12} + \text{LSP7[3]}\ x^{11} + \text{LSP8[3]}\ x^{10} \\ & + \text{LSP9[3]}\ x^{9} + \text{LSP10[3]}\ x^{8} + \text{CBGAIN1[1]}\ x^{7} + \text{CBGAIN2[1]}\ x^{6} \\ & + \text{CBGAIN3[1]}\ x^{5} + \text{CBGAIN4[1]}\ x^{4} + \text{CBGAIN5[1]}\ x^{3} \\ & + \text{CBGAIN6[1]}\ x^{2} + \text{CBGAIN7[1]}\ x^{1} + \text{CBGAIN8[1]}\ x^{0} \quad , \end{aligned} \qquad (2.4.7.1.1\text{-}1)$$

where LSPi[3] is the MSB of LSP code i, and CBGAINi[1] is the second MSB of CBGAIN code i. In effect, a(x) is made up of the MSBs of all ten LSP codes, and the second MSBs of the CBGAIN codes.

By using a(x) as the input, the first ten parity check bits are generated from the remainder polynomial r(x) of the following long division:

---

[9]The terminology BCH(n, k) implies that the BCH code word is n bits long and has k information bits. The BCH(28, 18) code is a shortened BCH(31, 21) code.

[10]GF(2) is the Galois Field of two elements. The multiplications and divisions are just ordinary multiplies and divides of one polynomial with another, except that the coefficients are restricted to be binary and the arithmetic is performed mod 2. There are no carries or borrows. See Lin, S. and Costello, D. J., *Error Control Coding: Fundamentals and Applications*, (New Jersey: Prentice-Hall Inc., 1983), pp. 19-29.

$$\frac{a(x)\, x^{10}}{g_{pc}(x)} = q(x) + \frac{r(x)}{g_{pc}(x)} \quad , \qquad (2.4.7.1.1\text{-}2)$$

where $g_{pc}(x)$ is the generator polynomial of the BCH (28,18) code,

$$g_{pc}(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1, \qquad (2.4.7.1.1.3)$$

and q(x) is the quotient polynomial which is not used in generating parity check bits. The parity check bits PCB[1] through PCB[10] are obtained as coefficients of r(x) as follows:[11]

$$\begin{aligned} r(x) = {} & \overline{PCB[10]}\ x^9 + \overline{PCB[9]}\ x^8 + \overline{PCB[8]}\ x^7 + \overline{PCB[7]}\ x^6 \\ & + \overline{PCB[6]}\ x^5 + \overline{PCB[5]}\ x^4 + \overline{PCB[4]}\ x^3 \\ & + \overline{PCB[3]}\ x^2 + \overline{PCB[2]}\ x^1 + \overline{PCB[1]}\ x^0. \end{aligned} \qquad (2.4.7.1.1\text{-}4)$$

The parity check bit, PCB[0], is a parity bit on the 18 protected bits in a(x) and the ten parity check bits in r(x). PCB[0] is ‘0’ if the exclusive-OR of all 28 bits results in ‘0’; PCB[0] is ‘1’ if the exclusive-OR of all 28 bits results in ‘1’. That is,

$$\begin{aligned} PCB[0] = {} & LSP1[3] \oplus LSP2[3] \oplus LSP3[3] \oplus LSP4[3] \oplus LSP5[3] \oplus LSP6[3] \oplus LSP7[3] \\ & \oplus LSP8[3] \oplus LSP9[3] \oplus LSP10[3] \oplus CBGAIN1[1] \oplus CBGAIN2[1] \oplus CBGAIN3[1] \\ & \oplus CBGAIN4[1] \oplus CBGAIN5[1] \oplus CBGAIN6[1] \oplus CBGAIN7[1] \oplus CBGAIN8[1] \\ & \oplus PCB[10] \oplus PCB[9] \oplus PCB[8] \oplus PCB[7] \oplus PCB[6] \oplus PCB[5] \oplus PCB[4] \\ & \oplus PCB[3] \oplus PCB[2] \oplus PCB[1]. \end{aligned} \qquad (2.4.7.1.1\text{-}5)$$

[11]Note that PCB[1] through PCB[10] are inverted before transmission.

2.4.7.1.2 Rate 1 Packing

The 171 Rate 1 bits shall be packed into a primary traffic packet as shown in Table 2.4.7.1.2-1. Bit 170 shall be the first primary traffic bit in the frame and bit 0 shall be the last primary traffic bit in the frame.

Table 2.4.7.1.2-1. Rate 1 Packet Structure (Part 1 of 2)

Bit	Code
170	LSP1[2]
169	LSP1[3]
168	LSP2[2]
167	LSP2[3]
166	LSP3[2]
165	LSP3[3]
164	LSP4[2]
163	LSP4[3]
162	LSP5[2]
161	LSP5[3]
160	LSP6[2]
159	LSP6[3]
158	LSP7[2]
157	LSP7[3]
156	LSP8[2]
155	LSP8[3]
154	LSP9[2]
153	LSP9[3]
152	LSP10[2]
151	LSP10[3]
150	LSP1[1]
149	LSP1[0]
148	LSP2[1]
147	LSP2[0]
146	LSP3[1]
145	LSP3[0]
144	LSP4[1]
143	CBGAIN1[1]
142	LSP4[0]
141	LSP5[1]
140	LSP5[0]
139	LSP6[1]
138	LSP6[0]
137	LSP7[1]
136	LSP7[0]
135	CBGAIN2[1]
134	LSP8[1]
133	LSP8[0]
132	LSP9[1]
131	LSP9[0]
130	LSP10[1]
129	LSP10[0]
128	PGAIN1[2]
127	CBGAIN3[1]
126	PGAIN1[1]
125	PGAIN1[0]
124	PLAG1[6]
123	PLAG1[5]
122	PLAG1[4]
121	PLAG1[3]
120	PLAG1[2]
119	CBGAIN4[1]
118	PLAG1[1]
117	PLAG1[0]
116	CBINDEX1[6]
115	CBINDEX1[5]
114	CBINDEX1[4]
113	CBINDEX1[3]
112	CBINDEX1[2]
111	CBGAIN5[1]
110	CBINDEX1[1]
109	CBINDEX1[0]
108	CBGAIN1[2]
107	CBGAIN1[0]
106	CBINDEX2[6]
105	CBINDEX2[5]
104	CBINDEX2[4]
103	CBGAIN6[1]
102	CBINDEX2[3]
101	CBINDEX2[2]
100	CBINDEX2[1]
99	CBINDEX2[0]
98	CBGAIN2[2]
97	CBGAIN2[0]
96	PGAIN2[2]
95	CBGAIN7[1]
94	PGAIN2[1]
93	PGAIN2[0]
92	PLAG2[6]
91	PLAG2[5]
90	PLAG2[4]
89	PLAG2[3]
88	PLAG2[2]
87	CBGAIN8[1]
86	PLAG2[1]
85	PLAG2[0]
84	CBINDEX3[6]
83	CBINDEX3[5]
82	CBINDEX3[4]
81	CBINDEX3[3]
80	CBINDEX3[2]
79	PCB[10]
78	CBINDEX3[1]
77	CBINDEX3[0]
76	CBGAIN3[2]
75	CBGAIN3[0]

Table 2.4.7.1.2-1. Rate 1 Packet Structure (Part 2 of 2)

Bit	Code	Bit	Code	Bit	Code	Bit	Code
74	CBINDEX4[6]	55	PCB[7]	36	CBINDEX6[1]	17	CBINDEX7[3]
73	CBINDEX4[5]	54	PLAG3[1]	35	CBINDEX6[0]	16	CBINDEX7[2]
72	CBINDEX4[4]	53	PLAG3[0]	34	CBGAIN6[2]	15	PCB[2]
71	PCB[9]	52	CBINDEX5[6]	33	CBGAIN6[0]	14	CBINDEX7[1]
70	CBINDEX4[3]	51	CBINDEX5[5]	32	PGAIN4[2]	13	CBINDEX7[0]
69	CBINDEX4[2]	50	CBINDEX5[4]	31	PCB[4]	12	CBGAIN7[2]
68	CBINDEX4[1]	49	CBINDEX5[3]	30	PGAIN4[1]	11	CBGAIN7[0]
67	CBINDEX4[0]	48	CBINDEX5[2]	29	PGAIN4[0]	10	CBINDEX8[6]
66	CBGAIN4[2]	47	PCB[6]	28	PLAG4[6]	9	CBINDEX8[5]
65	CBGAIN4[0]	46	CBINDEX5[1]	27	PLAG4[5]	8	CBINDEX8[4]
64	PGAIN3[2]	45	CBINDEX5[0]	26	PLAG4[4]	7	PCB[1]
63	PCB[8]	44	CBGAIN5[2]	25	PLAG4[3]	6	CBINDEX8[3]
62	PGAIN3[1]	43	CBGAIN5[0]	24	PLAG4[2]	5	CBINDEX8[2]
61	PGAIN3[0]	42	CBINDEX6[6]	23	PCB[3]	4	CBINDEX8[1]
60	PLAG3[6]	41	CBINDEX6[5]	22	PLAG4[1]	3	CBINDEX8[0]
59	PLAG3[5]	40	CBINDEX6[4]	21	PLAG4[0]	2	CBGAIN8[2]
58	PLAG3[4]	39	PCB[5]	20	CBINDEX7[6]	1	CBGAIN8[0]
57	PLAG3[3]	38	CBINDEX6[3]	19	CBINDEX7[5]	0	PCB[0]
56	PLAG3[2]	37	CBINDEX6[2]	18	CBINDEX7[4]		

2.4.7.2 Rate 1/2 Packing

The 80 Rate 1/2 bits shall be packed into a primary traffic packet as shown in Table 2.4.7.2-1. Bit 79 shall be the first primary traffic bit in the frame and bit 0 shall be the last primary traffic bit in the frame.

Table 2.4.7.2-1. Rate 1/2 Packet Structure

Bit	Code	Bit	Code	Bit	Code	Bit	Code
79	LSP1[1]	59	PGAIN1[2]	39	CBINDEX2[6]	19	CBINDEX3[6]
78	LSP1[0]	58	PGAIN1[1]	38	CBINDEX2[5]	18	CBINDEX3[5]
77	LSP2[1]	57	PGAIN1[0]	37	CBINDEX2[4]	17	CBINDEX3[4]
76	LSP2[0]	56	PLAG1[6]	36	CBINDEX2[3]	16	CBINDEX3[3]
75	LSP3[1]	55	PLAG1[5]	35	CBINDEX2[2]	15	CBINDEX3[2]
74	LSP3[0]	54	PLAG1[4]	34	CBINDEX2[1]	14	CBINDEX3[1]
73	LSP4[1]	53	PLAG1[3]	33	CBINDEX2[0]	13	CBINDEX3[0]
72	LSP4[0]	52	PLAG1[2]	32	CBGAIN2[2]	12	CBGAIN3[2]
71	LSP5[1]	51	PLAG1[1]	31	CBGAIN2[1]	11	CBGAIN3[1]
70	LSP5[0]	50	PLAG1[0]	30	CBGAIN2[0]	10	CBGAIN3[0]
69	LSP6[1]	49	CBINDEX1[6]	29	PGAIN2[2]	9	CBINDEX4[6]
68	LSP6[0]	48	CBINDEX1[5]	28	PGAIN2[1]	8	CBINDEX4[5]
67	LSP7[1]	47	CBINDEX1[4]	27	PGAIN2[0]	7	CBINDEX4[4]
66	LSP7[0]	46	CBINDEX1[3]	26	PLAG2[6]	6	CBINDEX4[3]
65	LSP8[1]	45	CBINDEX1[2]	25	PLAG2[5]	5	CBINDEX4[2]
64	LSP8[0]	44	CBINDEX1[1]	24	PLAG2[4]	4	CBINDEX4[1]
63	LSP9[1]	43	CBINDEX1[0]	23	PLAG2[3]	3	CBINDEX4[0]
62	LSP9[0]	42	CBGAIN1[2]	22	PLAG2[2]	2	CBGAIN4[2]
61	LSP10[1]	41	CBGAIN1[1]	21	PLAG2[1]	1	CBGAIN4[1]
60	LSP10[0]	40	CBGAIN1[0]	20	PLAG2[0]	0	CBGAIN4[0]

2.4.7.3 Rate 1/4 Packing

The 40 Rate 1/4 bits shall be packed into a primary traffic packet as shown in Table 2.4.7.3-1. Bit 39 shall be the first primary traffic bit in the frame and bit 0 shall be the last primary traffic bit in the frame.

Table 2.4.7.3-1. Rate 1/4 Packet Structure

Bit	Code	Bit	Code	Bit	Code	Bit	Code
39	LSP1[0]	29	PGAIN1[2]	19	CBINDEX1[6]	9	CBINDEX2[6]
38	LSP2[0]	28	PGAIN1[1]	18	CBINDEX1[5]	8	CBINDEX2[5]
37	LSP3[0]	27	PGAIN1[0]	17	CBINDEX1[4]	7	CBINDEX2[4]
36	LSP4[0]	26	PLAG1[6]	16	CBINDEX1[3]	6	CBINDEX2[3]
35	LSP5[0]	25	PLAG1[5]	15	CBINDEX1[2]	5	CBINDEX2[2]
34	LSP6[0]	24	PLAG1[4]	14	CBINDEX1[1]	4	CBINDEX2[1]
33	LSP7[0]	23	PLAG1[3]	13	CBINDEX1[0]	3	CBINDEX2[0]
32	LSP8[0]	22	PLAG1[2]	12	CBGAIN1[2]	2	CBGAIN2[2]
31	LSP9[0]	21	PLAG1[1]	11	CBGAIN1[1]	1	CBGAIN2[1]
30	LSP10[0]	20	PLAG1[0]	10	CBGAIN1[0]	0	CBGAIN2[0]

2.4.7.4 Rate 1/8 Packing

The 16 Rate 1/8 bits shall be packed into a primary traffic packet as shown in Table 2.4.7.4-1. Bit 15 shall be the first primary traffic bit in the frame and bit 0 shall be the last primary traffic bit in the frame.

Table 2.4.7.4-1. Rate 1/8 Packet Structure

Bit	Code	Bit	Code	Bit	Code	Bit	Code
15	CBSEED[3]	11	CBSEED[2]	7	CBSEED[1]	3	CBSEED[0]
14	LSP1[0]	10	LSP4[0]	6	LSP7[0]	2	LSP10[0]
13	LSP2[0]	9	LSP5[0]	5	LSP8[0]	1	CBGAIN1[1]
12	LSP3[0]	8	LSP6[0]	4	LSP9[0]	0	CBGAIN1[0]

2.4.8 Decoding at the Transmitting Speech Codec and the Receiving Speech Codec

At the encoder or transmit side, after each codebook subframe a version of the decoder (shown in Figure 2.4.8-1) is run to update the filter memories. At the decoder or receive side, the decoder (shown in Figure 2.4.8-2) decodes the received parameters to produce $s_d(n)$, the reconstructed speech. The two decoders are quite similar.

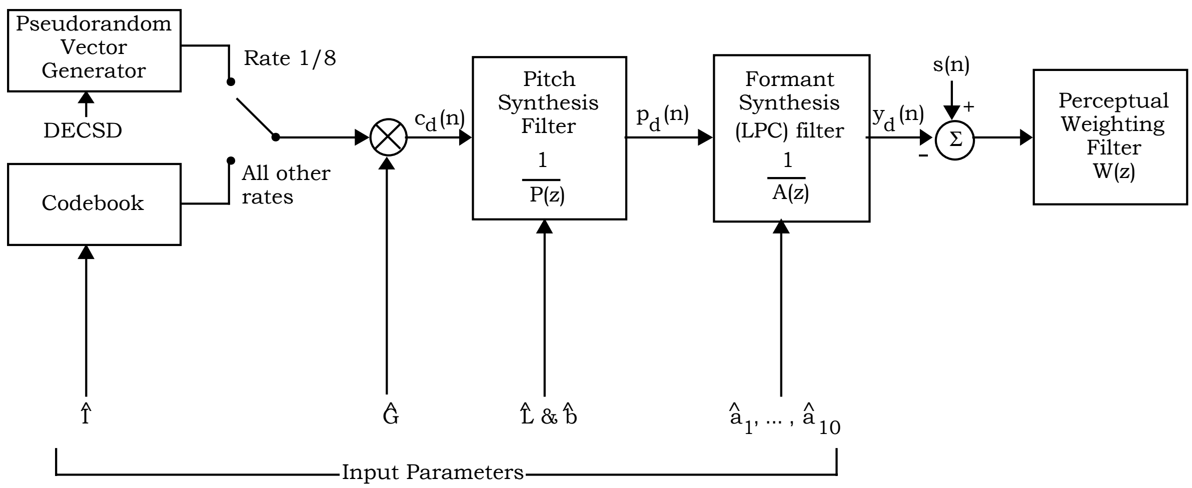


Figure 2.4.8-1. Decoding at the Transmitting Speech Codec

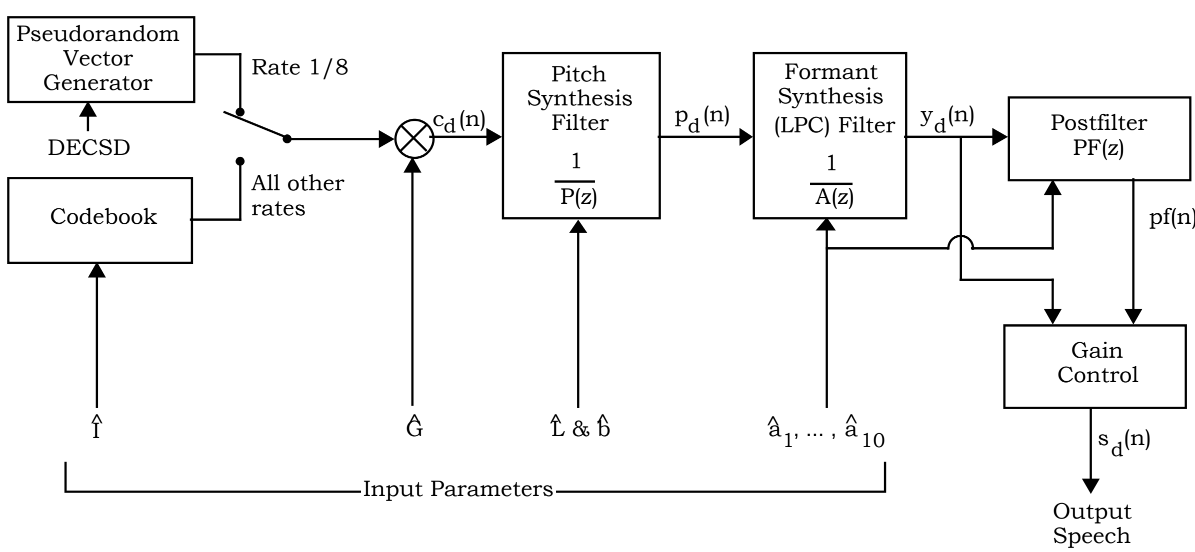


Figure 2.4.8-2. Decoding at the Receiving Speech Codec

2.4.8.1 Generating the Scaled Codebook Vector

Both the transmitting speech codec and the receiving speech codec generates the scaled codebook vector, $c_d(n)$. $c_d(n)$ is generated differently for a Rate 1/8 packet than for all other rate packets. $c_d(n)$ is generated in integer precision. All fractional bits are truncated. Fractional precision is used in computing $c_d(n)$; only the final result is truncated to integer precision.

2.4.8.1.1 Generating the Scaled Codebook Vector for All Rates Except Rate 1/8

First, $\hat{I}$ and $\hat{G}$ is decoded from CBGAIN and CBINDEX as previously described. $c_d(n)$ is then set to $\hat{G}\ c((n-\hat{I})\ \mathrm{mod}\ 128)$, where c(n) is the codebook in Table 2.4.6.1-1.

2.4.8.1.2 Generating the Scaled Codebook Vector for Rate 1/8

For Rate 1/8 frames, $c_d(n)$ is set to a pseudorandom white sequence. Both the transmitting speech codec and the receiving speech codec must produce exactly the same sequence. This requires that the pseudorandom number generators at both sides start with the exact same seed, hereafter referred to as DECSD.[12] DECSD is set to the 16-bit Rate 1/8 packet considered as an integer. rnd(n), a length 160 random sequence having the same energy as the codebook, is then generated using the following procedure:

```
{
  i=0
  decrv (old) = DECSD
  while (i < 160)
     {
       decrv (new) = (521 decrv (old) + 259) mod 2^16
       rnd = (decrv (new) + 2^15) mod 2^16 - 2^15
       rnd (i) = sqrt(1.887) rnd /32768.0
       decrv (old) = decrv (new)
       i = i + 1
     }
  }
```

The temporary variable decrv is an integer; the temporary variable rnd is an integer which is normalized to produce rnd(n), for each index n. In the procedure above, rnd(n) is considered as a real number whose precision in a fixed point representation is described below. Intermediate integer calculations should be kept in full precision.

The resulting code vector, $c_d(n)$, is determined as follows:

$$c_d(n) = \hat{G}\ \mathrm{rnd}(n)\ , \qquad (2.4.8.1.2\text{-}1)$$

---

[12]In an implementation which contains both the encoder and decoder operating in parallel, two distinct versions of DECSD must be kept so as not to confuse the pseudorandom number sequence generated at the encoder with the sequence generated at the decoder.

where $\hat{G}$ is the interpolated gain value for the appropriate subframe (see 2.4.6.2.2). Although $c_d(n)$ is computed without fractional bits, rnd(n) is computed with fractional precision, since the magnitude of rnd(n) is less than $\sqrt{1.887}$.[13] rnd(n) should be kept with at least 12 fractional bits, multiplied by the corresponding interpolated $\hat{G}$ value described above with three fractional bits, and then truncated to integer format.

2.4.8.2 Generating the Pitch Synthesis Filter Output

Both the transmitting speech codec and the receiving speech codec generate the output of the pitch synthesis filter, $p_d(n)$, identically. The filter 1/P(z) is initialized with the final state resulting from the last output sample generated, but using $\hat{b}$ and $\hat{L}$ appropriate for the current pitch subframe. $c_d(n)$ is filtered by 1/P(z), to produce $p_d(n)$. For Rate 1/8 frames, $\hat{b}$ is set to 0. The final state of the filter is saved for use in generating the output samples for the next pitch subframe, and for use in the searches for the next pitch subframe in the encoder. The filter memories of 1/P(z), and $p_d(n)$ are kept in integer format, without fractional bits.

2.4.8.3 Generating the Formant Synthesis Filter Output

Both the transmitting speech codec and the receiving speech codec generate the output of the formant synthesis filter, $y_d(n)$, identically. The LSP frequencies are interpolated appropriately for the codebook subframe being generated, as described in 2.4.3.3.4. The interpolated LSP frequencies are then converted back into LPC coefficients as described in 2.4.3.3.5. The filter 1/A(z) is initialized with the final state resulting from the last output sample generated, but using $\{\hat{a}_i\}$ equal to the LPC coefficients generated for the current codebook subframe. $p_d(n)$ is filtered by 1/A(z) to produce $y_d(n)$. The final state of the filter is saved for use in generating the output samples for the next codebook subframe, and for use in the searches for the next codebook subframe in the encoder. The filter memories of 1/A(z) and $y_d(n)$ are in integer format, without fractional bits.

2.4.8.4 Updating the Memories of W(z) in the Transmitting Speech Codec

At the encoder, $s(n) - y_d(n)$ is then filtered by W(z) to update the filter memories of W(z) for use in the searches of the next codebook subframe. The filter W(z) is initialized with the final state resulting from the last output sample generated, but with $\{\hat{a}_i\}$ equal to the LPC coefficients appropriate for the current codebook subframe. $s(n) - y_d(n)$ is filtered by W(z). The output of this filter may be discarded. The final state of the filter is saved for use in the searches for the next codebook subframe.

---

[13]The sequence rnd(i) should have the same energy as the codebook. A uniform probability density function from $\sqrt{3}$ to $-\sqrt{3}$ has variance 1, so rnd would be normalized to have variance 1 by multiplying the value by $\sqrt{3}$ /32768.0. However, the codebook has a variance of 0.629. To create a random sequence with variance of 0.629, rnd is multiplied by $\sqrt{3 * 0.629}$ /32768.0 $=\sqrt{1.887}$ /32768.0

2.4.8.5 The Adaptive Postfilter in the Receiving Speech Codec

At the decoder, an adaptive postfilter should be used to enhance the perceptual quality of the output speech. The postfilter has the form

$$PF(z) = B(z)\, A(z/p)/A(z/s), \qquad (2.4.8.5\text{-}1)$$

where A(z) is the formant prediction error filter defined in Equation 2.4.3.1-1; p = 0.5 and s = 0.8. B(z) is an anti-tilt filter designed to offset the spectral tilt introduced by A(z/p)/A(z/s). B(z) is as follows:

$$B(z) = \frac{1 - \gamma z^{-1}}{1 + \gamma z^{-1}}\ , \qquad (2.4.8.5\text{-}2)$$

where γ is a function of the average of the ten interpolated LSP frequencies $\hat{w}'_i$ as follows:

$$\gamma = \begin{cases} 0.25 & \text{if average}(\hat{w}'_i) \le 0.24 \\ -25\,(\text{average}(\hat{w}'_i) - 0.25) & \text{if } 0.24 < \text{average}(\hat{w}'_i) \le 0.26 \\ -0.25 & \text{if average}(\hat{w}'_i) > 0.26 \end{cases} \quad . \qquad (2.4.8.5\text{-}3)$$

Typically, the average of the ten LSP frequencies is less than 0.25, which results in a B(z) that is a high-pass brightener. Occasionally the average of the ten LSP frequencies is greater than 0.25, which results in a B(z) that is a low-pass dampener.

The filter PF(z) is initialized with the final state resulting from the last output sample. In this case, γ is a function of the current average of the ten interpolated LSP frequencies and the coefficients of A(z) are equal to the LPC coefficients appropriate for the current codebook subframe (see 2.4.3.2.4). $y_d(n)$ should then be filtered by PF(z) to produce pf(n).

A gain control should be put on the output of PF(z) to ensure that the energy of the output signal is roughly the same as the energy of the input signal. The input and output energies are computed on 40 sample intervals, regardless of the data rate selected. This is accomplished as follows:

First the energy of the 40 samples of the input, $y_d(n)$, is computed as follows:

$$E_{in} = \sum_{n=0}^{39} y_d^2(n)\ . \qquad (2.4.8.5\text{-}4)$$

The energy of the output, pf(n), is computed in the same manner:

$$E_{out} = \sum_{n=0}^{39} pf^{\,2}(n)\ . \qquad (2.4.8.5\text{-}5)$$

An initial scale factor, $SCALE_{init}$, is computed as follows:

$$\mathrm{SCALE}_{init} = \sqrt{\frac{E_{in}}{E_{out}}}\ . \tag{2.4.8.5-6}$$

SCALE_{init} is then filtered by a first order IIR filter to produce the final scale factor, SCALE_{fin}, by

$$\mathrm{SCALE}_{fin}(\text{current}) = 0.9375\ \mathrm{SCALE}_{fin}(\text{previous}) + 0.0625\ \mathrm{SCALE}_{init}(\text{current}). \tag{2.4.8.5-7}$$

SCALE_{init}(current) is the SCALE_{init} for the current 40 samples defined above and SCALE_{fin}(previous) is the SCALE_{fin} from the previous 40 samples.

The reconstructed speech $s_d(n)$ is then computed as

$$s_d(n) = \mathrm{SCALE}_{fin}(\text{current})\ pf(n). \tag{2.4.8.5-8}$$

2.4.8.6 Special Cases

2.4.8.6.1 Insufficient Frame Quality (Erased) Packets

If the transmission rate cannot be satisfactorily determined, the multiplex sublayer informs the receiving speech codec of an erasure (see 2.3.2.2). In addition, the receive speech codec may declare an erasure when it receives a Rate 1 packet and errors are detected (see 2.4.8.6.2), when it receives a Rate 1 packet with bit errors and the number of errors exceeds one (see 2.4.8.6.3), or when it receives a Rate 1/8 packet consisting of all ones (see 2.4.8.6.5).

When the receive speech codec receives or declares an erased packet, the decoder decays all the parameters toward their initialization levels. The current value of $\hat{G}_l$ is set to the largest integer less than 0.7 times the previous value of $\hat{G}_l$ (the previous codebook gain in dB). This value is entered into the predictor so that in the next frame, the output of the predictor will be a function of the average of the previous dB value and the largest integer less than 0.7 times the previous dB value. The linear codebook gain for the current frame, $\hat{G}_a$, is computed from the current value of $\hat{G}_l$ using Table 2.4.6.2.1-1, and this value of $\hat{G}_a$ is used for the entire frame.[14] $\hat{G}_s$ is set equal to 1. In this way, multiple erasures will result in a steadily decreasing volume.

The codebook index is randomly chosen. The random codebook index is used as if the codebook subframe size is 160 samples.

If the last frame received prior to an insufficient frame quality or Rate 1 with Bit Errors packet was not Rate 1/8, the following procedure is followed to determine the pitch gain and lag. The pitch lag is repeated from the last pitch subframe of the previous frame. The pitch gain depends on the number of consecutive insufficient frame quality or Rate 1 with

[14] The output of the predictor and inverse quantizer in Figure 2.4.6.2.1-1 are ignored.

Bit Errors packets received as follows. For the first such packet, the pitch gain is saturated at 0.9 (i.e., the pitch gain is set to the minimum of the previous subframe's and the value 0.9). For the second consecutive such packet, the pitch gain is saturated at 0.6. For the third consecutive such packet, the pitch gain is saturated at 0.3. If four or more consecutive such packets are received, the pitch gain is set to zero. If the last frame received prior to an insufficient frame quality or Rate 1 with Bit Errors packet was Rate 1/8, the pitch lag and gain are set to zero for all consecutive insufficient frame quality or Rate 1 with Bit Errors packets received. The value of the previous pitch lag and gain are initialized to zero.

The memories in the LSP predictors are multiplied by the predictor coefficient, 0.90625, and LSP frequencies are regenerated from these memories. The LSP frequencies are checked for stability and are low-pass filtered using SM = 0.875 (see 2.4.3.3.3). These uninterpolated LSP frequencies are then converted back into LPC coefficients, which are used for the entire frame of reconstructed speech.[15] In this way, multiple erasures will eventually lead to predictor memories equal to zero, resulting in LSP frequencies at their bias levels. This results in no shaping by the formant synthesis (LPC) filter, so erasures slowly move the LPC spectrum towards white noise.

These parameters are then used to reconstruct the current frame of speech and to update the filter memories for the next codebook subframe at the decoder.

2.4.8.6.2 Rate 1 Packets

The receiving speech codec evaluates a Rate 1 packet for bit errors. If the ten parity check bits PCB[1] through PCB[10] show that the packet has no errors,[16] the received frame is decoded as a normal Rate 1 frame. If the ten parity check bits PCB[1] through PCB[10] show that the packet has an error, the packet is declared an erased packet and handled as described in 2.4.8.6.1.

2.4.8.6.3 Rate 1 Packets with Bit Errors

In certain cases, the decoder may receive a Rate 1 packet with bit errors. This received packet is most likely a full rate frame with errors. The decoder evaluates this data, and reconstructs the speech in different ways depending on the assessed quality of the received packet.

The decoder first verifies the 11 parity check bits, PCB[0] through PCB[10] as follows.

- If the ten parity check bits PCB[1] through PCB[10] show that the packet has no errors, the received packet is decoded as a normal Rate 1 packet with the exception

---

[15]Note that this is equivalent to having the output of Q_{wi}^{-1} in Figure 2.4.3.3.1-1 equal to 0 for an erased packet.

[16]See Lin and Costello, *Error Control Coding: Fundamentals and Applications*, pp. 103-110 for a discussion of methods for determining whether there is an error and the number of errors.

that the pitch gain and lag are determined as stated below for all four pitch subframes. The codebook parameters and LSP parameters are decoded and used as in a typical Rate 1 packet to reconstruct the speech and update the filter memories.

If the last frame received prior to an insufficient frame quality or Rate 1 with Bit Errors packet was not Rate 1/8, the following procedure is followed to determine the pitch gain and lag. The pitch lag is repeated from the last pitch subframe of the previous frame. The pitch gain depends on the number of consecutive insufficient frame quality or Rate 1 with Bit Errors packets received as follows. For the first such packet, the pitch gain is saturated at 0.75 (i.e., the pitch gain is set to the minimum of the previous subframe's pitch gain and 0.75). For the second consecutive such packet, the pitch gain is saturated at 0.5. For the third consecutive such packet, the pitch gain is saturated at 0.25. If four or more consecutive such packets are received, the pitch gain is set to zero. If the last frame received prior to an insufficient frame quality or Rate 1 with Bit Errors packet was Rate 1/8, the pitch lag and gain are set to zero for all consecutive insufficient frame quality or Rate 1 with Bit Errors packets received. The value of the previous pitch lag and gain are initialized to zero.

- If the ten parity check bits PCB[1] through PCB[10] show that the packet has only one bit in error and the parity bit PCB[0] does not check, the bit in error is corrected. Speech reconstruction is then completed as described above.
- If the ten parity check bits PCB[1] through PCB[10] show that the packet has only one bit in error and the parity bit PCB[0] checks, or if the ten parity check bits PCB[1] through PCB[10] detect more than one bit in error, the frame is declared an erasure, and the speech is reconstructed as described in 2.4.8.6.1.

2.4.8.6.4 Blanked Packets

A blanked frame occurs when the transmitting station uses the entire frame for either signaling traffic or secondary traffic. Blanking differs from erasing in that the encoder is aware that the packet is blanked, whereas the encoder is unaware of erasures. As such, the blanking algorithm is used at both the encoder and decoder for reconstructing the speech and updating the filter memories.

For a blanked packet, the codebook gain is set to zero ($\hat{G}$ is set equal to 0), which effectively disables the codebook. However, changes are not made in the codebook predictor memories. The pitch parameters from the last pitch subframe of the previous frame are used, with the exception that if the pitch gain is greater than 1, it is set equal to 1. The previous frame's uninterpolated LSP frequencies, $\hat{w}_i$, are converted into LPC coefficients, which are used for the entire frame. However, changes are not made in the LSP predictor memories, $P_W(z)$.

From the above parameters, the speech is reconstructed and the filter memories are updated at both the encoder and the decoder.

2.4.8.6.5 All Ones Rate 1/8 Packets

A Rate 1/8 packet consisting of all ones is considered as null Traffic Channel data. This packet is declared an erased packet and handled as described in 2.4.8.6.1.

2.4.9 Speech Codec Initialization

Upon being commanded to initialize the receiving side, the speech codec sets all receiving parameters as follows:

- The filter and predictor memories are set to zero.
- The LSPs, $\hat{w}_i$(previous), are set to $Bias_i$ (see 2.4.3.2.7 and 2.4.3.3.1).
- The Rate 1/8 codebook gain, $\hat{G}'$ (old), is set to 0 (see 2.4.8.1.2).
- The adaptive postfilter gain, $SCALE_{fin}$ (previous), is set to 1.0 (see 2.4.8.5).
- The pitch gain and lag for the previous pitch subframe are set to zero (see 2.4.6.8.1 and 2.4.6.8.3).

Upon being commanded to initialize the transmitting side, the speech codec sets all transmitting parameters as follows:

- The filter and predictor memories are set to zero.
- The LSPs, $\hat{w}_i$(previous), are set to $Bias_i$ (see 2.4.3.2.7 and 2.4.3.3.1).
- The Rate 1/8 codebook gain, $\hat{G}'$ (old), is set to 0 (see 2.4.8.1.2).
- The background noise level, B_1, is set to 5059644, (see 2.4.4.3).
- The previous frame's energy estimate, $R(0)_{prev}$, is set to 5059644 (see 2.4.4.3).
- The Rate 1/8 random codebook seed, SD, is set to 0.

2.4.10 Output Audio Interface

2.4.10.1 Output Audio Interface in the Mobile Station

2.4.10.1.1 Band Pass Filtering

Output reconstruction filtering shall conform to CCITT Recommendation G.714 "Separate Performance Characteristics for the Encoding and Decoding Sides of PCM Channels Applicable to 4-Wire Voice-Frequency Interfaces." Additional reconstruction filtering may be provided by the manufacturer.

2.4.10.1.2 Receive Level Adjustment

The mobile station shall have a nominal receive objective loudness rating (ROLR) equal to 51 dB when receiving from a reference base station (see 2.4.2.2.2). The loudness ratings are described in IEEE Standard 661-1979 "IEEE Standard Method for Determining Objective Loudness Ratings of Telephone Connections." Measurement techniques and tolerances are described in TIA/EIA/IS-125.

2.4.10.2 Output Audio Interface in the Base Station

Details of the digital and analog interfaces to the network are outside the scope of this document.

2.4.10.2.1 Receive Level Adjustment

The base station shall set the audio level so that a received 1004 Hz tone 3.17 dB below maximum amplitude produces a level of 0 dBm0 at the network interface. Measurement techniques and tolerances are described in TIA/EIA/IS-125.

2.4.11 Summary of Encoding and Decoding

2.4.11.1 Encoding Summary

The following summarizes the steps taken to encode a frame:

1.0 Initial Computations

1.1 Remove the DC from the current input speech.

1.2 Compute the LPC coefficients for the current frame.

1.3 Compute the LSP frequencies from the LPC coefficients.

1.4 Compute the data rate.

1.5 Convert the LSP frequencies into transmission codes.

1.6 If the packet is Rate 1/2, go to 3.0.

1.7 If the packet is Rate 1/4, go to 4.0.

1.8 If the packet is Rate 1/8, go to 5.0.

1.9 If the packet is a Blank packet, go to 6.0.

1.10 Go to 2.0.

2.0 Rate 1 Packet Encoding

2.1 Start with the first pitch subframe.

2.2 Interpolate the LSPs for the pitch subframe and the two corresponding codebook subframes, then convert them back to LPC coefficients.

2.3 Find the optimal pitch gain and lag for the pitch subframe.

2.4 Find the optimal codebook gain and index for the first codebook subframe in the pitch subframe.

2.5 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

2.6 Find the optimal codebook gain and index for the second codebook subframe in the pitch subframe.

2.7 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

2.8 If all four pitch subframes for this frame have not been done, go to the next pitch subframe and go to 2.2.

2.9 Compute the CRC and pack the data into the 171-bit packet.

2.10 Done encoding.

3.0 Rate 1/2 Packet Encoding

3.1 Start with the first pitch subframe.

3.2 Interpolate the LSPs for the pitch subframe and the two corresponding codebook subframes, then convert them back to LPC coefficients.

3.3 Find the optimal pitch gain and lag for the pitch subframe.

3.4 Find the optimal codebook gain and index for the first codebook subframe in the pitch subframe.

3.5 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

3.6 Find the optimal codebook gain and index for the second codebook subframe in the pitch subframe.

3.7 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

3.8 If both pitch subframes for this frame have not been done, go to the next pitch subframe and go to 3.2.

3.9 Pack the data into the 80-bit packet.

3.10 Done encoding.

4.0 Rate 1/4 Packet Encoding

4.1 Interpolate the LSPs for the pitch subframe and the two corresponding codebook subframes, then convert them back to LPC coefficients.

4.2 Find the optimal pitch gain and lag for the pitch subframe.

4.3 Find the optimal codebook gain and index for the first codebook subframe.

4.4 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

4.5 Find the optimal codebook gain and index for the second codebook subframe.

4.6 Update the pitch synthesis filter, formant synthesis filter, and perceptual weighting filter memories.

4.7 Pack the data into the 40-bit packet.

4.8 Done encoding.

5.0 Rate 1/8 Packet Encoding

5.1 Interpolate the LSPs for the codebook subframe, then convert them back to LPC coefficients.

5.2 Find the optimal codebook gain and index for the codebook subframe.

5.3 Discard the index, generate CBSEED, and pack the data into the 16-bit packet.

5.4 Update the pitch, formant, and perceptual weighting filter memories using the 16-bit packet as the seed for the pseudo-random number generator in the decoder.

5.5 Done encoding.

6.0 Blank Packet Encoding

6.1 Set codebook gain to zero, without changing the predictor memories for the codebook gain.

6.2 Set the current pitch gain and lag to those of the last pitch subframe of the previous frame, and saturate the pitch gain to be no greater than 1.

6.3 Convert the previous frame's uninterpolated LSP frequencies to LPC coefficients, without changing the LSP predictor memories.

6.4 Update the pitch, formant, and perceptual weighting filter memories, using the codebook, pitch, and LPC parameters described above for the entire frame.

6.5 Done encoding.

2.4.11.2 Decoding Summary

The following summarizes the steps taken to decode a frame.

1.0 Initial Computations

1.1 If the received packet is Rate 1/2, go to 4.0.

1.2 If the received packet is Rate 1/4, go to 5.0.

1.3 If the received packet is Rate 1/8, go to 6.0.

1.4 If the received packet is blanked, go to 8.0.

1.5 If the received packet is of insufficient frame quality (erasure), go to 7.0.

1.6 If the received packet is Rate 1 with bit errors, go to 3.0.

1.7 Go to 2.0.

2.0 Rate 1 Packet Decoding

2.1 Unpack the 171-bit packet into the appropriate codes.

2.2 Check the internal packet parity check bits to determine if there are any detected errors. If there are any detected errors, then declare the packet has insufficient frame quality and go to 7.0.

2.3 Compute the speech codec parameters from the unpacked codes.

2.4 Compute the scaled codebook vector for all 160 samples using the codebook index and gain parameters for all eight codebook subframes.

2.5 Compute the output of the pitch synthesis filter for all 160 samples from the scaled codebook vector computed above using the pitch lag and gain parameters for all four pitch subframes.

2.6 Interpolate the LSP frequencies four times (once for each pitch subframe) and convert these frequencies to the LPC coefficients used in the formant synthesis and adaptive postfilter for the four pitch subframes.

2.7 Compute the output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the appropriate LPC coefficients for each pitch subframe of 40 samples.

2.8 Compute output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter using the appropriate LPC coefficients for each pitch subframe of 40 samples.

2.9 Done decoding.

3.0 Rate 1 Packet with Bit Errors Decoding

3.1 Check the internal packet parity check bits to see how many bit errors exist in the current packet.

3.2 If the number of bits in error is greater than one, or if there is one bit in error and the parity bit PCB[0] checks, go to 7.0.

3.3 If one bit is in error, correct it.

3.4 Unpack the 171-bit packet into the appropriate codes and compute the speech codec parameters from these codes.

3.5 Compute the output of the pitch synthesis filter for all 160 samples from the scaled codebook vector computed above using the smoothed pitch lag and gain parameters for all four pitch subframes (See 2.4.8.6.3).

3.6 Go to 2.4.

4.0 Rate 1/2 Packet Decoding

4.1 Unpack the 80-bit packet into the appropriate codes, and compute the speech codec parameters from these codes.

4.2 Compute the scaled codebook vector for all 160 samples using the codebook index and gain parameters for all four codebook subframes.

4.3 Compute output of the pitch synthesis filter for all 160 samples from the scaled codebook vector computed above using the pitch lag and gain parameters for both pitch subframes.

4.4 Interpolate the LSP frequencies two times (once for each pitch subframe) and convert these frequencies to the LPC coefficients used in formant synthesis and adaptive postfilter for the four codebook subframes.

4.5 Compute the output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the appropriate LPC coefficients for each codebook subframe of 40 samples.

4.6 Compute the output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter using the appropriate LPC coefficients for each codebook subframe of 40 samples.

4.7 Done decoding.

5.0 Rate 1/4 Packet Decoding

5.1 Unpack the 40-bit packet into the appropriate codes, and compute the speech codec parameters from these codes.

5.2 Compute the scaled codebook vector for all 160 samples using the codebook index and gain parameters for both codebook subframes.

5.3 Compute the output of the pitch synthesis filter for all 160 samples from the scaled codebook vector computed above using the pitch lag and gain parameters for the pitch subframe.

5.4 Interpolate the LSP frequencies once (for the pitch subframe) and convert these frequencies to the LPC coefficients used in formant synthesis and adaptive postfilter for the two codebook subframes.

5.5 Compute output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the appropriate LPC coefficients for each codebook subframe of 80 samples.

5.6 Compute the output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter using the appropriate LPC coefficients for each codebook subframe of 80 samples.

5.7 Done decoding.

6.0 Rate 1/8 Packet Decoding

6.1 If the packet is all 1’s (the frame was null Traffic Channel data), then go to 7.0.

6.2 Unpack the 16-bit packet into the appropriate codes and compute the speech codec parameters from these codes.

6.3 Compute the scaled codebook vector for all 160 samples using the 16-bit packet as the random seed for the pseudorandom number generator and low pass filtering and interpolating the codebook gain.

6.4 The output of the pitch synthesis filter equals the scaled codebook vector because the pitch synthesis filter is not used.

6.5 Interpolate the LSP frequencies once for the codebook subframe (the entire frame), and convert these frequencies to the LPC coefficients used in the formant synthesis and adaptive postfilter for the codebook subframes (the entire frame).

6.6 Compute the output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the appropriate LPC coefficients for the codebook subframe of 160 samples.

6.7 Compute the output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter again using the appropriate LPC coefficients for the codebook subframe of 160 samples.

6.8 Done decoding.

7.0 Insufficient Frame Quality (Erasure) Decoding

7.1 Decay the codebook gain magnitude in dB by 0.7, update the codebook gain magnitude predictor memories, and compute the resulting linear value of the codebook gain.

7.2 Select a random codebook index.

7.3 Compute the scaled codebook vector for all 160 samples using the codebook gain and index parameters generated above.

7.4 Compute the output of the pitch synthesis filter for all 160 samples from the scaled codebook vector computed above using the smoothed pitch lag and gain parameters for all four pitch subframes (See 2.4.8.6.1).

7.5 Decay the LSP predictor memories by 0.90625, compute the resulting LSP frequencies, and convert them into the LPC coefficients used in the formant synthesis and adaptive postfilter for the frame.

7.6 Compute the output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the LPC coefficients computed above.

7.7 Compute the output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter using the LPC coefficients computed above.

7.8 Done decoding.

8.0 Blank Packet Decoding

8.1 Set the codebook gain to zero, without changing the codebook gain predictor memories. The scaled codebook vector is the all zero vector.

8.2 Set the current pitch gain and lag to those of the last pitch subframe of the previous frame, and saturate the pitch gain to be at most unity.

8.3 Compute the output of the pitch synthesis filter for all 160 samples using the pitch gain and lag parameters defined above.

8.4 Set the current LSP frequencies to be the uninterpolated LSP frequencies from the previous frame, without changing the LSP predictor memories. Compute the LPC coefficients from these LSP frequencies.

8.5 Compute the output of the formant synthesis filter for all 160 samples from the output of the pitch synthesis filter using the LPC coefficients computed above.

8.6 Compute the output of the adaptive postfilter and the reconstructed speech for all 160 samples from the output of the formant synthesis filter using the LPC coefficients computed above.

8.7 Done decoding.

2.4.12 Allowable Delays

2.4.12.1 Allowable Transmitting Speech Codec Encoding Delay

The transmitting speech codec in the mobile station shall supply a packet to the multiplex sublayer not later than 20 ms after it has obtained the last input sample for the Hamming window (see 2.4.3.2.2).

2.4.12.2 Allowable Receiving Speech Codec Decoding Delay

The receiving decoder in the mobile station shall generate the first sample of speech using parameters from a packet supplied to it by the multiplex sublayer not later than 3 ms after being supplied the packet.

2.5 Summary of Service Option 1 Notation

Table 2.5-1 lists the notation used by Service Option 1, Variable Data Rate Two-Way Voice.

Table 2.5-1. Summary of Service Option 1 Notation (Part 1 of 5)

Parameter	Section	Name/Description
α_i	2.4.3.2.5	Linear predictive coding coefficients before bandwidth expansion.
a(x)	2.4.7.1.1	Input polynomial in GF(2); used for parity checking.
a_i	2.4.3.2.5	Linear predictive coding coefficients.
$\hat{a}_i$	2.4.3.3.5	Quantized, smoothed and interpolated LPC coefficients.
$a_{zir}(n)$	2.4.5.1.1	Zero input response of the formant synthesis filter.
A(z)	2.4.3.1	Formant prediction error filter.
1/A(z)	2.4.3.1	Formant synthesis filter.
b	2.4.5.1	Pitch gain.
b*	2.4.5.1.1	Optimal pitch gain.
$\hat{b}$	2.4.5.2	Pitch gain used for synthesis.
β	2.4.3.2.5	Scaling factor for bandwidth expansion.
B_i	2.4.4.3	Background noise estimate for the ith frame.
$Bias_i$	2.4.3.2.7	Line spectral pair bias for LSP frequency i.
B(z)	2.4.8.5	Anti-tilt filter.
CBGAINi	2.4.1	Codebook gain for the ith codebook subframe.
CBINDEXi	2.4.1	Codebook index for the ith codebook subframe.
CBSEED	2.4.1	Four bit value to randomize Rate 1/8 packets.
$c_d(n)$	2.4.8.1	Scaled codebook vector.
$c_I(n)$	2.4.6.1.1	The codebook vector for index I.
c(n)	2.4.6.1.1	Circular codebook values.
decrv	2.4.8.1.2	Random variable used in generating the Rate 1/8 code vector.
DECSD	2.4.8.1.2	The decoder seed for a Rate 1/8 packet. Equal to the entire packet.
e(n)	2.4.5.1.1 2.4.6.1.1	The error between the input speech signal and the response of the formant synthesis filter.
$E^{(i)}$	2.4.3.2.4	Energy of prediction error with formant synthesis (LPC) filter of order i.
E_{in}	2.4.8.5	Input energy to the adaptive postfilter.
E_{out}	2.4.8.5	Output energy of the adaptive postfilter.

1 **Table 2.5-1. Summary of Service Option 1 Notation (Part 2 of 5)**

Parameter	Section	Name/Description
E_{xyI}	2.4.6.1.1	The zero-offset cross correlation between the output of the perceptual weighting filter and the weighted synthesis filter for the codebook search.
E_{yyI}	2.4.6.1.1	The energy output of the weighted synthesis filter for the codebook search.
E_{xyL}	2.4.5.1.1	The zero-offset cross correlation between the output of the perceptual weighting filter and the weighted synthesis filter for the pitch search.
E_{yyL}	2.4.5.1.1	The energy output of the weighted synthesis filter for the pitch search.
$F_G(x)$	2.4.6.1.3	Codebook gain prediction filter function.
γ	2.4.8.5	Anti-spectral tilt filter coefficient.
G	2.4.6.1	Codebook gain.
G*	2.4.6.1	Optimal codebook gain.
$\hat{G}$	2.4.6.2.1	Decoded codebook gain (Decoded, filtered and interpolated for Rate 1/8).
$\hat{G}_a$	2.4.6.2	Decoded linear codebook gain magnitude.
$\hat{G}'$	2.4.6.2.2	Decoded and filtered codebook gain (used for Rate 1/8 and erased packets).
G_l	2.4.6.1.3	Codebook gain magnitude in dB.
$\hat{G}_l$	2.4.6.2.1	Decoded codebook gain magnitude in dB.
$g_{pc}(x)$	2.4.7.1.1	Parity check generator polynomial.
G_s	2.4.6.1.3	Sign of the codebook gain.
$\hat{G}_s$	2.4.6.2.1	Sign of the decoded codebook gain.
h(n)	2.4.5.1.2	Impulse response of H(z).
H(z)	2.4.5.1	Weighted synthesis filter. The combined formant synthesis filter and perceptual weighting filter.
i	All sections	Index.
I	2.4.6.1	Codebook index.
I*	2.4.6.1	Index of optimal codeword.
$\hat{I}$	2.4.6.2.1	Codebook index used for synthesis.
$\alpha_j^{(i)}$	2.4.3.2.4	LPC coefficient j of formant synthesis (LPC) filter of order i.
k	All sections	Index.

1 **Table 2.5-1. Summary of Service Option 1 Notation (Part 3 of 5)**

Parameter	**Section**	**Name/Description**
k_i	2.4.3.2.4	Partial correlation coefficients.
L	2.4.5.1	Pitch lag.
L*	2.4.5.1.1	Optimal pitch lag.
$\hat{L}$	2.4.5.2	Pitch lag used for synthesis.
L_A	2.4.1	LPC frame length in samples.
L_C	2.4.1	Codebook subframe length in samples.
L_P	2.4.1	Pitch subframe length in samples.
LSPi	2.4.1	Transmission code for Line spectral pair frequency i.
N	2.4.3.2.7	Number of bits of quantization in $Q_{wi}(x)$.
N_h	2.4.5.1.1	Number of samples that are used from the impulse response of the weighted synthesis filter.
P	2.4.3.1	Order of formant synthesis (LPC) filter.
$P_A(z)$	2.4.3.2.6	Intermediate polynomial used in transforming the LPC coefficients to LSP frequencies.
$\hat{P}_A(z)$	2.4.3.3.5	Intermediate polynomial used in transforming the interpolated LSP frequencies to LPC coefficients.
$pa_{zir}(n)$	2.4.6.1.1	Zero input response of the cascade of the pitch and formant synthesis filters.
PCB	2.4.1	Parity check bits for Rate 1 packets.
$p_c(n)$	2.4.5.1.1	Past outputs of the pitch synthesis filter.
$p_d(n)$	2.4.8.2	Output of the pitch synthesis filter.
PF(z)	2.4.8.5	Adaptive post filter.
pf(n)	2.4.8.5	Output of the adaptive post filter.
P_G	2.4.6.1.3	Codebook gain predictor.
$P_G(\mathbf{x},n)$	2.4.6.1.3	Output of P_G at time n for input sequence **x**.
PGAINi	2.4.1	Transmission code for the pitch gain for the ith pitch subframe.
p_i	2.4.3.2.6	Coefficients of $P_A(z)$.
$\hat{p}_i$	2.4.3.3.5	Coefficients of $\hat{P}_A(z)$.
p'_i	2.4.3.2.6	Coefficients of P'(w).
$p_L(n)$	2.4.5.1.1	Estimated output of the pitch synthesis filter for lag L with b = 1.

1 **Table 2.5-1. Summary of Service Option 1 Notation (Part 4 of 5)**

Parameter	Section	Name/Description
PLAGi	2.4.1	Pitch lag for the ith pitch subframe.
p(n)	2.4.5.1.1	Combined past outputs and estimated future outputs of the pitch synthesis filter.
$p_o(n)$	2.4.5.1.1	Estimate of the future outputs of the pitch synthesis filter.
P'(w)	2.4.3.2.6	Function used in computing LSP frequencies.
$P_w(z)$	2.4.3.2.7	Prediction filter used in converting LSP frequencies.
1/P(z)	2.4.1	Pitch synthesis filter.
$p_{zir}(n)$	2.4.6.1.1	Zero input response of the pitch synthesis filter.
$Q_A(z)$	2.4.3.2.6	Intermediate polynomial used in transforming the LPC coefficients to LSP frequencies.
$\hat{Q}_A(z)$	2.4.3.3.5	Intermediate polynomial used in transforming the interpolated LSP frequencies to LPC coefficients.
$Q_G(x)$	2.4.6.1.3	Codebook gain quantizer function.
q_i	2.4.3.2.6	Coefficients of $Q_A(z)$.
$\hat{q}_i$	2.4.3.3.5	Coefficients of $\hat{Q}_A(z)$.
q'_i	2.4.3.2.6	Coefficients in Q'(w).
Q'(w)	2.4.3.2.6	Function used in computing LSP frequencies.
$Q_{ti}(x)$	2.4.3.2.7	Quantizer without limiting for the ith LSP frequency.
$Q_{wi}(x)$	2.4.3.2.7	Quantizer for the ith LSP frequency.
Q_{wi}^{min}	2.4.3.2.7	Minimum LSP quantization level for the ith coefficient.
Q_{wi}^{max}	2.4.3.2.7	Maximum LSP quantization level for the ith coefficient.
$R(0)_{prev}$	2.4.4.2	R(0) for the previous frame.
R(k)	2.4.3.2.3	kth value of the autocorrelation function for the current frame.
SD	2.4.6.1.3	Random number used to generate CBSEED in a Rate 1/8 packet.
s(n)	2.4.3.2.2	Input speech samples corresponding to the frame or subframe with DC removed.
$s_d(n)$	2.4.8	Speech reconstructed by the receiving speech codec.
$s_w(n)$	2.4.3.2.2	Windowed input speech signal.
$SCALE_{fin}$	2.4.8.5	Final scale factor for the adaptive postfilter in the receiving speech codec.

Table 2.5-1. Summary of Service Option 1 Notation (Part 5 of 5)

Parameter	Section	Name/Description
$SCALE_{init}$	2.4.8.5	Initial scale factor for the adaptive postfilter in the receiving speech codec.
SM	2.4.3.3.3	Low-pass filter coefficient for the LSP frequency low-pass filter.
$T_k(B_i)$	2.4.4.1	Thresholds used to determine the data rate.
w_i	2.4.3.2.6	LSP frequencies.
$\tilde{w}_i$	2.4.3.3.1	Regenerated LSP frequencies.
$\hat{w}_i$	2.4.3.3.3	$\tilde{w}_i$ after stabilization and filtering.
$\hat{w}'_i$	2.4.3.3.4	$\hat{w}_i$ after interpolation.
$\Delta\tilde{w}_{min}$	2.4.3.3.2	Minimum LSP frequency spacing.
$W_H(n)$	2.4.3.2.2	Hamming window.
W(z)	2.4.5.1	Perceptual weighting filter.
x(n)	2.4.5.1.1	e(n) filtered by W(z).
$y_d(n)$	2.4.8.3	Formant synthesis filter output.
$y_I(n)$	2.4.6.1.1	$c_I(n)$ convolved by h(n).
$y_L(n)$	2.4.5.1.1	$p_L(n)$ convolved by h(n).
z	All sections	z transform variable.
ζ	2.4.5.1	Perceptual weighting parameter used in W(z).

ANNEX A BIBLIOGRAPHY

This is an informative annex. The documents listed in this annex are for information only and are not essential for the completion of the requirements of this standard.

—*Books:*

1. Rabiner, L. R. and Schafer, R. W., *Digital Processing of Speech Signals*, New Jersey, Prentice-Hall Inc., 1978.
2. Lin, S. and Costello, D. J., *Error Control Coding: Fundamentals and Applications*, New Jersey, Prentice-Hall Inc., 1983.
3. Oppenheim, A. V. and Schafer, R. W., *Digital Signal Processing*, New Jersey, Prentice-Hall, Inc., 1975.

No text.