3GPP2 C.S0014-A

Version 1.0

Version Date: April 2004

3rd Generation Partnership Project 2 "3GPP2"

Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems

Table of Contents

1 GENERAL

1.1 General Description

Service Option 3 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.† The receiving station generates a speech packet from every Traffic Channel frame and supplies it to the speech codec for decoding into voice samples.

Speech codecs communicate at one of three rates corresponding to the 9600 bps, 4800 bps, and 1200 bps frame rates.

The specifications defined in Sections 4 and 5 of this document provide the detailed algorithmic description of the EVRC. In the case of a discrepancy between the floating point and algorithmic descriptions, the bit-exact specification will prevail. The specifications defined in Input Signal Preprocessing (see 4.1), Determining the Data Rate (see 4.3), and Adaptive Postfilter (see 5.4) are optional for implementations intended for varying operational environments (such as in-vehicle hands-free). Any implementation, which deviates from the algorithm specified in this standard, shall meet the minimum performance requirements defined in 3GPP2 C.S0018-0-1.

1.2 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 3 and shall be called Service Option 3.

1.3 Allowable Delays

1.3.1 Allowable Transmitting Speech Codec Encoding Delay

The transmitting speech codec shall supply a packet to the multiplex sublayer no later than 20 ms after it has obtained the last input sample for the current speech frame.

1.3.2 Allowable Receiving Speech Codec Decoding Delay

The receiving decoder 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.

† IS-95-A 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 3, 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.

1.4 Special Cases

1.4.1 Blanked Packets

A blanked frame occurs when the transmitting station uses the entire frame for either signaling traffic or secondary traffic. The EVRC does no special encode processing during the generation of a blank packet, i.e., the generated voice packet is simply not used. The decoder, in turn, treats a blank packet in the same manner as a frame erasure.

1.4.2 Null Traffic Channel Data

A Rate 1/8 packet with all bits set to '1' is considered as null Traffic Channel data. This packet is declared an erased packet and handled as described in Section 5. If more than 2 consecutive all-ones Rate 1/8 packets are received, the decoder's output shall be muted until a valid packet is received.

1.4.3 All Zeros Packet

Rate 1 and Rate ½ packets with all bits set to '0' shall be considered erased frames and shall be handled as described in Section 5.

1.5 Terms and Numeric Information

ACB. Adaptive Codebook.

ACELP. Algebraic Code Excited Linear Predictive Coding, the algorithm that is used by the EVRC to generate the stochastic component of the excitation.

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 Radio Telecommunications Service, other than a mobile station, used for radio communications with mobile stations.

CCITT. New revisions of CCITT standards will have an ITU designation.

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 codecs 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 3. 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.

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.

DFT. See Discrete Fourier Transform.

Discrete Fourier Transform (DFT). A method of transforming a time domain sequence into a corresponding frequency domain sequence.

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.

EVRC. Enhanced Variable Rate Codec.

FCB. Fixed Codebook.

FFT. See Fast Fourier Transform.

Fast Fourier Transform (FFT). An efficient implementation of the Discrete Fourier Transform.

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

IDFT. See Inverse Discrete Fourier Transform.

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.

Interpolation. In the speech coder context, a means of smoothing the transitions of estimated parameters from one set to another. Usually a linear function.

Inverse Discrete Fourier Transform (IDFT). A method of transforming a frequency domain sequence into a corresponding time domain sequence.

ITU. International Telecommunication Union.

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 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.

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

PSTN. Public Switched Telephone Network.

Quantization. A process that allows one or more data elements to be represented at a lower resolution for the purpose of reducing the effective bandwidth required for transmission or storage of the respective data elements.

RCELP. Relaxed Code Excited Linear Predictive Coding, the speech coding algorithm on which the EVRC is based.

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.

ROLR. See Receive 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 REQUIRED MULTIPLEX OPTION SUPPORT

Service Option 3 shall support an interface with Multiplex Option 1. Speech packets for Service Option 3 shall only be transported as primary traffic.

2.1 Interface to Multiplex Option 1

2.1.1 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 four types as shown in Table 2.1.1-1. The number of bits supplied to the multiplex sublayer for each type of packet shall also be as shown in Table 2.1.1-1. Unless otherwise commanded, the speech codec may supply a Rate 1, Rate 1/2, 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.

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

Table 2.1.1-1 Packet Types Supplied by Service Option 3 to the Multiplex Sublayer

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

2.1.2 Received Packets

The multiplex sublayer in the receiving station categorizes every received Traffic Channel frame, and supplies the packet type and accompanying bits, if any, to the speech codec as shown in Table 2.1.1-1. The speech codec processes the bits of the packet as described in Section 4. The received packet types shown in Table 2.1.2-1 correspond to the transmitted packet types shown in Table 2.1.1-1. 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. Although the Service Option 3 does not utilize Rate 1/4 packets, Multiplex Option 1 is not required to recognize this fact; Service Option 3 is, therefore, responsible for declaring Rate 1/4 frames as having insufficient frame quality (erasure).

Table 2.1.2-1 Packet Types Supplied by the Multiplex Sublayer to Service Option 3

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

2.2 Negotiation for Service Option 3

The mobile station and base station can negotiate for Service Option 3 using either service option negotiation, as described in IS-95, or service negotiation, as described in IS-95 and ANSI J-STD-008.

2.2.1 Procedures Using Service Option Negotiation

The mobile station shall perform service option negotiation for Service Option 3 as described in 6.6.4.1.2 of IS-95-A. The base station shall perform service option negotiation for Service Option 3 as described in 7.6.4.1.2 of IS-95-A.

2.2.1.1 Initialization and Connection

2.2.1.1.1 Initialization and Connection in the Mobile Station

If the mobile station sends a *Service Option Response Order* accepting Service Option 3 in response to receiving a *Service Option Request Order*, (see 6.6.4.1.2.2.1 of IS-95-A), the mobile station shall initialize and connect Service Option 3 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 IS-95-A), or
    - The time that the mobile station sends the *Service Option Response Order* accepting Service Option 3,

  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 3, 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 3, or
  - If not in the *Conversation Substate*, the time that the mobile station enters the *Waiting for Answer Substate e*, whichever is later.

If the mobile station receives a *Service Option Response Order* accepting its request for Service Option 3 (see 6.6.4.1.2.2.2 of IS-95-A), the mobile station shall initialize and connect Service Option 3 according to the following:

- If the mobile station is in the *Conversation Substate e*, 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 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 3 initializations are described in Sections 4 and 5.

When the transmitting side of Service Option 3 is connected, Service Option 3 shall generate and transfer packets to the multiplex sublayer. When the receiving side is connected, Service Option 3 shall transfer and process packets from the multiplex sublayer. Refer to 6.1.3.3.11.3 of IS-95-A when the transmitting side of a service option is not connected.

2.2.1.1.2 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 3.

If the base station accepts Service Option 3 (by sending a *Service Option Response Order* as described in 7.6.4.1.2.2.1 of IS-95-A), it should initialize and connect both the transmitting and receiving side of Service Option 3 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 3 (by receiving a *Service Option Response Order* as described in 7.6.4.1.2.2.2 of IS-95-A), it should initialize and connect both the transmitting and receiving side of Service Option 3 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 3 is connected, Service Option 3 shall generate and transfer packets to the multiplex sublayer. When the receiving side is connected, Service Option 3 shall transfer and process packets from the multiplex sublayer. Refer to 7.1.3.5.11.3 of IS-95-A when the transmitting side of a service option is not connected.

2.2.1.2 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 IS-95-A). In addition to pending ACTION_TIMEs for messages or orders not related to the *Service Option Control Order* for Service Option 3, the mobile station shall support at least one pending ACTION_TIME for *Service Option Control Orders* for Service Option 3. The mobile station shall not send a *Service Option Control Order* for this service option.

If Service Option 3 is active, the mobile station shall treat 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 Section 4 and 5. The initializations shall begin at the implicit or explicit action time (see 6.6.4.1.5 of 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.

If Service Option 3 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 2.2.1.2-1, then the mobile station shall generate the fraction of those packets normally generated as Rate 1 packets (see 4.3) 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 3 is active, the mobile station receives a *Service Option Control Order* that specifies different fractions, or
- Service Option 3 is initialized.

Table 2.2.1.2-1. 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 as are formed as shown in Table 2.2.1.2-2. 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 4.3) 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 2.1.1).

Table 2.2.1.2-2. 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 3 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 IS-95-A).

2.2.2 Procedures Using Service Negotiation

The mobile station and base station shall perform service negotiation for Service Option 3 as described in IS-95 or J-STD-008, and the negotiated service configuration shall include only valid attributes for the service option as specified in Table 2.2.2-1.

Table 2.2.2-1. Valid Service Configuration Attributes for Service Option 3

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 rates enabled
Reverse Transmission Rates	Rate Set 1 with all rates enabled
Forward Traffic Type	Primary Traffic
Reverse Traffic Type	Primary Traffic

2.2.2.1 Initialization and Connection

2.2.2.1.1 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 3, 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 Sections 4 and 5) 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 3 shall process received packets and generate and supply packets for transmission as follows:
  - If the mobile station is in the *Conversation Substate*, Service Option 3 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 e*, Service Option 3 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' for transmission, except when commanded to generate a Blank packet.

2.2.2.1.2 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 3, 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 Sections 4 and 5) 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 3 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.

2.2.2.2 Service Option Control Messages

2.2.2.2.1 Mobile Station Requirements

The mobile station shall support one pending *Service Option Control Message* for Service Option 3.

If the mobile station receives a *Service Option Control Message* for Service Option 3, 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.

   If the MOBILE_TO_MOBILE field is equal to '0', the mobile station shall process each received packet as described in Section 5.

2. If the INIT_CODEC field is equal to '1', the mobile station shall perform speech codec initialization (see Sections 4 and 5). The mobile station shall complete the initialization within 40 ms.

3. If the RATE_REDUC field is equal to a value defined in Table 2.2.2.2.2-2, Service Option 3 shall generate the fraction of those packets normally generated as Rate 1 packets (see 4.3) at either Rate 1 or Rate 1/2 as specified by the corresponding line in Table 2.2.2.2.2-2. Service Option 3 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 3 is initialized.

   Service Option 3 may use the following procedure to perform this rate reduction: Sequences of *N* packets as are formed as shown in Table 2.2.2.2.1-1. 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 4.3) 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 2.1.1).

Table 2.2.2.2.1-1. 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 2.2.2.2.1-1, the mobile station shall reject the message by sending a *Mobile Station Reject Order* with the ORDQ field set equal to '00000100'.

2.2.2.2.2 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 3:

Table 2.2.2.2.2-1. 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 2.2.2.2.1-1 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 2.2.2.2.1), 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 the MOBILE_TO_MOBILE field to '0'.

INIT_CODEC - Initialize speech codec.

If the mobile station is to initialize the speech codec (see Sections 4 and 5), the base station shall set this field to '1'. Otherwise, the base station shall set this field to '0'.

Table 2.2.2.2.2-2. 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.		

3 AUDIO INTERFACES

3.1 Input Audio Interface

3.1.1 Input Audio Interface in the Mobile Station

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

3.1.1.1 Conversion and Scaling

Whether the input is analog or digital, the signal presented to the input of the speech codec shall be sampled at a rate of 8000 samples per second and shall be quantized to a uniform PCM format with at least 13 bits of dynamic range.

The quantities in this standard assume a 16-bit integer input normalization with a range from -32,768 through +32,767. The following speech codec discussion assumes this 16-bit integer normalization. If an input audio interface uses a different normalization scheme, then appropriate scaling should be used.

3.1.1.2 Digital Audio Input

If the input audio is an 8-bit μ-Law/A-Law 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.

3.1.1.3 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 3.1.1.1).

3.1.1.3.1 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. 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 3GPP2 C.S0018-0-1.

3.1.1.3.2 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.

3.1.1.3.3 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 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 3GPP2 C.S0018-0-1.

3.1.2 Input Audio Interface in the Base Station

3.1.2.1 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 3.1.1.1.

3.1.2.2 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 3GPP2 C.S0018-0-1.

3.1.2.3 Line Echo Canceling

The base station shall provide a method to cancel echoes returned by the PSTN interface.[†] 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; however, the latter requirement is subject to local PSTN configuration demands, i.e., a 64 ms (or greater) echo canceling capability may be required in cases where the PSTN does not provide echo cancellation for long distance service.

3.2 Output Audio Interface

3.2.1 Output Audio Interface in the Mobile Station

3.2.1.1 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.

3.2.1.2 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. 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 3GPP2 C.S0018-0-1.

3.2.2 Output Audio Interface in the Base Station

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

[†] 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.

3.2.2.1 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 3GPP2 C.S0018-0-1.

4 SPEECH ENCODER

The Enhanced Variable Rate Codec (EVRC) is based upon the RCELP algorithm, appropriately modified for variable rate operation and for robustness in the CDMA environment. RCELP is a generalization of the Code-Excited Linear Prediction (CELP) algorithm. Unlike conventional CELP encoders, RCELP does not attempt to match the original speech signal exactly. Instead of attempting to match the original residual signal, RCELP matches a time-warped version of the original residual that conforms to a simplified pitch contour. The pitch contour is obtained by estimating the pitch delay once in each frame and linearly interpolating the pitch from frame to frame. One benefit of using this simplified pitch representation is that more bits are available in each packet for the stochastic excitation and for channel impairment protection than would be if a traditional fractional pitch approach were used. This results in enhanced error performance without impacting perceived speech quality in clear channel conditions.

The encoder uses 3 of the 4 primary traffic packet types permitted by IS-95 Multiplex Option 1: Rate 1 (171 bits/packet), Rate 1/2 (80 bits/packet), and Rate 1/8 (16 bits/packet). Upon command, the encoder will produce a blank packet (which contains no bits) or other than a Rate 1 packet (i.e., Rate 1/2 maximum); otherwise, the encoder makes its own determination about what type of packet to generate. Bit allocations for each packet type are given in Table 4-1.

Table 4-1. Bit Allocations by Packet Type

Field	**Packet Type**			
(see 4.8)	**Rate 1**	**Rate 1/2**	**Rate 1/8**	**Blank**
Spectral Transition Indicator	1			
LSP	28	22	8	
Pitch Delay	7	7		
Delta Delay	5			
ACB Gain	9	9		
FCB Shape	105	30		
FCB Gain	15	12		
Frame Energy			8	
(reserved)	1			
Total	171	80	16	0

The algorithms in the sections that follow are presented in terms of block diagrams, mathematical equations where appropriate, and in pseudo-code where equations would be cumbersome. Inputs, outputs, and processing are defined for each processing block or module. Internal variables for each module are defined when used. Modules are decomposed into sub-modules at increasing levels of detail until a defining equation or block of pseudo-code describing the lowest level of processing is reached. The algorithm descriptions are designed to maximize clarity, not to illustrate the most efficient implementation. For example, no attempt is made to illustrate memory conserving techniques such as reuse of scratch variables or buffers. Operations such as Discrete Fourier Transforms (DFTs) or Finite Impulse Response (FIR) filters are described in terms of a defining equation rather than in terms of one of the fast algorithms available to carry them out.

Figure 4-1 is a high-level view of the EVRC speech encoder showing all major modules. Inputs to the encoder are the speech signal vector, $\{s(n)\}$, and an external rate command signal. The external rate command may direct the encoder to produce a blank packet or a packet other than Rate 1. If an external rate command is received, it will supersede the encoder's internal rate selection mechanism.

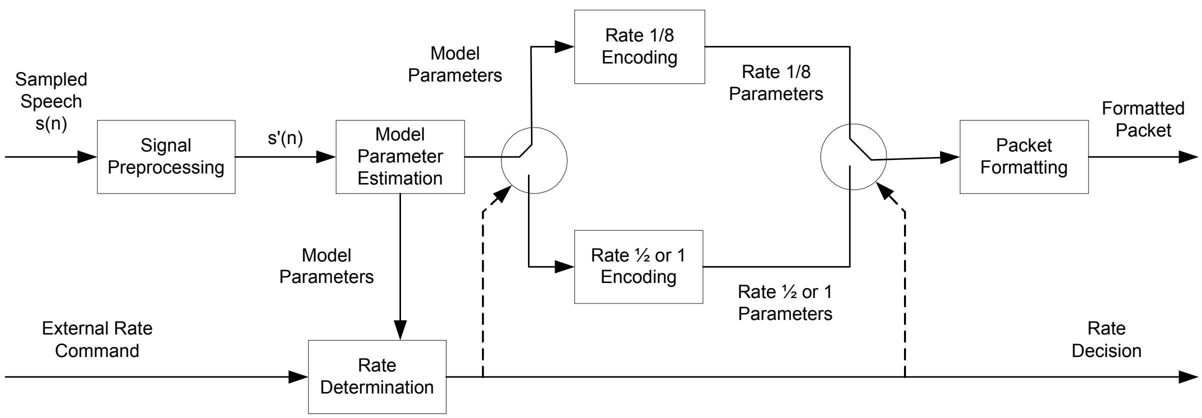

Figure 4-1. Speech Encoder Top-Level Diagram

The input speech vector, $\{s(n)\}$, is presented to the signal pre-processing module (see 4.1), which performs high-pass and adaptive noise suppression filtering. The pre-processed speech vector, $\{s'(n)\}$, is then presented to the model parameter estimation module. The model parameter estimation module (see 4.2) performs LPC analysis to determine a set of linear prediction coefficients (LPCs) and the optimal pitch delay (τ). It also converts the LPCs to line spectral pairs (LSPs) and calculates the long and short-term prediction gains. The rate determination module (see 4.3) applies a voice activity detection (VAD) algorithm and rate selection logic in order to determine the type of packet to generate.

After estimating the model parameters, the encoder will characterize the excitation signal and quantize parameters in a way appropriate to the selected rate (see 4.4 through 4.6). If Rate 1/8 is selected, the encoder will not attempt to characterize any periodicity in the speech residual, but will instead just characterize its energy contour. At Rates 1/2 and 1 the encoder will apply the RCELP algorithm to match a time-warped version of the original speech residual.

A number of parameters are included at Rate 1 to provide enhanced performance in poor channel conditions. These include the spectral transition indicator and the delay difference (Δ_{τ}). The packet formatting module (see 4.8) accepts all of the parameters calculated and quantized in the rate-specific encoding modules, and formats a packet appropriate to the selected rate. The formatted packet is then presented to the multiplex sub-layer. The rate decision is also presented to the multiplex sub-layer. Sections 4.1 through 4.8 describe each of the encoder modules in detail.

4.1 Input Signal Preprocessing

Input preprocessing is needed to condition the input signal against excessive low frequency and other background noises that can degrade the codec voice quality. Refer to 1.1 for guidelines concerning potential variations of the Input Signal Preprocessing functions.

4.1.1 High-Pass Filter

The input sampled speech shall be high-pass filtered as described below:

Inputs:

- The input sampled speech signal, $\{s(n)\}$

Outputs:

- The high-pass filtered speech signal, $\{s_{hp}(n)\}$

Initialization:

- The filter memory is set to all zeros at initialization.

Processing: The High-Pass Filter is a sixth order Butterworth filter implemented as three cascaded biquadratic sections. The cutoff frequency of the filter is 120 Hz. The transfer function of the filter is:

$$H_{hpf} = \prod_{j=1}^{3} H_j(z), \tag{4.1.1-1}$$

where each section, $H_j(z)$, is given by:

$$H_j(z) = \frac{a_{j0} + a_{j1}z^{-1} + a_{j2}z^{-2}}{1 + b_{j1}z^{-1} + b_{j2}z^{-2}}. \tag{4.1.1-2}$$

The filter coefficients are given as:

$a_{10} = 1.0$, $a_{11} = -2.000125721$, $a_{12} = 1.000125737$,
$b_{11} = -1.943779252$, $b_{12} = 0.952444269$,

$a_{20} = 1.0$, $a_{21} = -1.999873569$, $a_{22} = 0.999873585$,
$b_{21} = -1.866892280$, $b_{22} = 0.875214548$,

$a_{30} = 0.833469450$, $a_{31} = -1.666939491$, $a_{32} = 0.833470028$,
$b_{31} = -1.825209384$, $b_{32} = 0.833345838$.

4.1.2 Noise Suppression

Noise Suppression is used to improve the signal quality that is presented to the Model Parameter Estimator. The procedures by which the Noise Suppression shall be implemented are described in 4.1.2.1 to 4.1.2.11.

Input:

- The output of the High-Pass Filter, $\{s_{hp}(n)\}$

Output:

- The output of the Noise Suppressor is designated as $\{s'(n)\}$.

Initialization:

The following variables shall be set to zero at initialization (frame $m = 0$):

- The overlapped portion of the input frame buffer, $\{d(m)\}$
- The pre-emphasis and de-emphasis memories
- The overlap-and-add buffer, $h(n)$
- The output buffer history, $s'(n)$; $0 \le n < 160$

The following shall be initialized to a startup value other than zero:

- The channel energy estimate, $\boldsymbol{E}_{ch}(m)$, (see 4.1.2.2)
- The long-term power spectral estimate, $\overline{\mathbf{E}}_{dB}(m)$, (see 4.1.2.5)
- The channel noise estimate, $\boldsymbol{E}_n(m)$, (see 4.1.2.10)

Processing: Although the frame size of the speech codec is 20 ms, the Noise Suppressor frame size is 10 ms. Therefore, the following procedures shall be executed two times per 20 ms speech frame and the current 10 ms frame shall be denoted m. Figure 4.1.2-1 depicts the Noise Suppression system that is described in the following sections.

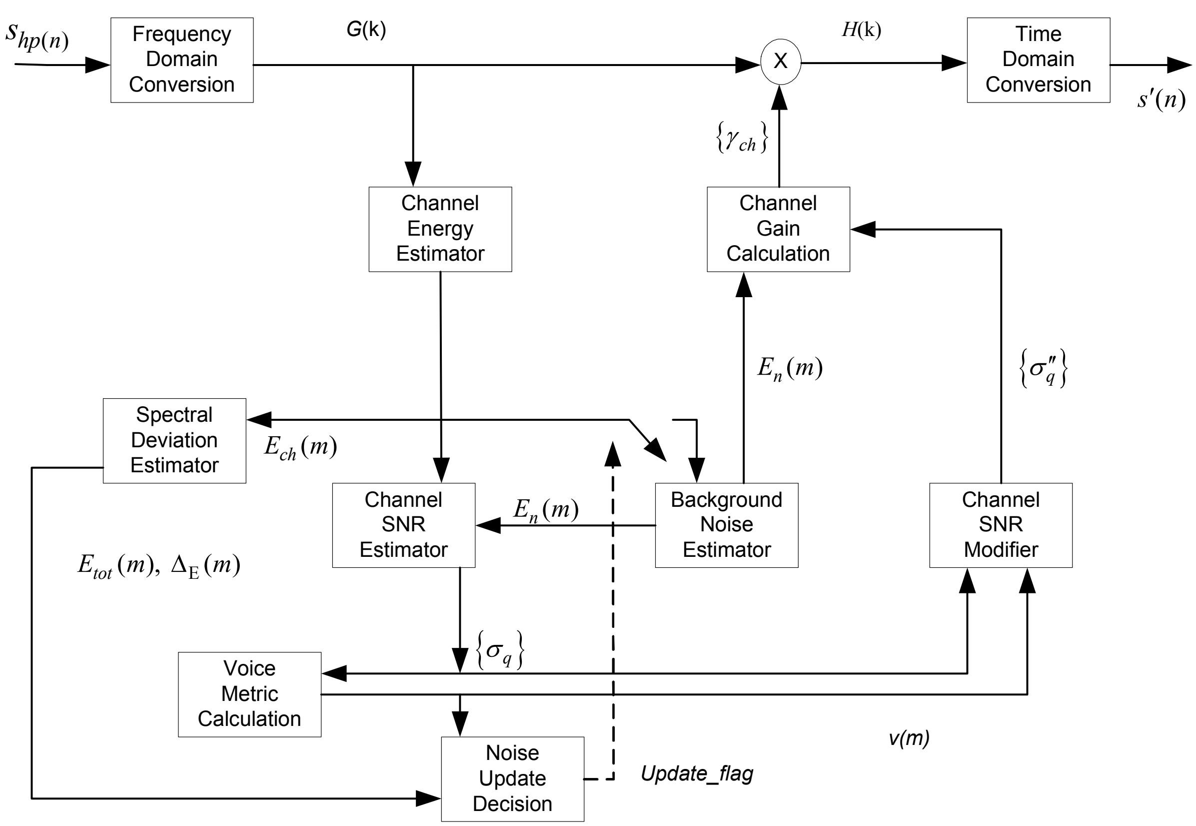

Figure 4.1. 2-1 Noise Suppression Block Diagram

4.1.2.1 Frequency Domain Conversion

The input signal is windowed using a smoothed trapezoid window, in which the first D samples of the input frame buffer $\{d(m)\}$ are overlapped from the last D samples of the previous frame. This overlap is described as:

$$d(m,n) + d(m-1, L+n); \qquad 0 \le n < D, \tag{4.1.2.1-1}$$

where m is the current frame, n is the sample index to the buffer $\{d(m)\}$, L = 80 is the frame length, and D = 24 is the overlap (or delay) in samples.

The remaining samples of the input buffer are then pre-emphasized according to the following:

$$d(m, D+n) = s_{hp}(n) + \zeta_p s_{hp}(n-1); \qquad 0 \le n < L, \tag{4.1.2.1-2}$$

where ζ_p = -0.8 is the pre-emphasis factor. This results in the input buffer containing $L + D$ = 104 samples in which the first D samples are the pre-emphasized overlap from the previous frame, and the following L samples are pre-emphasized input from the current frame.

Next, a smoothed trapezoidal window is applied to the input buffer to form the DFT data buffer, $\{g(n)\}$, defined as:

$$g(n) = \begin{cases} d(m,n)\sin^2(\pi(n+0.5)/2D) & ; 0 \le n < D, \\ d(m,n) & ; D \le n < L, \\ d(m,n)\sin^2(\pi(n-L+D+0.5)/2D) & ; L \le n < D+L, \\ 0 & ; D+L \le n < M, \end{cases} \tag{4.1.2.1-3}$$

where M = 128 is the DFT sequence length and all other terms are previously defined.

The transformation of $g(n)$ to the frequency domain is performed using the Discrete Fourier Transform (DFT) defined† as:

$$G(k) = \frac{2}{M} \sum_{n=0}^{M-1} g(n) e^{-j2\pi nk/M}; \qquad 0 \le k < M, \tag{4.1.2.1-4}$$

where $e^{j\omega}$ is a unit amplitude complex phasor with instantaneous radial position ω.

4.1.2.2 Channel Energy Estimator

Calculate the channel energy estimate $\boldsymbol{E}_{ch}(m)$ for the current frame, m, as:

---

† This atypical definition is used to exploit the efficiencies of the complex Fast Fourier Transform (FFT). The $2/M$ scale factor results from preconditioning the M point real sequence to form an $M/2$ point complex sequence that is transformed using an $M/2$ point complex FFT. Details on this technique can be found in Proakis, J. G. and Manolakis, D. G., *Introduction to Digital Signal Processing*, New York, Macmillan, 1988, pp. 721-722.

$$E_{ch}(m,i) = \max\left\{E_{\min}, \alpha_{ch}(m)E_{ch}(m-1,i) + (1-\alpha_{ch}(m))\frac{1}{f_H(i)-f_L(i)+1}\sum_{k=f_L(i)}^{f_H(i)}\left|G(k)\right|^2\right\} \quad ; 0 \le i < N_c, \tag{4.1.2.2-1}$$

where $E_{\min}$ = 0.0625 is the minimum allowable channel energy, $\alpha_{ch}(m)$ is the channel energy smoothing factor (defined below), N_c = 16 is the number of combined channels, and $f_L(i)$ and $f_H(i)$ are the i-th elements of the respective low and high channel combining tables, which are defined as:

$$\boldsymbol{f}_L = \{ 2, 4, 6, 8, 10, 12, 14, 17, 20, 23, 27, 31, 36, 42, 49, 56 \},$$

$$\boldsymbol{f}_H = \{ 3, 5, 7, 9, 11, 13, 16, 19, 22, 26, 30, 35, 41, 48, 55, 63 \}.$$

The channel energy smoothing factor, $\alpha_{ch}(m)$, is defined as:

$$\alpha_{ch}(m) = \begin{cases} 0 & ; m \le 1, \\ 0.45 & ; m > 1. \end{cases} \tag{4.1.2.2-2}$$

So, this means that $\alpha_{ch}(m)$ assumes a value of zero for the first frame (m = 1) and a value of 0.45 for all subsequent frames. This allows the channel energy estimate to be initialized to the unfiltered channel energy of the first frame.

4.1.2.3 Channel SNR Estimator

Estimate the quantized channel SNR indices as:

$$\sigma_q(i) = \max\left\{0, \min\left\{89, \text{round}\left\{10\log_{10}\left(\frac{E_{ch}(m,i)}{E_n(m,i)}\right)/0.375\right\}\right\}\right\}; \quad 0 \le i < N_c, \tag{4.1.2.3-1}$$

where $\mathbf{E}_n(m)$ is the current channel noise energy estimate (see 4.1.2.10), and the values of $\{ \sigma_q \}$ are constrained to be between 0 and 89, inclusive.

4.1.2.4 Voice Metric Calculation

Next, calculate the sum of voice metrics as:

$$v(m) = \sum_{i=0}^{N_c-1} V\left(\sigma_q(i)\right), \tag{4.1.2.4-1}$$

where $V(k)$ is the k^{th} value of the 90 element voice metric table $\boldsymbol{V}$, that is defined as:

$\boldsymbol{V}$ = { 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 7, 7, 7, 8, 8, 9, 9, 10, 10, 11, 12, 12, 13, 13, 14, 15, 15, 16, 17, 17, 18, 19, 20, 20, 21, 22, 23, 24, 24, 25, 26, 27, 28, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50 }.

4.1.2.5 Spectral Deviation Estimator

Calculate the estimated log power spectrum as:

$$E_{dB}(m,i) = 10\log_{10}\left(E_{ch}(m,i)\right) \quad ; \qquad 0 \le i < N_c \ . \tag{4.1.2.5-1}$$

Then, calculate the estimated spectral deviation between the current power spectrum and the average long-term power spectral estimate:

$$\Delta_E(m) = \sum_{i=0}^{N_c-1} \left| E_{dB}(m,i) - \bar{E}_{dB}(m,i) \right| , \tag{4.1.2.5-2}$$

where $\bar{\mathbf{E}}_{dB}(m)$ is the average long-term power spectral estimate calculated during the previous frame, as defined in Equation 4.1.2.5-7. The initial value of $\bar{E}_{dB}(m)$, however, is defined to be the estimated log power spectrum of frame 1, or:

$$\bar{E}_{dB}(m) = E_{dB}(m); \qquad \text{m=1}. \tag{4.1.2.5-3}$$

Calculate the total channel energy estimate, $E_{tot}(m)$, for the current frame, m, according to the following:

$$E_{tot}(m) = 10\log_{10}\left(\sum_{i=0}^{N_c-1} E_{ch}(m,i) \right). \tag{4.1.2.5-4}$$

Calculate the exponential windowing factor, $\alpha(m)$, as a function of total channel energy, $E_{tot}(m)$, as:

$$\alpha(m) = \alpha_H - \left(\frac{\alpha_H - \alpha_L}{E_H - E_L} \right)\left(E_H - E_{tot}(m) \right), \tag{4.1.2.5-5}$$

and then limit the result to be between α_H and α_L by

$$\alpha(m) = \max\left\{ \alpha_L, \min\left\{ \alpha_H, \alpha(m) \right\} \right\}, \tag{4.1.2.5-6}$$

where E_H and E_L are the energy endpoints (in dB) for the linear interpolation of $E_{tot}(m)$, that is transformed to $\alpha(m)$ which has the limits $\alpha_L \le \alpha(m) \le \alpha_H$. The values of these constants are defined as: $E_H = 50$, $E_L = 30$, $\alpha_H = 0.99$, $\alpha_L = 0.50$. As an example, a signal with relative energy of 40 dB would use an exponential windowing factor of $\alpha(m) = 0.745$ for the following calculation.

Update the average long-term power spectral estimate for the next frame by:

$$\bar{E}_{dB}(m+1,i) = \alpha(m)\bar{E}_{dB}(m,i) + \left(1 - \alpha(m)\right) E_{dB}(m,i) \, ; \qquad 0 \le i < N_c , \tag{4.1.2.5-7}$$

where all the variables are previously defined.

4.1.2.6 Background Noise Update Decision

The following logic, as shown in pseudo-code, demonstrates how the noise estimate update decision is ultimately made:

```
/* Normal update logic */
update_flag = FALSE
if ( v(m) ≤ UPDATE_THLD ) {
    update_flag = TRUE
    update_cnt = 0
}
/* Forced update logic */
else if (( E_tot(m) > NOISE_FLOOR_DB ) and ( Δ_E(m) < DEV_THLD )) {
    update_cnt = update_cnt + 1
    if ( update_cnt ≥UPDATE_CNT_THLD )
        update_flag = TRUE
}
/* "Hysteresis" logic to prevent long-term creeping of update_cnt */
if ( update_cnt == last_update_cnt )
    hyster_cnt = hyster_cnt + 1
else
    hyster_cnt = 0
last_update_cnt = update_cnt
if ( hyster_cnt > HYSTER_CNT_THLD )
    update_cnt = 0
```

The values of the previously used constants are UPDATE_THLD = 35, NOISE_FLOOR_DB = $10\log_{10}(E_{floor})$ (see 4.1.2.8), DEV_THLD = 28, UPDATE_CNT_THLD = 50, HYSTER_CNT_THLD = 6.

4.1.2.6 SNR Estimate Modification

Next, determine whether the channel SNR modification should take place, then proceed to modify the appropriate SNR indices:

/* Set or reset modify flag */
$index\_cnt = 0$
for ($i = N_M$ to N_c - 1 step 1) {
    if ($\sigma_q(i) \geq$ INDEX_THLD)
        $index\_cnt = index\_cnt + 1$
}
if ($index\_cnt <$ INDEX_CNT_THLD)
    $modify\_flag$ = TRUE
else
    $modify\_flag$ = FALSE

/* Modify the SNR indices to get $\{\sigma'_q\}$ */
if ($modify\_flag$ == TRUE)
    for ($i = 0$ to N_c - 1 step 1)
        if (($v(m) \leq$ METRIC_THLD) or ($\sigma_q(i) \leq$ SETBACK_THLD))
            $\sigma'_q(i) = 1$
        else
            $\sigma'_q(i) = \sigma_q(i)$
else
    $\{\sigma'_q\} = \{\sigma_q\}$

/* Limit $\{\sigma''_q(i)\}$ to SNR threshold σ_{th} */
for ($i = 0$ to N_c - 1 step 1)
    if $\left(\sigma'_q(i) < \sigma_{th}\right)$
        $\sigma''_q(i) = \sigma_{th}$
    else
        $\sigma''_q(i) = \sigma'_q(i)$

The previous constants and thresholds are given to be: $N_M = 5$, INDEX_THLD = 12, INDEX_CNT_THLD = 5, METRIC_THLD = 45, SETBACK_THLD = 12, $\sigma_{th} = 6$.

4.1.2.7 Channel Gain Computation

Compute the overall gain factor for the current frame as:

$$\gamma_n = \max\left\{\gamma_{\min}, -10\log_{10}\left\{\frac{1}{E_{floor}}\sum_{i=0}^{N_c-1} E_n(m,i)\right\}\right\}, \tag{4.1.2.8-1}$$

where $\gamma_{\min}$ = -13 is the minimum overall gain, E_{floor} = 1 is the noise floor energy, and $\boldsymbol{E}_n(m)$ is the estimated noise spectrum (see 4.1.2.10) calculated during the previous frame.

Next, calculate channel gains (in dB) as:

$$\gamma_{dB}(i) = \mu_g \left(\sigma_q''(i) - \sigma_{th}\right) + \gamma_n\,; \qquad 0 \le i < N_c\,, \tag{4.1.2.8-2}$$

where μ_g = 0.39 is the gain slope, and then convert to linear channel gains:

$$\gamma_{ch}(i) = \min\left\{1, 10^{\gamma_{dB}(i)/20}\right\}; \qquad 0 \le i < N_c\,, \tag{4.1.2.8-3}$$

4.1.2.8 Frequency Domain Filtering

Now, apply the channel gains to the transformed input signal $G(k)$:

$$H(k) = \begin{cases} \gamma_{ch}(i)G(k) & ; f_L(i) \le k \le f_H(i),\ 0 \le i < N_c, \\ G(k) & ; 0 \le k < f_L(0),\ f_H(N_c - 1) < k \le M/2, \end{cases} \tag{4.1.2.9-1}$$

where the bottom part of the equation represents the frequencies that are not altered by the channel gains. But, it is also required that the magnitude of $H(k)$ be even, and the phase be odd, so that the following condition is also imposed:

$$H(M-k) = H^*(k); \quad 0 < k < M/2\,, \tag{4.1.2.9-2}$$

where $*$ denotes complex conjugate. This guarantees that the imaginary part of the inverse DFT of $H(k)$ will be zero (in Equation 4.1.2.11-1).

4.1.2.9 Background Noise Estimate Update

If (and only if) the update flag is set (*update_flag* == TRUE), then update the channel noise estimate for the next frame by:

$$E_n(m+1,i) = \max\left\{E_{\min}, \alpha_n E_n(m,i) + (1-\alpha_n)E_{ch}(m,i)\right\}; \quad 0 \le i < N_c\,, \tag{4.1.2.10-1}$$

where $E_{\min}$ = 0.0625 is the minimum allowable channel energy, and σ_n = 0.9 is the channel noise smoothing factor. The channel noise estimate shall be initialized for each of the first four frames to the estimated channel energy, i.e.:

$$E_n(m,i) = \max\left\{E_{init}, E_{ch}(m,i)\right\} \quad ; 1 \le m \le 4,\ 0 \le i < N_c\,, \tag{4.1.2.10-2}$$

where E_{init} = 16 is the minimum allowable channel noise initialization energy.

4.1.2.10 Time Domain Signal Reconstruction

Convert the filtered signal to the time domain using the inverse DFT:

$$h(m,n) = \frac{1}{2}\sum_{k=0}^{M-1} H(k)e^{j2\pi nk/M}; \quad 0 \le n < M \quad . \tag{4.1.2.11-1}$$

Complete the frequency domain filtering process by applying overlap-and-add:

$$h'(n) = \begin{cases} h(m,n) + h(m-1, n+L) & ; \quad 0 \le n < M - L, \\ h(m,n) & ; \quad M - L \le n < L. \end{cases} \tag{4.1.2.11-2}$$

Finally, apply signal de-emphasis by:

$$s'(n+240) = h'(n) + \zeta_d s'(n+239); \quad 0 \le n < L\,, \tag{4.1.2.11-3}$$

where ζ_d = 0.8 is the de-emphasis factor and $\{s'(n)\}$ is the 320 element output buffer. Since the 10 ms per frame Noise Suppression is performed twice per 20 ms speech frame, the output presented to Model Parameter Estimation comprises $s'(n)$; $160 \le n < 320$.

4.2 Model Parameter Estimation

Model Parameter Estimation comprises the linear predictive analysis, residual calculation and long-term prediction that must be performed for each frame, independent of the rate decision.

Inputs: The inputs to model parameter estimation are:

- The current 20 ms frame number, m
- The pre-processed speech output vector, $\{s'(n)\}$
- The unquantized line spectral pairs, $\Omega(m\text{-}1)$, calculated in the previous frame
- The LPC prediction gain, $\gamma_{lpc}(m\text{-}1)$, calculated in the previous frame

The pre-processed speech buffer, $\{s'(n)\}$, contains 320 pre-processed speech samples, i.e., $0 \le n < 320$. Samples 0 through 79 are "lookback" from the previous frame, samples 80 through 239 are the current frame, and samples 240 through 319 are "lookahead" to the next frame. The last 160 samples from the preprocessing modules, then, constitute 80 samples for the last half of the "current" frame and 80 samples "lookahead" to the next frame.

Outputs: The outputs of model parameter estimation are:

- The unquantized linear predictive coefficients for the current frame, $\{a\}$
- The unquantized LSPs for the current frame, $\Omega(m)$
- The LPC prediction gain, $\gamma_{lpc}(m)$
- The prediction residual, $\{\varepsilon(n)\}$
- The long-term pitch delay estimate, τ
- The long-term prediction gain, β
- The spectral transition indicator, *LPCFLAG*
- The bandwidth expanded correlation coefficients, $\boldsymbol{R}_w$

Processing: The model parameter estimation module performs three major functions:

- Calculation of the formant filter parameters and prediction gain
- Generation of the short-term prediction residual signal
- Calculation of the delay estimate and long-term prediction gain

In addition to these functions, this module also generates a spectral transition indicator, which is used to improve channel impairment performance. The following sections describe each of the major model parameter estimation functions in detail.

4.2.1 Formant Filter Parameter Calculation

Inputs: The inputs to the formant filter parameter calculation are:

- The pre-processed speech input signal, $\{s'(n)\}$
- The LPC prediction gain from the previous frame, $\gamma_{lpc}(m\text{-}1)$

Outputs: The outputs of the formant filter parameter calculation are:

- The unquantized LPCs for the current frame $\{a\}$
- The unquantized LSPs for the current frame, $\Omega(m)$
- The LPC prediction gain, $\gamma_{lpc}(m)$
- The impulse response of the formant filter, $h(n)$
- The spectral transition indicator, *LPCFLAG*

Initialization:

- The LPC prediction gain, $\gamma_{lpc}(m) = 1$, for $m = 0$

Processing: The formant filter parameters are calculated from the pre-processed speech buffer, $\{s'(n)\}$, using the autocorrelation method and Durbin's Recursion. The details of the computation shall be executed as described in 4.2.1.1 through 4.2.1.3.

4.2.1.1 Direct Form LPC Parameter Calculation

The pre-processed speech buffer is windowed using a Hamming window centered at the end of the current frame, i.e., on sample 239. The equation for the Hamming window is:

$$W_H(k) = 0.54 - 0.46\cos\left(\frac{2\pi}{160}(k-160)\right); \quad 160 \le k < 320, \qquad (4.2.1.1\text{-}1)$$

and the expression for the windowed speech is:

$$s'_H(k-160) = W_H(k)s'(k); \quad 160 \le k < 320\,. \qquad (4.2.1.1\text{-}2)$$

The first 17 terms of the autocorrelation function of $\{s_H'(n)\}$, **R**, are calculated directly using:

$$R(k) = \sum_{i=0}^{159-k} s'_H(i)s'_H(i+k); \quad 0 \le k < 16. \qquad (4.2.1.1\text{-}3)$$

The first 11 are used for LPC analysis, while all 17 terms are used by the Rate Determination Algorithm (see 4.3). The autocorrelation terms are then spectrally smoothed by windowing as follows:

$$R_w(k) = \begin{cases} 1.00003R(k); & k = 0, \\ \exp\left[-\frac{1}{2}\left(\frac{40\pi k}{8000}\right)^2\right]R(k); & 1 \le k \le 16. \end{cases} \tag{4.2.1.1-4}$$

The unquantized, unweighted LPCs, {α} are then calculated from $\mathbf{R}_W$ using Durbin's Recursion†, as follows:

{

$E^{(0)} = R_w(0)$

$i = 1$

while ($i \le 10$)

{

$$k_i = \frac{1}{E^{(i-1)}}\left[R_w(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_w(i-j)\right]$$

$\alpha_i^{(i)} = k_w$

j = 1

while (j < i)

{

$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$

j = j + 1

}

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

i = i + 1

}

}

The LPC coefficients are $\alpha_j^{(10)}$, for $1 \le j \le 10$, or $\{\alpha\} = \{\alpha_1, \alpha_2, ..., \alpha_{10}\}$.

4.2.1.2 Generation of Spectral Transition Indicator (*LPCFLAG*)

The impulse response, $\{h_{raw}\}$, of the unweighted, unquantized formant filter is calculated to 54 terms. The unweighted, unquantized formant filter is given by:

$$\frac{1}{A_{raw}(z)} = \frac{1}{1 - \sum_{k=1}^{10} \alpha_k z^{-k}}, \tag{4.2.1.2-1}$$

† 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 *i*th stage.

where k is the LPC index. The energy of the raw impulse response is then calculated as an estimate of the LPC prediction gain. This is given by:

$$\gamma_{lpc}(m) = \sum_{k=0}^{53} h_{raw}^2(k)\,. \quad (4.2.1.2\text{-}2)$$

If the ratio $\gamma_{lpc}(m)/\gamma_{lpc}(m\text{-}1) > 10$ (where $\gamma_{lpc}(m\text{-}1)$ is the LPC prediction gain from the previous frame), then the spectral transition indicator, *LPCFLAG*, is set to 1, indicating that a large spectral transition has occurred. Otherwise, *LPCFLAG* is set to 0.

4.2.1.3 Direct Form LPC to LSP Conversion

The raw LPCs are bandwidth expanded using an exponential window:

$$\alpha_k = (0.994)^k \alpha_k \qquad ; 0 < k \le 10\,. \quad (4.2.1.3\text{-}1)$$

The unquantized, bandwidth expanded LPCs are then converted to LSPs. Here, $A(z)$ is given by:

$$A(z) = 1 - \sum_{k=1}^{10} \alpha_k z^{-k}\,, \quad (4.2.1.3\text{-}2)$$

where $\{a\}$ are the bandwidth expanded LPC coefficients.

Next, define $P_A(z)$ and $Q_A(z)$ as follows:

$$P_A(z) = A(z) + z^{-11}A(z^{-1}) = 1 + \sum_{i=1}^{5} p_i z^{-i} + \sum_{i=6}^{10} p_{11-i} z^{-i} + z^{-11}\,, \quad (4.2.1.3\text{-}3)$$

$$Q_A(z) = A(z) - z^{-11}A(z^{-1}) = 1 + \sum_{i=1}^{5} q_i z^{-i} - \sum_{i=6}^{10} q_{11-i} z^{-i} - z^{-11}\,, \quad (4.2.1.3\text{-}4)$$

where $p_i = -a_i - a_{11-i}$; $1 \le i \le 5$, and $q_i = -a_i + a_{11-i}$; $1 \le i \le 5$. The LSP frequencies are the ten roots which exist between $\omega = 0$ and $\omega = 0.5$ in the following two equations:

$$P'(\omega) = \cos 5(2\pi\omega) + p'_1 \cos 4(2\pi\omega) + \ldots + p'_4 \cos(2\pi\omega) + p'_5/2\,, \quad (4.2.1.3\text{-}5)$$

$$Q'(\omega) = \cos 5(2\pi\omega) + q'_1 \cos 4(2\pi\omega) + \ldots + q'_4 \cos(2\pi\omega) + q'_5/2\,, \quad (4.2.1.3\text{-}6)$$

where the p' and q' values are computed recursively as follows from the p and q values:

$$p'_0 = q'_0 = 1\,, \quad (4.2.1.3\text{-}7)$$

$$p'_i = p_i - p'_{i-1} \qquad 1 \le i \le 5\,, \quad (4.2.1.3\text{-}8)$$

$$q'_i = q_i + q'_{i-1} \qquad 1 \le i \le 5. \quad (4.2.1.3\text{-}9)$$

Since the formant synthesis (LPC) filter is stable, the roots of the two functions alternate; the smallest root, ω_1 is the lowest root of $P'(\omega)$, the next smallest root, ω_2 is the lowest root of $Q'(\omega)$, etc. Thus, ω_1, ω_3, ω_5, ω_7, and ω_9 are the roots of $P'(\omega)$, and ω_2, ω_4, ω_6, ω_8, and ω_{10} are the roots of $Q'(\omega)$. The corresponding unquantized LSP parameter vector is defined as:

$$\Omega(m) = \{ \omega_1, \omega_2, ..., \omega_{10} \} . \qquad (4.2.1.3\text{-}10)$$

4.2.2 Generation of the Short-Term Prediction Residual

Inputs: The inputs to the short-term prediction residual calculation are:

- The unquantized LSPs from the previous frame, $\Omega(m\text{-}1)$
- The unquantized LSPs calculated for the current frame, $\Omega(m)$
- The pre-processed speech input vector, $\{s'(n)\}$

Output: The output of short-term prediction residual calculation is:

- The residual vector, $\{\varepsilon(n)\}$

This vector contains 320 elements, partitioned in the same way as the pre-processed speech input vector. The first 80 samples represent the last 10 ms of the previous frame; the next 160 samples represent the current 20 ms frame, and the last 80 samples represent the 10 ms look-ahead.

Initialization: The unquantized LSPs, $\Omega(m,i) = 0.048i$; $m = 0$, $1 \le i \le 10$.

Processing: The short-term prediction residual signal, $\{\varepsilon(n)\}$, is generated by passing $\{s'(n)\}$ through the inverse filter created by interpolating between $\Omega(m\text{-}1)$ and $\Omega(m)$. The $\{s'(n)\}$ vector is divided into five segments, and a different set of interpolated LSPs is computed for each corresponding segment. The interpolated LSPs are converted to LPCs, and the appropriate segment of $\{s'(n)\}$ is convolved with the resulting filter to generate the corresponding samples of $\{\varepsilon(n)\}$. The initial state of the inverse filter shall be zero for each frame, m, i.e., this is a zero state filter. Also, throughout the following discussion, variables with a "dot" are implied to be interpolated, e.g., $\dot{\Omega}(m,k)$.

4.2.2.1 LSP Interpolation

For each segment k, the unquantized, interpolated LSP vector is:

$$\dot{\Omega}(m,k) = (1-\mu_k)\Omega(m-1) + \mu_k\Omega(m) \quad ; 0 \le k \le 4 \qquad (4.2.2.1\text{-}1)$$

where the interpolator constants, $\{\mu\}$, and their corresponding sets of sample indices for each segment of $\{s'(n)\}$ are given in Table 4.2.2.1-1.

Table 4.2.2.1-1. LSP Interpolation Constants

k	segment start sample	segment end sample	μ_k
0	0	79	0.0
1	80	132	0.1667

2	133	185	0.5
3	186	239	0.8333
4	240	319	1.0

4.2.2.2 LSP to Direct Form LPC Conversion

For each segment, $0 \le k \le 4$, the following process is executed. First, $\dot{P}_A(z)$ and $\dot{Q}_A(z)$ are computed from a set of interpolated LSP frequencies, $\dot{\Omega}(m,k)$, as follows:

$$\dot{P}_A(z) = \left(1+z^{-1}\right)\prod_{j=1}^{5}\left(1-2z^{-1}\cos\left(2\pi\,\dot{\omega}_{(2j-1)}\right)+z^{-2}\right), \qquad (4.2.2.2\text{-}1)$$

$$\dot{Q}_A(z) = \left(1-z^{-1}\right)\prod_{j=1}^{5}\left(1-2z^{-1}\cos\left(2\pi\,\dot{\omega}_{(2j)}\right)+z^{-2}\right), \qquad (4.2.2.2\text{-}2)$$

where $\left\{\dot{\omega}_1, \dot{\omega}_2, ..., \dot{\omega}_{10}\right\} = \dot{\Omega}(m,k)$, and the k is implied. These equations can also be expressed as:

$$\dot{P}_A(z) = 1+\sum_{i=1}^{5}\dot{p}_i\, z^{-i} + \sum_{i=6}^{10}\dot{p}_{11-i}\, z^{-i} + z^{-11}, \qquad (4.2.2.2\text{-}3)$$

$$\dot{Q}_A(z) = 1+\sum_{i=1}^{5}\dot{q}_i\, z^{-i} - \sum_{i=6}^{10}\dot{q}_{11-i}\, z^{-i} - z^{-11}. \qquad (4.2.2.2\text{-}4)$$

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

$$\dot{A}(z) = \frac{\dot{P}_A(z)+\dot{Q}_A(z)}{2}, \qquad (4.2.2.2\text{-}5)$$

$$= 1+\sum_{i=1}^{5}\frac{(\dot{p}_i+\dot{q}_i)}{2}z^{-i} + \sum_{i=6}^{10}\frac{(\dot{p}_{11-i}+\dot{q}_{11-i})}{2}z^{-i}, \qquad (4.2.2.2\text{-}6)$$

$$\dot{A}(z) = 1-\sum_{i=1}^{10}\dot{a}_i\, z^{-i}. \qquad (4.2.2.2\text{-}7)$$

So, the interpolated LPC coefficients for segment k of the current frame are given by:

$$\dot{a}_i(k) = \begin{cases} -\dfrac{\dot{p}_i + \dot{q}_i}{2} & ; 1 \le i \le 5 \\ -\dfrac{\dot{p}_{11-i} + \dot{q}_{11-i}}{2} & ; 6 \le i \le 10 \end{cases} \quad (4.2.2.2\text{-}8)$$

4.2.2.3 Generation of Residual Samples

The short-term prediction residual, $\varepsilon(n)$, is generated by passing $s'(n)$ through the inverse filter using the appropriate LPCs:

$$\varepsilon(n) = s'(n) - \sum_{i=1}^{10} \dot{a}_i(k) s'(n-i) , \quad (4.2.2.3\text{-}1)$$

where the value for k in $\dot{a}_i(k)$ is determined by the segment in which n lies as determined by the starting and ending sample numbers indicated in Table 4.1.2.1-1. All values of $s'(n)$ for $n < 0$ are zero.

4.2.3 Calculation of the Delay Estimate and Long-Term Prediction Gain

Inputs: The input to the delay estimate and long-term prediction gain calculation is:

- The short-term residual vector, $\{\varepsilon(n)\}$

Outputs: The outputs of the delay estimate and long-term prediction gain calculation are:

- The pitch delay estimate, τ
- The long-term prediction gain, β

Initialization: The values of all local state variables and buffers are set to zero at start-up.

Processing: The pitch delay shall be calculated using the method described in this section. The pitch delay is the delay that maximizes the autocorrelation function of the short-term prediction residual signal, subject to certain constraints. This calculation is carried out independently over two estimation windows. The first of these comprises the entire current frame; the second comprises the second half of the current frame and the look-ahead. Rules are then applied to combine the delay estimates and gains for the two estimation windows. The constrained search for the optimal delay in each window shall be carried out as follows:

The processes in 4.2.3.1 through 4.2.3.3 shall be performed once for each of the two estimation windows of each frame.

4.2.3.1 Non-exhaustive Open Loop Delay Search

The residual signal, $\{\varepsilon(n)\}$, is filtered and decimated by a factor of four to generate the decimated residual signal, $\{\varepsilon_d(n)\}$, by applying:

$$x(n) = \varepsilon(n + n_{start}) + 2.2875x(n-1) - 1.956x(n-2) + 0.5959x(n-3) ; \quad 0 \le n < 80, \quad (4.2.3.1\text{-}1)$$

then:

$$\varepsilon_d(n') = \varepsilon_d(n'+20) \quad , \quad 0 \le n' < 20 , \quad (4.2.3.1\text{-}2)$$

and finally:

$$\varepsilon_d(n') = x(4n'-77) - 0.312[x(4n'-78) + x(4n'-79)] + x(4n'-80) , \quad 20 \le n' < 40, \quad (4.2.3.1\text{-}3)$$

where $\{x(n)\}$ is the decimator filter memory, $\{\varepsilon_d(n)\}$ is the decimated residual buffer, and n_{start} is 160 for the first estimation window and 240 for the second estimation window. The autocorrelation function of this decimated residual signal is generated using:

$$r(d) = \sum_{k=0}^{40-d} \varepsilon_d(k)\varepsilon_d(k+d) \qquad ; 5 \le \mathrm{d} \le 30, \tag{4.2.3.1-4}$$

and the delay, d_{max}, corresponding to the maximum positive correlation, r_{max}, is found. The following operations are then performed to determine whether to use d_{max} as the decimated delay estimate, or to use an estimate in the neighborhood of the smoothed delay estimate, $\tilde{\tau}$, instead:

```
if (τ̃ ≠ 0 AND |τ̃ − 4d_max| > 2) {
    if(r'_max > 0.835 r_max) {
        d_max = d'_max
    }
}
```

where d'_{max} is calculated as the delay corresponding to the maximum positive value of $\{r(d)\}$ in the range: $\max\{5, \lfloor \tilde{\tau}/4 \rfloor - 2\} \le d \le \min\{30, 4 + \max\{5, \lfloor \tilde{\tau}/4 \rfloor - 2\}\}$, and $\tilde{\tau}$ is updated per Equation 4.2.3.3-4.

The optimal delay estimate, D_{max}, is calculated as the index corresponding to R_{max}, which is the maximum positive value of:

$$R(D) = \sum_{n=0}^{159-D} \varepsilon(n+n_0)\varepsilon(n+n_0+D) \quad ; \max\{20, 4d_{\max}-3\} \le D \le \min\{120, 4d_{\max}+3\}, \tag{4.2.3.1-5}$$

where n_0 = 80 and 160 for the respective first and second estimation windows.

4.2.3.2 Long-Term Prediction Gain Calculation

The energy of the undecimated residual is calculated as follows:

$$R_\varepsilon(d) = \sqrt{\sum_{i=0}^{159-D_{\max}} \varepsilon^2(i+n_0) \sum_{j=0}^{159-D_{\max}} \varepsilon^2(j+n_0+D_{\max})}, \tag{4.2.3.2-1}$$

from which the long-term prediction gain can be derived by:

$$\beta = \max\left\{0, \min\left\{1, \frac{R_{\max}}{R_\varepsilon(0)}\right\}\right\}. \tag{4.2.3.2-2}$$

4.2.3.3 Smoothed Delay Estimate and LTP Gain

The following operations are performed to determine whether to replace the values of β and D_{max} calculated in 4.2.3.1 and 4.2.3.2 by values obtained in the neighborhood of the smoothed delay estimate, $\tilde{\tau}$:

```
if( τ̃ > 0 ) {
    if(   (D_max > min{120, τ̃ + 6}   AND β' > 0.6β) OR
          (D_max < min{20, τ̃ − 6}  AND  (β' > 1.2β) ) {
              β = β'
```

$D_{max} = D'_{max}$

}

}

where D'_{max} is the index of R'_{max}, the maximum positive value of:

$$R'(D) = \sum_{n=0}^{159-D} \varepsilon(n+n_0)\varepsilon(n+n_0+D) \quad ; \max\left\{20, \tilde{\tau}-6\right\} \le D \le \min\left\{120, \tilde{\tau}+6\right\} \qquad (4.2.3.3\text{-}1)$$

and:

$$\beta' = \max\left\{0, \min\left\{1, \frac{R'_{\max}}{\sum_{i=0}^{159-D'_{\max}} \varepsilon^2(i+n_0) \sum_{j=0}^{159-D'_{\max}} \varepsilon^2(j+n_0+D'_{\max})}\right\}\right\}. \qquad (4.2.3.3\text{-}2)$$

The smoothed delay and long-term prediction gain estimates are updated as follows:

$$\tilde{\beta} = \begin{cases} \beta & ; \beta > 0.4, \\ 0.75\tilde{\beta} & ; \text{otherwise}, \end{cases} \qquad (4.2.3.3\text{-}3)$$

then:

$$\tilde{\tau} = \begin{cases} \mathrm{D}_{\max} & ; \beta > 0.4, \\ \tilde{\tau} & ; \beta \le 0.4 \text{ AND } \tilde{\beta} \ge 0.3, \\ 0 & ; \text{otherwise} \end{cases} \qquad (4.2.3.3\text{-}4)$$

4.2.3.4 Composite Delay and Gain Calculations

The delay estimates and long-term prediction gains for the two estimation windows are combined as follows: Let (D_0, β_0) be the optimal delay and gain found for the window comprising the current frame, and let (D_1, β_1) be the optimal delay and gain found for the window comprising the second half of the current frame and the look-ahead. Then:

if ($\beta_0 > \beta_1 + 0.4$) {
    if ($|D_0 - D_1| > 15$) {
        $\tau = D_0$
        $\beta = \beta_0$
    }
    else {
        $\tau = (D_0 + D_1) / 2$
        $\beta = (\beta_0 + \beta_1) / 2$
    }
}
else {
    $\tau = D_1$

$\beta = \beta_1$

}

4.3 Determining the Data Rate

The rate determination algorithm (RDA) is used to select one of three encoding rates: Rate 1, Rate 1/2, and Rate 1/8. Active speech is encoded at Rate 1 or Rate 1/2, and background noise is encoded at Rate 1/8. The actual data rate used for encoding the input signal may be modified to be Rate 1/2 maximum for the purpose of inserting a signaling message (see 2.2.1). Refer to 1.1 for guidelines concerning potential variations of the RDA.

Inputs:

- The bandwidth expanded correlation coefficients, $\boldsymbol{R}_W$
- The long-term prediction gain, β

Outputs:

- The speech encoder rate for the current frame, $Rate(m)$

Initialization:

- The rate histories, $Rate(m\text{-}1)$ and $Rate(m\text{-}2)$, are set to Rate 1/8 frames.
- The band energy, $E^{sm}_{f(i)}(m)$, initialization is given in 4.3.2.1.
- The background noise estimate, $B_{f(i)}(m)$, initialization is given in 4.3.2.2.
- The signal energy estimate, $S_{f(i)}(m)$, initialization is given in 4.3.2.3.

Processing:

- The data rate shall be determined by the processing steps defined in 4.3.1 and 4.3.2.

4.3.1 Estimating the Data Rate Based on Current Signal Parameters

The encoding rate is determined by comparing the current frame energy in each of two frequency bands, $f(1)$ and $f(2)$, to background noise estimates in these respective bands. Thresholds above the background noise in each band are determined by an estimated signal-to-noise ratio in each band. These thresholds are set for Rate 1, Rate 1/2, and Rate 1/8 encoding. The highest rate calculated from the two frequency bands is then selected as the encoding rate for the current frame.

4.3.1.1 Computing Band Energy

The rate determination algorithm uses energy thresholds to determine the encoding rate for the current frame. The input speech is divided into two distinct bands: band $f(1)$ spans 0.3-2.0 kHz, band $f(2)$ spans 2.0-4.0 kHz.† The band energy for band $f(i)$, $BE_{f(i)}$, is calculated as

---

† Whenever a variable (or symbol or value) with a subscript $f(i)$ appears in any equation or pseudocode in 4.1.3.4, 4.3.2.2, and 4.3.2.3, it refers to a variable (or symbol or value) associated with either band $f(1)$ or band $f(2)$.

$$BE_{f(i)} = R_w(0)R_{f(i)}(0) + 2.0 \sum_{k=1}^{L_h-1} R_w(k)R_{f(i)}(k) \qquad (4.3.1.1\text{-}1)$$

where:

$$R_{f(i)}(k) = \sum_{n=0}^{L_h-1-k} h_i(n)h_i(n+k) \quad ; 0 \le k < L_h \qquad (4.3.1.1\text{-}2)$$

and $h_i(n)$ is the impulse response of the band-pass filter i, where i = 1, 2. $R_W(k)$ is the autocorrelation sequence defined in Equation 4.2.1.1-4, and L_h = 17 is the length of the impulse response of the band-pass filters.

The band-pass filters used for both frequency bands are defined in Table 4.3.1.1-1.

Table 4.3.1.1-1. FIR Filter Coefficients Used for Band Energy Calculations

n	$h_1(n)$ **(lower band)**	***n***	$h_2(n)$ **(upper band)**
0	-5.557699E-02	0	-1.229538E-02
1	-7.216371E-02	1	4.376551E-02
2	-1.036934E-02	2	1.238467E-02
3	2.344730E-02	3	-6.243877E-02
4	-6.071820E-02	4	-1.244865E-02
5	-1.398958E-01	5	1.053678E-01
6	-1.225667E-02	6	1.248720E-02
7	2.799153E-01	7	-3.180645E-01
8	4.375000E-01	8	4.875000E-01
9	2.799153E-01	9	-3.180645E-01
10	-1.225667E-02	10	1.248720E-02
11	-1.398958E-01	11	1.053678E-01
12	-6.071820E-02	12	-1.244865E-02
13	2.344730E-02	13	-6.243877E-02
14	-1.036934E-02	14	1.238467E-02
15	-7.216371E-02	15	4.376551E-02
16	-5.557699E-02	16	-1.229538E-02

4.3.1.2 Calculating Rate Determination Thresholds

The rate determination thresholds for each frequency band $f(i)$ are a function of both the background noise estimate, $B_{f(i)}(m\text{-}1)$, and the signal energy estimate, $S_{f(i)}(m\text{-}1)$, of the previous or $(m\text{-}1)$th frame. Two thresholds for each band are computed as

$$T_1\left(B_{f(i)}(m-1), SNR_{f(i)}(m-1)\right) = k1\left(SNR_{f(i)}(m-1)\right)B_{f(i)}(m-1), \qquad (4.3.1.2\text{-}1)$$

$$T_2\left(B_{f(i)}(m-1), SNR_{f(i)}(m-1)\right) = k2\left(SNR_{f(i)}(m-1)\right)B_{f(i)}(m-1), \qquad (4.3.1.2\text{-}2)$$

where the integer $SNR_{f(i)}(m\text{-}1)$ is:

$$SNR_{f(i)}(m-1) = \begin{cases} 0 & ; QSNRU_{f(i)}(m-1) < 0, \\ QSNRU_{f(i)}(m-1) & ; 0 \le QSNRU_{f(i)}(m-1) \le 7 \\ 7 & ; QSNRU_{f(i)}(m-1) > 7 \end{cases} \quad (4.3.1.2\text{-}3)$$

and where:

$$QSNRU_{f(i)}(m-1) = round\left\{\left(10\log_{10}\left(S_{f(i)}(m-1)/B_{f(i)}(m-1)\right)-20\right)/5\right\}. \quad (4.3.1.2\text{-}4)$$

The functions $k1(\bullet)$ and $k2(\bullet)$ are defined in Table 4.3.1.2-1, and $B_{f(i)}(m\text{-}1)$ and $S_{f(i)}(m\text{-}1)$ are defined in 4.3.2.2 and 4.3.2.3, respectively.

Table 4.3.1.2-1. Threshold Scale Factors as a Function of SNR

$SNR_{f(i)}(m\text{-}1)$	$k1(SNR_{f(i)}(m\text{-}1))$	$k2(SNR_{f(i)}(m\text{-}1))$
0	7.0	9.0
1	7.0	12.6
2	8.0	17.0
3	8.6	18.5
4	8.9	19.4
5	9.4	20.9
6	11.0	25.5
7	31.6	79.6

The threshold scale factors are identical for the low- and high-frequency bands.

4.3.1.3 Comparing Thresholds

Band energy, $BE_{f(i)}$, is compared with two thresholds: $T_1(B_{f(i)}(m\text{-}1),SNR_{f(i)}(m\text{-}1))$ and $T_2(B_{f(i)}(m\text{-}1),SNR_{f(i)}(m\text{-}1))$. If $BE_{f(i)}$ is greater than both thresholds, Rate 1 is selected. If $BE_{f(i)}$ is greater than only one threshold, Rate 1/2 is selected. If $BE_{f(i)}$ is at or below both thresholds, Rate 1/8 is selected. This procedure is performed for both frequency bands and the higher of the two encoding rates selected from the individual bands is chosen as the preliminary data rate, *Rate*(m), of the current frame m.

4.3.1.4 Performing Hangover

If the preliminary data rate decision (from 4.3.1.3) transitions from at least two consecutive Rate 1 frames to a lower rate frame, then the next M frames are encoded as Rate 1 before allowing the encoding rate to drop to Rate 1/2 and finally to Rate 1/8. The number of hangover frames, M, is a function of the $SNR_{f(1)}(m\text{-}1)$ (the SNR in the lower frequency band) and is denoted as $Hangover(SNR_{f(1)}(m\text{-}1))$ in Table 4.3.1.4-1. $SNR_{f(1)}(m\text{-}1)$ is calculated as defined in Equation 4.3.1.2-3. The hangover algorithm is defined by the following pseudocode:

```
{
    if ( Rate(m) == Rate 1 )
        count = 0
    if ( Rate(m-1) == Rate 1 and Rate(m-2) == Rate 1 and Rate(m) != Rate 1) {
        if ( count == 0 )
            M = Hangover(SNRf(1)(m-1))
        if ( count < M ) {
            Rate(m) = Rate 1
            count = count + 1
        }
    }
}
```

where *Rate*(*m*) is the rate of the current frame and *Rate*(*m*-1) and *Rate*(*m*-2) are the rates of the previous two frames, respectively. Also, the rates for the previous frames, as used here, are those generated by the RDA prior to override logic, and are, therefore, not subject to modification by external rate control as described in 4.3.1.5.

Table 4.3.1.4-1. Hangover Frames as a Function of SNR

$SNR_{f(1)}(m\text{-}1)$	Hangover $(SNR_{f(1)}(m\text{-}1))$
0	7
1	7
2	7
3	3
4	0
5	0
6	0
7	0

4.3.1.5 Constraining Rate Selection

The rate selected by the procedures described in 4.3.1.3 and 4.3.1.4 are used for the current frame except where it is modified by the following constraints.

If the previous frame was selected as Rate 1 and the current frame is selected as Rate 1/8, then the encoding rate of the current frame should be modified to Rate 1/2. There are no other restrictions on encoding rate transitions.

If the speech codec has been commanded to generate a Rate 1/2 maximum packet and the rate determined is Rate 1, it generates a Rate 1/2 packet. If the speech codec has been requested to generate a Blank packet, it generates a packet based on the rate determined by the rate determination algorithm (RDA).

4.3.2 Updating RDA Parameters

After the RDA is complete, RDA parameters shall be updated as described in 4.3.2.1 through 4.3.2.3.

4.3.2.1 Updating the Smoothed Band Energy

The band energy, $BE_{f(i)}$, calculated in Equation 4.3.1.1-1 is smoothed and used to estimate both the background noise energy (see 4.3.2.2) and signal energy (see 4.3.2.3) in each band. The smoothed band energy, $E^{sm}_{f(i)}(m)$, is computed as:

$$E^{sm}_{f(i)}(m) = \alpha_E(m) E^{sm}_{f(i)}(m-1) + \left(1-\alpha_E(m)\right) BE_{f(i)} \quad ; \; 1 \le \text{i} \le 2 \qquad (4.3.2.1\text{-}1)$$

where m refers to the current frame, and

$$\alpha_E(m) = \begin{cases} 0 & ; \text{m} \le 1, \\ 0.6 & ; \text{m} > 1. \end{cases} \qquad (4.3.2.1\text{-}2)$$

This allows the smoothed band energy to be initialized to the band energy of the first frame (m = 1).

4.3.2.2 Updating the Background Noise Estimate

An estimate of the background noise level, $B_{f(i)}(m)$, is computed for the current, or mth, frame using $B_{f(i)}(m\text{-}1)$, $E^{sm}_{f(i)}(m)$ (see 4.3.2.1) and $SNR_{f(i)}(m\text{-}1)$ (see 4.3.1.2). Pseudocode describing the background noise update for band $f(i)$ is given as

{
    if (β < 0.30 for 8 or more consecutive frames)
        $B_{f(i)}(m)$= min{ $E^{sm}_{f(i)}(m)$, 80954304, max{ 1.03$B_{f(i)}(m\text{-}1)$, $B_{f(i)}(m\text{-}1)$+ 1 } }
    else {
        if ($SNR_{f(i)}(m\text{-}1)$ > 3)
            $B_{f(i)}(m)$= min{ $E^{sm}_{f(i)}$(m), 80954304, max{ 1.00547B$_{f(i)}$(m-1), $B_{f(i)}$(m-1)+ 1 } }
        else
            $B_{f(i)}$ (m)= min{ Esmf(i)(m), 80954304, $B_{f(i)}$ (m-1) }
    }
    if ($B_{f(i)}$ (m) < lownoise(i))
        $B_{f(i)}$ (m) = lownoise(i)
}

where β, $E^{sm}_{f(i)}(m)$, and $SNR_{f(i)}(m\text{-}1)$ are defined in 4.2 and Equations 4.3.2.1-1 and 4.3.1.2-3, respectively. *lownoise*(1) equals 160.0 and *lownoise*(2) equals 80.0.

At initialization, the background noise estimate for the first frame, $B_{f(i)}(0)$, is set to 80,954,304 for both frequency bands, and the consecutive frame counter for β is set to zero. Initialization also occurs if the audio input to the encoder is disabled and then enabled.†

4.3.2.3 Updating the Signal Energy Estimate

The signal energy, $S_{f(i)}(m)$, is computed as

---

† This prevents the silence before the audio is connected from being mistaken as unusually low background noise.

{
If (β > 0.5 for 5 or more consecutive frames)
$S_{f(i)}(m)= \max\{ E^{sm}{}_{f(i)}(m), \min\{ 0.97S_{f(i)}(m-1), S_{f(i)}(m-1) - 1 \} \}$
else
$S_{f(i)}(m)= \max\{ E^{sm}{}_{f(i)}(m), S_{f(i)}(m-1) \}$
}

where β and $E^{sm}_{f(i)}(m)$ are defined in Section 4.2 and Equation 4.3.2.1-1, respectively.

At initialization, the signal energy estimates for the first frame, $S_{f(1)}(0)$ and $S_{f(2)}(0)$, are set to 51,200,000 and 5,120,000, respectively, and the consecutive frame counter for β is set to zero. Initialization also occurs if the audio input to the encoder is disabled and then enabled.

4.4 Quantization of LSP Parameters

Inputs: The inputs to the LSP quantizer are:

- The unquantized LSPs for the current frame, $\Omega(m)$
- The rate of the current frame, *Rate*(m)

Outputs: The outputs from the LSP quantizer are:

- The quantized LSPs for the current frame, $\Omega_q(m)$

Processing: The quantizer takes the form of a weighted split vector LSP quantizer. The splits for the quantizer for a given *Rate* are given in parameter Table 4.4-1. All of the sub-matrices are coded using the number of bits found in this table. The codebooks $\boldsymbol{q}_{rate}(k)$, that are used to quantize each set of LSPs can be found in Tables B-1 through B-9. Each LSP set has one codebook with dimensions $n_{sub}(k)$ by $n_{size}(k)$. $n_{sub}(k)$ is the number of LSP parameters in the set and $n_{size}(k)$ is the codebook size.

Table 4.4-1. LSP Parameter Splits

Codebook Number k	LSP Parameters in set $\Omega\,(m)$	LSP Split Starting Parameter Index B(k)	Number of LSP Parameters $n_{sub}(k)$	Code book Size $n_{size}(k)$	Number of bits allocated
Rate 1					
1	$\Omega\,(m, i);\, i\in(1, 2)$	1	2	64	6
2	$\Omega\,(m, i);\, i\in(3, 4)$	3	2	64	6
3	$\Omega\,(m, i);\, i\in(5, 6, 7)$	5	3	512	9
4	$\Omega\,(m, i);\, i\in(8, 9, 10)$	8	3	128	7
Rate 1/2					
1	$\Omega\,(m, i);\, i\in(1, 2, 3)$	1	3	128	7
2	$\Omega\,(m, i);\, i\in(4, 5, 6)$	4	3	128	7
3	$\Omega\,(m, i);\, i\in(7, ..., 10)$	7	4	256	8
Rate 1/8					
1	$\Omega\,(m, i);\, i\in(1, ..., 5)$	1	5	16	4
2	$\Omega\,(m, i);\, i\in(6, ..., 10)$	6	5	16	4

The number of codebooks, k_{num}, for each rate is shown in Table 4.4-2.

Table 4.4-2. Number of LSP Codebooks

Rate	Number of Codebooks k_{num}
1	4
1/2	3
1/8	2

4.4.1 Computation of Weights

The weights, $w(i)$, for the LSP frequencies are defined as follows:

$$\omega(i) = \begin{cases} \dfrac{50}{2\pi} + 1 & ;\ \Delta_{\Omega}(m,i) = 0, \\ \dfrac{0.5}{2\pi\Delta_{\Omega}(m,i)} + 1 & ;\ \text{otherwise}, \end{cases} \qquad 1 \le \mathrm{i} \le 10, \qquad (4.4.1\text{-}1)$$

where:

$$\Delta_{\Omega}(m,i)=\begin{cases}\Omega(m,i+1)-\Omega(m,i) & ;\ i=1,\\ \min\left(\left(\Omega(m,i)-\Omega(m,i-1)\right),\left(\Omega(m,i+1)-\Omega(m,i)\right)\right) & ;\ 2\le i\le 9,\\ \Omega(m,i)-\Omega(m,i-1) & ;\ i=10.\end{cases}\quad (4.4.1\text{-}2)$$

4.4.2 Error Matrix Computation

Compute the error matrix, $e(k)$, for each set of LSP frequencies k:

$$e(k,j)=\sum_{i=1}^{nsub(k)}\omega\left(B(k)+i-1\right)\left(\Omega\left(m,B(k)+i-1\right)-q_{rate}(k,i,j)\right)^{2}\quad ;1\le j\le n_{size}(k),\ \ 1\le k\le k_{\mathrm{num}}\ ,\quad (4.4.2\text{-}1)$$

where i is the sub-index of each LSP parameter set, and j is the index of the codebook entries. All the other parameters are defined in Tables 4.4-1 and 4.4-2.

4.4.3 Adjustment of Quantization Error

The error associated between codebooks is adjusted to prevent excessively small (or negative) gaps between LSPs at codebook seams. If the difference between a particular $q(k, 1, j)$ and $\Omega_q(m, B(k)\text{-}1)$ is less than δ, $e(k, j)$ is assigned a value that will make it invalid for the search. The error matrix is adjusted by:

$$e(k,j)=\begin{cases}MAXFLOAT & ;\left(q_{rate}(k,l,j)-\Omega_q(m,B(k)-1)\right)\le\delta,\\ e(k,j) & ;\text{otherwise},\end{cases}\quad 1\le j\le n_{size}(k),\ 1\le k\le k_{num}\quad (4.4.3\text{-}1)$$

where $\delta = 0.05 / 2\pi$

4.4.4 Quantization Search

Each error matrix, $e(k,i)$, is searched for the minimum error, $e_{min}(k)$, as defined by:

$$e_{\min}(k)=\min\left\{e(k,j)MAXFLOAT\right\}\quad ;1\le j\le n_{size}(k),\ \ 1\le k\le k_{num}\ .\quad (4.4.4\text{-}1)$$

The codebook index to be transmitted, *LSPIDX*(k), is defined as the index, j, where $e_{min}(k)$ was found above for all k.

4.4.5 Generation of Quantized LSP Parameters

The codebook indices, *LSPIDX*(k), are used to generate a quantized set of LSP parameters:

$$\Omega_q(m,B(k)+i-1)=q_{rate}(k,i,LSPIDX(k))\ ;1\le i\le n_{sub}(k),\ \ 1\le k\le k_{num}\ .\quad (4.4.5\text{-}1)$$

4.5 Encoding at Rates 1/2 and 1

The algorithm for encoding at Rate 1 is similar to that used for encoding at Rate 1/2 except for the quantization tables used and the fact that certain quantities (e.g., the spectral transition indicator, *LPCFLAG*, and the delay difference, *DDELAY*) are transmitted only at Rate 1. Encoding is accomplished as follows:

Inputs: The inputs to encoding are:

- The quantized LSPs from the previous frame, $\Omega_q(m\text{-}1)$
- The unquantized LSPs from the current and previous frames, $\Omega\,(m)$ and $\Omega\,(m\text{-}1)$
- The short-term prediction residual, $\varepsilon(n)$
- The pitch delay estimate for the previous frame, $\tau(m\text{-}1)$
- The pitch delay estimate for the current frame, $\tau(m)$
- The long term prediction gain, β
- The rate of the current frame, $Rate(m)$

Outputs: The outputs of encoding are:

- The quantized LSPs for the current frame, $\Omega_q(m)$
- The LSP indices corresponding to the quantized LSPs, $LSPIDX(k)$
- The adaptive codebook gain index, $ACBGIDX(m')$, for each subframe, m'
- The fixed codebook shape indices, $FCBSIDX(m')$, for each subframe, m'
- The fixed codebook gain indices, $FCBGIDX(m',k)$, for each subframe, m'
- The delay transmission code, *DELAY*
- The spectral transition indicator, *LPCFLAG*, Rate 1 only
- The delay difference, *DDELAY*, Rate 1 only

State Variables Affected:

- The adaptive codebook excitation signal, $E(n)$
- The accumulated shift counter, τ_{acc}
- The pointer to the last sample in the shifted residual, n_m
- The state of the shifted residual buffer, *shiftstate*
- The filter memories in $H_{wq}(z)$ and $H_w(z)$
- The modified residual buffer, $\hat{\varepsilon}(n)$

Initialization:

- The state of the shifted residual buffer, *shiftstate* = CENTER
- The quantized LSPs, $\Omega_q(m,k) = 0.048k$; $m = 0$, $1 \le k \le 10$

Processing:

EVRC encoding shall comprise the procedures described in 4.5.1 through 4.5.4. A block diagram of the RCELP encoding process is shown in Figure 4.5-1.

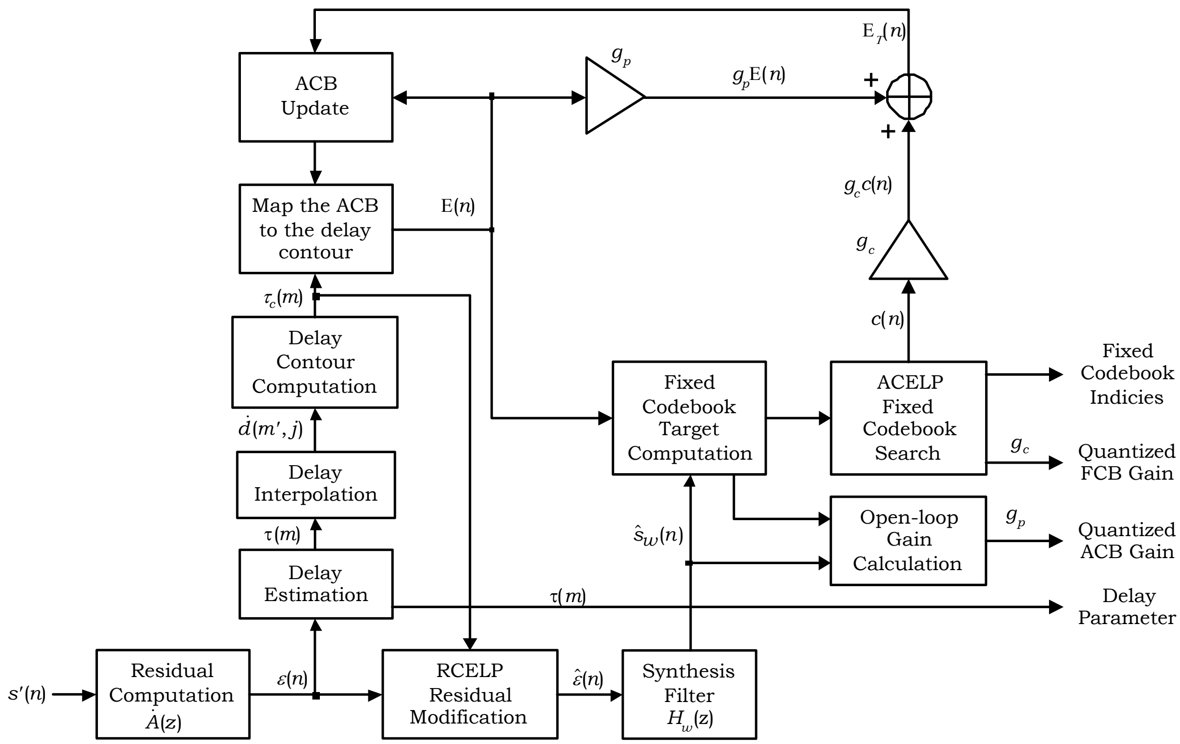

Figure 4.5-1. RCELP Encoding Block Diagram

4.5.1 LSP Quantization

The encoder shall vector quantize the unquantized LSPs, Ω(*m*), using the procedure found in 4.4 for Rates 1 or 1/2.

4.5.2 RCELP Shift State Update

The encoder shall update the shift state with hysteresis and control the accumulated shift. This step prevents the asynchrony between the original signals and the modified signals from increasing over time. It shall be implemented as a state machine, given by:

```
/* if signal is not at all periodic, reset */
if ( β < 0.1 ) {
      τacc = 0
      nm = 0
      shiftstate = CENTER
}
/* update shift state with hysteresis */
if ( τacc > 20 ) {
      shiftstate = RIGHT
}
if ( τacc < -20 ) {
      shiftstate = LEFT
```

```
}
if ( τacc <= 10 && shiftstate = RIGHT ) {
    shiftstate = CENTER
}
if ( τacc >= -10 && shiftstate = LEFT) {
    shiftstate = CENTER
}
/* control accumulated shift by adjusting the delay */
if ( shiftstate == LEFT && β < 0.4 ) {
    τ(m) = τ(m) + 1
}
else if ( shiftstate == RIGHT && β < 0.4 ) {
    τ(m) = τ(m) - 1
}
/* check to keep delay within bounds */
τ(m) = min{ τ(m), 120 }
τ(m) = max{ τ(m), 20 }
```

4.5.3 Delay Encoding

The delay parameter, $\tau(m)$, determined in 4.2.3, shall be encoded for transmission by:

$$DELAY = \tau(m) - \tau_{\min,}\ , \tag{4.5.3-1}$$

where τ_{min}=20.

In Rate 1 processing only, a differential delay parameter shall be transmitted for use in reconstructing the delay contour in the decoder after frame erasures. The transmission code, *DDELAY*, is defined as:

$$DDELAY = \begin{cases} 0 & ;\left|\Delta_\tau\right| > 15, \\ \Delta_\tau + 16 & ;\text{otherwise} \end{cases} \tag{4.5.3-2}$$

where

$$\Delta_\tau = \tau(m) - \tau(m-1)\ . \tag{4.5.3-3}$$

4.5.4 Rates 1/2 and 1 Subframe Processing

The procedures described in 4.5.4.1 through 4.5.4.14 shall be carried out for three subframes, $0 \le m' < 3$. The subframe size, *L*, is 53 for subframes 0 and 1, and 54 for subframe 2.

4.5.4.1 Interpolation of LSP Parameters

Interpolate the unquantized and quantized LSPs over the three subframes, *m′*, in the current frame, *m*. The form of the unquantized, interpolated LSP vector for subframe *m′* is given by:

$$\dot{\Omega}(m') = (1-\mu_{m'})\Omega(m-1) + \mu_{m'}\Omega(m), \tag{4.5.4.1-1}$$

and form of the quantized, interpolated LSP vector is given by:

$$\dot{\Omega}_q(m') = (1-\mu_{m'})\Omega_q(m-1) + \mu_{m'}\Omega_q(m), \tag{4.5.4.1-2}$$

where the subframe interpolator constants are defined as $\mu = \{\ 0.1667, 0.5, 0.8333\ \}$.

4.5.4.2 LSP to LPC Conversion

Convert the unquantized, interpolated LSPs, $\dot{\Omega}(m')$, to unquantized, interpolated LPC parameters, $\left\{\dot{a}\right\}$, as described in 4.2.2.2 for each subframe. Convert the quantized, interpolated LSPs, $\dot{\Omega}_q(m')$, to quantized, interpolated LPC parameters, $\left\{\dot{a}_q\right\}$, as described in 4.2.2.2, using the quantized LSPs at the end-points instead of the unquantized LSPs.

4.5.4.3 Zero Input Response Calculation

Calculate the zero-input response, $a_{zir}(n)$, of the weighted synthesis filter, $H_{wq}(z)$. The weighted synthesis filter is defined as:

$$H_{wq}(z) = \frac{1}{\dot{A}_q(z)}\left[\frac{\dot{A}\left(\gamma_1^{-1}z\right)}{\dot{A}\left(\gamma_2^{-1}z\right)}\right], \tag{4.5.4.3-1}$$

where

$$\dot{A}_q(z) = 1 - \sum_{k=1}^{10} \dot{a}_q(k) z^{-k}, \tag{4.5.4.3-2}$$

and

$$\dot{A}\left(\gamma_1^{-1}z\right) = 1 - \sum_{k=1}^{10} \dot{a}(k)\gamma_l^k z^{-k} \quad ; \text{l=1,2}, \tag{4.5.4.3-3}$$

where $\gamma_1 = 0.9, \quad \gamma_2 = 0.5$.

4.5.4.4 Impulse Response Calculation

Calculate the impulse response, $h_{wq}(n)$, of the weighted synthesis filter, $H_{wq}(z)$, to 54 terms.

4.5.4.5 Interpolated Delay Estimate Calculation

Calculate a set of three interpolated delay estimates, $\dot{d}(m', j)$, for each subframe m' as follows:

$$\dot{d}(m', j) = \begin{cases} \tau(m) & ;\left|\tau(m) - \tau(m-1)\right| > 15, \\ \left(1 - f(m'+j)\right)\tau(m-1) + f(m'+j)\tau(m) & ;\text{otherwise}, \end{cases} \qquad 0 \le m' < 3,\ 0 \le j < 3, \quad (4.5.4.5\text{-}1)$$

where m is the current frame, m' is the subframe index, and $m' + j$ is an index into an array of interpolator coefficients, $\boldsymbol{f}$, defined by:

$$\boldsymbol{f} = \{\ 0.0,\ 0.3313,\ 0.6625,\ 1.0,\ 1.0\ \}. \qquad (4.5.4.5\text{-}2)$$

4.5.4.6 Calculation of the Adaptive Codebook Contribution

Compute the adaptive codebook contribution, $E(n)$, described in 4.5.5.

4.5.4.7 Modification of the Original Residual

Generate a modified residual signal, $\hat{\varepsilon}(n)$, by shifting the current residual signal, $\varepsilon(n)$, to match the delay contour for the current subframe. See 4.5.6 for details on this operation.

4.5.4.8 Generation of the Weighted Modified Original Speech Vector

Generate the weighted modified original speech vector, $\hat{s}_w(n)$, by filtering the modified residual, $\hat{\varepsilon}(n)$, through the weighted synthesis filter, $H_w(z)$, given by:

$$H_w(z) = \frac{1}{\dot{A}(z)}\left[\frac{\dot{A}\left(\gamma_1^{-1}z\right)}{\dot{A}\left(\gamma_2^{-1}z\right)}\right], \qquad (4.5.4.8\text{-}1)$$

where

$$\dot{A}\left(z\right) = 1 - \sum_{k=1}^{10} \dot{a}(k) z^{-k}\ , \qquad (4.5.4.8\text{-}2)$$

and all other terms are defined in 4.5.4.3.

4.5.4.9 Closed-Loop Gain Calculation

Subtract the zero-input response, $a_{zir}(n)$, of the weighted synthesis filter, $H_{wq}(z)$, from the weighted modified original speech vector, $\hat{s}_w(n)$:

$$\hat{s}_w(n) = \hat{s}_w(n) - a_{zir}(n) \qquad ;0 \le n < L. \qquad (4.5.4.9\text{-}1)$$

Convolve the impulse response, $h_{wq}(n)$, with the adaptive codebook excitation, $E(n)$, generating the filtered adaptive codebook excitation, $\lambda(n)$:

$$\lambda(n) = \sum_{j=0}^{n} h_{wq}(j) E(n-j) \qquad ;0 \le n < L. \qquad (4.5.4.9\text{-}2)$$

The closed-loop quantized gain, g_p , is calculated by first determining the cross-correlation, $R_{s\lambda}(0)$, of $\hat{s}_w(n)$ and $\lambda(n)$:

$$R_{s\lambda}(0) = \sum_{n=0}^{L-1} \hat{s}_w(n)\lambda(n), \tag{4.5.4.9-3}$$

the auto-correlation of $\lambda(n)$ is then calculated by:

$$R_{\lambda\lambda}(0) = \sum_{n=0}^{L-1} \lambda^2(n). \tag{4.5.4.9-4}$$

The quantized closed loop gain, g_p, is defined by:

$$g_p = \begin{cases} g_{pcb}(0) & ;R_{s\lambda}(0) \le 0, \\ g_{pcb}(ACBGIDX(m')) & ;R_{s\lambda}(0) > 0, \end{cases} \tag{4.5.4.9-5}$$

where $g_{pcb}(k)$ is an entry in the adaptive codebook gain quantization Table 4.5.4-1, and $ACBGIDX(m')$ is the optimal index into $g_{pcb}(k)$ calculated by:

$ACBGIDX(m') = 0$
for ($k = 1$; $k < g_{pcb\_size}$; $k = k + 1$) {
    if $\left(R_{s\lambda}(0) > \left(\frac{g_{pcb}(k) + g_{pcb}(k-1)}{2} R_{\lambda\lambda}(0) \right) \right)$ {
        $ACBGIDX(m') = k$
    }
}

where $g_{pcb\_size} = 8$.

Table 4.5.4-1. Adaptive Codebook Gain Quantization Table

k	$g_{pcb}(k)$
0	0.00
1	0.30
2	0.55
3	0.70
4	0.80
5	0.90
6	1.00
7	1.20

4.5.4.10 Fixed Codebook Search Target Vector Generation

4.5.4.10.1 Perceptual Domain Target Vector

Calculate the target vector in the perceptual domain, $x_w(n)$, defined by:

$$x_w(n) = \hat{s}_w(n) - g_p \lambda(n) \quad ;0 \le n < L , \tag{4.5.4.10.1-1}$$

where $\hat{s}_w(n)$ and $\lambda(n)$ are defined in 4.5.4.9.

4.5.4.10.2 Conversion of the Target Vector to the Residual Domain

Convert the target vector from the perceptual domain to the residual domain by filtering $x_w(n)$ through the zero-state inverse of the weighting filter, $H_{wq}(z)$:

$$H_{wq}^{-1}(z) = \dot{A}_q(z) \frac{\dot{A}\left(\gamma_2^{-1} z\right)}{\dot{A}\left(\gamma_1^{-1} z\right)} , \tag{4.5.4.10.2-1}$$

and all other terms are defined in 4.5.4.3.

The resulting target vector in the residual domain is defined as $x(n)$.

4.5.4.10.3 Delay Calculation for Current Subframe

The fixed codebook search requires an estimate of the delay, $\dot{\tau}$, for the current subframe, defined by:

$$\dot{\tau} = round\left\{ \frac{\dot{d}(m',0) + \dot{d}(m',1)}{2} \right\} , \tag{4.5.4.10.3-1}$$

where $\dot{d}(m', j)$ is defined in 4.5.4.5.

4.5.4.11 Fixed Codebook Search

Perform the fixed codebook search as described in 4.5.7.

4.5.4.12 Fixed codebook gain quantization

The fixed codebook gain, g_c, computed in 4.5.4.11, is constrained by:

$$g_c = \begin{cases} \left(1.0 - 0.15 g_p\right) g_c & ;\text{Rate(m)==Rate1,} \\ \left(0.9 - 0.1 g_p\right) g_c & ;\text{Rate(m)==Rate1/2,} \end{cases} \tag{4.5.4.12-1}$$

which is then quantized by the following process:

$FCBGIDX(m') = 0$
for ($k = 1$; $k < g_{ccb\_size}$; $k = k + 1$)

$$\text{if}\left(g_c > \left(\frac{g_{ccb}(k)+g_{ccb}(k-1)}{2}\right)\right)\ \{$$

$$FCBGIDX(m') = k$$

$$\}$$

$$g_c = g_{ccb}(FCBGIDX(m'))$$

where $g_{\text{ccb}}(k)$ is a fixed codebook gain quantization entry defined in Table B-12 and B-13 for Rate 1 and Rate 1/2, respectively, and $FCBGIDX(m')$ is the optimal index into the appropriate table for each subframe m'.

4.5.4.13 Combined Excitation Vector Computation

The combined excitation vector, $E_T(n)$, for the current subframe, m' is defined as the sum of the adaptive codebook contribution, $E(n)$, and the fixed codebook contribution, $c(n)$:

$$E_T(n) = g_p E(n) + g_c c(n) \quad ;0 \le n < L. \tag{4.5.4.13-1}$$

4.5.4.14 Encoder State Variable Update

Update the weighted synthesis filter memory by filtering the combined excitation vector, $E_T(n)$, through the weighted synthesis filter, $H_{wq}(z)$, which is given in Equation 4.5.4.3-1.

Update the modified residual memory by copying n_m elements of the current modified residual to the beginning of the modified residual buffer:

$$\hat{\varepsilon}(n) = \hat{\varepsilon}(n+L); \quad 0 \le \mathrm{n} < \mathrm{n_m}. \tag{4.5.4.14-1}$$

Update the excitation memory for the adaptive codebook with the combined excitation for the current subframe:

$$E(n) = \begin{cases} E(n+L) & ;-128 \le n < -L \\ E_T(n+L) & ;-L \le n < 0 \end{cases} \tag{4.5.4.14-2}$$

Conditionally reset the accumulated shift counter, τ_{acc}, and the pointer to the last sample in the shifted residual, n_m, according to the following:

$$\text{if}\left(\left(\textstyle\sum_{i=0}^{2} ACBGIDX(i) \le 1\right) and \left(ACBGIDX(2) \ne 1\right)\right)\ \{$$

$$\tau_{acc} = 0$$

$$n_m = 0$$

$$\}$$

4.5.5 Computation of the Adaptive Codebook Contribution

Inputs: The inputs to the adaptive codebook computation are:

- The vector of interpolated delays, $d(m', j)$, calculated in 4.5.4.5
- The current subframe number, m'
- The current subframe size, L

Outputs:

- The adaptive codebook excitation, $E(n)$, where $-128 \le n \le 64$

Initialization: The variables used in the computation of the adaptive codebook contribution are initialized as follows:

- The adaptive codebook excitation memory, $E(n) = 0$; $-128 \le n \le 64$

Processing:

Computation of the adaptive codebook contribution shall be performed according to the procedures described in 4.5.5.1 and 4.5.5.2.

4.5.5.1 Delay Contour Computation

Compute the 8-times oversampled delay contour vector, $\tau_c(n)$, defined by:

$$\tau_c(n) = \begin{cases} \dot{d}(m',0) + \dfrac{n\left[\dot{d}(m',1) - \dot{d}(m',0)\right]}{L} & ;0 \le n < L, \\ \dot{d}(m',1) + \dfrac{(n-L)\left[\dot{d}(m',2) - \dot{d}(m',1)\right]}{L} & ;L \le n < (L+10) \end{cases} \quad (4.5.5.1\text{-}1)$$

where m' is the current subframe, L is the current subframe size, $\dot{d}(m', j)$ is a set of interpolated delays calculated in 4.5.4.5.

4.5.5.2 Mapping of the Adaptive Codebook to the Delay Contour

The new adaptive codebook contribution, E(n), is calculated by:

$$E(n) = \sum_{i=0}^{2f_l} E(i + n - T_E(n) - f_l) I_E\left(i + \left(2f_l + 1\right) T_{IE}(n)\right) \quad ;0 \le n < (L+10), \quad (4.5.5.2\text{-}1)$$

where $T_E(n)$ and $T_{IE}(n)$ are calculated for every n by:

$$T_E(n) = \begin{cases} round\left\{\tau_c(n)\right\} & ;\tau_c(n) > 0, \\ -round\left\{-\tau_c(n)\right\} & ;\tau_c(n) \le 0, \end{cases}$$

$$T_{IE}(n) = trunc\left\{\left(T_E(n) - \tau_c(n) + 0.5\right) R + 0.5\right\},$$

```
if ( TIE(n) == R ) {
    TIE(n) = 0
    TE(n) = TE(n) - 1
}
```

and L is the current subframe size, $f_l = 8$, $R = 8$, and $\{I_E(n)\}$ is a set of interpolation coefficients found in Tables B-11.

4.5.6 Modification of the Residual

Inputs: The inputs to the residual modification are:

- The vector of interpolated delays, $d(m', j)$, calculated in 4.5.4.5
- The short-term prediction residual, $\varepsilon(n)$
- The pitch prediction gain, β
- The current subframe number, m'
- The current subframe size, L

Outputs: The outputs from the residual modification are:

- The modified residual signal, $\hat{\varepsilon}(n)$

State Variables Affected:

- The modified target residual buffer, $\hat{\varepsilon}_t(n)$ where $-128 \le n \le 64$
- The pointer to the end of the last residual frame modified, n_m
- The accumulated shift, τ_{acc}

Initialization: The variables used in the residual modification are initialized as follows:

- The modified target residual buffer, $\hat{\varepsilon}_t(n) = 0$; $-128 \le n \le 64$
- The pointer to the end of the last residual frame modified, $n_m = 0$
- The accumulated shift, $\tau_{acc} = 0$

Processing: The modification of the residual signal to generate a target residual for the fixed codebook search shall be computed according to 4.5.6.1 through 4.5.6.4. A shifted target residual, $\hat{\varepsilon}_t(n)$, is generated in 4.5.6.1 using the past modified residual buffer and the delay contour of the current frame. This shifted residual is used as a target for shifting the residual of the current subframe. All the pitch pulses in the original residual, $\varepsilon(n)$, must be shifted individually to match the delay contour of the modified target residual, $\hat{\varepsilon}_t(n)$. The modification is performed by finding a shift frame containing each pulse in the original residual by the process described in 4.5.6.2. Once the shift frame is found, the accumulated shift, τ_{acc}, must be adjusted as described in 4.5.6.3 to determine the proper shift needed to match the pulse to the delay contour. The shift frame containing the pulse is then shifted by τ_{acc} as described in 4.5.6.4. The procedures described in 4.5.6.2 through 4.5.6.4 must be carried out for all pulses found in the residual of the current subframe.

4.5.6.1 Mapping of The Past Modified Residual to the Delay Contour

Calculate the target for the residual modification process by warping the past modified residual buffer, $\hat{\varepsilon}_t(n)$, by the delay contour, $\tau_c(n)$, calculated in 4.5.5.1.

The new target residual buffer, $\hat{\varepsilon}_t(n)$, is calculated by:

$$\hat{\varepsilon}_t(n) = \sum_{i=0}^{2f_l} \hat{\varepsilon}_t(i + n - T_\varepsilon(n) - f_l) I_\varepsilon\left(i + \left(2f_l + 1\right) T_{I\varepsilon}(n)\right) \quad ;0 \le n < (L+10)\,, \tag{4.5.6.1-1}$$

where $T_\varepsilon(n)$ and $T_{I\varepsilon}(n)$ are calculated for every n by:

$$T_\varepsilon(n) = \begin{cases} round\{\tau_c(n)\} & ;\tau_c(n) > 0, \\ -round\{-\tau_c(n)\} & ;\tau_c(n) \le 0, \end{cases}$$

$$T_{I\varepsilon}(n) = trunc\{(T_\varepsilon(n) - \tau_c(n) + 0.5)R + 0.5\},$$

```
if ((T_Iε(n) == R)
{
    T_Iε(n) = 0
    T_ε(n) = T_ε(n) - 1
}
```

and L is the current subframe size, $f_l = 3$, $R = 8$, $\{I_\varepsilon(n)\}$ is a set of interpolation coefficients found in Tables B-10, and $\tau_c(n)$ is computed in Equation 4.5.5.1-1.

4.5.6.2 Calculation of the Residual Shift Frame Parameters

First define the current subframe residual as:

$$\varepsilon_s(n) = \varepsilon(n + 80 + 53m') \quad ;0 \le n < 240\text{-}53m', \tag{4.5.6.2-1}$$

and proceed with the following sections.

4.5.6.2.1 Search for Pulses in the Subframe Residual

Calculate the location, n_{emax}, that contains the pulse having maximum energy in the subframe residual. The search window is centered at 1/2 the pitch lag, $\dot{d}(m',1)$, after the last modified residual sample, and is 1.5 times the pitch lag in length. This location, n_{emax}, is defined as the location, n, that maximizes:

$$E_\varepsilon(n) = \sum_{i=-2}^{2} \varepsilon_s^2(T + n + i) \quad ;0 \le n < 1, \tag{4.5.6.2.1-1}$$

where the index of the start of the energy search, T, is defined by:

$$T = \max\left\{\left(n_m + trunc\{-\tau_{acc} + 0.5\} - trunc\left\{\frac{1}{4}\dot{d}(m',1)\right\}\right), (\text{-}78 - 53m')\right\}, \tag{4.5.6.2.1-2}$$

where n_m is an index to the last residual sample modified, and τ_{acc} is the accumulated shift; the size of the window to be searched, l, is defined by:

$$l = \min\left\{trunc\left\{1.5\dot{d}(m',1)\right\}, 238 - 53m' - T\right\}. \tag{4.5.6.2.1-3}$$

Adjust n_{emax} by τ_{acc}:

$$n_{emax} = n_{emax} + T - trunc\{-\tau_{acc} + 0.5\}. \tag{4.5.6.2.1-4}$$

4.5.6.2.2 Location of the First Pulse in the Residual

The location of maximum pulse energy, n_{emax}, is checked to insure it is the first pitch pulse in the residual. If ($n_{emax} < n_m$), the pulse located is in a region that already has been modified, so the residual must be searched again using a smaller window size. The new search window is centered one pitch lag, $\dot{d}(m',1)$, after the pulse location found in 4.5.6.2.1, and is one half of a pitch lag in length. The new location, n_{emax}, is defined as the location, n, that maximizes:

$$E_{\varepsilon}(n) = \sum_{i=-2}^{2} \varepsilon_s^2 (T+n+i) \quad ;0 \le \mathrm{n} < 1, \qquad (4.5.6.2.2\text{-}1)$$

where the index of the start of the energy search, T, is defined by:

$$T = \max\left\{\left(n_{emax} + trunc\left\{-\tau_{acc} + 0.5\right\} + trunc\left\{\frac{3}{4}\dot{d}(m',1) + 0.5\right\}\right), \left(\text{-78 - 53m'}\right)\right\}, \qquad (4.5.6.2.2\text{-}2)$$

where n_{emax} is the index of the pulse located in 4.5.6.2.1, τ_{acc} is the accumulated shift; the size of the window to be searched, l, is defined by:

$$l = \min\left\{trunc\left\{1.5\,\dot{d}(m',1)\right\}, 238 - 53m' - T\right\}. \qquad (4.5.6.2.2\text{-}3)$$

The location of the maximum pulse, n_{emax}, is then re-defined and adjusted by:

$$n_{emax} = n_{emax} + T - trunc\left\{-\tau_{acc} + 0.5\right\}. \qquad (4.5.6.2.2\text{-}4)$$

4.5.6.2.3 Location of a Pulse Inside of the Lag Window

If $n_{emax} > \left(n_m + \dot{d}(m',1)\right)$, where n_{emax} was determined in 4.5.6.2.1 or 4.5.6.2.2, the pulse found has a larger lag than expected. A final pulse energy search of the lag window must be made to insure that the pulse found at n_{emax} is the desired pulse to be shifted. The new search window is centered one pitch lag, $\dot{d}(m',1)$, before the pulse location found in 4.5.6.2.1 or 4.5.6.2.2, and is one half of a pitch lag in length. The location, n''_{emax}, of the pulse with maximum energy in the current lag window is defined as the location, n, that maximizes:

$$E_{\varepsilon}(n) = \sum_{i=-2}^{2} \varepsilon_s^2 (T+n+i) \quad ;0 \le \mathrm{n} < 1, \qquad (4.5.6.2.3\text{-}1)$$

where the index of the start of the energy search, T, is defined by:

$$T = \max\left\{\left(n_{emax} + trunc\left\{-\tau_{acc} + 0.5\right\} - trunc\left\{\frac{5}{4}\dot{d}(m',1) + 0.5\right\}\right), \left(\text{-78 - 53m'}\right)\right\}, \qquad (4.5.6.2.3\text{-}2)$$

where n_{emax} is the index of the pulse determined in 4.5.6.2.1 or 4.5.6.2.2, τ_{acc} is the accumulated shift; the size of the window to be searched, l, is defined by:

$$l = \min\left\{ trunc\left\{ 0.5\,\dot{d}(m',1) \right\}, 238 - 53m' - T \right\} \qquad (4.5.6.2.3\text{-}3)$$

The location, n''_{emax}, of the maximum pulse in the lag window, is adjusted by τ_{acc}:

$$n''_{emax} = n''_{emax} + T - trunc\left\{ -\tau_{acc} + 0.5 \right\}, \qquad (4.5.6.2.3\text{-}4)$$

The location of the pulse to be shifted, n_{emax}, is then re-defined by:

$$n_{emax} = \begin{cases} n''_{emax} & ; n''_{emax} \ge n_m, \\ n_{emax} & ;\text{otherwise.} \end{cases} \qquad (4.5.6.2.3\text{-}5)$$

4.5.6.2.4 Shift Frame Boundary Calculation

The boundaries, T_{start} and T_{end}, for the frame in the residual to be shifted are calculated by:

$$T_{start} = n_m, \qquad (4.5.6.2.4\text{-}1)$$

$$T_{end} = \begin{cases} L & ; L\text{-}10 < n_{emax} < L - 5, \\ L + 10 & ; L < n_{emax} < L + 5, \\ L & ; n_{emax} \ge L + 5, \\ n_{emax} + 10 & ;\text{otherwise.} \end{cases} \qquad (4.5.6.2.4\text{-}2)$$

4.5.6.2.5 Shift Decision

Calculate a shift decision flag, ε_{shift}, by:

$$\varepsilon_{shift} = \begin{cases} FALSE & ; \left(n_{emax} \ge T_{end} \right) or \left(n_{emax} < T_{start} \right), \\ TRUE & ;\text{otherwise.} \end{cases} \qquad (4.5.6.2.5\text{-}1)$$

4.5.6.2.6 Peak to Average Ratio Calculation

If the shift decision flag is true (ε_{shift} == $TRUE$) as evaluated in 4.5.6.2.5, the ratio of the peak energy to the average energy in the residual frame to be shifted must be calculated to insure that the frame contains a valid pulse. The residual frame to be shifted is defined as $\varepsilon_s(n)$, where $T_{start} \le n \le T_{end}$ (see 4.5.6.2 through 4.5.6.4).

Calculate a vector of smoothed residual energies, $E_{win}(n)$, by:

$$E_{win}(n) = \sum_{i=0}^{4} \varepsilon_s^2 \left(T_{adj} + i + n \right), \qquad 0 \le n \le l, \qquad (4.5.6.2.6\text{-}1)$$

where the window size is calculated by:

$$l = T_{end} - T_{start} - 5, \qquad (4.5.6.2.6\text{-}2)$$

and T_{adj} is calculated as:

$$T_{adj} = T_{start} - trunc\{\tau_{acc}\}. \quad (4.5.6.2.6\text{-}3)$$

The peak energy, E_{peak}, is then calculated as:

$$E_{peak} = \max\{E_{win}(n)\} \quad ;0 \le n \le 1, \quad (4.5.6.2.6\text{-}4)$$

and the average energy, $E_{avg}(n)$, is calculated as:

$$E_{avg}(n) = \begin{cases} \alpha E_{avg}(n-1) + (1-\alpha)\varepsilon_s^2\left(T_{adj}+n+4\right); & \varepsilon_s^2\left(T_{adj}+n+4\right) < 4E_{avg}(n-1), \\ E_{avg}(n-1) & ;\text{otherwise} \end{cases} \quad 0 < n \le 1, \quad (4.5.6.2.6\text{-}5)$$

where α=0.875, and $E_{avg}(0) = E_{win}(0)$.

The ratio of the peak to average energy is then calculated by:

$$E_{ratio} = \begin{cases} 0 & ;E_{avg}(l) = 0, \\ \left(\dfrac{E_{peak}}{E_{avg}(l)}\right)\left(\dfrac{54}{T_{end} - T_{start}}\right) & ;\text{otherwise,} \end{cases} \quad (4.5.6.2.6\text{-}6)$$

The shift decision flag, ε_{shift}, is then re-defined depending on the peak-to-average ratio by:

$$\varepsilon_{shift} = \begin{cases} FALSE & ;\left(E_{ratio} < 16.0\right), \\ TRUE & ;\text{otherwise.} \end{cases} \quad (4.5.6.2.6\text{-}7)$$

4.5.6.3 Matching the Residual to the Delay Contour

If the shift decision flag is true (ε_{shift} == $TRUE$) for the current shift frame, update the accumulated shift, τ_{acc}. The accumulated shift is adjusted by the shift required to match the residual shift frame determined in 4.5.6.2 to the modified residual target, $\hat{\varepsilon}_t(n)$. This operation is detailed in 4.5.6.3.1 through 4.5.6.3.4.

4.5.6.3.1 Computation of the Shift Range

The size of the residual frame to be shifted, l, is calculated by:

$$l = T_{end} - T_{start}, \quad (4.5.6.3.1\text{-}1)$$

The residual frame is shifted between T_{srl} and T_{srr}, calculated by:

$$T_{srl} = \begin{cases} S_r + 1 & ;\tau_{acc} < 0, \\ S_r & ;\text{otherwise,} \end{cases} \quad (4.5.6.3.1\text{-}2)$$

$$T_{srr} = \begin{cases} S_r + 1 & ;\tau_{acc} > 0, \\ S_r & ;\text{otherwise,} \end{cases} \quad (4.5.6.3.1\text{-}3)$$

where $S_r = 3$.

For non-periodic signals, T_{srl} and T_{srr} are limited as defined by:

```
if [ ( β < 0.2 ) and ( |τacc| > 15 ) ] or [ ( β < 0.3 ) and ( |τacc| > 30 ) ]
    if ( τacc < 0 )
        Tsrr = 1
    else
        Tsrl = 1
```

For both periodic and non-periodic signals, limit the shift bounds, T_{srl} and T_{srr}, as defined by:

$$T_{srl} = \min\left\{72 - trunc\left\{\tau_{acc}\right\}, T_{srl}\right\}, \quad (4.5.6.3.1\text{-}4)$$

$$T_{srr} = \min\left\{72 + trunc\left\{\tau_{acc}\right\}, T_{srr}\right\}. \quad (4.5.6.3.1\text{-}5)$$

4.5.6.3.2 Generation of a Temporary Modified Residual Signal for Matching

Generate a temporary modified residual, $\hat{\varepsilon}_{tmp}(n)$, by shifting the current residual shift frame, $\varepsilon_s(n)$, by $\tau_{acc}+T_{srl}$:

$$\hat{\varepsilon}_{tmp}(n) = \sum_{i=0}^{2f_l} \hat{\varepsilon}_s(i + T_{start} - T - f_l + n) I_\varepsilon\left(i + \left(2f_l + 1\right)T_l\right) \quad ;0 \le n \le (l + T_{srl} + T_{srr}), \quad (4.5.6.3.2\text{-}1)$$

where T and T_I are defined by:

$$T_E(n) = \begin{cases} round\left\{\tau_{acc} + T_{srl}\right\} & ;\tau_{acc} + T_{srl} > 0, \\ -round\left\{-(\tau_{acc} + T_{srl})\right\} & ;\tau_{acc} + T_{srl} \le 0, \end{cases}$$

$$T_I = trunc\left\{\left(T - \left(\tau_{acc} + T_{srl}\right) + 0.5\right) R + 0.5\right\},$$

```
if ( TI = R ) {
    TI = 0
    T = T - 1
}
```

and l is the shift frame size, $f_l = 3$, $R = 8$, and $\{I_\varepsilon(k)\}$ is a set of interpolation coefficients found in Table B-10.

4.5.6.3.3 Matching the Temporary Modified Residual to the Target Residual

Generate an integer energy correlation vector, $E_I(n)$:

$$E_I = \sum_{i=0}^{l-1} \hat{\varepsilon}_{tmp}(n+i)\hat{\varepsilon}_t(T_{start} + i) \quad ;0 \le n \le T_{srl} + T_{srr}, \quad (4.5.6.3.3\text{-}1)$$

where $\{\hat{\varepsilon}_t\}$ was obtained in 4.5.6.1.

Interpolate $E_I(n)$ to obtain the fractional energy correlation vector, $E_f(n)$:

$$E_f\left(8\left(k-1\right) + j\right) = \sum_{i=-1}^{1} I_f(i,j) E_I(i+K) \quad ;0 \le j \le 7, 0 < T_{srl} + T_{srr}, \quad (4.5.6.3.3\text{-}2)$$

where $\{I_f(i, j)\}$ is a set of interpolation coefficients found in Table B-14.

The optimal shift, τ_{opt}, that will match the temporary modified residual to the target residual is then defined as the index, n, that maximizes $E_f(n)$.

4.5.6.3.4 Adjustment of the Accumulated Shift

The accumulated shift is then adjusted by:

$$\tau_{acc} = \begin{cases} \tau_{acc} - \left\lfloor \dfrac{\tau_{acc} - RT_{srl} + R/2}{R} \right\rfloor & ;\alpha > 0.7, \\ \tau_{acc} & ;\text{otherwise}, \end{cases} \quad (4.5.6.3.4\text{-}1)$$

where R = 8, and the gain, α, of the new shift, τ_{opt}, is calculated by:

$$\alpha = \begin{cases} 0 & ;E_\varepsilon E_T = 0, \\ \dfrac{E_f(\tau_{opt})}{\sqrt{E_\varepsilon E_T}} & ;\text{otherwise}. \end{cases} \quad (4.5.6.3.4\text{-}2)$$

where the energy of the temporary modified residual is calculated by:

$$E_\varepsilon = \sum_{i=T_{srl}}^{l+T_{srl}-1} \hat{\varepsilon}_{tmp}^2(i + T_{start}), \quad (4.5.6.3.4\text{-}3)$$

and the energy of the target modified residual is defined by:

$$E_T = \sum_{i=0}^{l-1} \hat{\varepsilon}_t^2(i + T_{start}). \quad (4.5.6.3.4\text{-}4)$$

4.5.6.4 Modification of the Residual

The current subframe residual, $\varepsilon_s(n)$, is shifted by τ_{acc}, to create the modified residual, $\hat{\varepsilon}(n)$, for the fixed codebook search:

$$\hat{\varepsilon}(n + n_m) = \sum_{i=0}^{2f_l} \varepsilon_s(i + n + n_m - T - f_l) I_E(i + T_I(2f_l + 1)) \quad ;0 \le n \le T_{end} - T_{start}, \quad (4.5.6.4\text{-}1)$$

where T and T_I are calculated by:

$$T = \begin{cases} round\{\tau_{acc}\} & ;\tau_{acc} > 0, \\ -round\{-\tau_{acc}\} & ;\tau_{acc} \le 0, \end{cases}$$

$$T_I = trunc\{(T - \tau_{acc} + 0.5)R + 0.5\},$$

```
if (TI = R) {
    TI = 0
    T = T - 1
}
```

and f_l = 8, R = 8, and $\{I_{\square}(n)\}$ is a set of interpolation coefficients found in Table B-11.

Next, use the following pseudo-code to update n_m and determine whether the pulse searching procedure is complete:

```
if (T_end < L) {
    n_m = T_end
    Go to 4.5.6.2
} else {
    n_m = T_end - L
    Go to 4.5.6.5
}
```

4.5.6.5 Modified Target Residual Update

After having completed the modification of the residual, $\hat{\varepsilon}(n)$, for the entire subframe, the modified target residual buffer, $\hat{\varepsilon}_t(n)$, is updated as follows:

$$\hat{\varepsilon}_t(n) = \begin{cases} \hat{\varepsilon}_t(n+L) & ; -128 \le n < -L \\ \hat{\varepsilon}(n+L) & ; -L \le n < 0 \end{cases}, \qquad (4.5.6.5\text{-}1)$$

4.5.7 Computation of the ACELP Fixed Codebook Contribution

The fixed codebook is based on an algebraic codebook structure, which has advantages in terms of storage, search complexity, and robustness. The codebook structure is based on an interleaved single-pulse permutation (ISPP) design. The codebook is searched on a subframe basis for the best index and gain to minimize the mean-squared weighted error between the original and synthesis speech.

Inputs: The inputs to the algebraic codebook coding routine are:

- The length L impulse response of the weighted synthesis filter, $h_{wq}(k)$, zero extended to length 55
- The length L residual domain target vector, $x(k)$, zero extended to length 55
- The length L perceptual domain target vector, $x_w(k)$, zero extended to length 55
- The average pitch delay for the current subframe, $\dot{\tau}$
- The quantized pitch prediction gain, g_p

Outputs: The outputs of the algebraic codebook coding routine are:

- The codeword of the algebraic codebook, $FCBSIDX(m',i)$; $0 \le i \le 3$
- The fixed codebook excitation vector, $\mathbf{c}_k$
- The fixed codebook gain, g_c

Processing: The algebraic codebook shall be implemented using the procedures described in 4.5.7.1 through 4.5.7.4.

4.5.7.1 Algebraic Codebook Structure, Rate 1

The Rate 1 fixed codebook is a 35-bit algebraic codebook. In this codebook, every codebook vector of length 55 contains at most 8 non-zero pulses. All pulses can have the amplitudes +1 or -1. The 55 positions in a subframe are divided into 5 tracks, as shown in Table 4.5.7.1-1. The algebraic codebook search always examines all the 55 positions as shown, regardless of subframe size. Based on the possible subframe size of 53, 53, and 54, the extra positions are ignored.

Table 4.5.7.1-1. Positions of Individual Pulses in the Rate 1 Algebraic Codebook

Track	Positions
T0	0, 5, 10, 15, 20, 25, 30, 35,40,45,50
T1	1, 6, 11, 16, 21, 26, 31, 36,41,46,51
T2	2, 7, 12, 17, 22, 27, 32, 37,42,47,52
T3	3, 8, 13, 18, 23, 28, 33, 38,43,48,53
T4	4, 9, 14, 19, 24, 29, 34, 39,44,49,54

Of the 5 tracks, 3 are allocated two pulses each and 2 are allocated one pulse each. This accounts for a total of 8 pulses. The single-pulse tracks can be either T3-T4, T4-T0, T0-T1, or T1-T2. The choice of single-pulse tracks is encoded with 2 bits. The positions of the pulses in the 2 single-pulse tracks are encoded with 7 bits (11x11=121<128) and their signs are encoded with 2 bits. For each double-pulse track, both positions and signs of the two pulses are encoded with 8 bits, which will be explained in more detail in 4.5.7.3. This gives a total of 35 bits (2+7+2+8x3).

The codebook vector, $\mathbf{c}_k$, is constructed according to:

$$c_k(j) = \sum_{i=0}^{N_p-1} s_i \delta(j - p_i) \quad ; 0 \le j \le 54 \, , \tag{4.5.7.1-1}$$

where $\delta(j - p_i)$ is a unit pulse at the i-th pulse position p_i of the k-th codevector, s_i is the sign of the i-th pulse, and N_p is the number of pulses, and k is the range of all possible code vectors.

A special feature incorporated in the algebraic codebook is that the selected codebook vector is dynamically shaped by filtering it through an adaptive pre-filter. In this implementation, the prefilter:

$$F(z) = \begin{cases} 1 & ;\dot{\tau} \ge L, \\ \dfrac{1}{1 - g_p z^{-\dot{\tau}}} & ;\text{otherwise}, \end{cases} \tag{4.5.7.1-2}$$

shall be used, where $\dot{\tau}$ is the average subframe pitch delay and g_p is the pitch gain. The pitch gain is the quantized pitch gain in the current subframe bounded by [0.2, 0.9]. For delays less than 55, the codebook vector, $\mathbf{c}_k$, is modified according to:

$$c_k(j) = \begin{cases} c_k(j) & ; 0 \le j < \tau, \\ c_k(j) + g_p c_k(j-\tau) & ; \tau \le j \le 54. \end{cases} \quad (4.5.7.1\text{-}3)$$

This modification is incorporated in the fixed codebook search by including the prefilter in the impulse response $h_{wq}(j)$. That is, prior to codebook search, the impulse response $h_{wq}(j)$ shall be modified according to:

$$h_{wq}(j) = \begin{cases} h_{wq}(j) & ; 0 \le j < \tau, \\ h_{wq}(j) + g_p h_{wq}(j-\tau) & ; \tau \le j \le 54. \end{cases} \quad (4.5.7.1\text{-}4)$$

4.5.7.2 Algebraic Codebook Search

The algebraic codebook is searched by minimizing the mean-squared error between the weighted input speech and the weighted synthesis speech. The perceptual domain target signal $x_w(n)$ is used in the closed-loop fixed-codebook search and is given by Equation 4.5.4.10.1-1.

Let $\mathbf{c}_k$ be the algebraic codebook vector at index k. The algebraic codebook is searched by maximizing the term:

$$T_k = \frac{C_k}{E_k} = \frac{\left(\mathbf{d}^2 \mathbf{c}_k\right)^2}{\mathbf{c}_k^t \Phi \mathbf{c}_k}, \quad (4.5.7.2\text{-}1)$$

where $\mathbf{d} = \mathbf{H}^t \mathbf{x}_w$ is the cross-correlation between the perceptual domain target signal $x_w(n)$ and the impulse response $h_{wq}(n)$, $\Phi = \mathbf{H}^t\mathbf{H}$ is the correlation matrix of the impulse response $h_{wq}(n)$, and $\mathbf{H}$ is a lower triangular Toeplitz matrix with diagonal $h_{wq}(0)$ and lower diagonals $h_{wq}(1)$,..., $h_{wq}(54)$, i.e.:

$$\mathbf{H} = \begin{bmatrix} h_{wq}(0) & 0 & 0 & \cdots & 0 \\ h_{wq}(1) & h_{wq}(0) & 0 & \cdots & 0 \\ h_{wq}(2) & h_{wq}(1) & h_{wq}(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{wq}(54) & h_{wq}(53) & h_{wq}(52) & \cdots & h_{wq}(0) \end{bmatrix}. \quad (4.5.7.2\text{-}2)$$

The cross-correlation vector $\mathbf{d}$ and the matrix Φ are computed prior to the codebook search. The elements of the vector $\mathbf{d}$ are computed by:

$$d(n) = \sum_{j=n}^{54} x_w(j) h_{wq}(j-n) \quad ; 0 \le n \le 54, \quad (4.5.7.2\text{-}3)$$

and the (i, j)-th element of the symmetric matrix Φ is computed by:

$$\phi(i,j)=\begin{cases}\sum_{n=\max\{i,j\}}^{L-1} h_{wq}(n-i)h_{wq}(n-j) & ;(0\le j<L)\, and\, (0\le i<L)\\ 0 & ;(L\le j<55)\, and\, (L\le i<55)\end{cases}. \quad (4.5.7.2\text{-}4)$$

The algebraic structure of the codebook allows for very fast search procedures since the innovation vector, $\mathbf{c}_k$, contains only a few non-zero pulses. The correlation in the numerator of Equation 4.5.7.2-1 is given by:

$$c_k=\left(\sum_{i=0}^{N_p-1} s_i d(p_i)\right)^2. \quad (4.5.7.2\text{-}5)$$

The energy in the denominator of Equation 4.5.7.2-1 is given by:

$$E_k=\sum_{i=0}^{N_p-1}\phi(p_i,p_i)+2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1} s_i s_j \phi(p_i,p_j). \quad (4.5.7.2\text{-}6)$$

In order to determine the optimal algebraic codebook vector which maximizes the term in Equation 4.5.7.2-1, the correlation and energy terms in Equation 4.5.7.2-5 and Equation 4.5.7.2-6 should be computed for all possible combinations of pulse positions and signs. This, however, is a prohibitive task. In order to simplify the search, two strategies for searching the pulse signs and positions as explained below shall be used.

4.5.7.2.1 Pre-setting of Pulse Signs

In order to simplify the search procedure, the pulse signs are preset (outside the closed loop search) by considering the sign of an appropriate reference signal. In this case, the signal $e(i)$, given by:

$$e_i=\sqrt{\frac{\sum_{j=0}^{54} d^2(j)}{\sum_{j=0}^{54} x^2(j)}}x(i)+2d(i); \qquad 0\le \mathrm{i}\le 54, \quad (4.5.7.2.1\text{-}1)$$

shall be used, where $x(i)$ is the residual domain target vector which is described in 4.5.4.10.

Amplitude pre-setting shall be done by setting the amplitude of a pulse at position i equal to the sign of $e(i)$. Hence, once the sign signal $s_i = \mathrm{sign}\{e(i)\}$ and the signal $d'(i) = d(i)s_i$ are computed, then the matrix Φ shall be modified by including the sign information, that is, $\phi'(i,j) = s_i s_j \phi(i,j)$. The correlation in Equation 4.5.7.2-5 is now given by:

$$c_k=\left(\sum_{i=0}^{N_p-1} d'(p_i)\right)^2, \quad (4.5.7.2.1\text{-}2)$$

and the energy in Equation 4.5.7.2-6 is given by:

$$E_k=\sum_{i=0}^{N_p-1}\phi'(p_i,p_i)+2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1}\phi'(p_i,p_j). \quad (4.5.7.2.1\text{-}3)$$

4.5.7.2.2 Non-Exhaustive Pulse Position Search

Having preset the pulse amplitudes, as explained in 4.5.7.2.1, the optimal pulse positions shall be determined using an efficient non-exhaustive analysis-by-synthesis search technique. In this technique, the term in Equation 4.5.7.2-1 is tested for a small percentage of position combinations, using an iterative "depth-first" tree search strategy. In this approach, the 8 pulses are grouped into 4 pairs of pulses. The pulse positions shall be determined sequentially one pair at a time. In the first iteration, the single-pulse tracks are T3 and T4. The search process shall be repeated for other 3 iterations, by assigning the single-pulse tracks to T4-T0, T0-T1, and T1-T2 respectively. The codeword, *q*, used to represent the various chosen tracks are given in Table 4.5.7.2.2-1.

Table 4.5.7.2.2-1. Codeword for the Track Orders

Double-pulse Track Order $(p_0, p_1), (p_2, p_3), (p_4, p_5)$	**Single-pulse Track Order** (p_6, p_7)	**Codeword** (q)
T0-T1-T2	T3-T4	00
T1-T2-T3	T4-T0	01
T2-T3-T4	T0-T1	10
T3-T4-T0	T1-T2	11

Once the positions and signs of the excitation pulses are determined, the codebook vector, $\mathbf{c}_k$, shall be built as in Equation 4.5.7.1-1 and shall be modified according to Equation 4.5.7.1-3.

4.5.7.3 Codeword Computation of the Algebraic Codebook

Upon completion of algebraic codebook search, the positions and signs of the pulses are encoded into 35 bits. The algebraic codebook routine outputs 4 indices: the first 3 indices (8 bits each) represent the positions and signs of each double-pulse track and the 4th index (11 bits) represents the positions and signs of pulses in the two single-pulse tracks (9 bits) along with the codeword for the two single-pulse tracks (2 bits). For the codeword computations, the following variables are defined for each of the single and double pulse tracks:

$$\lambda_0 = \lfloor p_i / 5 \rfloor; \quad \mathrm{i} \in \{0,2,4,6\}, \qquad (4.5.7.3\text{-}1)$$

$$\lambda_1 = \lfloor p_j / 5 \rfloor; \quad \mathrm{j} \in \{1,3,5,7\}. \qquad (4.5.7.3\text{-}2)$$

These are the decimated pulse positions and $s_{\lambda 0}$ and $s_{\lambda 1}$ are the signs of the respective decimated positions, and $b_{\lambda 0}$ and $b_{\lambda 1}$ are defined as:

$$b_{\lambda i} = \begin{cases} 0 & ;\mathrm{s}_{\lambda i} > 0, \\ 1 & ;\mathrm{s}_{\lambda i} < 0 \end{cases} \quad 0 \le \mathrm{i} \le 1. \qquad (4.5.7.3\text{-}3)$$

In the case of coding the 3 double-pulse tracks, each 8 bit codeword *FCBSIDX*(*m'*,*i*), is $(0 \le \mathrm{i} \le 2)$ calculated according to:

$$FCBSIDX(m',i) = \begin{cases} b_{\lambda k} \times 128 + \min\{\lambda_0,\lambda_1\} \times 11 + \max\{\lambda_0,\lambda_1\}; & \text{if } s_{\lambda 0} = s_{\lambda 1}, \\ b_{\lambda k} \times 128 + \max\{\lambda_0,\lambda_1\} \times 11 + \min\{\lambda_0,\lambda_1\}; & \text{if } s_{\lambda 0} = -s_{\lambda 1}, \end{cases} \quad (4.5.7.3\text{-}4)$$

where $b_{\lambda k}$ follows the sign of the pulse located at the larger decimated position, i.e., $\max\{\lambda_0, \lambda_1\}$. The three double-pulse codewords $FCBSIDX(m',i)$, $0 \le i \le 2$, are packed according to the order specified in the Table 4.5.7.2.2-1, which depends on the positions of the single-pulse tracks.

In the case of coding the two single-pulse tracks, the 11 bit codeword $FCBSIDX(m',3)$ is calculated according to:

$$FCBSIDX(m',3) = q \times 512 + b_{\lambda 0} \times 256 + b_{\lambda 1} \times 128 + \lambda_0 \times 11 + \lambda_1, \quad (4.5.7.3\text{-}5)$$

where $0 \le q \le 3$ is the single-pulse track codeword in Table 4.5.7.2.2-1.

4.5.7.4 Algebraic Codebook Structure, Rate 1/2

A 10 bit algebraic codebook is used for Rate 1/2 packets. The innovation vector contains 3 non-zero pulses. Each pulse has 8 possible positions, which are coded by 3 bits. The pulse positions and corresponding codewords are given in Table 4.5.7.4-1. All pulses have a fixed signs (+1 for T_0 and T_2 and -1 for T_1). An additional bit, however, is used to change the signs of all three pulses simultaneously, i.e., s = 1 indicates polarity inversion, s = 0 otherwise. Therefore, the total bits per subframe for Rate 1/2 is 3*3+1=10.

The codebook is searched using techniques similar to that for Rate 1. However, in the Rate 1/2 case, all the pulse position combinations are exhaustively searched by maximizing Equation 4.5.7.2-1. Upon completion of the search, the sign information is processed as follows:

```
if ( sign{ d^t c_k } < 0 ) {
    c_k = -c_k
    s = 1
} else {
    s = 0
}
```

By referencing Table 4.5.7.4-1, the 10-bit codeword can then be calculated by:

$$FCBSIDX(m',0) = s \times 512 + q_{T0} \times 64 + q_{T1} \times 8 + q_{T2}\ , \quad (4.5.7.4\text{-}1)$$

Table 4.5.7.4-1. Positions of Individual Pulses in the Rate 1/2 Algebraic Codebook

Pulse	Positions							
T_0	0	7	14	21	28	35	42	49
T_1	2	9	16	23	30	37	44	51
T_2	4	11	18	25	32	39	46	53
Codewords q_{Ti} ; $0 \le i \le 2$	000	001	010	011	100	101	110	111

4.5.7.5 Fixed Codebook Gain Calculation

The fixed codebook gain for both Rate 1 and Rate 1/2 shall be found by:

$$g_c = \frac{\mathbf{d}^t \mathbf{c}_k}{\mathbf{c}_k^t \Phi \mathbf{c}_k}, \qquad (4.5.7.5\text{-}1)$$

where all variables are previously defined.

4.6 Encoding at Rate 1/8

Inputs: The inputs to Rate 1/8 encoding are:

- The unquantized LSPs from the current and previous frames, $\Omega(m)$ and $\Omega(m\text{-}1)$
- The quantized LSPs from the previous frame, $\Omega_q(m\text{-}1)$
- The short-term prediction residual, $\varepsilon(n)$

Outputs: The outputs of Rate 1/8 encoding are:

- The quantized LSPs for the current frame, $\Omega_q(m)$
- The quantization index corresponding to these LSPs, *LSPIDX*(k)
- The vector quantized frame energy index, *FGIDX*

State Variables Affected:

- The accumulated shift counter, τ_{acc}, is set to 0
- The pointer to the last element in shifted residual, n_m, is set to 0

Processing: Rate 1/8 encoding shall comprise 4.6.1 to 4.6.8.

4.6.1 LSP Quantization

Vector quantize the unquantized LSPs, $\Omega(m)$, using the procedure found in 4.4 for Rate 1/8.

4.6.2 Interpolation of LSP Parameters

Interpolate the quantized LSPs over the three subframes, m', in the current frame, m, as defined in 4.2.2.1.

4.6.3 LSP to LPC Conversion

Convert the quantized, interpolated LSPs, $\dot{\Omega}_q(m)$, to quantized, interpolated LPC parameters, $\left\{\dot{a}_q\right\}$, as described in Section 4.2.2.2 for each subframe.

4.6.4 Impulse Response Computation

Calculate the unweighted impulse response, $h(n)$, of $1/\dot{A}_q(z)$ to 54 terms for each subframe, where $\dot{A}_q(z)$ is defined as:

$$\dot{A}_q(z) = 1 - \sum_{i=1}^{10} \dot{a}_q(i) z^{-i} . \qquad (4.6.4\text{-}1)$$

4.6.5 Calculation of the Frame Energy Gain

The frame energy gain is defined as the ratio of the energy of the impulse response to the mean of the residual for each subframe m':

$$\gamma(m') = \frac{E_\varepsilon(m')}{E_h(m')} , \qquad (4.6.5\text{-}1)$$

where the energy of the impulse response is given by:

$$E_h(m') = \sqrt{\sum_{i=0}^{L-1} h^2(n)} , \qquad (4.6.5\text{-}2)$$

and the mean $E_\varepsilon(m')$ of the residual signal $\varepsilon(n)$ is defined as:

$$E_\varepsilon(m') = \max\left\{1, \frac{1}{L} \sum_{i=0}^{L-1} \left|\varepsilon(n)\right|\right\} , \qquad (4.6.5\text{-}3)$$

where L is the subframe size (53 for subframes 0 and 1, 54 for subframe 2), $n = 0$ is defined as the index of the first sample in the current subframe, and $\varepsilon(n)$ is the residual at the current subframe.

4.6.6 Gain Quantization

The gain vector, $\gamma(m')$, shall be quantized to 8 bits using a vector quantizer. The quantizer will assign one index for the best three-element gain vector corresponding to $\gamma(m')$. The best vector is found by calculating the error vector, $e_g(k)$, as defined by:

$$e_g(k) = \sum_{m'=0}^{2} \left(\log_{10}\left(\gamma(m')\right) - q_{\log}\left(m', k\right)\right)^2 , \qquad (4.6.6\text{-}1)$$

where $q_{log}(m',k)$ is an entry in the gain quantization codebook found in Table B-15. The best codebook index, *FGIDX*, is defined as the index, k, at which $e_g(k)$ is minimized. This codebook index, *FGIDX*, shall be used to determine a set of quantized gains $\gamma_q(m')$:

$$\gamma_q(m') = 10^{q\log(m', FGIDX)}; \quad 0 \le m' < 3 . \qquad (4.6.6\text{-}2)$$

4.6.7 Generation of Rate 1/8 Excitation

The excitation for each subframe at Rate 1/8 is generated by using a zero-mean, unit variance pseudo-Gaussian white noise sequence which is scaled by the quantized frame energy gain $\gamma_q(m')$ for each subframe m':

$$E(n) = \gamma_q ran\_g\left\{seed\right\}; \quad 0 \le n < L , \qquad (4.6.7\text{-}1)$$

where ran_g{*seed*} is the unit variance pseudo-random Gaussian white noise generator (see 4.7.2) and *seed* is a unique seed value defined at reset of the system.

4.6.8 Perceptual Weighting Filter Update

Although there is no closed-loop search at Rate 1/8, it is still necessary to update the memory of the perceptual weighting filter with the new excitation determined in 4.6.7 for each subframe. Update the weighted synthesis filter memory by filtering the excitation vector, E(n), through the weighted synthesis filter, $H_{wq}(z)$, which is given in Equation 4.5.4.3-1.

4.7 Random Number Generation

Zero mean, unit variance gaussian pseudo-random numbers are obtained by generating uniform pseudo-random numbers and appropriately transforming them. The following two sub-sections describe the uniform pseudo-random number generator and the algorithm for effecting the uniform to gaussian transformation.

4.7.1 Uniform Pseudo-Random Number Generation Algorithm

The algorithm for generating the uniform pseudo-random numbers is initialized with a seed value, and produces a new seed value with each successive invocation, as well as producing the desired pseudo-random number. The state of the uniform random number generator is captured by the current value of the seed. Different modules making use of the random number generator should maintain their own seeds. Uniform pseudo-random numbers are generated as follows:

Inputs: The input to the uniform number generator is:

- The seed value, *seed0*

Outputs: The outputs of the uniform pseudo-random number generator are:

- The uniform pseudo-random number, *ran0*
- The modified seed value, *newseed0*

Processing: Uniform pseudo-random number generation is accomplished as described in the following pseudo-code:

```
newseed0 = seed0 ⊕ 23148373
temp = trunc(newseed0 / 127773)
newseed0 = 16807 * (newseed0 - temp * 127773) - 2836 * temp
if(newseed0 < 0)
    newseed0 = newseed0 + 2147483647
ran0 = newseed0 / 2147483647
newseed0 = seed0 ⊕ 23148373
```

4.7.2 Gaussian Pseudo-Random Number Generator

The gaussian pseudo-random number generator makes use of the uniform pseudo-random number generator described in 4.7.1. Gaussian pseudo-random numbers are generated in pairs. While the algorithm that performs the transformation from uniform to gaussian does not itself have any memory, the values produced by it are a function of the seed value, which is used in invocations of the uniform pseudo-random number generator described in 4.7.1. Consequently, it is important for each module that makes use of the random number generator to maintain its own seed value. The transformation from uniform to gaussian is described as follows:

Inputs: The input to the gaussian pseudo-random number generator is

- The seed value, *seed*

Outputs: The outputs of the gaussian pseudo-random number generator are:

- The pair of gaussian pseudo-random numbers generated, *ran_g0* and *ran_g1*
- The new value of the seed, *newseed*.

Processing: Generation of gaussian pseudo-random numbers is described in the following pseudocode. Note that *ran_u0* and *ran_u1* are the uniform pseudo-random numbers produced by two different invocations of the algorithm of 4.7.1, and *newseed0* and *newseed1* are the new seed values produced by each of these respective invocations. Each invocation of the uniform pseudo-random number generator uses the current value of *seed* as its input.

```
do {
        v1 = 2.0 * ran_u0  - 1.0
        seed = newseed0
        v2 = 2.0 * ran_u1  - 1.0
        seed = newseed1
        rsq = v1 * v1 + v2 * v2
} while (rsq >= 1.0 OR rsq == 0.0)
fac = (-2.0 * log(rsq) / rsq)^(1/2)
ran_g0 = v1 * fac
ran_g1= v2* fac
```

4.8 Packet Formatting

After encoding, the encoded speech packets shall be formatted as described in Table 4.8-1. For each parameter, bit index 0 corresponds to the most significant bit. The packet buffer bit index 1 indicates the bit position corresponding to the beginning of the packet buffer. Parameter notations are defined as follows:

- *LPCFLAG*: spectral change indicator,
- *LSPIDX*(*k*)[*j*]: the *j*-th bit of the *k*-th LSP codebook for the entire frame, (4 codebooks for Rate 1, 3 for Rate 1/2, and 2 for Rate 1/8).
- *DELAY*[*j*]: the *j*-th bit of the pitch delay estimate for the entire frame,
- *DDELAY*[*j*]: the *j*-th bit of the delay difference for the entire frame,
- *ACBGIDX*(*m'*)[*j*]: the *j*-th bit of the adaptive codebook gain index for the *m'*-th subframe,
- *FCBSIDX*(*m',k*)[*j*]: the *j*-th bit of the *k*-th fixed codebook shape index for the *m'*-th subframe, (4 codebooks for Rate 1, 1 for Rate 1/2),
- *FCBGIDX*(*m'*)[*j*]: the j-th bit of the fixed codebook gain index for the *m'*-th subframe,
- *FGIDX*[*j*]: the *j*-th bit of the frame energy gain

In the case where a Rate 1/8 packet with all bits set to '1' (null Traffic Channel data) is generated, *FGIDX*[0] is set to zero.

Table 4.8-1. Packet Formats

Bit Index	Rate 1 Packet Bits	Rate 1/2 Packet Bits	Rate 1/8 Packet Bits
1	*LPCFLAG*	*LSPIDX*(1)[0]	*LSPIDX*(1)[0]
2	*LSPIDX*(1)[0]	*LSPIDX*(1)[1]	*LSPIDX*(1)[1]
3	*LSPIDX*(1)[1]	*LSPIDX*(1)[2]	*LSPIDX*(1)[2]
4	*LSPIDX*(1)[2]	*LSPIDX*(1)[3]	*LSPIDX*(1)[3]
5	*LSPIDX*(1)[3]	*LSPIDX*(1)[4]	*LSPIDX*(2)[0]
6	*LSPIDX*(1)[4]	*LSPIDX*(1)[5]	*LSPIDX*(2)[1]
7	*LSPIDX*(1)[5]	*LSPIDX*(1)[6]	*LSPIDX*(2)[2]
8	*LSPIDX*(2)[0]	*LSPIDX*(2)[0]	*LSPIDX*(2)[3]
9	*LSPIDX*(2)[1]	*LSPIDX*(2)[1]	*FGIDX*[0]
10	*LSPIDX*(2)[2]	*LSPIDX*(2)[2]	*FGIDX*[1]
11	*LSPIDX*(2)[3]	*LSPIDX*(2)[3]	*FGIDX*[2]
12	*LSPIDX*(2)[4]	*LSPIDX*(2)[4]	*FGIDX*[3]
13	*LSPIDX*(2)[5]	*LSPIDX*(2)[5]	*FGIDX*[4]
14	*LSPIDX*(3)[0]	*LSPIDX*(2)[6]	*FGIDX*[5]
15	*LSPIDX*(3)[1]	*LSPIDX*(3)[0]	*FGIDX*[6]
16	*LSPIDX*(3)[2]	*LSPIDX*(3)[1]	*FGIDX*[7]
17	*LSPIDX*(3)[3]	*LSPIDX*(3)[2]	
18	*LSPIDX*(3)[4]	*LSPIDX*(3)[3]	
19	*LSPIDX*(3)[5]	*LSPIDX*(3)[4]	
20	*LSPIDX*(3)[6]	*LSPIDX*(3)[5]	
21	*LSPIDX*(3)[7]	*LSPIDX*(3)[6]	
22	*LSPIDX*(3)[8]	*LSPIDX*(3)[7]	
23	*LSPIDX*(4)[0]	*DELAY*[0]	
24	*LSPIDX*(4)[1]	*DELAY*[1]	
25	*LSPIDX*(4)[2]	*DELAY*[2]	
26	*LSPIDX*(4)[3]	*DELAY*[3]	
27	*LSPIDX*(4)[4]	*DELAY*[4]	
28	*LSPIDX*(4)[5]	*DELAY*[5]	
29	*LSPIDX*(4)[6]	*DELAY*[6]	
30	*DELAY*[0]	*ACBGIDX*(0)[0]	
31	*DELAY*[1]	*ACBGIDX*(0)[1]	
32	*DELAY*[2]	*ACBGIDX*(0)[2]	
33	*DELAY*[3]	*FCBSIDX*(0,0)[0]	
34	*DELAY*[4]	*FCBSIDX*(0,0)[1]	
35	*DELAY*[5]	*FCBSIDX*(0,0)[2]	
36	*DELAY*[6]	*FCBSIDX*(0,0)[3]	
37	*DDELAY*[0]	*FCBSIDX*(0,0)[4]	
38	*DDELAY*[1]	*FCBSIDX*(0,0)[5]	
39	*DDELAY*[2]	*FCBSIDX*(0,0)[6]	
40	*DDELAY*[3]	*FCBSIDX*(0,0)[7]	
41	*DDELAY*[4]	*FCBSIDX*(0,0)[8]	
42	*ACBGIDX*(0)[0]	*FCBSIDX*(0,0)[9]	
43	*ACBGIDX*(0)[1]	*FCBGIDX*(0)[0]	
44	*ACBGIDX*(0)[2]	*FCBGIDX*(0)[1]	
45	*FCBSIDX*(0,0)[0]	*FCBGIDX*(0)[2]	
46	*FCBSIDX*(0,0)[1]	*FCBGIDX*(0)[3]	
47	*FCBSIDX*(0,0)[2]	*ACBGIDX*(1)[0]	
48	*FCBSIDX*(0,0)[3]	*ACBGIDX*(1)[1]	

49	*FCBSIDX*(0,0)[4]	*ACBGIDX*(1)[2]	
50	*FCBSIDX*(0,0)[5]	*FCBSIDX*(1,0)[0]	
51	*FCBSIDX*(0,0)[6]	*FCBSIDX*(1,0)[1]	
52	*FCBSIDX*(0,0)[7]	*FCBSIDX*(1,0)[2]	
53	*FCBSIDX*(0,1)[0]	*FCBSIDX*(1,0)[3]	
54	*FCBSIDX*(0,1)[1]	*FCBSIDX*(1,0)[4]	
55	*FCBSIDX*(0,1)[2]	*FCBSIDX*(1,0)[5]	
56	*FCBSIDX*(0,1)[3]	*FCBSIDX*(1,0)[6]	
57	*FCBSIDX*(0,1)[4]	*FCBSIDX*(1,0)[7]	
58	*FCBSIDX*(0,1)[5]	*FCBSIDX*(1,0)[8]	
59	*FCBSIDX*(0,1)[6]	*FCBSIDX*(1,0)[9]	
60	*FCBSIDX*(0,1)[7]	*FCBGIDX*(1)[0]	
61	*FCBSIDX*(0,2)[0]	*FCBGIDX*(1)[1]	
62	*FCBSIDX*(0,2)[1]	*FCBGIDX*(1)[2]	
63	*FCBSIDX*(0,2)[2]	*FCBGIDX*(1)[3]	
64	*FCBSIDX*(0,2)[3]	*ACBGIDX*(2)[0]	
65	*FCBSIDX*(0,2)[4]	*ACBGIDX*(2)[1]	
66	*FCBSIDX*(0,2)[5]	*ACBGIDX*(2)[2]	
67	*FCBSIDX*(0,2)[6]	*FCBSIDX*(2,0)[0]	
68	*FCBSIDX*(0,2)[7]	*FCBSIDX*(2,0)[1]	
69	*FCBSIDX*(0,3)[0]	*FCBSIDX*(2,0)[2]	
70	*FCBSIDX*(0,3)[1]	*FCBSIDX*(2,0)[3]	
71	*FCBSIDX*(0,3)[2]	*FCBSIDX*(2,0)[4]	
72	*FCBSIDX*(0,3)[3]	*FCBSIDX*(2,0)[5]	
73	*FCBSIDX*(0,3)[4]	*FCBSIDX*(2,0)[6]	
74	*FCBSIDX*(0,3)[5]	*FCBSIDX*(2,0)[7]	
75	*FCBSIDX*(0,3)[6]	*FCBSIDX*(2,0)[8]	
76	*FCBSIDX*(0,3)[7]	*FCBSIDX*(2,0)[9]	
77	*FCBSIDX*(0,3)[8]	*FCBGIDX*(2)[0]	
78	*FCBSIDX*(0,3)[9]	*FCBGIDX*(2)[1]	
79	*FCBSIDX*(0,3)[10]	*FCBGIDX*(2)[2]	
80	*FCBGIDX*(0)[0]	*FCBGIDX*(2)[3]	
81	*FCBGIDX*(0)[1]		
82	*FCBGIDX*(0)[2]		
83	*FCBGIDX*(0)[3]		
84	*FCBGIDX*(0)[4]		
85	*ACBGIDX*(1)[0]		
86	*ACBGIDX*(1)[1]		
87	*ACBGIDX*(1)[2]		
88	*FCBSIDX*(1,0)[0]		
89	*FCBSIDX*(1,0)[1]		
90	*FCBSIDX*(1,0)[2]		
91	*FCBSIDX*(1,0)[3]		
92	*FCBSIDX*(1,0)[4]		
93	*FCBSIDX*(1,0)[5]		
94	*FCBSIDX*(1,0)[6]		
95	*FCBSIDX*(1,0)[7]		
96	*FCBSIDX*(1,1)[0]		
97	*FCBSIDX*(1,1)[1]		
98	*FCBSIDX*(1,1)[2]		
99	*FCBSIDX*(1,1)[3]		
100	*FCBSIDX*(1,1)[4]		
101	*FCBSIDX*(1,1)[5]		
102	*FCBSIDX*(1,1)[6]		

103	*FCBSIDX*(1,1)[7]		
104	*FCBSIDX*(1,2)[0]		
105	*FCBSIDX*(1,2)[1]		
106	*FCBSIDX*(1,2)[2]		
107	*FCBSIDX*(1,2)[3]		
108	*FCBSIDX*(1,2)[4]		
109	*FCBSIDX*(1,2)[5]		
110	*FCBSIDX*(1,2)[6]		
111	*FCBSIDX*(1,2)[7]		
112	*FCBSIDX*(1,3)[0]		
113	*FCBSIDX*(1,3)[1]		
114	*FCBSIDX*(1,3)[2]		
115	*FCBSIDX*(1,3)[3]		
116	*FCBSIDX*(1,3)[4]		
117	*FCBSIDX*(1,3)[5]		
118	*FCBSIDX*(1,3)[6]		
119	*FCBSIDX*(1,3)[7]		
120	*FCBSIDX*(1,3)[8]		
121	*FCBSIDX*(1,3)[9]		
122	*FCBSIDX*(1,3)[10]		
123	*FCBGIDX*(1)[0]		
124	*FCBGIDX*(1)[1]		
125	*FCBGIDX*(1)[2]		
126	*FCBGIDX*(1)[3]		
127	*FCBGIDX*(1)[4]		
128	*ACBGIDX*(2)[0]		
129	*ACBGIDX*(2)[1]		
130	*ACBGIDX*(2)[2]		
131	*FCBSIDX*(2,0)[0]		
132	*FCBSIDX*(2,0)[1]		
133	*FCBSIDX*(2,0)[2]		
134	*FCBSIDX*(2,0)[3]		
135	*FCBSIDX*(2,0)[4]		
136	*FCBSIDX*(2,0)[5]		
137	*FCBSIDX*(2,0)[6]		
138	*FCBSIDX*(2,0)[7]		
139	*FCBSIDX*(2,1)[0]		
140	*FCBSIDX*(2,1)[1]		
141	*FCBSIDX*(2,1)[2]		
142	*FCBSIDX*(2,1)[3]		
143	*FCBSIDX*(2,1)[4]		
144	*FCBSIDX*(2,1)[5]		
145	*FCBSIDX*(2,1)[6]		
146	*FCBSIDX*(2,1)[7]		
147	*FCBSIDX*(2,2)[0]		
148	*FCBSIDX*(2,2)[1]		
149	*FCBSIDX*(2,2)[2]		
150	*FCBSIDX*(2,2)[3]		
151	*FCBSIDX*(2,2)[4]		
152	*FCBSIDX*(2,2)[5]		
153	*FCBSIDX*(2,2)[6]		
154	*FCBSIDX*(2,2)[7]		
155	*FCBSIDX*(2,3)[0]		

156	*FCBSIDX*(2,3)[1]		
157	*FCBSIDX*(2,3)[2]		
158	*FCBSIDX*(2,3)[3]		
159	*FCBSIDX*(2,3)[4]		
160	*FCBSIDX*(2,3)[5]		
161	*FCBSIDX*(2,3)[6]		
162	*FCBSIDX*(2,3)[7]		
163	*FCBSIDX*(2,3)[8]		
164	*FCBSIDX*(2,3)[9]		
165	*FCBSIDX*(2,3)[10]		
166	*FCBGIDX*(2)[0]		
167	*FCBGIDX*(2)[1]		
168	*FCBGIDX*(2)[2]		
169	*FCBGIDX*(2)[3]		
170	*FCBGIDX*(2)[4]		
171	TTY Baud Rate Bit		

5 SPEECH DECODER

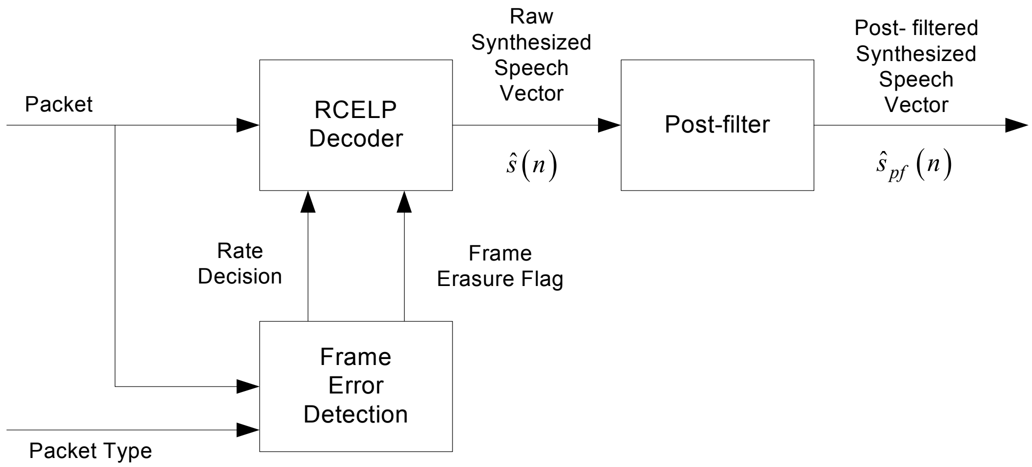

Figure 5-1. Speech Decoder Top-Level Diagram

Figure 5-1 presents a top-level view of the EVRC speech decoder. The inputs to the decoder are the received speech packet, and a packet type indicator from the multiplex sub-layer. The frame error detection module uses the packet type indicator to determine the data rate and whether or not there was a frame error detected by the multiplex sub-layer. The decoder also applies rules to detect some channel errors not detected by the multiplex sub-layer.

The decoder uses the parameters contained in the received packet to re-synthesize the speech frame based on the rate decision. It uses the frame erasure flag to trigger frame error recovery logic. The raw synthesized speech is then post-filtered and output.

5.1 Frame Error Detection

Inputs: The inputs to frame error detection are:

The packet type supplied by the multiplex sublayer

- The delay transmission code, *DELAY*

Outputs: The outputs of frame error detection are:

- The frame erasure flag, *FER_FLAG*(*m*)
- The rate of operation for the decoder, *Rate*

Processing: The frame error detector shall comprise 5.1.1 and 5.1.2.

Initialization:

- The last valid rate of operation, *last_valid_rate*, is initialized to Rate 1/8
- The frame erasure flag, *FER_FLAG*(*m*) = FALSE ; *m* = 0

5.1.1 Received Packet Type Processing

The received packet type from the multiplex sublayer (see 2.1.2) is used to generate the decoder rate of operation, *Rate*, as well as the frame erasure flag, *FER_FLAG*(*m*). *FER_FLAG*(*m*), is defined for each received packet type in Table 5.1.1-1.

Table 5.1.1-1 Received Packet Type Decoding

Packet Type	*FER_FLAG*(*m*)
Rate 1	FALSE
Rate 1/2	FALSE
Rate 1/4	TRUE
Rate 1/8	FALSE
Blank	TRUE
Rate 1 with bit errors	TRUE
Insufficient frame quality (erasure)	TRUE
Rate 1/8 with all bits set to '1'	TRUE

The decoder rate of operation for the current frame, *Rate*, is then defined by the pseudo-code:

```
if ( FER_FLAG(m) == TRUE ) {
    if (last_valid_rate == Rate 1/8) {
        Rate = Rate 1/8
    }
    else {
        Rate = Rate 1
    }
}
if ( (Rate == Rate 1/8) and (last_valid_rate == Rate 1) and
    (FER_FLAG(m-1) == FALSE) ) {
        FER_FLAG(m) = TRUE
        Rate = Rate 1
}
if ( FER_FLAG(m) == FALSE ) {
    Rate = Rate from received packet type
    last_valid_rate = Rate
}
```

5.1.2 Delay Parameter Checking

If *Rate* is determined to be Rate 1/2 or 1, and the delay transmission code, *DELAY*, is greater than 100, the *FER_FLAG* (*m*) flag shall be set to TRUE.

5.1.3 Delta Delay Parameter Checking

If *Rate* is determined to be Rate 1, the delta delay transmission code, DDELAY, shall be sanity checked. The last frame's delay shall be computed according to Eq. 5.2.2.2.1-2. If the computed delay is less than DMIN or greater than DMAX, then the *FER_FLAG(m)* flag shall be set to TRUE.

5.2 Rate 1/2 and 1 Decoding

Inputs: The inputs to Rate 1/2 and 1 decoding are:

- The quantized LSP indices from the current frame, *LSPIDX*(k)
- The quantized LSPs from the previous frame, $\Omega_q(m-1)$
- The adaptive codebook gain index, *ACBGIDX*(m'), for each subframe, m',
- The fixed codebook shape indices, *FCBSIDX*(m', k), for each subframe, m',
- The fixed codebook gain indices, *FCBGIDX*(m', k), for each subframe, m',
- The delay transmission code, *DELAY*
- The spectral transition indicator, *LPCFLAG*, Rate 1 only
- The delay difference, *DDELAY*, Rate 1 only
- The frame erasure flag, *FER_FLAG*(m), for the current frame, m

Outputs: The outputs from Rate 1/2 and 1 decoding are:

- The post-filtered synthesized speech signal, $\hat{s}_{pf}(n)$

State Variables:

- The adaptive codebook excitation, E(n)

Initialization:

- The adaptive codebook excitation memory, E(n) = 0; $-128 \le n < 64$
- The memory for the last valid adaptive codebook buffer, $\{E_{ly}(n)\} = 0$; $0 \le n < 128$
- The delay, $\tau(m) = 40$; $m = 0$
- The quantized LSPs, $\Omega_q(m,k) = 0.048k$; $1 \le k \le 10$, $m = 0$
- The codebook and pitch gains for the last valid frame (see 5.2.3.5 and 5.2.3.6)
  $g_c(m') = g_p(m') = 0$; $0 \le m' \le 2$
- The fader scaler variable, $\alpha_f = 1$

Processing:

Rate 1/2 and 1 decoding shall comprise 5.2.1 to 5.2.3

5.2.1 Decoding of the LSP Parameters

If (*FER_FLAG*(m) = *TRUE*), use the LSP parameters from the last frame to generate bandwidth expanded LSPs for the current frame as follows: $\Omega_q(m) = 0.875\ \Omega_q(m\text{-}1) + 0.125\ \Omega_{spread}$ where Ω_{spread} are the initial values of the LSPs, as defined in Section 5.2.

Otherwise, decode the LSP parameters, $\Omega_q(m)$, as defined in 4.4.5, using the decoded LSP indices, *LSPIDX*(k). In the event that the decoded LSPs are not strictly ascending, *FER_FLAG*(m) shall be set to *TRUE*, where "strictly ascending" is defined as $\Omega_q(m, i) < \Omega_q(m, i+1)\ ;\ 1 \le i \le 9$.

5.2.2 Delay Decoding and Frame Erasure Delay Contour Reconstruction

5.2.2.1 Delay Decoding

The delay parameter, $\tau(m)$, for the current frame is defined as:

$$\tau(m) = \begin{cases} \tau(m-1) & ; FER\_FLAG(m) = TRUE, \\ DELAY + 20 & ; otherwise, \end{cases} \tag{5.2.2.1-1}$$

where *DELAY* is the pitch delay transmission code.

5.2.2.2 Frame Erasure Delay Contour Reconstruction for Rate 1

If (*FER_FLAG*(m) = *FALSE*) and (*FER_FLAG*(m-1) = *TRUE*) and (*Rate* = Rate 1), the delay contour shall be reconstructed (for Rate 1 operation only) by warping the last valid adaptive codebook memory buffer, $E_{lv}(n)$, using a recovered delay contour. $E_{lv}(n)$ is first copied to the current adaptive codebook excitation memory, $E(n)$:

$$E(n\text{-}128) = E_{lv}(n); \quad 0 \le n < 128 \tag{5.2.2.2-1}$$

where $E_{lv}(n)$ was determined in 5.2.3.9 for the last valid frame, and $E(n)$ is the adaptive codebook excitation memory.

5.2.2.2.1 Delay Reconstruction

Calculate a delay, τ', which corresponds to the delay of the last valid frame:

$$\tau' = \tau\,(m\text{-}1), \tag{5.2.2.2.1-1}$$

and recover the last frame's delay as computed in the encoder by using the delay difference:

$$\tau\,(m\text{-}1) = \tau\,(m) - DELAY - 16, \tag{5.2.2.2.1-2}$$

where *DDELAY* is the received delay difference transmission code, and $\tau\,(m)$ is the current frame delay parameter.

Limit τ' by:

$$\tau' = \begin{cases} \tau\,(m\text{-}1) & ; \left|\tau(m\text{-}1) - \tau'\right| > 15, \\ \tau' & ; otherwise, \end{cases} \tag{5.2.2.2.1-3}$$

E(n) is then warped using τ' and $\tau(m\text{-}1)$ in 5.2.2.2.2 and 5.2.2.2.3 for each subframe $0 \le m' < 3$.

5.2.2.2.2 Reconstruction of the Delay Contour

The reconstructed delay contour, $\tau_c(n)$, is generated as defined in 4.5.5.1 of the encoder, using a set of delay interpolations computed by:

$$\dot{d}(m', j) = \begin{cases} \tau(m-1) & ; \left| \tau(m-1) - \tau' \right| > 15, \\ \left(1 - f(m'+j)\right)\tau(m-1) + f(m'+j)\tau' & ; otherwise, \end{cases}$$

$$0 \le j < 3, \qquad (5.2.2.2.2\text{-}1)$$

where m is the current frame, m' is the subframe index, and $m'+j$ is an index into an array of interpolator coefficients, $\boldsymbol{f}$, defined by:

$$\boldsymbol{f} = \{\ 0.0,\ 0.3313,\ 0.6625,\ 1.0,\ 1.0\ \}. \qquad (5.2.2.2.2\text{-}2)$$

5.2.2.2.3 Warping of the Adaptive Codebook Memory

The adaptive codebook memory, E(n), is then mapped to the delay contour, $\tau_c(n)$, as described in 4.5.5.2 of the encoder.

5.2.2.3 Smoothing of the Decoded Delay

The delay of the previous frame, $\tau(m\text{-}1)$, shall be smoothed for delay interpolation in 5.2.3.4 by:

$$\tau(m-1) = \begin{cases} \tau(m) & ; \left| \tau(m) - \tau(m-1) - \tau' \right| > 15, \\ \tau(m-1) & ; otherwise, \end{cases} \qquad (5.2.2.3\text{-}1)$$

5.2.3 Rates 1/2 and 1 Subframe Decoding

Compute the decoded synthesized speech signal for each subframe, $0 \le m' < 3$, as described in 5.2.3.1 through 5.2.3.10. The subframe size, L, is 53 for subframes 0 and 1, and 54 for subframe 2.

5.2.3.1 Interpolation of LSP Parameters

Interpolate the quantized LSPs over the three subframes, m', in the current frame, m. The form of the quantized, interpolated LSP vector is:

$$\dot{\Omega}_q(m') = (1 - \mu_{m'})\Omega_q(m-1) + \mu_{m'}\Omega_q(m), \qquad (5.2.3.1\text{-}1)$$

where the subframe interpolator constants are defined as $\mu = \{\ 0.1667,\ 0.5,\ 0.8333\ \}$.

5.2.3.2 LSP to LPC Conversion

Convert the quantized, interpolated LSPs, $\dot{\Omega}_q(m)$, to quantized, interpolated LPC parameters, $\left\{\dot{a}_q\right\}$, as described in 4.2.2.2 for each subframe.

5.2.3.3 Bandwidth Expansion

If the decoded *LPCFLAG* transmission code = *TRUE*, and (*FER_FLAG*(m - 1) = *TRUE*), the interpolated LPC parameters, $\left\{ \dot{a}_q \right\}$, shall be bandwidth expanded using:

$$\dot{a}_q(k) = (0.75)^K \ \dot{a}_q(k); \qquad 1 \le k < 10. \qquad (5.2.3.3\text{-}1)$$

5.2.3.4 Interpolated Delay Estimate Calculation.

Calculate the set of interpolated delay estimates, $\dot{d}(m', j)$, as in 4.5.4.5 of the encoder. If (*FER_FLAG*(m) = *TRUE*), and the average adaptive codebook gain (see 5.2.3.5) for the last valid frame, (g_{pavg} < 0.3) , the interpolated delay estimates are defined as:

$$\dot{d}(m', j) = d_{md}(m' + j); \qquad 0 \le j < 3 \qquad (5.2.3.4\text{-}1)$$

where $d_{rnd}(k)$ is the k^{th} element of:

$$d_{md} = \{55.0, 80.0, 39.0, 71.0, 33.0\}, \qquad (5.2.3.4\text{-}2)$$

5.2.3.5 Calculation of the Adaptive Codebook Contribution

The average adaptive codebook gain, g_{pavg} is computed by:

$$\mathrm{g_{pavg}} = \begin{cases} \mathrm{g_{pavg}}(m\text{-}1) & ; FER\_FLAG(m) = TRUE \ and \ FER\_FLAG(m\text{-}1)=\mathrm{FALSE}, \\ 0.75\mathrm{g_{pavg}}(m\text{-}1) & ; FER\_FLAG(m) = TRUE \ and \ FER\_FLAG(m\text{-}1)=\mathrm{TRUE}, \\ \frac{1}{3}\sum_{i=0}^{2} g_p(i) & ; FER\_FLAG(m) = \mathrm{FALSE}, \end{cases} \qquad (5.2.3.5\text{-}1)$$

where $g_{pcb}(k)$ is an entry in the adaptive codebook gain quantization Table 4.5.4-1, and *ACBGIDX*(m') is the decoded adaptive codebook gain index.

The adaptive codebook gain, $g_p(m')$, shall be defined as:

$$g_p(m') = \begin{cases} g_{pavg}(m) & ; FER\_FLAG(m) = TRUE, \\ \mathrm{g_{pcb}}\left(ACBGIDX(m')\right) & ; FER\_FLAG(m) = FALSE, \end{cases} \qquad (5.2.3.5\text{-}2)$$

Calculate the adaptive codebook excitation, E(n), as described in 4.5.5 of the encoder.

5.2.3.6 Calculation of the Fixed Codebook Gain

The fixed codebook gain, $g_c(m')$, is defined as:

$$g_c(m') = \begin{cases} g_{cavg} & ; FER\_FLAG(m) = TRUE\ , \\ g_{ccb}\left(FCBGIDX(m')\right) & ; otherwise\ , \end{cases} \quad (5.2.3.6\text{-}1)$$

where the average fixed codebook gain, g_{cavg} , is defined by:

$$g_{cavg} = \frac{1}{3}\sum_{i=0}^{2} g_c(i)\ , \quad (5.2.3.6\text{-}2)$$

calculated over $g_c(i)$ for the last valid frame; $g_{ccb}(k)$ is an entry in the fixed codebook gain quantization Table B-12, and $FCBGIDX(m')$ I s the fixed codebook gain index.

5.2.3.7 Computing of the Reconstructed ACELP Fixed Codebook Excitation

If ($FER\_FLAG(m) = TRUE$), the ACELP contribution is $c(n) = 0$; $0 \le n \le 54$, otherwise generate the fixed codebook contribution as follows:

For Rate 1 frames, the position of the two-single-pulse tracks, q, is determined by using $FCBSIDX(m',3)[9,10]$ as given in Table 4.5.7.2.2-1. The remaining three tracks correspond to double-pulse tracks.

The single and double-pulse track positions in the decimated domain, for $0 \le i \le 3$, are determined by:

$$\lambda_0 = \left\lfloor \frac{FCBSIDX(m',i)[0,...,6]}{11} \right\rfloor, \quad (5.2.3.7\text{-}1)$$

$$\lambda_1 = FCBSIDX(m',i)[0,...,6] - 11\lambda_0\ . \quad (5.2.3.7\text{-}2)$$

The sign of the three double-pulse tracks is determined by using the relationship between λ_0 and λ_1 as well as $FCBSIDX(m', i)$ [7] for i = 0, 1, 2. If $\lambda_0 < \lambda_1$, then the two pulses have the same sign, as specified by $FCBSIDX(m', i)$ [7]. If $\lambda_0 < \lambda_1$, then the two pulses have different signs, and $FCBSIDX(m', i)$ [7] specifies the sign of the pulse at position λ_0 as described in 4.5.7.3. The sign of the two single-pulse tracks is given by the codeword $FCBSIDX(m', 3)$ [7] and $FCBSIDX(m', 3)$ [8], accordingly.

Next, referring to Equations 4.5.7.3-1 and 4.5.7.3-2 and Tables 4.5.7.1-1 and 4.5.7.2.2-1, the pulse positions in the decimated domain shall be converted to the undecimated domain.

For Rate 1/2, $FCBSIDX(m', 0)$ [9] indicates the sign of the first and the third pulse track, and the opposite sign of the 2nd pulse track, and $FCBSIDX(m', 0)$ [0,...,8] indicates the positions of three pulses in the three tracks accordingly, as given in the Table 4.5.7.4-1.

The received algebraic codebook index is used to extract the positions and amplitudes (signs) of the excitation pulses and to construct the algebraic codevector, c, according to Equation 4.5.7.1-1. If the pitch delay, τ, is less than the subframe size 55, the pitch sharpening processing shall be applied which modifies $c(n)$ by $c(n)=c(n)+g_P c(n-\tau)$, where g_P is the decoded pitch gain bounded by [0.2, 0.9].

5.2.3.8 Decoder Total Excitation Generation

Add the fixed and adaptive codebook contributions to obtain the total excitation, $E_T(n)$, for the current frame as defined by:

$$E_T(n) = \begin{cases} g_p E(n); & FER\_FLAG(m) = TRUE \\ g_p E(n) + g_c c(n); & otherwise, \end{cases} \quad ; 0 \le n < L. \qquad (5.2.3.8\text{-}1)$$

5.2.3.9 Adaptive Codebook Memory Update

Update the adaptive codebook excitation memory with the combined excitation for the current subframe, $E_T(n)$, excluding any random excitations added in 5.2.3.10:

$$E_T(n) = \begin{cases} E(n+L); & -128 \le n < -L \\ E_T(n+L); & -L \le n < 0. \end{cases} \qquad (5.2.3.9\text{-}1)$$

If ($FER\_FLAG(m) = FALSE$), and the current subframe, $m' = 2$, update the last valid excitation memory buffer, $E_{lv}(n)$ with the adaptive codebook memory:

$$E_{lv}(n) = E(n-128); \quad 0 \le n < 128. \qquad (5.2.3.9\text{-}2)$$

5.2.3.10 Additional Excitation Frame Processing

A fade scaling variable, α_f, shall be calculated every subframe as defined by:

$$\alpha_f = \begin{cases} \alpha_f - 0.05; & FER\_FLAG(m) = TRUE \\ \alpha_f + 0.2; & otherwise, \end{cases} \qquad (5.2.3.10\text{-}1)$$

where α_f is bounded by $0 \le \alpha_f \le 1$, and is used to scale the combined excitation:

$$E_T(n) = \alpha_f E_T(n); \quad 0 \le n < L. \qquad (5.2.3.10\text{-}2)$$

If ($FER\_FLAG(m) = TRUE$), and the average adaptive codebook gain, ($g_{pavg} < 0.4$), a random fixed codebook excitation is added to the combined excitation:

$$E_T(n) = E_T(n) + 0.1 g_{cavg} \text{ran\_g}\{seed\}; \quad 0 \le n < L, \qquad (5.2.3.10\text{-}3)$$

where g_{cavg} is calculated in 5.2.3.6 for the last valid frame, and ran_g{*seed*} is a unit variance pseudo-random Gaussian white noise sequence (see 4.7.2).

5.2.3.11 Synthesis of the Decoder Output Signal

Filter the combined excitation, $E_T(m)$, through the synthesis filter using the interpolated LPCs generated in 5.2.3.2, creating the synthesized speech signal, $\hat{s}(n)$. The synthesis filter for the decoder, $H_q(z)$, is defined by:

$$H_q(z) = \frac{1}{A_q(z)} = \frac{1}{1 - \sum_{k=1}^{10} a_q(k) z^{-k}}. \qquad (5.2.3.11\text{-}1)$$

The signal, $\hat{s}(n)$, shall be post-filtered according to 5.4, producing the post-filtered synthesized speech signal, $\hat{s}_{pf}(n)$.

5.3 Rate 1/8 Decoding

Inputs: The inputs to Rate 1/8 decoding are:

- The quantized LSP indices from the current frame, *LSPIDX*(*k*)
- The quantized LSPs from the previous frame, $\Omega_q(m\text{-}1)$
- The vector quantized frame energy index, *FGIDX*(*m*)
- The vector quantized frame energy index from the last frame, *FGIDX*(*m* - 1)
- The frame erasure flag, *FER_FLAG*(*m*), for the current frame *m*

Outputs: The outputs from Rate 1/8 decoding are:

- The post-filtered synthesized speech signal, $\hat{s}_{pf}(n)$

State Variables:

None.

Processing:

Rate 1/8 decoding shall comprise 5.3.1 to 5.3.3.

5.3.1 Decoding of the LSP parameters

If (*FER_FLAG*(*m*) = *TRUE*), use the LSP parameters from the last frame, $\Omega_q(m) = \Omega_q(m\text{-}1)$. Otherwise, decode the LSP parameters, $\Omega_q(m)$, as defined in 4.4.5, using the decoded *LSPIDX*(*k*). In the event that the decoded LSPs are not strictly ascending, *FER_FLAG*(*m*) shall be set to *TRUE*, where "strictly ascending" is defined as $\Omega_q(m, i) < \Omega_q(m, i+1)\ ; 1 \le i \le 9$.

5.3.2 Decoding of the Frame Energy Vector

If (*FER_FLAG*(*m*) = *FALSE*), the frame energy vector, $\gamma_q(m')$, shall be defined as:

$$\gamma_q(m') = 10^{q\log(m', FGIDX(m))}; \quad 0 \le m' < 3, \tag{5.3.2-1}$$

where $q_{log}(m', FGIDX)$ is an entry in the gain quantization codebook found in Table B-15.

If (*FER_FLAG*(*m*) = *TRUE*), the frame energy vector, $\gamma_q(m')$, is calculated by:

$$\gamma_q(m') = \frac{1}{3}\sum_{i=0}^{2} 10^{q\log(i, FGIDX(m-1))}; \quad 0 \le m' < 3, \tag{5.3.2-2}$$

where *FGIDX*(*m*-1) is the codebook index from the last Rate 1/8 packet in which *FER_FLAG* was *FALSE*.

5.3.3 Rate 1/8 Subframe Decoding

Compute the decoded synthesized speech signal for each subframe, $0 \le m' < 3$, as described in 5.3.3.1 to 5.3.3.4. The subframe size, *L*, is 53 for subframes 0 and 1, and 54 for subframe 2.

5.3.3.1 Rate 1/8 Excitation Generation

Obtain the Rate 1/8 excitation by generating a zero-mean, unit variance pseudo-random Gaussian white noise process (see 4.7.2) scaled by $\gamma_q(m')$:

$$E(n) = \gamma_q(m')\text{ran\_g}\{seed\}; \quad 0 \le n < \text{L}, \tag{5.3.3.1-1}$$

No attempt is made to synchronize the random time series generated at the receiver with that generated at the transmitter.

5.3.3.2 Interpolation of LSP Parameters

Interpolate the quantized LSPs as in 5.2.3.1.

5.3.3.3 LSP to LPC conversion

Perform the LSP to LPC conversion as defined in 5.2.3.2.

5.3.3.4 Synthesis of Decoder Output Signal

Filter the Rate 1/8 excitation, E(n), through the synthesis filter as in 5.2.3.11. The signal, $\hat{s}(n)$, shall be post-filtered according to 5.4, producing the post-filtered synthesized speech signal, $\hat{s}_{pf}(n)$.

5.4 Adaptive Postfilter

The Adaptive Postfilter function improves the perceived speech quality of the decoder output. Refer to 1.1 for guidelines concerning potential variations of the Adaptive Postfilter.

Inputs:

- The decoder synthesis output signal, $\hat{s}(n)$
- The quantized, interpolated LPC coefficients, $\dot{A}_q(n)$
- The interpolated pitch delays, $\dot{d}(m')$

Output:

The post-filtered synthesis signal, $\hat{s}_{pf}(n)$

Initialization: All filter memories shall be initialized to zero at start-up.

Processing: The decoded speech signal, $\hat{s}(n)$, shall be post-filtered by applying the following operations in the sequence specified:

- A tilt-compensation filter $H_t(z)$
- A short term residual filter, $H_\varepsilon(z)$
- A long term filter, $H_p(z)$
- Gain normalization, g_s
- A short term filter, $H_s(z)$

The postfilter shall be applied once for each of the three subframes and shall take the form:

$$PF(z) = H_S(z) g_S H_p(z) H_\varepsilon(z) H_t(z) . \quad (5.4\text{-}1)$$

The postfilter residual memory buffer is then updated with the current subframe's residual signal. Each of these steps shall be carried out as specified in Sections 5.4.1 to 5.4.5.

5.4.1 Tilt Compensation Filter

Generate the tilt-compensated speech signal, $\hat{s}_t(n)$, by filtering the decoded speech signal, $\hat{s}(n)$, through the tilt compensation filter given by:

$$H_t(z) = 1 - \mu z^{-1} , \quad (5.4.1\text{-}1)$$

where μ is the tilt coefficient calculated by determining if $\hat{s}(n)$ is voiced or un-voiced. The signal, $\hat{s}(n)$, is defined as being voiced if:

$$R = \left(\sum_{i=0}^{L-2} \hat{s}(i)\hat{s}(i+1) \right) \geq 0 . \quad (5.4.1\text{-}2)$$

Using the above result, the tilt coefficient, μ, is defined as:

$$\mu = \begin{cases} 0 & ; R < 0, \\ tilt \text{ (Rate)} & ; R \geq 0, \end{cases} \quad (5.4.1\text{-}3)$$

where *tilt*(*Rate*) is defined in Table 5.4.1-1.

Table 5.4.1-1. Postfilter Coefficients

Rate	*tilt*(*Rate*)	γ_{p1}	γ_{p2}
1	0.20	0.57	0.75
1/2	0.35	0.50	0.75
1/8	0	0.57	0.57

5.4.2 The Short Term Residual Filter

Compute the postfilter residual, $\varepsilon_{pf}(n)$, by filtering the tilted speech signal, $\hat{s}_t(n)$, through $H_\varepsilon(z)$, given by:

$$H_\varepsilon(z) = \dot{A}_q\left(\gamma_{p1}^{-1} z\right) = 1 - \sum_{k-1}^{10} \dot{a}_q(k) \gamma_{p1}^{k} z^{-k} , \quad (5.4.2\text{-}1)$$

where γ_{p1} is given for each *Rate* in Table 5.4.1.1-1, and $\left\{ \dot{a}_q \right\}$ are the interpolated LPC parameters for the current subframe.

5.4.3 The Long-term Postfilter

Generate the long-term post-filtered signal, $\hat{s}_p(n)$, by filtering the residual signal, $\varepsilon_{pf}(n)$, through the long term post filter given by:

$$H_p(z) = \begin{cases} 1.0 + g_p g_{lt} z^{-d_{opt}} & ; 0.5 \le g_p < 1.0, \\ 1.0 + g_{lt} z^{-d_{opt}} & ; 1.0 \le g_p, \\ 1.0 & ; \text{otherwise}, \end{cases} \tag{5.4.3-1}$$

where g_{lt} = 0.5 and where the long term predication gain, g_p, is calculated by:

$$g_p = \frac{\sum_{n=0}^{L-1} \varepsilon_{pf}(n)\varepsilon_{pf}(n-d_{opt})}{\sum_{n=0}^{L-1} \varepsilon_{pf}(n-d_{opt})\varepsilon_{pf}(n-d_{opt})}, \tag{5.4.3-2}$$

and where the long-term delay, d_{opt}, is calculated from $\varepsilon_{pf}(n)$ by finding the best integer delay around $d_I = \left(\frac{\dot{d}(m',0) + \dot{d}(m',1)}{2} \right)$. The decoded interpolated integer delay d_I is searched from (d_I - 3) to (d_I + 3) to find the best integer delay for the long-term postfilter. The best integer delay d_{opt} is computed by maximizing the correlation:

$$R(d_{opt}) = \sum_{n=0}^{L-1} \varepsilon_{pf}(n)\varepsilon_{pf}(n-d_{opt}); \quad (d_I - 3) \le d_{opt} \le (d_I + 3). \tag{5.4.3-3}$$

5.4.4 Gain Normalization and Short-Term Postfilter

Compute the temporary signal, $\hat{s}_s(n)$, by filtering the long-term postfilter output signal, $\hat{s}_p(n)$, through the short-term postfilter, $H_s(z)$, given by:

$$H_s(z) = \frac{1}{A_q\left(\gamma_{p2}^{-1} z\right)}, \tag{5.4.4-1}$$

where γ_{p2} is found in Table 5.4.1-1.

Then compute the short-term postfilter gain, g_s, by using the temporary signal, $\hat{s}_s(n)$:

$$g_s = \sqrt{\frac{\sum_{n=0}^{L-1} \hat{s}^2(n)}{\sum_{n=0}^{L-1} \hat{s}_s^2(n)}}, \tag{5.4.4-2}$$

where g_s is upper bound limited at 1.0. The signal $g_s \hat{s}_p(n)$ is then filtered through the short-term postfilter in Equation 5.4.4-1 to produce the final post-filtered speech signal, $\hat{s}_{pf}(n)$.

6 TTY/TDD EXTENSION

6.1 Introduction

This section provides an option to reliably transport the TTY/TDD 45.45 bps and 50 bps Baudot code, making digital wireless technology accessible to TTY/TDD users. This section is separated into two major components. Section 6.3 describes the new interface between the encoder and the decoder for transporting the TTY information. Section 6.3.4 is a description of the TTY/TDD software simulation of this section, and is offered only as a recommendation for implementation. However, in the event of ambiguous or contradictory information, the software simulation shall be used to resolve any conflicts.

This section is an extension of the previous version of the TTY/TDD extension for EVRC, 3GPP2 C.S0014-0-3. It extends the previous version by adding the 50 bps Baudot functionality, but is capable with interoperating with 3GPP2 C.S0014-0-3 when used at 45.45 bps. See Section 6.3.3.2 for details regarding interoperability with 3GPP2 C.S0014-0-3.

This section 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.

6.2 Overview

The following sections provide a method for reliably transporting the 45.45 bps and 50 bps Baudot code in the audio path, making digital wireless telephony accessible to TTY/TDD users. The following extension is robust to frame and bit errors and is completely interoperable with the pre-existing 3GPP2 C.S0014-0 speech-coding standard. The solution supports voice carryover/hearing carryover (VCO/HCO). VCO allows a TTY/TDD user to switch between receiving TTY and talking into the phone. Similarly, HCO allows a user to switch between transmitting TTY characters and picking up the phone to listen. When Baudot tones are not present, the vocoder operates as usual, there is no modification or added delay to the voice path when speech is present.

The TTY/TDD audio solution transports Baudot signals through the vocoder by detecting the characters, and their baud rate, that are being transmitted by the TTY/TDD in the encoder and conveying those characters to the decoder. Because one Baudot character spans at least 7 speech processing frames, the character being transmitted shall be sent a minimum of 6 times to the decoder, allowing the decoder to correctly regenerate the character despite frame errors and random bit errors in the speech packet.

The TTY characters are concealed in the speech packet in a way that interoperates with legacy vocoders that have not been modified for TTY. This is made possible because, when Baudot tones are present, the TTY information replaces the pitch lag bits for the adaptive codebook (ACB) and the ACB gain is set to zero so that an unmodified decoder ignores the TTY information. The rest of the bits in the speech packet contain information for an unmodified decoder to reconstruct the Baudot signal with the fixed codebook and the linear prediction (LPC) filter at least as well as if the encoder was not modified. Furthermore, the encoder shall disable noise suppression, and the rate shall be set to full rate when the Baudot tones are present. This further enhances the system’s performance when a modified encoder is interoperating with an unmodified decoder.

A decoder modified with this extension maintains a history buffer to monitor the ACB gain and pitch lag in the speech packets. When the decoder detects that the ACB gain has been set to zero, and the pitch lag contains information consistent with TTY, the decoder stops decoding speech and begins regenerating the Baudot tones. When the decoder stops detecting TTY information, it resumes processing speech.

When Baudot tones are not present, the modified vocoder operates on speech in exactly the same way as the unmodified vocoder. The TTY processing does not add any additional delay to the speech path.

6.3 TTY/TDD Extension

The TTY processing in the encoder shall process the received PCM one frame at a time and label each frame as NON_TTY, or as TTY_SILENCE, or as a TTY character. The vocoder will be in one of two states: TTY_MODE or NON_TTY_MODE. In the absence of Baudot tones, the encoder and decoder shall be in the NON_TTY_MODE, and the encoder and decoder shall process the frame as speech. When Baudot tones are present, the encoder and decoder shall enter TTY_MODE and process the TTY information as described below.

There shall exist a mechanism to disable the TTY/TDD extension in the vocoder, reverting the vocoder to its unmodified state.

6.3.1 TTY Onset Procedure

The TTY Onset Procedure describes the process by which the vocoder shall transition from the speech mode to the TTY mode.

6.3.1.1 Encoder TTY Onset Procedure

When the TTY encoder processing initially detects that Baudot tones are present, the encoder shall label each frame as TTY_SILENCE until it buffers enough frames to detect the character being sent. The TTY_SILENCE message shall be sent to the decoder according to the method described below. Because of the delay caused by the buffering in the encoder and decoder to detect TTY characters, it is necessary to alert the decoder to mute its output when Baudot tones are first detected. This prevents the Baudot tones from getting through the speech path before the TTY decoder processing is able to detect the TTY characters and regenerate the tones. The TTY_SILENCE message shall be sent to the decoder within 2 frames after the PCM containing the Baudot tones initially enters the encoder. However, in order to reduce the risk of false alarms, the TTY encoder may delay sending the TTY_SILENCE message for the very first character in a call. The TTY_SILENCE message shall be sent for a minimum of 4 frames and shall continue to be sent until a TTY character is detected, or until a NON_TTY frame is detected.

6.3.1.2 Decoder TTY Onset Procedure

When the decoder is in NON_TTY_MODE, the packet shall be decoded in the usual manner for speech. Because there are no bits in the packet to switch the decoder's state, the decoder shall infer the presence of TTY information from the ACB gain and pitch information. The decoder shall recognize when TTY_SILENCE messages are being sent in the packets and transition from NON_TTY_MODE to TTY_MODE before the decoder's speech path reconstructs a TTY character from the audio information in the speech packets. When the decoder makes the transition to TTY_MODE, it shall mute its output until it detects TTY characters or until it transitions back to NON_TTY_MODE. Refer to the implementation recommendation in Section 6.3.4 for an example of the TTY decoder processing.

6.3.1.3 TTY_MODE PROCESSING

The format of the Baudot code can be found in ITU-T Recommendation V.18. The Baudot code is a carrierless, binary FSK signaling scheme. A 1400 Hz. tone is used to signal a logical “1” and an 1800 Hz. tone is used to signal a logical “0”. A TTY bit has a duration of 22 ± 0.4 ms for 45.45 baud and 20 ± 0.4 ms for 50 baud. A character consists of 1 start bit, 5 data bits, and 1.5 – 2 stop bits. When a character is not being transmitted, silence, or a noisy equivalent, is transmitted. Hence, a TTY character spans a minimum of 7 speech processing frames. When the TTY encoder processing detects a character, it shall send the character, its baud rate, and its header (see Section 6.3.2) for a description of the header) to the decoder over a minimum of 6 consecutive frames and a maximum of 16 frames. Because channel impairments cause frame errors and bit errors, the decoder may not receive all of the packets sent by the encoder. The decoder shall use the redundancy to correct any corrupted TTY information. Once the decoder recognizes the TTY character being sent, the decoder’s TTY repeater shall regenerate the Baudot tones corresponding to that character.

At call startup, the encoder processing may initialize its baud rate to either 45.45 baud or 50 baud. It is recommended that the default baud rate be set to the predominant baud rate of the region. Because the encoder may require several characters to determine the correct baud rate, it should be expected that the baud rate may change 3-7 characters into the TTY call.

6.3.1.4 TTY_SILENCE Processing

In order to reduce the average data rate of a TTY call, the TTY processing shall be capable of transmitting 1/8 rate packets to the decoder when the encoder is processing the silence periods between characters. Since no TTY information is in the 1/8 packet, the decoder shall infer TTY_SILENCE from an 1/8 rate packet when it is in TTY_MODE. The TTY_SILENCE message may also be sent to the decoder using a full rate frame, as described in Section 6.3.2. When setting the rate to accommodate the TTY information, care shall be taken so that a full rate frame is not immediately followed by an 1/8 rate frame. This is an illegal rate transition according to 3GPP2 C.S0014-0, and will force an unmodified decoder to declare a frame erasure.

6.3.2 TTY Header, Baud Rate, and Character Format

The TTY information put into the speech packet contains header character, and baud rate information. When the encoder is transmitting a TTY character, the header shall contain a sequence number to distinguish that character from its preceding and following neighbors. The same header, character, and baud rate information shall be transmitted for each instance of a character for a minimum of 6 frames and a maximum of 16 frames. The header shall cycle through its range of valid values, one value for each instance of a character. The header and character field shall be assigned a value to correspond to the TTY_SILENCE message. TTY_SILENCE may also be conveyed by 1/8 rate packets (see Section 6.3.1.4).

The rate bit shall specify the baud rate to be regenerated by the decoder. The rate bit shall be set to ‘0’ to denote 45.45 baud and ‘1’ to denote 50 baud. During the TTY_SILENCE message, the baud rate bit shall be set to its last transmitted value. In the case where the baud rate has not yet been determined, the default startup baud rate shall be used. The encoder processing may initialize its baud rate to either 45.45 baud or 50 baud.

Table 6.3.2-1: TTY Header and Character Fields

Description	Range		
	Header (2 bits)	Character (5 bits)	Baud Rate (1 bit)
Reserved	0	0 – 31	0-1
TTY Character	1 – 2	0 – 31	0-1
TTY_SILENCE	3	4	0-1
Reserved	3	0 – 3, 5 – 31	0-1

The combinations of valid values for the TTY header, character, and baud rate fields are specified in Table 6.3.2-1. Note that the value for TTY_SILENCE corresponds to the index for the maximum pitch value allowed by the vocoder. All unused values are reserved for future use and shall be considered as invalid values for the purposes of this annex.

6.3.3 Transporting the TTY Information in the Speech Packet.

In full rate and half rate, there are 7 bits per frame assigned to the pitch lag. In full rate, the last bit of the speech packet, Bit 171, is reserved by the speech coder. This bit plus the 7 bits assigned to the pitch lag shall be used to convey TTY information from the encoder to the decoder using full rate frames. Half rate packets may also be used for conveying TTY information. Because half rate packets have only 7 bits available, i.e., the 7 pitch lag bits, half rate frames shall contain the TTY header and TTY character information only and the decoder shall use its last valid baud rate to regenerate TTY characters. In order to improve interoperability between a modified encoder and an unmodified decoder, it is recommended to transport the TTY information in a full rate packet; however, a modified decoder shall be capable of detecting TTY information in both full rate and half rate packets.

The TTY information replaces the pitch lag bits and the reserved Bit 171. The ACB gain shall be set to zero for each subframe in packets containing TTY information. Bit 171 shall be used to convey the baud rate information and the 7 pitch lag bits shall be used to convey the TTY header and character information. The 5 least significant bits of the pitch bits shall be used for the 5-bit Baudot code. Two remaining pitch bits shall be used for the TTY header information. The TTY information is assigned to the pitch lag bits according to Table 6.3.3-1.

Table 6.3.3-1: TTY Header and Character Bit Assignment

PITCH LAG BIT ASSIGNMENT						
MSB						LSB
6	5	4	3	2	1	0
TTY HEADER		5 BIT BAUDOT CODE				
MSB	LSB	MSB				LSB
1	0	4	3	2	1	0

6.3.3.1 Half Rate TTY Mode

In the case where the encoder and decoder are both modified for TTY, it is possible to reduce the average data rate by using half rate packets to transport TTY information. Half rate packets should only be used to transport TTY information during Dim & Burst signaling, or when it is determined that both the near-end and far-end vocoders are TTY capable. When the near-end (far-end) decoder recognizes that the far-end (near-end) encoder is sending TTY information, the near-end (far-end) encoder may be notified by the near-end (far-end) decoder to send TTY information in half rate packets. When the near-end (far-end) decoder receives a NON_TTY packet, the near-end (far-end) encoder should exit half rate TTY mode. This preserves interoperability in the event of a hard handoff from a modified vocoder to an unmodified vocoder. Because half rate packets do not convey the baud rate, the vocoder should return to full rate for at least 4 characters after a change in baud rate before returning to half rate TTY mode, whenever possible. In Dim & Burst signaling, for example, it may be more desirable to stay in half rate TTY mode to complete the signaling message rather than go to full rate to convey a baud rate change.

6.3.3.2 Interoperability with 45.45 Baud-Only TTY Extensions

Previous TTY extensions support 45.45 bps Baudot code only. This extension supports both 45.45 baud and 50 baud by using an additional bit to convey the baud rate. Note, the other 7 bits used for the character information and header are the same as the 45.45 baud-only TTY extensions.

Because the baud rate uses a bit in the encoded speech packet that was not previously used for TTY, some level of interoperability is achieved (See Table 6.3.3.2-1). When a 45.45 baud decoder is receiving 50 baud TTY packets, it will regenerate the characters at 45.45 baud. In this case, the regenerator may fall behind and drop characters because it is regenerating characters at a slower rate than the encoder is sending them.

In the case where the encoder is 45.45 baud-only and the decoder is capable of regenerating either 45.45 or 50 baud, the 2 header bits and the 5 TTY character bits are the same, so the decoder is able to decode these bits correctly. The baud rate bit, however, has no meaning to the 45.45 baud-only encoder, so it will be set randomly, depending on the implementation, and the baud rate used by the decoder will depend on the value of the baud rate bit.

Table 6.3.3.2-1: Baud Rate Interoperability Matrix

	45.45 baud-only decoder	45.45 and 50 baud decoder
45.45 baud-only encoder	Compatible	EITHER 45.45 OR 50 BAUD WILL BE REGENERATED BY THE DECODER, DEPENDING ON THE VALUE OF THE BAUD RATE BIT SET BY THE ENCODER.
45.45 and 50 baud encoder	BAUD RATE IS IGNORED AND DECODER ALWAYS REGENERATES 45.45 BAUD.	Compatible

In order to address the interoperability issues between legacy 45.45 baud-only TTY solutions and 45.45/50 solutions, network implementations of this specification may provide a means for carriers to manually set the baud rate to an appropriate rate for the region. Implementations of this specification, however, must implement 45.45 baud and 50 baud, as well as the auto-baud rate detection algorithm, in order to be compliant with this specification.

In order for 45.45 baud-only implementations to interoperate with this extension, it is recommended that the 45.45 baud-only encoder implementations set the reserved Bit 171 to zero so that the baud rate bit is set to 45.45 baud.

6.3.3.3 Reflected Baudot Tones

It is possible for the signal generated by the TTY solution to reflect back to the near-end TTY detector, either on the network side or the mobile side. Possible sources of the reflected signal are crosstalk, impedance mismatch, or hybrid echo. To prevent the detector from detecting Baudot tones, that are not intentionally originated by the TTY device, the input PCM shall be muted, before being processed by the near-end encoder, whenever Baudot tones are being regenerated. This requirement applies to both the network and mobiles. Note, the sample solution described in Section 6.3.4 does not contain a mechanism for blocking reflected Baudot tones.

6.3.4 TTY/TDD Processing Recommendation

The following describes the software simulation of this annex. It is intended as a recommendation for implementation only and is not required to satisfy compliance with this annex. However, the software shall be used to resolve ambiguous or incomplete statements that may exist in the sections above. The TTY/TDD processing is divided into 2 major components, encoder processing and decoder processing. The TTY encoder process detects the presence of Baudot tones and decodes the TTY character being transmitted. It then conveys that information to the decoder. The TTY decoder processing shall detect the presence of TTY information and regenerate the Baudot tones corresponding to that character. Refer to Figure 6.3.4-1 for a block diagram of the TTY processing.

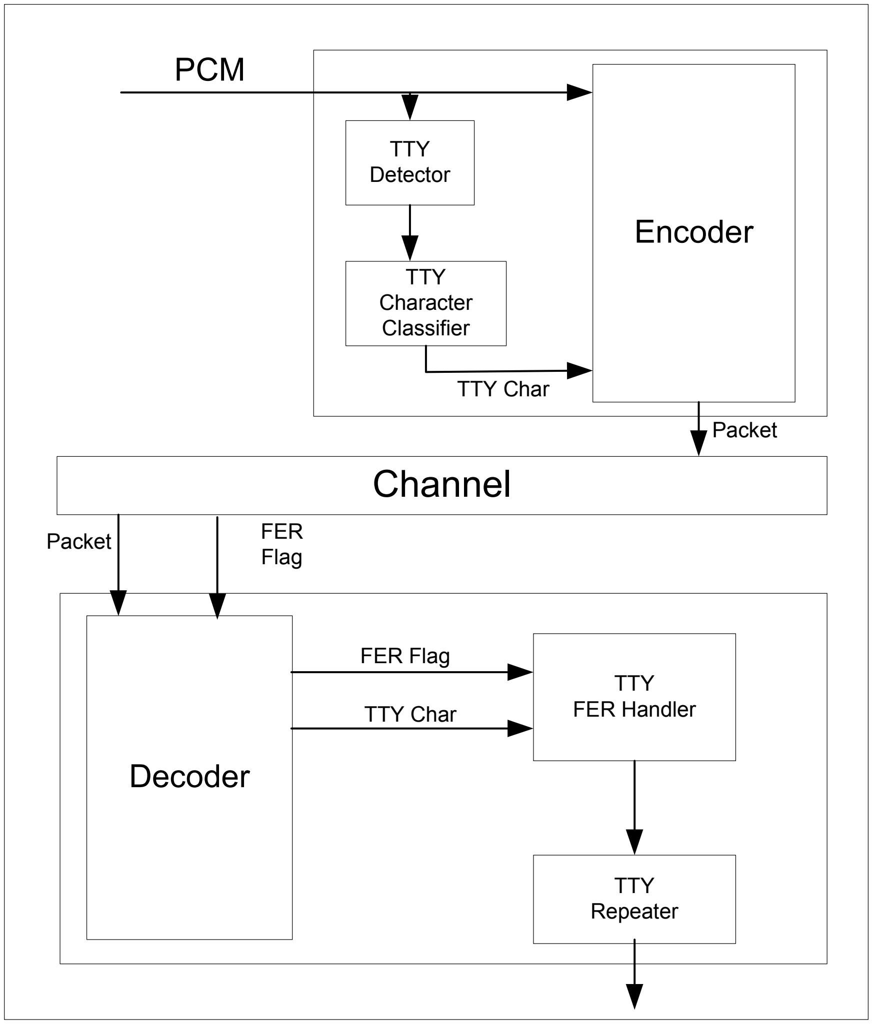

Figure 6.3.4-1. TTY/TDD Processing Block Diagram

6.3.5 TTY Encoder Processing

The TTY encoder processing takes the larger task of detecting TTY characters and divides it into a series of smaller tasks, creating different levels of detection. It is through this divide and conquer approach that the tty_enc( ) routine has low complexity in the absence of Baudot tones.

The first level of detection is to divide the 160 samples in the speech frame into 10 blocks of 16 samples. These blocks are called detection intervals, or dits. Each dit is classified as NON_TTY, LOGIC_0, or LOGIC_1.

The next level of detection is to determine if there are enough LOGIC_0 or LOGIC_1 dits in a row to form a TTY bit. The transition from a "0" bit to a "1" bit or the detection of two consecutive "0" bits signals the onset of a character and the TTY_SILENCE message is sent to the decoder. The TTY_SILENCE message shall continue to be sent until a TTY character is detected, or until a NON_TTY frame is detected. When enough bits are detected to form a character, the rate of the character is determined and the TTY character information is sent to the decoder.

6.3.5.1 TTY Encoder Inputs

- TTY character information from the previous frame (header, TTY character, and baud rate).
- 160 PCM samples from the output of the high pass filter.

6.3.5.2 Dit Classification

The 160 samples from the high pass filter's output are converted into 10 dits. For every block of 16 samples, it computes the spectral energy at the mark and space frequencies using a 16-point DFT at 1400 Hz and 1800 Hz with a rectangular window. The ratio of the maximum energy between the mark and space energy and the total energy is compared to a threshold; i.e.,

$$\frac{\max(mark\_energy, space\_energy)}{total\_energy} > \text{THRESH}$$

If that threshold is exceeded, the dit is labeled as either a mark or a space, whichever one has the greater energy. If the threshold is not met, the dit is labeled as NON_TTY.

6.3.5.3 Dits to Bits

The dits are used to form a TTY bit. Dits are classified as LOGIC_0, LOGIC_1, or UNKNOWN. A nominal bit consists of 11 dits for 45.45 baud and 10 dits for 50 baud. A bit is not required to have a continuous run of LOGIC_0's or LOGIC_1's in order to be detected. Spurious UNKNOWN detections are permitted, up to a threshold.

A TTY bit is searched by looking at the dits within a variable-length sliding window. The window varies in length from 8 dits to 13 dits. The number of LOGIC_0's, LOGIC_1's, and UNKNOWN dit detections are counted in the window. The dits in the search window must meet the following criteria in order to detect a TTY bit:

- Minimum of 6 LOGIC_0 (LOGIC_1) dits
- Maximum of 2 LOGIC_1 (LOGIC_0) dits
- Maximum of 5 UNKNOWN dits

Heuristics are also applied so that the sliding window is centered over the TTY bit being detected. If all of the thresholds are met, a "0" ("1") TTY bit is declared, and the overall length of the bit and the number of bad dits within the window are recorded. If the dits in the search window do not meet the criteria for a TTY bit, a gap between TTY bits is declared and the gap is recorded. The window is slid by one dit and the search is repeated.

The length of the search window is allowed to vary. The length is adjusted depending on the previous character's baud rate, and where the mark/space transitions occur.

Two history buffers are maintained to record the detected TTY bits and gaps. The array tty_bit_hist[ ] maintains a history of the bits that are detected and tty_bit_len_hist[ ] stores the lengths, in dits, of the gaps and TTY bits that are detected. Both arrays are 9 elements long, there is one element for the start bit, five for the data bits, one for the stop bit, and one for the memory bit, the bit before the start bit, as depicted in Figure 6.3.5.3-1. The last element in the array is used to record partially detected TTY bits.

Figure 6.3.5.3-1 TTY Bit History Buffer

0	1	2	3	4	5	6	7	8
Memory Bit	Start Bit	LSB Data 0	Data 1	Data 2	Data 3	MSB Data 4	Stop Bit	Next Bit

When a TTY bit or a gap is detected, its value is recorded in tty_bit_hist[ ] and its length is stored in tty_bit_len_hist[ ]. The length is used by get_tty_char( ) to threshold the length of a candidate character, and by tty_rate( ) to determine if a detected character is 45.45 baud or 50 baud. Since the length of the memory bit is irrelevant, tty_bit_len_hist[0] is used to count the number of NON_TTY and TTY_SILENCE dits that were detected within the TTY bits.

Because the speech frames may not coincide with the boundaries of the TTY bits, it is possible that a bit may straddle two speech frames. It is possible, therefore, that the sliding window may contain only a partial bit within a frame. These partial detections are recorded in element 8 of the TTY bit history buffers, labeled "Next Bit" in Figure 6.3.5.3-1.

6.3.5.4 TTY Character Classification

The routine get_tty_char( ) checks if the detected bits form a TTY character. The following conditions must be met in order for a character to be declared.

- The bit preceding the start bit must not be a "0".
- The start bit must be a "0"
- The stop bit must be a "1"
- The length of the candidate character, in dits, must be within a maximum and minimum threshold.
- The number of bad dit detections within the character must be within a threshold.

If all of the conditions are met, a character is declared.

Once a character is found, the character information and its header are sent to the decoder a minimum of 6 frames and a maximum of 16 frames. The constant FRAMING_HANGOVER dictates the maximum number of times the information for the same character is sent. If a new character is framed before FRAMING_HANGOVER is reached, the information for the old character is terminated and the new information is sent to the decoder.

6.3.5.5 TTY Baud Rate Determination

After a character has been detected, tty_rate( ) determines the baud rate of the character by counting the length, in dits, of the start bit and the five data bits. The length of the stop bit is not used because its length is too variable.

A character with a length of 63 dits or greater is declared 45.45 baud, otherwise it is declared 50 baud. A hangover of three characters is maintained before tty_rate( ) will switch from one baud rate to the other. That is to say, if the baud rate changes in the middle of a call, three consecutive characters must be detected at the new baud rate in order for tty_rate( ) to declare the new baud rate.

6.3.5.6 TTY State Machine

The routine get_tty_state( ) is responsible for changing the state of the TTY encoder processing. There are 3 states, NON_TTY_MODE, TTY_ONSET, and TTY_MODE. Get_tty_state( ) is responsible for determining NON_TTY_MODE and TTY_ONSET.

Changing TTY state from NON_TTY_MODE to TTY_ONSET requires that a “0” bit is followed by a “1” bit. This rule requires the presence of both the space tone and the mark tone and for the tones to be the correct duration. This test must be met in order to declare TTY_ONSET for the first time.

NON_TTY_MODE is declared by get_tty_state( ) whenever a non-TTY bit is detected or when a “0” bit occupies the bit preceding the start bit.

6.3.6 TTY/TDD Decoder Processing

The TTY decoder processing must recognize when TTY/TDD information is in the packet, recover from channel impairments to decode the TTY character being sent, and regenerate the Baudot tones corresponding to that character.

When the decoder is in NON_TTY_MODE, the packet is decoded in the usual manner for speech. When the decoder makes the transition to TTY_MODE, it mutes its output until it receives TTY character information or until it transitions back to NON_TTY_MODE.

6.3.6.1 TTY Decoder Inputs

- TTY Information (header, TTY character, baud rate).
- Bad frame indicator

6.3.6.2 Decoding the TTY/TDD Information

The task of detecting the presence of TTY information, recovering from frame and bit errors, and decoding the TTY character is performed in tty_dec( ). If the routine detects that TTY information is being sent, the tty_dec( ) flag is set to non-zero and the PCM buffer is filled with the appropriate Baudot tones. If TTY is not detected, the flag is set to zero and the PCM buffer is returned unmodified.

Table 6.3.6.2-1: tty_dec( ) History Buffer

Frame:	0	1	2	3	4	5	6	7	8	9	10
Description:	Lookahead									Current Frame	Lookback

The routine labels each frame as NON_TTY, TTY_SILENCE, FER, or a TTY character, and maintains a history buffer of these classifications for 11 frames: 9 frames of lookahead, 1 current frame, and 1 frame of lookback (see Table 6.3.6.2-1). The most recent packet enters the buffer at location 0, but the decision for the current frame is based on the contents of element 9. The buffer is updated at the end of each frame, shifting its contents to the right by one. The buffer is initialized to NON_TTY.

At the start of each frame, the most recent information is sanity checked to see if it is consistent with TTY information. If the frame erasure flag is set, the frame is labeled FER, otherwise the TTY/TDD information is checked to see if the header and TTY character fields fall within the allowed range of values. If all of the tests pass, Frame 0 is labeled with the TTY character in the history buffer.

The TTY decoder processing can reliably regenerate the TTY characters despite channel impairments because the character information is transmitted a minimum of 6 times from the encoder. Errors are corrected by a voting process. The current frame and nine frames of lookahead are used to determine the correct TTY header, character, and baud rate. Errors are replaced with the winner of the voting process.

Voting is conducted under the following conditions:

- Any time the current frame is labeled as FER.

- Every time the current frame contains information for the start of a new character. Because a new character must contain a minimum of 6 frames of the same information, a vote is taken to verify that the information is present before it will generate the tones for that character. Once a character wins the vote, any frame errors, bit errors, or other inconsistencies are corrected in the frame window where the character information is expected, i.e. the current frame and the adjacent frames of lookahead will contain the same header and character information.
- Any time the current frame contains TTY_SILENCE or a TTY character, and the frame of lookback contains NON_TTY. This makes it harder for the decoder to erroneously go into TTY_MODE, thus preventing false alarms when speech is present.

In the normal course of TTY transmissions, TTY_SILENCE messages are sent first, followed by the TTY character information. When 3 TTY_SILENCE messages are received in a window of 5 frames, the decoder's output is muted until a NON_TTY frame is received or until the voting results in a TTY character to be regenerated. If a new character is detected, it will only be regenerated if it was preceded by TTY_SILENCE. If it is not, the character information is ignored and the decoder returns to NON_TTY mode. This is done to prevent false alarms from packets that look like TTY packets but are really speech packets with the ACB gain coincidentally set to zero.

Once tty_dec( ) makes its decision on the current frame, tty_dec( ) calls tty_gen( ) to generate the appropriate PCM samples.

6.3.6.3 Baudot Generator

Once the current frame is labeled by tty_dec( ), tty_gen( ) is called to fill the PCM buffer with the appropriate Baudot tones. In the case of NON_TTY, the PCM buffer is returned unmodified. In the case of TTY_SILENCE, the PCM is muted.

Generating TTY characters is more involved because one character spans many frames, so tty_gen( ) must generate the Baudot tones one subframe at a time. When a TTY character needs to be regenerated, tty_gen( ) puts a subframe's worth of samples in the PCM buffer. It keeps track of which bit it is in the middle of generating and the number of samples left to generate for that bit, so that the next time it is called, it can pick up where it left off. Once tty_gen( ) begins to generate a character, it will generate the entire character before it will generate the next character. This is done so that the repeater will only generate valid TTY characters.

There exists logic in tty_gen( ) to detect when the next character arrives before the current one is finished. If the next character arrives before the current one can be regenerated, a minimum of 1 stop bit is generated.

There exists a provision in the ITU-T Recommendation V.18 for the TTY/TDD device to extend its stop bit in order to prevent a TTY/TDD device from detecting its own echo. This routine will extend the stop bit a maximum of 300 ms if a TTY character is followed by silence. If a new character arrives before 300 ms has elapsed, the extended stop bit is terminated and the new character is generated immediately.

The tones themselves are generated by tone_gen( ). Before tty_gen( ) returns, it updates the decoder's lookback field in the TTY history buffer with the information corresponding to the last samples generated. For example, if tty_gen( ) finished generating a character in the middle of the subframe and started generating silence, the lookback field is updated with TTY_SILENCE.

6.3.6.4 Tone Generator

The routine tone_gen( ) is a sine wave generator. Given a frequency and the number of samples, it will generate the PCM samples by using a 2 tap marginally stable IIR filter. The filter implements the trigonometric identity:

$$\cos(k\omega) = 2 \cdot \cos(\omega) \cdot \cos\big((k-1)\omega\big) - \cos\big((k-2)\omega\big)$$

It is a zero excitation filter, using only its past 2 samples and the cosine of the frequency to be generated, to produce the next sample.

7 APPENDIX A. SUMMARY OF NOTATION

Table 7-1. Summary of Noise Suppression Notation

Parameter	Section	Name/Description
$\alpha(m)$	4.1.2.5	Exponential windowing factor.
$\alpha_{ch}(m)$	4.1.2.2	Channel energy smoothing factor.
$\Delta_E(m)$	4.1.2.5	Spectral deviation for the current frame.
$\{\gamma_{ch}\}$	4.1.2.8	Linear channel gains.
$\{\gamma_{dB}\}$	4.1.2.8	Channel gains (in dB).
γ_n	4.1.2.8	Overall gain factor.
μ_g	4.1.2.8	Gain slope.
$\{\sigma_q\}$	4.1.2.3	Quantized channel SNR indices, in 0.375 dB increments.
$\{\sigma'_q\}$	4.1.2.7	Modified, quantized channel SNR indices.
$\{\sigma''_q\}$	4.1.2.7	Limited, modified, quantized channel SNR indices.
ζ_d	4.1.2.11	Deemphasis factor.
ζ_p	4.1.2.1	Preemphasis factor.
D	4.1.2.1	Input overlap (delay).
$\{d(m)\}$	4.1.2.1	Input overlap buffer.
$E_{ch}(m)$	4.1.2.2	Channel energy estimate vector.
$E_{dB}(m)$	4.1.2.4	Estimated log power spectra for the current frame.
$\overline{E}_{dB}(m)$	4.1.2.4	Estimated long-term log power spectra for the current frame.
E_{init}	4.1.2.2	Minimum allowable channel noise initialization energy.
$E_{\min}$	4.1.2.2	Minimum allowable channel energy.
$E_n(m)$	4.1.2.10	Channel noise energy estimate vector.
$E_{tot}(m)$,	4.1.2.4	Total channel energy for the current frame.
f_H	4.1.2.2	Frequency combining table, high limit.
f_L	4.1.2.2	Frequency combining table, low limit.
$\{G(k)\}$	4.1.2.1	Frequency domain input DFT buffer.
$\{g(n)\}$	4.1.2.1	Time domain input DFT buffer.
$\{h'(n)\}$	4.1.2.11	Overlap-and-add output buffer.
$\{H(k)\}$	4.1.2.9	Frequency domain output DFT buffer.
$\{h(m)\}$	4.1.2.11	Time domain output DFT buffer.
L	4.1.2.1	Frame length = 80.
M	4.1.2.1	DFT sequence length = 128.
m	4.1.2.1	The current 10 ms frame.

N_c	4.1.2.2	Number of channels = 16.
$\{s'(n)\}$	4.1.2	Noise suppressor output signal.
$\{s_{hp}(n)\}$	4.1.2	High pass filter output/noise suppressor input signal.
V	4.1.2.3	Voice metric table.
$v(m)$	4.1.2.3	Voice metric sum.

Table 7-2. Summary of Model Parameter Estimation Notation

Parameter	Section	Name/Description
$\{s'(n)\}$	4.2	Noise suppressor output signal.
$\Omega(m)$	4.2	Vector of unquantized line spectral pairs for frame *m*.
$\gamma_{lpc}(m)$	4.2	LPC prediction gain for frame *m*.
$\mathrm{a}(m)$	4.2	Vector of unquantized linear predictive coefficients for frame *m*.
$\{\varepsilon(n)\}$	4.2	Prediction residual.
τ	4.2	Long-term predictor (pitch) delay.
β	4.2	Long-term prediction gain.
LPCFLAG	4.2	Spectral transition indicator.
$\{h(n)\}$	4.2.1	Impulse response of the formant filter.

Table 7-3. Summary of Rate Determination Algorithm Notation

Parameter	Section	Name/Description
β	4.2.3	Long-term prediction gain
$BE_{f(i)}$	4.3.1.1	Energy in the *i*th frequency band.
$B_{f(i)}(m)$	4.3.1.2	Background noise estimate for the *i*th frequency band in the *m*th frame.
$E^{sm}{}_{f(i)}(m)$	4.3.2.1	Smoothed energy estimate for the *i*th frequency band in the *m*th frame.
$f(i)$	4.3.1	Frequency span of band-pass filter *i*.
$h_i(n)$	4.3.1.1	Impulse response of the *i*th frequency band filter.
Hangover	4.3.1.4	Number of frames after a Rate 1 frame required before a non-Rate 1 frame can be encoded.
L_h	4.3.1.1	The length of the impulse response of the band-pass filters.
lownoise(i)	4.3.2.2	Lower bound on the background noise estimate in the *i*th frequency band.
$R_W(k)$	4.3.1.1	*k*th value of the bandwidth expanded autocorrelation function for the current frame.
$R_{f(i)}(k)$	4.3.1.1	Autocorrelation function of the *i*th frequency band impulse response.
Rate(m)	4.3.1.3	Encoding rate for the *m*th frame
$S_{f(i)}(m)$	4.3.1.2	Signal energy estimate in the *i*th frequency band for the *m*th frame.
$SNR_{f(i)}(m)$	4.3.1.2	Quantized Signal-to-Noise Ratio in the *i*th frequency band for the *m*th frame.
$T_i(B,SNR)$	4.3.1.2	Thresholds used to determine the data rate as a function of the background noise and the quantized SNR in each frequency band *i*.

1 **Table 7-4. Summary of Encoder/Decoder Notation**

Parameter	Section	Name/Description
q_{rate}	4.4	Quantized LSP codebooks for each rate.
$\Omega_q(m)$	4.5, 5.2, 5.3	Vector of quantized LSPs for frame m.
$\Omega_{nq}(m)$	4.5	Vector of unquantized LSPs for frame m.
m	4.5, 5.2	Current frame number.
M'	4.5, 5.2	Current subframe number.
L	4.5, 5.2	Subframe size.
LSPIDX	4.5	Vector of LSP indices corresponding to the quantized LSPs.
$ACBGIDX(m')$	4.5	Adaptive codebook index.
$FCBSIDX(m')$	4.5	Fixed codebook shape index.
$FCBGIDX(m')$	4.5	Fixed codebook gain index
$\tau(m)$	4.5, 5.2	Pitch delay estimate for frame m.
DELAY	4.5	Delay transmission code.
DDELAY	4.5	Delay difference.
β	4.5, 5.2	Long-term prediction gain
$\{\varepsilon(n)\}$	4.5, 5.2	Short-term prediction residual signal.
$\{\alpha_{zir}(n)\}$	4.5	Zero-input response of weighted filter.
$E(n)$	4.5, 5.2	Adaptive codebook excitation signal.
$E_I(n)$	4.5, 5.2	Adaptive codebook excitation signal.
τ_{acc}	4.5	Accumulated shift counter.
shiftstate	4.5	State of the shifted residual buffer.
$\dot{d}(m', j)$	4.5, 5.2	Interpolated delay estimates for each subframe, m'.
$g_c(m')$	4.5, 5.2	Fixed codebook gain for each subframe, m'.
$g_p(m')$	4.5, 5.2	Adaptive codebook gain for each subframe, m'.
$\left\{\hat{s}_w(n)\right\}$	4.5.4.9	Weighted modified original speech vector.
$\left\{x_w(n)\right\}$	4.5.4.10.1	Target vector in the perceptual domain.
n_m	4.5.6	Pointer to the last sample in the shifted residual.
$\left\{\hat{\varepsilon}(n)\right\}$	4.5.6	Modified residual signal.
$\left\{\tilde{\varepsilon}_t(n)\right\}$	4.5.6	Modified residual target.
$FER\_FLAG(m)$	5.1, 5.2, 5.3	Frame erasure flag for frame m.
Rate	5.1	Decoder rate of operation.

last_valid_rate	5.1	Last valid decoder rate of operation.
$E_{lv}(n)$	5.2.2.2	Last valid codebook memory.
$\left\{\hat{s}(n)\right\}$	5.2, 5.3	Decoder synthesized speech signal.
$\left\{\hat{s}_{pf}(n)\right\}$	5.2, 5.3	Decoder post-filtered synthesized speech signal.
$g_{pavg}(m')$	5.2.3	Decoder average fixed codebook gain.
$g_{pavg}(m')$	5.2.3	Decoder average adaptive codebook gain.

8 APPENDIX B. CODEBOOK MEMORIES AND CONSTANTS

Table 8-1. LSP Quantization Table, Rate 1, Codebook 1

j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$	j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$	j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$
1	1.420163E-2	1.938816E-2	23	3.891726E-2	5.657889E-2	45	3.185912E-2	4.621231E-2
2	2.916675E-2	6.517492E-2	24	6.048006E-2	1.045370E-1	46	6.189098E-2	7.332318E-2
3	2.066932E-2	4.975649E-2	25	2.691566E-2	3.571689E-2	47	4.417184E-2	5.792409E-2
4	3.947198E-2	9.558509E-2	26	4.111172E-2	7.333230E-2	48	7.935962E-2	1.411774E-1
5	2.270125E-2	3.966258E-2	27	4.126607E-2	4.851652E-2	49	2.474123E-2	3.236294E-2
6	5.387895E-2	6.283478E-2	28	7.180496E-2	1.062024E-1	50	3.365639E-2	8.046506E-2
7	2.905255E-2	5.734358E-2	29	3.380379E-2	4.243004E-2	51	3.379437E-2	5.449772E-2
8	4.482806E-2	1.153646E-1	30	5.918182E-2	7.974680E-2	52	6.536490E-2	9.527759E-2
9	1.941107E-2	3.468897E-2	31	4.701079E-2	6.285638E-2	53	2.933642E-2	4.284110E-2
10	4.375030E-2	6.752285E-2	32	9.420119E-2	1.300532E-1	54	5.278705E-2	8.161594E-2
11	3.554973E-2	4.940868E-2	33	1.942443E-2	2.727323E-2	55	4.007249E-2	6.181447E-2
12	6.992199E-2	8.672798E-2	34	3.708317E-2	6.648982E-2	56	6.758486E-2	1.171961E-1
13	2.778802E-2	4.657485E-2	35	2.801364E-2	5.159849E-2	57	3.030650E-2	3.869141E-2
14	5.791110E-2	6.745425E-2	36	5.344610E-2	9.259042E-2	58	4.831063E-2	7.423830E-2
15	4.746644E-2	5.502715E-2	37	2.549592E-2	4.328448E-2	59	4.375483E-2	5.228423E-2
16	7.888989E-2	1.224430E-1	38	5.518607E-2	7.361823E-2	60	8.323100E-2	1.098820E-1
17	2.217159E-2	3.026288E-2	39	3.398511E-2	6.053291E-2	61	3.756006E-2	4.532172E-2
18	3.391345E-2	7.177040E-2	40	6.181821E-2	1.345813E-1	62	6.601132E-2	7.975802E-2
19	3.179891E-2	4.989961E-2	41	2.356692E-2	3.552420E-2	63	5.032251E-2	5.901763E-2
20	6.115560E-2	8.733612E-2	42	5.108042E-2	6.795625E-2	64	8.771333E-2	1.631874E-1
21	2.675065E-2	3.967359E-2	43	3.834650E-2	5.234694E-2			
22	4.441010E-2	8.267313E-2	44	7.442758E-2	9.661083E-2			

Table 8-2. LSP Quantization Table, Rate 1, Codebook 2

j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$	j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$	j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$
1	5.219596E-2	8.384457E-2	23	9.266608E-2	1.467255E-1	45	7.772978E-2	1.202872E-1
2	1.058741E-1	1.286942E-1	24	1.792855E-1	2.197060E-1	46	1.306480E-1	1.843318E-1
3	5.483239E-2	1.338429E-1	25	7.064585E-2	9.999245E-2	47	6.919396E-2	1.842180E-1
4	1.177685E-1	1.940373E-1	26	1.065005E-1	1.794434E-1	48	2.039041E-1	2.497152E-1
5	5.360865E-2	1.113987E-1	27	8.792497E-2	1.252877E-1	49	7.076717E-2	9.031861E-2
6	1.199897E-1	1.474747E-1	28	1.536402E-1	1.978527E-1	50	1.084716E-1	1.619665E-1
7	8.003736E-2	1.429997E-1	29	8.884301E-2	1.124657E-1	51	7.168864E-2	1.510932E-1
8	1.640866E-1	2.098218E-1	30	1.482867E-1	1.675170E-1	52	1.387795E-1	2.188018E-1
9	5.210592E-2	9.952294E-2	31	8.165681E-2	1.692740E-1	53	6.759071E-2	1.267403E-1
10	8.675680E-2	1.859665E-1	32	2.078105E-1	2.310336E-1	54	1.334124E-1	1.688389E-1
11	7.773411E-2	1.315069E-1	33	6.149280E-2	8.362632E-2	55	9.618226E-2	1.587287E-1
12	1.605455E-1	1.819303E-1	34	1.144733E-1	1.367800E-1	56	1.864856E-1	2.365609E-1
13	7.422437E-2	1.104371E-1	35	6.871299E-2	1.380991E-1	57	8.234471E-2	1.021260E-1
14	1.186351E-1	1.753068E-1	36	1.105114E-1	2.153529E-1	58	1.003366E-1	1.949185E-1
15	6.615578E-2	1.644419E-1	37	5.556523E-2	1.222428E-1	59	9.959820E-2	1.364251E-1
16	1.968109E-1	2.166820E-1	38	1.205576E-1	1.610725E-1	60	1.824485E-1	2.036552E-1
17	6.053178E-2	9.454086E-2	39	8.322497E-2	1.554755E-1	61	9.788907E-2	1.211455E-1

18	1.062714E-1	1.480139E-1	40	1.616385E-1	2.282689E-1	62	1.454531E-1	1.836045E-1
19	5.874866E-2	1.477246E-1	41	6.291523E-2	1.062296E-1	63	9.583955E-2	1.721949E-1
20	1.348165E-1	2.015180E-1	42	8.291869E-2	2.067745E-1	64	2.232959E-1	2.464186E-1
21	6.596983E-2	1.164474E-1	43	8.847569E-2	1.358000E-1			
22	1.322972E-1	1.532673E-1	44	1.697722E-1	1.937739E-1			

Table 8-3. LSP Quantization Table, Rate 1, Codebook 3

j	$q_{rate}(3, 1, j)$	$q_{rate}(3, 2, j)$	$q_{rate}(3, 3, j)$	j	$q_{rate}(3, 1, j)$	$q_{rate}(3, 2, j)$	$q_{rate}(3, 3, j)$
1	1.364258E-1	1.686519E-1	2.046882E-1	257	1.476240E-1	1.812728E-1	2.047079E-1
2	1.857176E-1	2.287562E-1	2.519580E-1	258	1.937514E-1	2.209740E-1	2.617752E-1
3	1.227602E-1	1.859507E-1	2.794467E-1	259	1.320898E-1	1.948516E-1	2.835476E-1
4	1.964685E-1	2.644844E-1	2.893189E-1	260	2.077394E-1	2.705968E-1	2.922648E-1
5	1.256537E-1	1.505293E-1	2.761443E-1	261	1.277334E-1	1.668960E-1	2.838914E-1
6	1.963016E-1	2.417000E-1	2.882307E-1	262	2.053094E-1	2.478075E-1	2.836328E-1
7	1.400994E-1	2.223656E-1	2.746666E-1	263	1.542119E-1	2.250141E-1	2.700820E-1
8	2.599523E-1	2.753950E-1	3.109759E-1	264	2.675741E-1	2.844269E-1	3.093348E-1
9	1.584522E-1	1.885910E-1	2.073392E-1	265	1.688469E-1	1.870045E-1	2.024332E-1
10	1.956162E-1	2.213795E-1	2.870229E-1	266	2.024411E-1	2.167331E-1	2.930792E-1
11	1.694246E-1	2.016147E-1	2.756692E-1	267	1.636213E-1	2.156165E-1	2.827929E-1
12	2.123938E-1	2.642507E-1	3.179675E-1	268	2.255093E-1	2.662830E-1	3.178866E-1
13	1.829651E-1	1.995476E-1	2.295388E-1	269	1.891103E-1	2.056094E-1	2.221136E-1
14	2.152007E-1	2.624094E-1	2.824327E-1	270	2.212402E-1	2.602889E-1	2.925411E-1
15	1.464046E-1	2.369667E-1	2.900671E-1	271	1.555634E-1	2.468508E-1	2.896488E-1
16	2.453386E-1	3.033581E-1	3.422602E-1	272	2.484062E-1	3.052919E-1	3.553167E-1
17	1.374790E-1	1.582766E-1	2.392172E-1	273	1.271222E-1	1.580537E-1	2.541644E-1
18	2.019990E-1	2.201026E-1	2.695469E-1	274	2.049988E-1	2.194769E-1	2.783420E-1
19	1.183500E-1	2.302064E-1	2.835548E-1	275	1.333023E-1	2.296140E-1	2.869472E-1
20	2.255193E-1	2.722721E-1	3.060730E-1	276	2.367771E-1	2.679182E-1	3.082309E-1
21	1.356614E-1	1.916340E-1	2.659120E-1	277	1.408536E-1	2.034147E-1	2.732571E-1
22	1.957331E-1	2.319262E-1	3.143761E-1	278	2.076843E-1	2.345200E-1	3.245833E-1
23	1.679990E-1	2.277063E-1	2.769478E-1	279	1.771817E-1	2.295954E-1	2.835392E-1
24	2.501706E-1	3.016271E-1	3.210842E-1	280	2.613784E-1	3.011602E-1	3.217071E-1
25	1.334923E-1	2.012231E-1	2.338940E-1	281	1.485957E-1	2.077720E-1	2.469461E-1
26	2.064421E-1	2.387042E-1	2.775601E-1	282	2.143348E-1	2.480613E-1	2.722592E-1
27	1.790488E-1	1.957766E-1	2.806566E-1	283	1.763803E-1	1.968979E-1	2.922869E-1
28	2.061936E-1	2.640554E-1	3.330984E-1	284	1.981935E-1	2.754833E-1	3.490376E-1
29	1.751853E-1	1.911663E-1	2.575403E-1	285	1.761532E-1	1.932490E-1	2.695485E-1
30	2.283986E-1	2.452967E-1	3.089808E-1	286	2.369686E-1	2.500658E-1	3.068208E-1
31	1.808598E-1	2.435791E-1	2.966311E-1	287	1.760607E-1	2.540376E-1	3.035668E-1
32	2.761530E-1	3.082561E-1	3.468226E-1	288	2.829529E-1	3.017651E-1	3.539563E-1
33	1.371157E-1	1.800578E-1	2.209535E-1	289	1.453537E-1	1.836788E-1	2.347501E-1
34	1.813701E-1	2.267701E-1	2.703927E-1	290	1.938426E-1	2.306356E-1	2.678178E-1
35	1.252465E-1	1.796069E-1	3.103764E-1	291	1.389590E-1	1.867608E-1	3.131132E-1
36	1.907084E-1	2.877342E-1	3.134762E-1	292	1.999445E-1	2.776248E-1	3.250463E-1
37	1.304861E-1	1.604353E-1	3.002437E-1	293	1.429661E-1	1.713108E-1	3.030134E-1
38	1.973186E-1	2.563785E-1	2.784743E-1	294	2.077417E-1	2.586918E-1	2.887670E-1
39	1.585971E-1	2.373814E-1	2.629103E-1	295	1.717769E-1	2.402461E-1	2.732845E-1
40	2.618259E-1	2.777172E-1	3.313822E-1	296	2.710466E-1	2.851709E-1	3.274011E-1

41	1.641607E-1	1.858415E-1	2.356159E-1	297	1.698546E-1	1.875458E-1	2.244847E-1
42	2.094861E-1	2.214528E-1	2.921539E-1	298	2.152220E-1	2.273397E-1	2.950088E-1
43	1.668079E-1	2.136418E-1	2.706759E-1	299	1.755966E-1	2.179366E-1	2.748796E-1
44	2.298343E-1	2.883746E-1	3.062383E-1	300	2.346654E-1	2.895309E-1	3.164944E-1
45	1.821543E-1	2.008225E-1	2.401694E-1	301	1.899470E-1	2.049538E-1	2.469552E-1
46	2.249447E-1	2.698139E-1	2.914012E-1	302	2.372978E-1	2.683167E-1	2.906843E-1
47	1.639406E-1	2.503412E-1	2.783078E-1	303	1.699632E-1	2.533675E-1	2.925330E-1
48	2.567280E-1	2.951038E-1	3.532971E-1	304	2.706599E-1	2.971461E-1	3.561840E-1
49	1.402188E-1	1.766877E-1	2.467733E-1	305	1.525397E-1	1.701390E-1	2.527039E-1
50	2.152913E-1	2.292160E-1	2.642836E-1	306	2.191192E-1	2.359007E-1	2.697391E-1
51	1.210027E-1	2.183338E-1	3.223413E-1	307	1.422457E-1	2.181846E-1	3.282181E-1
52	2.542432E-1	2.739862E-1	2.962625E-1	308	2.614728E-1	2.780257E-1	3.023759E-1
53	1.603854E-1	1.837629E-1	2.815987E-1	309	1.535260E-1	1.907277E-1	2.928208E-1
54	1.878322E-1	2.374204E-1	3.297775E-1	310	2.092410E-1	2.498087E-1	3.247091E-1
55	1.777884E-1	2.267035E-1	3.023225E-1	311	1.751764E-1	2.386468E-1	3.063927E-1
56	2.751082E-1	2.937306E-1	3.123738E-1	312	2.732189E-1	3.039550E-1	3.205139E-1
57	1.701164E-1	1.852321E-1	2.461250E-1	313	1.639116E-1	1.896116E-1	2.562725E-1
58	2.217548E-1	2.399122E-1	2.868919E-1	314	2.269538E-1	2.401202E-1	2.927285E-1
59	1.950837E-1	2.083379E-1	2.883497E-1	315	1.955657E-1	2.119562E-1	2.973747E-1
60	2.375365E-1	2.750045E-1	3.397860E-1	316	2.410456E-1	2.884970E-1	3.363523E-1
61	1.883693E-1	2.043718E-1	2.573750E-1	317	1.949483E-1	2.094753E-1	2.563097E-1
62	2.472502E-1	2.605518E-1	3.021375E-1	318	2.478846E-1	2.633564E-1	3.112709E-1
63	1.669442E-1	2.469124E-1	3.188944E-1	319	1.691897E-1	2.358646E-1	3.362494E-1
64	2.781186E-1	3.130111E-1	3.653293E-1	320	2.860016E-1	3.254238E-1	3.596074E-1
65	1.452135E-1	1.630515E-1	2.249126E-1	321	1.562586E-1	1.767049E-1	2.143934E-1
66	2.056925E-1	2.208315E-1	2.528178E-1	322	2.089969E-1	2.239687E-1	2.608868E-1
67	1.211257E-1	1.963741E-1	3.001227E-1	323	1.357654E-1	2.035801E-1	3.055032E-1
68	2.155668E-1	2.656573E-1	2.992029E-1	324	2.189614E-1	2.794635E-1	2.994508E-1
69	1.091342E-1	1.784721E-1	2.883232E-1	325	1.340648E-1	1.783321E-1	2.901696E-1
70	2.035085E-1	2.403479E-1	2.963097E-1	326	2.132984E-1	2.400315E-1	3.003459E-1
71	1.531018E-1	2.254153E-1	2.848437E-1	327	1.643734E-1	2.264387E-1	2.871712E-1
72	2.502334E-1	2.777369E-1	3.248407E-1	328	2.507396E-1	2.808125E-1	3.353494E-1
73	1.663089E-1	1.941734E-1	2.116354E-1	329	1.636495E-1	1.971080E-1	2.211652E-1
74	2.012895E-1	2.260622E-1	2.932465E-1	330	2.081396E-1	2.308698E-1	2.961371E-1
75	1.495188E-1	2.142017E-1	2.838948E-1	331	1.591131E-1	2.181892E-1	2.955320E-1
76	2.218361E-1	2.852315E-1	3.200826E-1	332	2.398835E-1	2.818312E-1	3.260456E-1
77	1.895732E-1	2.065776E-1	2.303323E-1	333	1.893947E-1	2.081271E-1	2.384464E-1
78	2.312477E-1	2.468643E-1	2.898466E-1	334	2.329957E-1	2.596035E-1	2.934280E-1
79	1.391169E-1	2.591899E-1	2.980196E-1	335	1.605588E-1	2.551648E-1	3.028729E-1
80	2.445126E-1	2.826714E-1	3.612583E-1	336	2.535093E-1	2.960285E-1	3.677216E-1
81	1.225310E-1	1.685148E-1	2.708793E-1	337	1.301244E-1	1.748390E-1	2.604860E-1
82	2.043728E-1	2.303984E-1	2.717929E-1	338	2.102040E-1	2.335708E-1	2.830619E-1
83	1.426439E-1	2.224056E-1	2.920572E-1	339	1.523655E-1	2.253388E-1	3.037210E-1
84	2.426437E-1	2.774294E-1	2.971355E-1	340	2.405586E-1	2.771922E-1	3.058919E-1
85	1.520486E-1	1.969211E-1	2.610132E-1	341	1.637288E-1	1.947794E-1	2.692536E-1
86	2.178750E-1	2.458404E-1	3.081386E-1	342	2.257094E-1	2.409027E-1	3.180606E-1
87	1.901093E-1	2.310991E-1	2.801782E-1	343	1.920551E-1	2.298578E-1	2.898267E-1
88	2.543142E-1	2.940798E-1	3.396492E-1	344	2.627597E-1	3.042922E-1	3.356806E-1
89	1.566986E-1	2.085975E-1	2.280108E-1	345	1.660712E-1	2.068192E-1	2.397125E-1

90	2.250887E-1	2.500145E-1	2.762502E-1	346	2.239156E-1	2.501069E-1	2.852962E-1
91	1.782190E-1	1.982282E-1	3.041989E-1	347	1.884023E-1	2.037937E-1	3.030411E-1
92	2.085672E-1	2.923954E-1	3.467869E-1	348	2.306990E-1	2.870441E-1	3.498028E-1
93	1.710521E-1	2.034388E-1	2.626443E-1	349	1.820254E-1	2.140735E-1	2.634700E-1
94	2.302755E-1	2.588175E-1	3.119865E-1	350	2.372978E-1	2.650254E-1	3.178155E-1
95	1.853336E-1	2.457602E-1	3.105540E-1	351	1.892787E-1	2.588022E-1	3.048662E-1
96	2.894139E-1	3.110956E-1	3.464762E-1	352	2.972431E-1	3.171531E-1	3.565839E-1
97	1.503324E-1	1.675382E-1	2.401829E-1	353	1.586075E-1	1.786598E-1	2.419194E-1
98	1.799717E-1	2.371686E-1	2.608997E-1	354	1.948874E-1	2.416959E-1	2.621767E-1
99	1.498662E-1	1.978901E-1	3.079166E-1	355	1.581244E-1	2.117531E-1	3.113522E-1
100	2.107997E-1	2.881801E-1	3.297472E-1	356	2.169027E-1	2.987968E-1	3.209941E-1
101	1.317111E-1	1.659065E-1	3.228980E-1	357	1.492728E-1	1.749641E-1	3.153344E-1
102	2.148320E-1	2.528221E-1	2.975471E-1	358	2.216223E-1	2.561791E-1	3.039030E-1
103	1.837604E-1	2.375236E-1	2.746100E-1	359	1.759796E-1	2.435055E-1	2.858017E-1
104	2.555752E-1	2.754392E-1	3.460219E-1	360	2.645904E-1	2.855416E-1	3.451078E-1
105	1.826622E-1	1.994709E-1	2.160517E-1	361	1.801371E-1	2.052794E-1	2.222560E-1
106	2.092403E-1	2.224067E-1	3.023829E-1	362	2.107962E-1	2.263154E-1	3.144269E-1
107	1.840883E-1	2.113278E-1	2.825381E-1	363	1.791512E-1	2.094397E-1	2.932809E-1
108	2.411711E-1	2.970360E-1	3.159793E-1	364	2.497190E-1	2.912577E-1	3.271623E-1
109	1.968047E-1	2.118159E-1	2.416477E-1	365	1.987002E-1	2.158968E-1	2.499602E-1
110	2.427620E-1	2.585866E-1	2.932044E-1	366	2.407264E-1	2.648577E-1	2.996396E-1
111	1.589055E-1	2.650770E-1	2.898813E-1	367	1.712497E-1	2.681662E-1	3.035727E-1
112	2.580606E-1	3.189032E-1	3.478468E-1	368	2.695556E-1	3.161006E-1	3.565707E-1
113	1.487664E-1	1.668539E-1	2.668274E-1	369	1.505648E-1	1.841909E-1	2.686748E-1
114	2.159423E-1	2.299383E-1	2.760416E-1	370	2.169412E-1	2.408140E-1	2.789422E-1
115	1.384105E-1	2.392834E-1	3.279724E-1	371	1.353995E-1	2.605865E-1	3.326049E-1
116	2.437653E-1	2.884085E-1	3.060487E-1	372	2.561510E-1	2.878229E-1	3.061564E-1
117	1.701571E-1	1.899863E-1	2.812192E-1	373	1.663988E-1	1.887218E-1	2.930237E-1
118	2.191170E-1	2.580053E-1	3.265720E-1	374	2.292141E-1	2.615654E-1	3.274941E-1
119	1.921636E-1	2.236142E-1	2.986831E-1	375	1.982666E-1	2.329705E-1	2.991343E-1
120	2.735454E-1	3.120781E-1	3.307666E-1	376	2.870463E-1	3.071038E-1	3.272981E-1
121	1.624521E-1	2.049309E-1	2.533377E-1	377	1.758987E-1	2.118986E-1	2.513329E-1
122	2.238553E-1	2.376711E-1	3.032020E-1	378	2.320674E-1	2.446222E-1	2.994437E-1
123	1.939553E-1	2.123356E-1	3.075669E-1	379	1.907801E-1	2.120900E-1	3.250593E-1
124	2.299127E-1	2.975811E-1	3.374992E-1	380	2.315312E-1	3.141661E-1	3.427359E-1
125	1.893354E-1	2.041481E-1	2.786098E-1	381	1.950999E-1	2.095543E-1	2.794835E-1
126	2.423036E-1	2.731631E-1	3.153617E-1	382	2.404161E-1	2.696048E-1	3.280155E-1
127	1.550097E-1	2.880952E-1	3.359964E-1	383	1.718009E-1	2.822331E-1	3.147493E-1
128	2.737162E-1	3.312155E-1	3.625391E-1	384	2.692438E-1	3.384625E-1	3.799357E-1
129	1.523894E-1	1.726191E-1	1.905857E-1	385	1.599346E-1	1.779668E-1	2.008186E-1
130	1.969883E-1	2.263098E-1	2.461975E-1	386	2.019797E-1	2.306685E-1	2.567733E-1
131	1.205552E-1	2.063698E-1	2.811998E-1	387	1.340243E-1	2.109616E-1	2.846877E-1
132	1.937094E-1	2.719005E-1	3.013329E-1	388	2.037129E-1	2.830531E-1	3.033094E-1
133	1.367012E-1	1.540932E-1	2.822584E-1	389	1.445289E-1	1.647281E-1	2.850794E-1
134	1.972992E-1	2.536563E-1	2.903151E-1	390	2.062856E-1	2.486490E-1	2.963831E-1
135	1.434638E-1	2.438729E-1	2.755337E-1	391	1.581382E-1	2.343177E-1	2.796500E-1
136	2.584773E-1	2.732799E-1	3.211191E-1	392	2.649956E-1	2.799007E-1	3.186194E-1
137	1.544062E-1	1.937935E-1	2.158842E-1	393	1.665375E-1	1.842794E-1	2.145475E-1

138	2.059795E-1	2.242770E-1	2.857324E-1	394	2.030519E-1	2.351105E-1	2.887560E-1
139	1.745353E-1	2.084824E-1	2.796685E-1	395	1.684227E-1	2.039462E-1	2.874789E-1
140	2.188446E-1	2.724863E-1	3.270956E-1	396	2.317270E-1	2.740864E-1	3.247552E-1
141	1.776097E-1	2.129902E-1	2.391197E-1	397	1.853562E-1	2.141131E-1	2.290304E-1
142	2.291638E-1	2.591659E-1	2.835147E-1	398	2.424826E-1	2.606555E-1	2.830303E-1
143	1.573532E-1	2.399613E-1	3.042631E-1	399	1.675623E-1	2.420275E-1	2.994620E-1
144	2.456138E-1	3.168245E-1	3.429094E-1	400	2.388099E-1	3.190039E-1	3.584159E-1
145	1.429532E-1	1.619054E-1	2.537102E-1	401	1.379083E-1	1.547878E-1	2.656112E-1
146	2.101928E-1	2.228477E-1	2.711038E-1	402	2.110193E-1	2.246073E-1	2.799547E-1
147	1.268439E-1	2.167091E-1	2.977347E-1	403	1.375699E-1	2.251285E-1	3.093129E-1
148	2.310001E-1	2.801091E-1	2.997074E-1	404	2.292399E-1	2.761510E-1	3.152418E-1
149	1.529805E-1	1.939969E-1	2.728957E-1	405	1.604875E-1	1.954612E-1	2.831695E-1
150	2.128607E-1	2.415454E-1	3.165188E-1	406	2.185057E-1	2.381972E-1	3.303401E-1
151	1.711547E-1	2.224697E-1	2.937865E-1	407	1.819913E-1	2.330270E-1	2.932760E-1
152	2.519882E-1	3.042550E-1	3.312700E-1	408	2.545523E-1	3.143942E-1	3.363923E-1
153	1.331889E-1	2.079250E-1	2.553621E-1	409	1.440958E-1	2.266402E-1	2.505951E-1
154	2.120449E-1	2.421897E-1	2.889037E-1	410	2.151880E-1	2.514173E-1	2.850440E-1
155	1.846125E-1	2.011436E-1	2.863608E-1	411	1.876744E-1	2.044589E-1	2.941690E-1
156	2.182867E-1	2.767524E-1	3.445815E-1	412	2.304948E-1	2.684524E-1	3.523701E-1
157	1.835622E-1	1.994785E-1	2.621566E-1	413	1.850221E-1	1.990753E-1	2.719306E-1
158	2.331305E-1	2.495969E-1	3.158428E-1	414	2.425694E-1	2.553892E-1	3.113993E-1
159	1.898990E-1	2.468749E-1	2.971325E-1	415	1.951661E-1	2.491021E-1	2.989985E-1
160	2.750225E-1	3.224903E-1	3.469777E-1	416	2.836542E-1	3.146003E-1	3.556194E-1
161	1.423053E-1	1.926892E-1	2.161559E-1	417	1.514900E-1	1.977298E-1	2.324675E-1
162	1.956762E-1	2.222686E-1	2.765874E-1	418	2.000299E-1	2.301013E-1	2.819339E-1
163	1.332415E-1	1.977918E-1	3.228979E-1	419	1.387113E-1	1.918166E-1	3.457804E-1
164	1.848651E-1	2.971062E-1	3.261052E-1	420	1.965804E-1	3.047148E-1	3.405534E-1
165	1.502037E-1	1.767813E-1	2.915362E-1	421	1.381543E-1	1.885431E-1	2.994612E-1
166	2.031445E-1	2.596162E-1	2.991560E-1	422	2.056665E-1	2.689049E-1	3.055372E-1
167	1.654890E-1	2.383421E-1	2.874939E-1	423	1.724479E-1	2.335584E-1	2.936252E-1
168	2.710713E-1	2.895445E-1	3.195210E-1	424	2.701454E-1	2.986548E-1	3.285564E-1
169	1.685984E-1	1.988256E-1	2.303476E-1	425	1.754894E-1	1.913616E-1	2.355853E-1
170	2.138117E-1	2.344718E-1	2.909596E-1	426	2.205488E-1	2.347740E-1	2.953977E-1
171	1.746054E-1	2.172560E-1	2.856881E-1	427	1.856524E-1	2.223491E-1	2.798839E-1
172	2.285035E-1	2.961903E-1	3.165347E-1	428	2.294570E-1	3.045463E-1	3.246843E-1
173	1.871726E-1	2.205474E-1	2.396887E-1	429	1.869008E-1	2.154694E-1	2.518568E-1
174	2.288848E-1	2.635832E-1	3.013295E-1	430	2.349105E-1	2.712174E-1	2.998947E-1
175	1.778971E-1	2.581315E-1	2.814877E-1	431	1.851424E-1	2.560710E-1	2.932913E-1
176	2.595136E-1	3.072044E-1	3.487936E-1	432	2.638837E-1	3.071275E-1	3.625467E-1
177	1.452244E-1	1.787160E-1	2.591870E-1	433	1.609976E-1	1.789379E-1	2.558083E-1
178	2.190623E-1	2.382235E-1	2.604615E-1	434	2.256711E-1	2.437351E-1	2.686250E-1
179	1.436509E-1	2.097608E-1	3.158302E-1	435	1.550762E-1	2.303962E-1	3.210056E-1
180	2.501275E-1	2.791823E-1	3.051536E-1	436	2.517605E-1	2.796534E-1	3.142022E-1
181	1.489864E-1	2.012268E-1	2.825437E-1	437	1.569888E-1	2.074669E-1	2.899340E-1
182	2.083878E-1	2.356039E-1	3.453639E-1	438	2.174795E-1	2.596264E-1	3.406591E-1
183	1.858303E-1	2.216073E-1	3.107736E-1	439	1.768115E-1	2.310871E-1	3.175625E-1
184	2.809047E-1	2.954698E-1	3.254995E-1	440	2.829526E-1	2.998444E-1	3.368229E-1
185	1.729673E-1	1.970781E-1	2.458011E-1	441	1.820603E-1	1.987347E-1	2.519803E-1
186	2.194957E-1	2.447671E-1	2.935878E-1	442	2.258742E-1	2.524692E-1	2.933564E-1

187	1.839096E-1	2.150043E-1	3.003345E-1	443	2.008000E-1	2.177869E-1	3.022101E-1
188	2.453386E-1	2.685953E-1	3.483304E-1	444	2.474238E-1	2.868829E-1	3.478206E-1
189	1.929574E-1	2.066251E-1	2.673364E-1	445	2.011281E-1	2.147469E-1	2.622697E-1
190	2.548456E-1	2.686423E-1	3.035479E-1	446	2.539634E-1	2.694780E-1	3.121338E-1
191	1.768531E-1	2.593310E-1	3.162008E-1	447	1.910349E-1	2.557382E-1	3.325596E-1
192	2.909291E-1	3.156348E-1	3.687235E-1	448	2.910537E-1	3.314584E-1	3.685885E-1
193	1.571170E-1	1.735529E-1	2.287365E-1	449	1.572299E-1	1.853741E-1	2.253613E-1
194	2.125093E-1	2.305012E-1	2.522180E-1	450	2.080513E-1	2.383509E-1	2.642129E-1
195	1.425218E-1	2.019799E-1	2.930122E-1	451	1.468483E-1	2.130001E-1	3.001926E-1
196	2.149197E-1	2.780651E-1	3.141761E-1	452	2.186306E-1	2.902638E-1	3.090458E-1
197	1.359473E-1	1.810559E-1	2.754754E-1	453	1.436992E-1	1.878152E-1	2.837699E-1
198	1.984167E-1	2.416738E-1	3.051734E-1	454	2.073280E-1	2.450887E-1	3.089564E-1
199	1.595173E-1	2.315801E-1	2.954125E-1	455	1.642281E-1	2.278267E-1	3.089079E-1
200	2.582036E-1	2.873481E-1	3.203520E-1	456	2.619197E-1	2.913337E-1	3.315280E-1
201	1.748407E-1	1.928833E-1	2.112500E-1	457	1.706489E-1	2.021575E-1	2.178278E-1
202	2.021685E-1	2.270257E-1	3.048841E-1	458	2.077961E-1	2.347048E-1	3.067838E-1
203	1.695321E-1	2.118262E-1	2.973554E-1	459	1.721188E-1	2.140574E-1	3.101518E-1
204	2.300337E-1	2.915044E-1	3.265894E-1	460	2.291162E-1	2.809499E-1	3.337743E-1
205	1.950461E-1	2.117092E-1	2.277058E-1	461	1.966222E-1	2.166531E-1	2.332797E-1
206	2.379269E-1	2.524116E-1	2.977522E-1	462	2.377892E-1	2.589713E-1	3.046092E-1
207	1.537629E-1	2.465416E-1	3.147689E-1	463	1.551820E-1	2.630326E-1	3.189431E-1
208	2.360757E-1	3.035689E-1	3.706245E-1	464	2.493888E-1	3.169709E-1	3.777625E-1
209	1.386603E-1	1.679500E-1	2.735153E-1	465	1.513636E-1	1.750107E-1	2.782458E-1
210	2.138062E-1	2.272672E-1	2.862763E-1	466	2.198102E-1	2.323602E-1	2.850349E-1
211	1.250806E-1	2.440984E-1	3.025488E-1	467	1.426306E-1	2.406029E-1	3.041251E-1
212	2.357149E-1	2.812088E-1	3.089037E-1	468	2.427649E-1	2.837621E-1	3.154812E-1
213	1.516914E-1	2.108778E-1	2.638130E-1	469	1.574675E-1	2.075241E-1	2.756749E-1
214	2.207304E-1	2.527779E-1	3.164137E-1	470	2.287586E-1	2.490922E-1	3.281394E-1
215	1.849247E-1	2.394248E-1	2.851208E-1	471	1.908727E-1	2.381252E-1	2.948946E-1
216	2.595485E-1	3.098099E-1	3.264237E-1	472	2.663893E-1	3.143214E-1	3.386695E-1
217	1.629307E-1	2.199009E-1	2.361486E-1	473	1.706442E-1	2.259800E-1	2.473724E-1
218	2.341942E-1	2.499443E-1	2.775491E-1	474	2.364428E-1	2.530035E-1	2.882204E-1
219	1.708702E-1	1.982916E-1	3.214126E-1	475	1.854238E-1	2.048889E-1	3.146088E-1
220	2.315669E-1	2.750151E-1	3.697104E-1	476	2.173790E-1	2.945536E-1	3.678310E-1
221	1.800024E-1	2.067010E-1	2.712049E-1	477	1.885640E-1	2.151743E-1	2.729997E-1
222	2.380753E-1	2.540062E-1	3.238276E-1	478	2.451021E-1	2.597704E-1	3.218856E-1
223	1.991483E-1	2.542739E-1	3.074797E-1	479	1.984442E-1	2.611607E-1	3.170979E-1
224	2.874286E-1	3.250451E-1	3.486346E-1	480	2.990139E-1	3.289653E-1	3.566812E-1
225	1.452850E-1	1.913592E-1	2.496914E-1	481	1.582488E-1	1.922057E-1	2.460591E-1
226	1.946593E-1	2.408212E-1	2.773027E-1	482	2.023854E-1	2.479658E-1	2.717497E-1
227	1.531510E-1	1.943757E-1	3.275504E-1	483	1.617108E-1	2.137081E-1	3.273847E-1
228	2.040858E-1	2.985957E-1	3.214801E-1	484	2.144197E-1	3.055525E-1	3.337216E-1
229	1.560097E-1	1.810127E-1	3.009317E-1	485	1.618200E-1	1.898973E-1	3.105016E-1
230	2.109624E-1	2.557703E-1	3.080861E-1	486	2.194363E-1	2.650296E-1	3.092888E-1
231	1.854441E-1	2.490213E-1	2.740298E-1	487	1.883039E-1	2.496332E-1	2.854995E-1
232	2.744935E-1	2.894420E-1	3.387949E-1	488	2.693254E-1	2.998070E-1	3.417225E-1
233	1.769419E-1	1.944769E-1	2.220773E-1	489	1.724060E-1	2.109773E-1	2.277732E-1
234	2.163775E-1	2.307358E-1	3.036893E-1	490	2.202815E-1	2.340158E-1	3.128461E-1

235	1.896835E-1	2.146608E-1	2.884454E-1	491	1.832573E-1	2.220620E-1	2.910524E-1
236	2.408273E-1	2.981418E-1	3.273784E-1	492	2.425312E-1	3.095276E-1	3.303897E-1
237	2.017878E-1	2.194418E-1	2.393275E-1	493	2.075467E-1	2.246626E-1	2.444201E-1
238	2.488125E-1	2.658659E-1	2.933824E-1	494	2.458582E-1	2.702860E-1	3.051321E-1
239	1.820278E-1	2.682791E-1	2.939914E-1	495	1.848406E-1	2.720968E-1	3.125311E-1
240	2.564986E-1	3.199845E-1	3.626632E-1	496	2.742526E-1	3.212524E-1	3.746582E-1
241	1.587993E-1	1.754337E-1	2.673899E-1	497	1.664258E-1	1.844916E-1	2.682781E-1
242	2.242593E-1	2.366683E-1	2.776391E-1	498	2.284237E-1	2.430254E-1	2.811849E-1
243	1.492034E-1	2.265853E-1	3.452556E-1	499	1.600913E-1	2.529533E-1	3.358223E-1
244	2.506558E-1	2.922649E-1	3.135743E-1	500	2.621100E-1	2.955819E-1	3.133541E-1
245	1.580963E-1	2.021932E-1	2.987117E-1	501	1.677028E-1	2.015369E-1	3.018016E-1
246	2.288209E-1	2.485573E-1	3.447265E-1	502	2.378230E-1	2.598948E-1	3.382311E-1
247	1.879721E-1	2.341094E-1	3.042356E-1	503	1.972062E-1	2.454909E-1	3.178954E-1
248	2.856571E-1	3.148781E-1	3.369315E-1	504	2.984553E-1	3.192098E-1	3.409717E-1
249	1.626800E-1	2.178201E-1	2.574368E-1	505	1.711953E-1	2.243278E-1	2.627361E-1
250	2.240498E-1	2.467398E-1	3.007959E-1	506	2.306269E-1	2.533102E-1	3.012068E-1
251	2.013546E-1	2.182867E-1	3.130363E-1	507	2.048142E-1	2.218816E-1	3.259666E-1
252	2.380285E-1	2.981035E-1	3.535038E-1	508	2.229875E-1	3.063391E-1	3.507172E-1
253	1.988300E-1	2.128771E-1	2.729808E-1	509	2.008554E-1	2.153599E-1	2.841435E-1
254	2.506165E-1	2.676600E-1	3.206119E-1	510	2.509517E-1	2.661893E-1	3.333606E-1
255	1.709018E-1	2.693304E-1	3.344282E-1	511	1.756103E-1	2.937913E-1	3.403269E-1
256	3.049889E-1	3.361967E-1	3.652354E-1	512	2.917451E-1	3.406025E-1	3.813972E-1

Table 8-4. LSP Quantization Table, Rate 1, Codebook 4

j	$q_{rate}(4, 1, j)$	$q_{rate}(4, 2, j)$	$q_{rate}(4, 3, j)$	j	$q_{rate}(4, 1, j)$	$q_{rate}(4, 2, j)$	$q_{rate}(4, 3, j)$
1	2.774615E-1	3.169721E-1	3.954983E-1	65	2.491101E-1	3.256969E-1	4.117283E-1
2	3.365604E-1	3.601570E-1	3.814730E-1	66	3.459292E-1	3.685775E-1	3.884733E-1
3	3.105093E-1	3.317324E-1	3.668644E-1	67	3.132197E-1	3.392295E-1	3.875979E-1
4	3.374710E-1	3.967953E-1	4.123563E-1	68	3.514540E-1	3.987303E-1	4.126562E-1
5	2.796604E-1	3.665201E-1	3.853135E-1	69	2.934871E-1	3.757631E-1	3.944881E-1
6	3.160390E-1	3.856093E-1	4.013048E-1	70	3.244708E-1	3.942029E-1	4.088827E-1
7	3.099604E-1	3.434107E-1	4.247455E-1	71	3.127108E-1	3.577203E-1	4.140612E-1
8	3.542436E-1	4.086993E-1	4.221680E-1	72	3.665072E-1	4.081713E-1	4.238914E-1
9	2.955872E-1	3.337411E-1	3.874217E-1	73	2.999657E-1	3.319934E-1	4.078602E-1
10	3.334464E-1	3.869748E-1	4.013531E-1	74	3.349252E-1	3.861430E-1	4.115381E-1
11	3.234128E-1	3.652697E-1	3.851933E-1	75	3.347880E-1	3.661962E-1	3.933471E-1
12	3.427320E-1	4.031925E-1	4.199204E-1	76	3.478479E-1	4.059265E-1	4.305073E-1
13	2.776818E-1	3.824950E-1	4.042742E-1	77	2.859529E-1	3.952833E-1	4.161193E-1
14	3.182480E-1	3.959853E-1	4.313532E-1	78	3.238674E-1	4.064767E-1	4.424828E-1
15	3.037114E-1	3.803197E-1	4.371736E-1	79	3.167167E-1	3.844516E-1	4.394110E-1
16	3.782888E-1	4.070773E-1	4.226791E-1	80	3.867729E-1	4.118246E-1	4.278315E-1
17	2.381165E-1	3.424543E-1	4.246247E-1	81	2.380724E-1	3.623424E-1	4.309317E-1
18	3.456157E-1	3.686811E-1	4.008173E-1	82	3.464500E-1	3.790829E-1	4.065678E-1
19	3.176881E-1	3.419027E-1	4.056018E-1	83	3.165766E-1	3.564686E-1	3.962183E-1
20	3.663690E-1	3.890399E-1	4.061545E-1	84	3.665392E-1	3.895909E-1	4.210556E-1
21	2.993980E-1	3.520217E-1	3.999557E-1	85	3.082914E-1	3.713243E-1	4.078674E-1
22	3.249919E-1	3.900288E-1	4.194787E-1	86	3.364352E-1	3.915144E-1	4.229771E-1
23	3.230258E-1	3.681143E-1	4.020878E-1	87	3.230355E-1	3.804473E-1	4.095502E-1
24	3.623263E-1	4.169280E-1	4.327737E-1	88	3.652281E-1	4.279104E-1	4.436913E-1

25	2.726964E-1	3.592050E-1	4.268807E-1	89	2.720380E-1	3.765968E-1	4.336859E-1
26	3.465399E-1	3.696166E-1	4.156212E-1	90	3.576658E-1	3.777616E-1	4.091790E-1
27	3.341091E-1	3.557363E-1	3.967496E-1	91	3.364986E-1	3.642159E-1	4.092555E-1
28	3.374690E-1	4.103927E-1	4.259868E-1	92	3.480824E-1	4.176318E-1	4.332845E-1
29	2.994690E-1	3.806483E-1	4.182841E-1	93	3.027545E-1	3.959748E-1	4.337173E-1
30	3.213782E-1	4.111980E-1	4.287925E-1	94	3.316763E-1	4.175872E-1	4.362398E-1
31	3.278412E-1	3.693451E-1	4.343956E-1	95	3.332876E-1	3.807991E-1	4.396207E-1
32	3.806691E-1	4.260864E-1	4.427546E-1	96	3.881120E-1	4.369336E-1	4.508293E-1
33	2.689437E-1	3.429430E-1	3.986815E-1	97	2.560266E-1	3.480152E-1	4.229226E-1
34	3.381029E-1	3.763388E-1	3.920432E-1	98	3.457740E-1	3.817258E-1	3.967941E-1
35	3.235935E-1	3.487421E-1	3.725520E-1	99	3.256238E-1	3.503919E-1	3.873307E-1
36	3.475508E-1	3.928854E-1	4.211699E-1	100	3.568681E-1	3.985748E-1	4.231772E-1
37	3.041828E-1	3.598167E-1	3.816333E-1	101	3.012262E-1	3.869070E-1	4.033356E-1
38	3.142214E-1	4.021086E-1	4.200853E-1	102	3.281784E-1	4.020902E-1	4.193893E-1
39	3.013066E-1	3.626627E-1	4.292628E-1	103	3.143854E-1	3.690439E-1	4.343753E-1
40	3.717703E-1	3.986964E-1	4.314390E-1	104	3.723211E-1	4.116724E-1	4.405187E-1
41	2.745913E-1	3.355955E-1	4.200797E-1	105	2.904797E-1	3.481219E-1	4.262165E-1
42	3.445408E-1	3.904518E-1	4.064121E-1	106	3.444388E-1	3.826664E-1	4.173211E-1
43	3.252398E-1	3.783445E-1	3.946733E-1	107	3.348668E-1	3.762357E-1	4.044752E-1
44	3.566835E-1	3.905742E-1	4.338511E-1	108	3.590254E-1	4.047219E-1	4.348384E-1
45	2.635013E-1	3.952601E-1	4.231164E-1	109	2.791280E-1	4.111066E-1	4.353606E-1
46	3.375207E-1	3.925635E-1	4.434158E-1	110	3.481255E-1	3.987321E-1	4.469274E-1
47	3.145223E-1	3.809686E-1	4.226764E-1	111	3.270189E-1	3.901073E-1	4.417075E-1
48	3.762351E-1	4.172987E-1	4.314513E-1	112	3.908584E-1	4.198139E-1	4.351535E-1
49	2.618550E-1	3.686461E-1	4.042606E-1	113	2.553193E-1	3.704060E-1	4.321886E-1
50	3.555802E-1	3.779945E-1	3.958682E-1	114	3.546520E-1	3.883327E-1	4.029561E-1
51	3.277428E-1	3.538728E-1	4.110406E-1	115	3.216082E-1	3.544898E-1	4.282998E-1
52	3.629606E-1	3.994670E-1	4.146902E-1	116	3.751635E-1	3.988340E-1	4.141774E-1
53	3.094109E-1	3.737961E-1	3.926725E-1	117	3.119536E-1	3.914307E-1	4.125525E-1
54	3.310163E-1	4.008016E-1	4.317593E-1	118	3.425288E-1	3.963653E-1	4.324974E-1
55	3.235731E-1	3.686196E-1	4.174551E-1	119	3.337444E-1	3.764224E-1	4.205370E-1
56	3.491159E-1	4.268401E-1	4.439140E-1	120	3.535291E-1	4.292311E-1	4.596993E-1
57	2.897386E-1	3.637593E-1	4.105118E-1	121	2.880179E-1	3.780000E-1	4.340117E-1
58	3.552865E-1	3.893313E-1	4.134324E-1	122	3.556835E-1	3.807800E-1	4.231455E-1
59	3.365659E-1	3.602225E-1	4.241790E-1	123	3.443583E-1	3.721849E-1	4.312654E-1
60	3.399327E-1	4.092288E-1	4.401849E-1	124	3.539661E-1	4.141667E-1	4.429413E-1
61	3.008897E-1	4.000811E-1	4.179557E-1	125	3.047702E-1	4.125175E-1	4.341831E-1
62	3.170521E-1	4.222881E-1	4.422296E-1	126	3.359134E-1	4.245908E-1	4.463785E-1
63	3.273368E-1	3.843117E-1	4.302886E-1	127	3.437382E-1	3.847662E-1	4.352714E-1
64	3.989909E-1	4.294984E-1	4.434752E-1	128	4.109413E-1	4.406630E-1	4.521134E-1

Table 8-5. LSP Quantization Table, Rate 1/2, Codebook 1

j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$	$q_{rate}(1, 3, j)$	j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$	$q_{rate}(1, 3, j)$
1	1.352263E-2	1.820813E-2	3.939407E-2	65	2.104905E-2	2.918910E-2	4.600358E-2
2	2.293929E-2	3.578312E-2	1.053529E-1	66	3.648633E-2	4.623870E-2	1.070442E-1
3	2.091065E-2	3.041591E-2	8.939411E-2	67	2.686521E-2	3.929376E-2	8.411799E-2
4	1.889090E-2	3.827222E-2	1.378204E-1	68	2.729040E-2	5.538051E-2	1.415862E-1
5	2.051438E-2	2.854812E-2	7.397622E-2	69	2.484767E-2	3.632777E-2	7.624309E-2

6	4.695103E-2	6.840319E-2	1.091238E-1	70	5.254308E-2	7.757787E-2	1.145680E-1
7	3.155572E-2	5.691400E-2	8.570576E-2	71	4.077414E-2	5.399238E-2	9.076405E-2
8	3.811819E-2	7.777847E-2	1.925329E-1	72	5.730433E-2	7.658031E-2	1.795790E-1
9	2.162972E-2	2.929089E-2	6.250420E-2	73	2.460324E-2	3.414084E-2	6.789908E-2
10	3.114140E-2	5.990793E-2	1.028607E-1	74	4.082201E-2	6.297838E-2	9.951913E-2
11	3.027993E-2	5.350124E-2	7.809258E-2	75	3.830250E-2	5.528575E-2	7.900193E-2
12	6.508462E-2	9.066247E-2	1.428510E-1	76	7.241113E-2	1.019039E-1	1.469796E-1
13	3.273404E-2	5.040278E-2	6.264923E-2	77	3.739022E-2	4.704639E-2	6.546845E-2
14	5.274399E-2	6.225743E-2	1.221983E-1	78	5.273975E-2	6.727704E-2	1.396804E-1
15	3.488404E-2	6.422224E-2	9.160246E-2	79	4.053654E-2	7.050813E-2	9.256686E-2
16	4.889844E-2	1.050580E-1	1.688135E-1	80	4.434253E-2	1.103672E-1	1.996363E-1
17	2.357911E-2	3.210347E-2	5.608996E-2	81	2.549207E-2	3.476040E-2	6.059020E-2
18	2.772528E-2	4.872818E-2	1.012242E-1	82	4.354655E-2	5.323695E-2	1.083260E-1
19	2.743480E-2	4.049659E-2	9.349261E-2	83	2.795998E-2	4.913248E-2	8.842845E-2
20	4.383601E-2	6.032613E-2	1.524009E-1	84	4.980519E-2	8.817289E-2	1.525973E-1
21	2.689949E-2	4.529064E-2	6.498004E-2	85	3.193463E-2	4.621693E-2	6.852064E-2
22	5.160590E-2	6.083122E-2	1.087996E-1	86	5.802463E-2	6.842687E-2	1.150853E-1
23	4.200649E-2	6.118451E-2	8.544740E-2	87	4.339047E-2	6.905756E-2	8.449844E-2
24	7.135027E-2	1.019721E-1	1.746410E-1	88	7.396916E-2	1.192405E-1	1.773402E-1
25	2.889067E-2	4.139644E-2	5.259280E-2	89	3.187675E-2	4.596974E-2	5.723726E-2
26	3.163645E-2	6.635321E-2	1.249503E-1	90	4.508738E-2	5.665094E-2	1.320058E-1
27	4.302895E-2	5.140233E-2	7.968777E-2	91	4.590970E-2	5.455804E-2	8.614233E-2
28	5.709708E-2	1.084445E-1	1.440756E-1	92	7.446858E-2	1.138154E-1	1.615706E-1
29	3.388403E-2	5.047469E-2	7.297654E-2	93	3.975096E-2	4.953595E-2	7.225423E-2
30	6.542657E-2	7.909877E-2	1.155706E-1	94	6.762578E-2	8.310290E-2	1.279901E-1
31	3.854235E-2	7.331254E-2	1.023075E-1	95	5.762581E-2	6.953264E-2	1.050130E-1
32	6.578245E-2	1.029098E-1	2.118744E-1	96	6.853135E-2	1.217588E-1	2.206266E-1
33	1.547279E-2	2.045597E-2	5.461213E-2	97	2.184805E-2	2.991309E-2	5.162080E-2
34	2.279502E-2	3.909542E-2	1.194438E-1	98	3.643432E-2	4.917951E-2	1.232772E-1
35	3.068892E-2	4.545402E-2	8.204189E-2	99	3.896113E-2	4.766350E-2	8.617166E-2
36	2.259572E-2	4.791017E-2	1.718444E-1	100	4.146352E-2	6.880070E-2	1.693562E-1
37	2.710880E-2	4.017396E-2	7.019229E-2	101	3.355146E-2	4.178152E-2	7.371594E-2
38	4.957894E-2	7.929633E-2	1.048625E-1	102	5.802247E-2	8.703142E-2	1.129175E-1
39	3.060959E-2	5.640594E-2	9.495841E-2	103	4.802431E-2	5.694865E-2	1.007557E-1
40	6.342246E-2	9.116555E-2	1.847244E-1	104	5.988731E-2	8.579423E-2	2.013889E-1
41	2.433424E-2	3.919983E-2	6.314062E-2	105	2.993100E-2	3.948284E-2	6.463761E-2
42	3.380120E-2	6.608465E-2	1.110315E-1	106	3.886266E-2	8.074436E-2	1.155198E-1
43	3.517841E-2	5.793973E-2	7.207029E-2	107	3.494440E-2	6.289110E-2	8.049820E-2
44	6.490541E-2	8.658319E-2	1.546487E-1	108	6.888179E-2	9.924311E-2	1.603933E-1
45	2.919347E-2	5.162046E-2	6.944373E-2	109	3.642377E-2	5.340165E-2	6.701520E-2
46	5.945228E-2	7.198293E-2	1.274345E-1	110	5.834927E-2	7.852858E-2	1.417467E-1
47	5.318885E-2	6.381821E-2	9.882185E-2	111	4.864696E-2	7.267369E-2	9.483159E-2
48	8.682910E-2	1.411354E-1	1.917285E-1	112	5.855336E-2	1.362898E-1	1.986397E-1
49	2.499911E-2	3.625560E-2	5.037240E-2	113	2.608885E-2	3.734068E-2	5.578532E-2
50	2.822464E-2	5.445723E-2	1.126635E-1	114	4.585044E-2	5.605125E-2	1.179279E-1
51	3.626181E-2	4.590732E-2	9.433436E-2	115	4.288013E-2	5.147391E-2	9.753090E-2
52	5.704553E-2	7.463004E-2	1.591572E-1	116	6.376116E-2	8.735529E-2	1.683349E-1
53	2.729875E-2	4.566259E-2	7.525297E-2	117	3.767099E-2	4.582160E-2	7.865281E-2
54	5.128602E-2	8.511270E-2	1.235880E-1	118	6.751946E-2	8.986979E-2	1.194181E-1

55	4.914520E-2	5.934831E-2	9.226860E-2	119	5.463743E-2	6.668059E-2	8.938138E-2
56	7.069619E-2	1.054520E-1	1.926021E-1	120	7.730866E-2	1.217544E-1	1.995792E-1
57	2.807338E-2	4.185092E-2	5.871598E-2	121	3.156213E-2	4.517022E-2	6.257685E-2
58	4.644490E-2	7.066988E-2	1.260384E-1	122	3.787827E-2	8.034865E-2	1.389617E-1
59	4.184537E-2	6.304453E-2	7.661699E-2	123	5.083033E-2	6.187406E-2	8.311538E-2
60	8.424164E-2	1.132829E-1	1.436871E-1	124	8.963114E-2	1.287538E-1	1.648916E-1
61	4.176156E-2	5.594729E-2	7.098728E-2	125	4.735035E-2	5.757244E-2	7.652646E-2
62	5.551614E-2	9.501267E-2	1.277272E-1	126	7.168986E-2	9.898957E-2	1.300784E-1
63	5.909355E-2	7.367300E-2	9.659359E-2	127	6.290826E-2	7.907788E-2	1.051111E-1
64	7.841367E-2	1.414324E-1	2.174286E-1	128	8.806498E-2	1.652062E-1	2.132142E-1

Table 8-6. LSP Quantization Table, Rate 1/2, Codebook 2

j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$	$q_{rate}(2, 3, j)$	j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$	$q_{rate}(2, 3, j)$
1	9.759153E-2	1.237015E-1	1.694380E-1	65	1.140764E-1	1.330714E-1	1.731815E-1
2	9.495363E-2	2.010818E-1	2.268553E-1	66	1.135758E-1	1.903073E-1	2.416812E-1
3	9.004966E-2	1.491649E-1	2.265328E-1	67	8.591653E-2	1.639202E-1	2.379345E-1
4	1.703027E-1	1.972229E-1	2.499748E-1	68	1.929169E-1	2.150824E-1	2.391281E-1
5	1.087736E-1	1.519724E-1	1.751234E-1	69	1.372918E-1	1.594233E-1	1.797222E-1
6	1.302789E-1	2.132292E-1	2.296464E-1	70	1.404354E-1	2.220923E-1	2.409608E-1
7	1.249180E-1	1.873478E-1	2.047120E-1	71	1.403872E-1	1.896012E-1	2.056357E-1
8	2.006702E-1	2.289636E-1	2.694208E-1	72	2.116955E-1	2.365784E-1	2.812489E-1
9	8.983756E-2	1.253328E-1	2.105394E-1	73	9.030106E-2	1.271574E-1	2.335679E-1
10	9.623767E-2	2.071859E-1	2.541745E-1	74	1.101181E-1	2.093284E-1	2.728363E-1
11	1.056946E-1	1.788564E-1	2.001210E-1	75	1.167104E-1	1.778540E-1	2.228088E-1
12	1.560490E-1	2.195737E-1	2.910794E-1	76	1.816915E-1	2.322652E-1	2.749912E-1
13	1.373923E-1	1.599933E-1	1.946985E-1	77	1.465535E-1	1.694747E-1	1.902460E-1
14	1.072625E-1	2.377910E-1	2.707408E-1	78	1.092138E-1	2.632920E-1	2.884908E-1
15	1.429765E-1	2.015505E-1	2.184689E-1	79	1.498151E-1	2.113427E-1	2.288995E-1
16	2.142705E-1	2.718814E-1	3.012002E-1	80	1.976455E-1	2.832300E-1	3.148823E-1
17	1.107292E-1	1.336882E-1	1.548772E-1	81	1.244956E-1	1.460980E-1	1.661252E-1
18	1.066677E-1	1.766788E-1	2.627989E-1	82	1.348786E-1	1.830301E-1	2.892883E-1
19	9.163529E-2	1.745928E-1	2.193293E-1	83	9.330321E-2	1.839622E-1	2.385430E-1
20	1.840386E-1	2.279641E-1	2.477622E-1	84	1.928443E-1	2.395883E-1	2.584215E-1
21	1.105724E-1	1.582072E-1	1.960131E-1	85	1.237968E-1	1.655566E-1	2.084084E-1
22	1.335434E-1	2.322697E-1	2.518282E-1	86	1.511443E-1	2.358011E-1	2.592806E-1
23	1.559223E-1	1.779413E-1	2.180966E-1	87	1.506577E-1	1.900525E-1	2.283626E-1
24	1.922601E-1	2.495125E-1	2.899115E-1	88	1.981810E-1	2.567942E-1	3.089756E-1
25	1.137089E-1	1.378724E-1	2.029299E-1	89	1.284900E-1	1.490840E-1	1.983765E-1
26	1.025575E-1	1.848201E-1	2.921646E-1	90	9.205958E-2	2.122313E-1	2.929488E-1
27	1.365956E-1	1.586874E-1	2.413996E-1	91	1.416981E-1	1.723567E-1	2.584541E-1
28	1.728138E-1	2.493034E-1	3.004586E-1	92	1.967335E-1	2.297097E-1	2.957802E-1
29	1.368712E-1	1.572498E-1	2.109132E-1	93	1.470622E-1	1.689181E-1	2.073636E-1
30	1.289748E-1	2.451679E-1	2.676536E-1	94	1.363099E-1	2.603731E-1	2.826074E-1
31	1.668123E-1	1.889980E-1	2.313459E-1	95	1.810411E-1	2.018261E-1	2.388676E-1
32	2.322485E-1	2.631961E-1	3.167549E-1	96	2.453263E-1	2.801831E-1	3.119543E-1
33	9.245610E-2	1.199775E-1	1.912623E-1	97	1.041318E-1	1.330407E-1	1.898347E-1
34	1.130853E-1	2.084615E-1	2.293681E-1	98	1.232982E-1	2.096211E-1	2.478132E-1
35	1.007164E-1	1.406701E-1	2.580630E-1	99	1.240408E-1	1.598274E-1	2.588561E-1
36	1.670104E-1	2.181055E-1	2.625925E-1	100	1.870489E-1	2.124881E-1	2.596291E-1

37	1.254872E-1	1.626870E-1	1.844092E-1	101	1.242553E-1	1.737690E-1	1.928500E-1
38	1.524066E-1	2.071317E-1	2.475824E-1	102	1.589178E-1	2.253898E-1	2.432848E-1
39	1.374412E-1	1.802624E-1	2.176988E-1	103	1.534212E-1	1.918073E-1	2.092495E-1
40	2.078535E-1	2.492095E-1	2.698301E-1	104	2.271545E-1	2.511812E-1	2.726004E-1
41	9.352573E-2	1.491974E-1	2.046520E-1	105	1.099221E-1	1.571003E-1	2.200250E-1
42	1.119972E-1	2.252331E-1	2.470031E-1	106	1.327824E-1	2.194855E-1	2.670289E-1
43	1.093150E-1	1.938119E-1	2.138022E-1	107	1.268575E-1	1.988363E-1	2.179285E-1
44	1.751186E-1	2.525203E-1	2.750828E-1	108	1.914150E-1	2.524242E-1	2.726528E-1
45	1.369187E-1	1.774406E-1	1.979311E-1	109	1.552776E-1	1.795735E-1	2.007736E-1
46	1.368112E-1	2.374262E-1	2.847378E-1	110	1.175477E-1	2.478699E-1	3.082793E-1
47	1.607598E-1	2.008332E-1	2.180845E-1	111	1.657070E-1	2.103395E-1	2.291993E-1
48	2.337102E-1	2.663726E-1	2.918021E-1	112	2.256949E-1	2.844382E-1	3.121061E-1
49	1.191711E-1	1.397032E-1	1.877233E-1	113	1.295032E-1	1.484201E-1	1.801804E-1
50	1.310500E-1	1.936967E-1	2.604270E-1	114	1.547525E-1	1.977485E-1	2.672750E-1
51	1.082671E-1	1.651948E-1	2.395230E-1	115	1.285902E-1	1.761784E-1	2.399059E-1
52	2.031950E-1	2.259422E-1	2.494032E-1	116	2.149268E-1	2.376344E-1	2.587940E-1
53	1.238429E-1	1.457946E-1	2.156356E-1	117	1.283223E-1	1.593385E-1	2.266266E-1
54	1.712263E-1	2.380545E-1	2.579756E-1	118	1.557476E-1	2.477405E-1	2.737268E-1
55	1.669238E-1	1.886047E-1	2.111242E-1	119	1.757417E-1	1.979524E-1	2.191159E-1
56	2.106208E-1	2.624427E-1	2.831280E-1	120	2.186264E-1	2.458092E-1	3.004797E-1
57	1.057488E-1	1.362865E-1	2.200502E-1	121	1.177090E-1	1.455129E-1	2.380445E-1
58	9.729458E-2	2.334715E-1	2.961140E-1	122	1.180069E-1	2.237755E-1	2.941751E-1
59	1.342984E-1	1.939554E-1	2.391488E-1	123	1.513492E-1	1.881578E-1	2.487433E-1
60	1.642293E-1	2.700678E-1	2.941425E-1	124	1.893122E-1	2.695805E-1	2.937860E-1
61	1.427603E-1	1.650334E-1	2.241004E-1	125	1.498956E-1	1.745373E-1	2.374300E-1
62	1.464145E-1	2.479423E-1	3.007081E-1	126	1.397755E-1	2.717094E-1	3.078395E-1
63	1.747788E-1	2.193493E-1	2.381630E-1	127	1.839457E-1	2.077172E-1	2.267222E-1
64	2.363111E-1	2.906697E-1	3.280110E-1	128	2.545522E-1	2.966409E-1	3.248015E-1

Table 8-7. LSP Quantization Table, Rate 1/2, Codebook 3

j	$q_{rate}(3, 1, j)$	$q_{rate}(3, 2, j)$	$q_{rate}(3, 3, j)$	$q_{rate}(3, 4, j)$
1	2.369047E-1	2.561044E-1	3.169558E-1	4.075205E-1
2	2.975969E-1	3.234825E-1	3.476675E-1	3.745512E-1
3	2.737212E-1	2.982975E-1	3.299239E-1	3.835991E-1
4	3.078496E-1	3.328363E-1	3.893403E-1	4.055760E-1
5	2.338036E-1	2.602965E-1	3.673520E-1	4.043883E-1
6	2.975137E-1	3.153566E-1	3.851352E-1	4.021971E-1
7	2.856188E-1	3.108728E-1	3.650224E-1	3.848168E-1
8	3.352716E-1	3.552222E-1	3.819211E-1	3.986858E-1
9	2.002656E-1	2.505023E-1	3.703982E-1	4.320127E-1
10	3.079821E-1	3.337677E-1	3.581991E-1	3.783868E-1
11	2.600861E-1	3.255203E-1	3.568733E-1	3.847378E-1
12	3.013564E-1	3.413694E-1	4.002968E-1	4.173372E-1
13	2.670810E-1	2.976744E-1	3.697020E-1	3.891392E-1
14	2.726699E-1	3.497041E-1	3.919253E-1	4.063833E-1
15	2.528259E-1	3.496366E-1	3.845510E-1	4.059310E-1
16	3.429271E-1	3.742740E-1	4.054682E-1	4.203519E-1
17	2.524087E-1	2.803758E-1	3.214366E-1	3.884369E-1
18	2.969702E-1	3.171736E-1	3.653426E-1	4.027368E-1

19	2.819052E-1	3.014792E-1	3.343356E-1	4.076335E-1
20	3.268729E-1	3.471777E-1	3.750177E-1	4.053724E-1
21	2.363711E-1	3.164411E-1	3.487070E-1	3.820304E-1
22	2.878176E-1	3.136270E-1	4.051297E-1	4.233797E-1
23	2.775025E-1	3.018438E-1	3.722509E-1	4.192128E-1
24	3.289889E-1	3.619011E-1	4.020155E-1	4.192298E-1
25	2.249605E-1	2.746364E-1	3.770161E-1	3.947265E-1
26	3.010455E-1	3.404862E-1	3.748881E-1	4.025322E-1
27	2.598980E-1	3.303350E-1	3.574938E-1	4.086580E-1
28	3.009619E-1	3.564491E-1	4.047795E-1	4.225090E-1
29	2.209796E-1	3.164777E-1	4.017441E-1	4.207358E-1
30	2.797550E-1	3.307761E-1	4.111529E-1	4.326870E-1
31	2.642469E-1	3.166106E-1	3.838767E-1	4.366838E-1
32	3.443812E-1	3.853657E-1	4.249495E-1	4.415602E-1
33	2.194883E-1	2.364599E-1	3.424660E-1	4.249900E-1
34	2.914651E-1	3.222820E-1	3.728528E-1	3.916359E-1
35	2.747924E-1	3.165363E-1	3.453926E-1	3.745552E-1
36	3.105835E-1	3.352649E-1	3.875272E-1	4.230762E-1
37	2.232115E-1	2.984976E-1	3.684262E-1	3.902137E-1
38	2.890788E-1	3.265128E-1	3.763087E-1	4.095537E-1
39	2.638301E-1	3.089773E-1	3.814530E-1	4.046608E-1
40	3.470736E-1	3.647978E-1	3.867635E-1	4.045117E-1
41	2.184527E-1	2.756141E-1	3.627111E-1	4.182790E-1
42	3.150428E-1	3.408132E-1	3.786272E-1	3.963168E-1
43	2.797277E-1	3.312597E-1	3.600613E-1	3.811755E-1
44	3.186024E-1	3.380443E-1	4.090108E-1	4.303004E-1
45	2.641969E-1	2.906725E-1	3.685950E-1	4.318568E-1
46	2.726456E-1	3.635148E-1	3.965188E-1	4.200912E-1
47	2.265410E-1	3.500551E-1	3.938515E-1	4.125970E-1
48	3.530539E-1	3.699296E-1	4.096561E-1	4.263873E-1
49	2.607884E-1	2.851725E-1	3.459433E-1	3.975007E-1
50	3.011131E-1	3.282019E-1	3.560680E-1	4.108038E-1
51	2.881016E-1	3.095596E-1	3.437568E-1	4.248729E-1
52	3.104894E-1	3.514219E-1	3.937174E-1	4.155505E-1
53	2.223083E-1	3.267982E-1	3.779817E-1	3.986350E-1
54	3.029155E-1	3.227819E-1	3.985589E-1	4.254896E-1
55	2.771368E-1	3.199926E-1	3.774909E-1	4.291775E-1
56	3.387318E-1	3.581644E-1	4.083864E-1	4.254954E-1
57	2.187262E-1	2.843850E-1	3.940537E-1	4.163470E-1
58	3.010060E-1	3.440937E-1	3.690137E-1	4.150913E-1
59	2.807837E-1	3.330537E-1	3.767262E-1	3.975269E-1
60	3.143941E-1	3.626788E-1	4.236690E-1	4.418992E-1
61	2.664536E-1	3.085138E-1	3.974072E-1	4.174502E-1
62	2.942227E-1	3.419044E-1	4.127269E-1	4.348889E-1
63	2.873007E-1	3.324346E-1	3.788567E-1	4.382340E-1
64	3.571466E-1	3.981471E-1	4.298757E-1	4.442439E-1
65	2.296713E-1	2.510186E-1	3.410466E-1	4.043763E-1
66	2.944726E-1	3.349446E-1	3.604097E-1	3.836829E-1

67	2.882510E-1	3.117227E-1	3.316801E-1	3.651047E-1
68	3.248816E-1	3.456567E-1	3.883064E-1	4.059549E-1
69	2.508292E-1	2.776235E-1	3.707995E-1	3.904792E-1
70	2.935234E-1	3.283192E-1	3.921123E-1	4.094641E-1
71	2.836088E-1	3.038856E-1	3.785044E-1	3.973106E-1
72	3.340398E-1	3.528374E-1	3.972729E-1	4.143220E-1
73	2.218919E-1	2.518775E-1	3.717235E-1	4.317910E-1
74	3.132014E-1	3.411754E-1	3.655036E-1	3.885672E-1
75	2.713305E-1	3.391637E-1	3.626164E-1	3.957360E-1
76	3.075501E-1	3.477777E-1	4.010496E-1	4.327675E-1
77	2.593874E-1	2.872438E-1	3.868173E-1	4.060427E-1
78	2.854852E-1	3.440950E-1	4.020505E-1	4.194138E-1
79	2.657814E-1	3.400844E-1	3.694077E-1	4.270317E-1
80	3.537409E-1	3.844633E-1	4.117478E-1	4.261818E-1
81	2.438665E-1	2.683502E-1	3.422020E-1	3.984572E-1
82	2.931452E-1	3.347542E-1	3.617028E-1	3.984166E-1
83	2.913430E-1	3.131552E-1	3.365259E-1	3.877486E-1
84	3.056568E-1	3.629046E-1	3.881534E-1	4.055432E-1
85	2.174923E-1	3.117235E-1	3.759848E-1	4.289978E-1
86	2.911493E-1	3.293809E-1	4.039004E-1	4.223332E-1
87	2.903621E-1	3.095310E-1	3.789942E-1	4.136884E-1
88	3.295649E-1	3.774047E-1	4.065849E-1	4.247397E-1
89	2.464616E-1	2.715933E-1	3.663383E-1	4.307538E-1
90	3.141077E-1	3.370119E-1	3.804097E-1	4.110994E-1
91	2.765684E-1	3.273207E-1	3.588443E-1	4.289495E-1
92	3.171791E-1	3.589724E-1	4.047658E-1	4.403763E-1
93	2.427778E-1	3.349548E-1	3.969435E-1	4.133184E-1
94	2.888955E-1	3.256912E-1	4.228596E-1	4.437587E-1
95	2.775833E-1	3.254790E-1	3.891447E-1	4.410759E-1
96	3.591257E-1	3.906941E-1	4.210095E-1	4.357085E-1
97	2.201724E-1	2.477193E-1	3.543819E-1	4.253981E-1
98	3.060468E-1	3.279247E-1	3.669928E-1	3.931926E-1
99	2.708056E-1	3.168266E-1	3.456487E-1	4.117176E-1
100	3.231889E-1	3.454631E-1	3.897788E-1	4.215708E-1
101	2.461361E-1	3.123920E-1	3.721886E-1	3.958427E-1
102	3.038567E-1	3.243548E-1	3.857473E-1	4.141550E-1
103	2.810754E-1	3.186085E-1	3.856469E-1	4.027036E-1
104	3.535171E-1	3.727025E-1	3.962646E-1	4.130749E-1
105	2.092211E-1	2.952622E-1	3.803143E-1	4.312782E-1
106	3.253136E-1	3.467355E-1	3.707240E-1	3.910456E-1
107	2.863965E-1	3.435600E-1	3.697136E-1	3.898678E-1
108	3.277947E-1	3.473678E-1	4.054651E-1	4.245662E-1
109	2.530550E-1	3.026563E-1	3.821651E-1	4.298983E-1
110	2.944185E-1	3.707454E-1	3.954433E-1	4.195148E-1
111	2.628731E-1	3.450692E-1	4.041409E-1	4.219021E-1
112	3.650635E-1	3.824351E-1	4.134248E-1	4.312417E-1
113	2.487885E-1	2.823728E-1	3.657723E-1	4.109811E-1
114	3.072888E-1	3.278289E-1	3.776650E-1	4.362209E-1
115	2.985423E-1	3.206273E-1	3.505697E-1	4.276202E-1

116	3.162580E-1	3.629038E-1	3.882251E-1	4.256089E-1
117	2.390779E-1	3.313105E-1	3.703179E-1	4.159959E-1
118	3.037358E-1	3.328061E-1	4.102328E-1	4.277511E-1
119	2.960025E-1	3.190148E-1	3.810625E-1	4.269550E-1
120	3.325089E-1	3.625170E-1	4.233151E-1	4.409952E-1
121	2.351287E-1	2.747311E-1	4.120706E-1	4.354788E-1
122	2.980738E-1	3.553388E-1	3.790878E-1	4.153188E-1
123	2.834298E-1	3.452649E-1	3.703763E-1	4.099008E-1
124	3.235931E-1	3.654128E-1	4.128131E-1	4.310235E-1
125	2.766264E-1	3.005084E-1	4.022369E-1	4.266388E-1
126	2.945129E-1	3.614432E-1	4.196352E-1	4.369992E-1
127	2.908073E-1	3.416894E-1	3.927793E-1	4.434903E-1
128	3.593915E-1	4.039851E-1	4.408438E-1	4.530286E-1
129	2.232955E-1	2.391925E-1	3.237680E-1	4.216895E-1
130	2.947781E-1	3.187987E-1	3.532178E-1	3.919064E-1
131	2.590321E-1	3.102405E-1	3.435690E-1	3.950642E-1
132	3.164747E-1	3.385444E-1	3.933290E-1	4.122356E-1
133	2.401082E-1	2.846312E-1	3.602810E-1	3.799738E-1
134	2.969091E-1	3.157983E-1	3.949643E-1	4.151276E-1
135	2.854341E-1	3.049215E-1	3.619747E-1	4.057673E-1
136	3.374071E-1	3.566722E-1	3.851551E-1	4.111867E-1
137	2.240149E-1	2.601162E-1	3.947725E-1	4.195859E-1
138	3.006479E-1	3.416407E-1	3.702235E-1	3.895201E-1
139	2.659460E-1	3.250392E-1	3.743399E-1	3.923461E-1
140	3.160293E-1	3.404913E-1	4.023553E-1	4.204842E-1
141	2.698415E-1	2.945624E-1	3.623418E-1	4.064155E-1
142	2.788973E-1	3.598310E-1	3.820258E-1	4.105775E-1
143	2.607608E-1	3.310885E-1	3.888263E-1	4.054866E-1
144	3.433723E-1	3.826470E-1	4.147166E-1	4.315929E-1
145	2.479981E-1	2.733932E-1	3.311604E-1	4.189432E-1
146	3.035796E-1	3.252025E-1	3.709844E-1	4.144205E-1
147	2.768969E-1	3.004995E-1	3.541782E-1	4.288070E-1
148	3.236556E-1	3.598170E-1	3.895254E-1	4.092887E-1
149	2.389278E-1	3.099192E-1	3.539156E-1	4.166343E-1
150	2.811717E-1	3.075203E-1	4.162649E-1	4.385238E-1
151	2.888587E-1	3.098108E-1	3.678452E-1	4.360356E-1
152	3.384235E-1	3.706344E-1	4.154500E-1	4.315345E-1
153	2.412604E-1	2.736179E-1	3.895546E-1	4.125395E-1
154	2.980467E-1	3.401221E-1	3.861837E-1	4.138264E-1
155	2.824364E-1	3.315975E-1	3.579414E-1	4.121152E-1
156	3.038202E-1	3.705886E-1	4.057750E-1	4.315171E-1
157	2.390777E-1	3.116385E-1	4.139358E-1	4.353041E-1
158	2.671168E-1	3.419379E-1	4.174094E-1	4.391848E-1
159	2.679468E-1	3.333439E-1	3.864814E-1	4.374625E-1
160	3.405110E-1	3.908780E-1	4.354851E-1	4.491019E-1
161	2.100699E-1	2.325245E-1	3.617814E-1	4.313579E-1
162	2.945099E-1	3.337098E-1	3.822786E-1	3.986389E-1
163	2.805252E-1	3.259052E-1	3.506470E-1	3.928739E-1

164	3.199996E-1	3.436747E-1	3.910705E-1	4.375011E-1
165	2.205810E-1	3.031519E-1	3.817655E-1	4.044882E-1
166	2.861227E-1	3.297465E-1	3.881028E-1	4.242477E-1
167	2.698071E-1	3.253323E-1	3.791545E-1	4.151382E-1
168	3.348589E-1	3.692584E-1	3.947431E-1	4.119222E-1
169	2.071098E-1	2.727795E-1	3.785664E-1	4.345800E-1
170	3.064662E-1	3.466957E-1	3.871383E-1	4.035583E-1
171	2.701486E-1	3.466545E-1	3.776967E-1	3.964345E-1
172	3.187459E-1	3.402257E-1	4.149916E-1	4.415788E-1
173	2.585928E-1	3.143701E-1	3.650838E-1	4.216152E-1
174	2.827130E-1	3.541371E-1	4.067460E-1	4.292679E-1
175	2.520218E-1	3.591051E-1	3.951029E-1	4.181484E-1
176	3.549062E-1	3.749529E-1	4.189660E-1	4.361444E-1
177	2.648411E-1	2.929418E-1	3.277515E-1	4.087905E-1
178	3.077743E-1	3.355862E-1	3.622096E-1	4.253942E-1
179	2.884663E-1	3.160757E-1	3.609896E-1	4.195514E-1
180	3.171284E-1	3.557722E-1	4.058088E-1	4.239730E-1
181	2.470897E-1	3.381846E-1	3.718596E-1	3.959715E-1
182	3.079817E-1	3.326918E-1	4.005342E-1	4.382737E-1
183	2.794848E-1	3.161835E-1	3.972377E-1	4.347466E-1
184	3.444905E-1	3.661532E-1	4.109594E-1	4.417271E-1
185	2.357418E-1	2.945873E-1	3.980725E-1	4.168334E-1
186	3.140385E-1	3.522720E-1	3.791389E-1	4.109691E-1
187	2.830025E-1	3.381363E-1	3.886419E-1	4.061933E-1
188	3.236253E-1	3.502434E-1	4.280896E-1	4.466304E-1
189	2.612521E-1	3.249710E-1	4.002145E-1	4.253218E-1
190	3.052845E-1	3.421642E-1	4.244751E-1	4.438310E-1
191	2.873748E-1	3.325006E-1	3.943083E-1	4.425385E-1
192	3.740754E-1	4.020264E-1	4.309335E-1	4.441600E-1
193	2.345040E-1	2.562186E-1	3.412388E-1	4.230453E-1
194	3.054926E-1	3.291570E-1	3.527098E-1	3.924391E-1
195	2.813236E-1	3.032923E-1	3.489254E-1	3.931639E-1
196	3.218935E-1	3.504199E-1	3.973175E-1	4.145603E-1
197	2.396846E-1	2.924515E-1	3.789374E-1	3.965355E-1
198	3.073072E-1	3.291279E-1	3.984556E-1	4.161433E-1
199	2.852746E-1	3.087745E-1	3.929165E-1	4.144377E-1
200	3.444464E-1	3.622019E-1	3.976198E-1	4.177436E-1
201	2.320831E-1	2.678080E-1	3.780757E-1	4.345609E-1
202	3.047387E-1	3.518653E-1	3.759732E-1	3.952937E-1
203	2.619909E-1	3.462073E-1	3.712969E-1	4.124389E-1
204	3.110809E-1	3.510409E-1	4.160828E-1	4.343401E-1
205	2.749804E-1	2.966315E-1	3.875205E-1	4.092438E-1
206	2.909391E-1	3.544556E-1	3.934270E-1	4.082203E-1
207	2.718719E-1	3.455108E-1	3.871253E-1	4.225906E-1
208	3.632459E-1	3.819322E-1	4.041149E-1	4.183707E-1
209	2.457707E-1	2.729093E-1	3.483179E-1	4.251618E-1
210	3.141390E-1	3.378723E-1	3.651952E-1	4.044234E-1
211	2.940758E-1	3.169355E-1	3.430472E-1	4.061304E-1
212	3.146275E-1	3.724134E-1	4.006607E-1	4.179308E-1

213	2.340142E-1	3.140072E-1	3.830035E-1	4.348292E-1
214	2.936357E-1	3.205300E-1	4.108374E-1	4.363931E-1
215	2.895058E-1	3.118289E-1	3.863115E-1	4.387713E-1
216	3.263174E-1	3.808582E-1	4.197214E-1	4.387955E-1
217	2.508095E-1	2.830181E-1	3.822474E-1	4.342444E-1
218	3.189941E-1	3.448551E-1	3.726901E-1	4.230670E-1
219	2.883801E-1	3.366222E-1	3.697423E-1	4.250576E-1
220	3.061077E-1	3.818569E-1	4.182062E-1	4.328684E-1
221	2.338983E-1	3.448618E-1	4.121766E-1	4.292162E-1
222	2.859809E-1	3.429038E-1	4.251129E-1	4.442997E-1
223	2.798588E-1	3.387893E-1	3.920854E-1	4.405410E-1
224	3.645093E-1	3.822027E-1	4.298306E-1	4.458184E-1
225	2.343923E-1	2.573774E-1	3.595671E-1	4.300886E-1
226	3.050319E-1	3.275894E-1	3.783056E-1	4.010261E-1
227	2.775226E-1	3.181303E-1	3.677943E-1	4.015430E-1
228	3.330358E-1	3.558210E-1	3.875489E-1	4.246287E-1
229	2.450210E-1	3.125607E-1	3.911476E-1	4.087628E-1
230	2.970591E-1	3.402469E-1	3.929193E-1	4.288997E-1
231	2.778393E-1	3.250198E-1	3.974364E-1	4.159209E-1
232	3.494653E-1	3.703625E-1	3.954825E-1	4.319234E-1
233	2.314856E-1	2.910234E-1	3.779095E-1	4.322597E-1
234	3.192835E-1	3.536711E-1	3.809829E-1	3.978434E-1
235	2.896892E-1	3.502657E-1	3.807297E-1	3.979694E-1
236	3.289873E-1	3.520054E-1	4.125572E-1	4.375979E-1
237	2.762733E-1	3.022672E-1	3.817234E-1	4.349891E-1
238	2.796273E-1	3.737273E-1	4.123746E-1	4.306263E-1
239	2.534428E-1	3.659400E-1	4.149370E-1	4.327436E-1
240	3.761072E-1	3.951420E-1	4.167877E-1	4.330236E-1
241	2.628158E-1	2.882705E-1	3.473972E-1	4.241826E-1
242	3.019313E-1	3.436526E-1	3.770313E-1	4.342045E-1
243	2.978343E-1	3.234954E-1	3.644924E-1	4.335508E-1
244	3.317745E-1	3.643249E-1	3.982436E-1	4.350783E-1
245	2.490497E-1	3.278708E-1	3.835870E-1	4.355581E-1
246	3.046534E-1	3.276712E-1	4.184847E-1	4.413788E-1
247	2.969609E-1	3.238989E-1	3.904637E-1	4.399160E-1
248	3.439238E-1	3.671005E-1	4.295232E-1	4.452150E-1
249	2.593997E-1	2.916027E-1	4.043725E-1	4.314132E-1
250	2.975375E-1	3.575738E-1	3.889918E-1	4.300070E-1
251	2.840689E-1	3.495746E-1	3.810428E-1	4.297128E-1
252	3.257163E-1	3.748759E-1	4.319593E-1	4.472908E-1
253	2.653030E-1	3.147460E-1	4.167035E-1	4.372947E-1
254	3.003986E-1	3.541473E-1	4.285381E-1	4.603364E-1
255	2.980772E-1	3.493049E-1	4.004293E-1	4.482135E-1
256	3.755762E-1	4.166573E-1	4.421368E-1	4.527286E-1

Table 8-8. LSP Quantization Table, Rate 1/8, Codebook 1

j	$q_{rate}(1, 1, j)$	$q_{rate}(1, 2, j)$	$q_{rate}(1, 3, j)$	$q_{rate}(1, 4, j)$	$q_{rate}(1, 5, j)$
1	4.209106E-2	6.947497E-2	1.116895E-1	1.457197E-1	2.089358E-1

2	5.494466E-2	9.824226E-2	1.100788E-1	1.589078E-1	2.054824E-1
3	4.518857E-2	7.519943E-2	1.142339E-1	1.546973E-1	1.974671E-1
4	4.947500E-2	7.966750E-2	1.257135E-1	1.694478E-1	2.077532E-1
5	4.178938E-2	6.345956E-2	1.206803E-1	1.585077E-1	2.040682E-1
6	4.715924E-2	7.912955E-2	1.218311E-1	1.565005E-1	2.230923E-1
7	5.453992E-2	8.034305E-2	1.294776E-1	1.518615E-1	2.017172E-1
8	5.585208E-2	9.411485E-2	1.401603E-1	1.780708E-1	2.295549E-1
9	4.544353E-2	7.354141E-2	1.193766E-1	1.544203E-1	2.101075E-1
10	6.317801E-2	9.523149E-2	1.236498E-1	1.767254E-1	2.174373E-1
11	5.276537E-2	8.435144E-2	1.158909E-1	1.579092E-1	2.073235E-1
12	5.186575E-2	8.132854E-2	1.375623E-1	1.832288E-1	2.164007E-1
13	4.441953E-2	6.887446E-2	1.311525E-1	1.626358E-1	2.165910E-1
14	4.937844E-2	8.188255E-2	1.306717E-1	1.682190E-1	2.313608E-1
15	5.590978E-2	9.078330E-2	1.334885E-1	1.629847E-1	2.096152E-1
16	6.137821E-2	9.860277E-2	1.479333E-1	1.928319E-1	2.315651E-1

Table 8-9. LSP Quantization Table, Rate 1/8, Codebook 2

j	$q_{rate}(2, 1, j)$	$q_{rate}(2, 2, j)$	$q_{rate}(2, 3, j)$	$q_{rate}(2, 4, j)$	$q_{rate}(2, 5, j)$
1	2.682296E-1	3.058530E-1	3.111035E-1	3.682334E-1	4.077447E-1
2	2.441801E-1	2.897017E-1	3.257376E-1	3.902148E-1	4.134584E-1
3	2.334183E-1	3.007829E-1	3.289390E-1	3.855733E-1	4.106846E-1
4	2.590587E-1	2.975686E-1	3.419662E-1	3.853117E-1	4.129523E-1
5	2.429045E-1	2.922362E-1	3.271855E-1	3.778814E-1	4.033293E-1
6	2.467419E-1	2.974937E-1	3.363123E-1	3.942606E-1	4.225895E-1
7	2.137760E-1	3.314042E-1	3.406769E-1	3.822208E-1	4.093902E-1
8	2.667348E-1	3.079165E-1	3.441972E-1	3.961151E-1	4.238752E-1
9	2.612143E-1	3.049254E-1	3.299724E-1	3.848680E-1	4.202374E-1
10	2.495487E-1	2.937286E-1	3.338274E-1	3.785067E-1	4.171406E-1
11	2.415889E-1	3.017342E-1	3.412825E-1	3.842858E-1	4.161965E-1
12	2.581891E-1	3.173641E-1	3.490434E-1	3.876993E-1	4.155136E-1
13	2.445059E-1	3.067345E-1	3.357932E-1	3.784443E-1	4.055705E-1
14	2.516403E-1	3.122508E-1	3.384779E-1	3.955419E-1	4.239680E-1
15	2.278799E-1	3.177920E-1	3.383191E-1	4.004411E-1	4.118556E-1
16	2.789686E-1	3.226197E-1	3.565811E-1	4.020676E-1	4.237045E-1

Table 8-10a. Interpolation Filter Coefficients, Cutoff=0.5 (1/2)

n	$I_{\varepsilon}7n)$	$I_{\varepsilon}(1 + 7n)$	$I_{\varepsilon}(2 + 7n)$	$I_{\varepsilon}(3 + 7n)$
0	-2.279553E-2	9.638780E-2	4.296515E-1	4.296515E-1
1	-2.165019E-2	6.437141E-2	3.936069E-1	4.594094E-1
2	-1.928710E-2	3.758768E-2	3.527558E-1	4.816282E-1
3	-1.634340E-2	1.617878E-2	3.087218E-1	4.953565E-1
4	-1.332172E-2	0.000000E+0	2.631803E-1	5.000000E-1
5	-1.057099E-2	-1.134067E-2	2.177682E-1	4.953565E-1
6	-8.282481E-3	-1.842952E-2	1.739981E-1	4.816282E-1
7	-6.500361E-3	-2.198557E-2	1.331849E-1	4.594094E-1

Table 8-10b. Interpolation Filter Coefficients, Cutoff=0.5 (2/2)

n	$I_{\varepsilon}(4 + 7n)$	$I_{\varepsilon}(5 + 7n)$	$I_{\varepsilon}(6 + 7n)$
0	9.638780E-2	-2.279553E-2	-5.144665E-3
1	1.331849E-1	-2.198557E-2	-6.500361E-3
2	1.739981E-1	-1.842952E-2	-8.282481E-3
3	2.177682E-1	-1.134067E-2	-1.057099E-2
4	2.631803E-1	0.000000E+0	-1.332172E-2
5	3.087218E-1	1.617878E-2	-1.634340E-2
6	3.527558E-1	3.758768E-2	-1.928710E-2
7	3.936069E-1	6.437141E-2	-2.165019E-2

Table 8-11a. Interpolation Filter Coefficients, Cutoff=0.9 (1/3)

n	$I_E(17n)$	$I_E(1+17n)$	$I_E(2+17n)$	$I_E(3+17n)$	$I_E(4+17n)$
0	3.333057E-3	-4.447694E-3	3.032594E-3	5.505681E-3	-2.749339E-2
1	1.814939E-3	-1.051010E-3	-3.490370E-3	1.602235E-2	-4.161043E-2
2	3.143855E-4	1.903512E-3	-8.614938E-3	2.320955E-2	-4.897606E-2
3	-1.011791E-3	4.128520E-3	-1.191523E-2	2.662804E-2	-4.954024E-2
4	-2.054155E-3	5.477784E-3	-1.325900E-2	2.636338E-2	-4.408088E-2
5	-2.749624E-3	5.939437E-3	-1.278129E-2	2.294780E-2	-3.401131E-2
6	-3.074976E-3	5.612335E-3	-1.082686E-2	1.723493E-2	-2.113197E-2
7	-3.038022E-3	4.671966E-3	-7.873206E-3	1.024916E-2	-7.363276E-3

Table 8-11b. Interpolation Filter Coefficients, Cutoff=0.9 (2/3)

n	$I_E(5+17n)$	$I_E(6+17n)$	$I_E(7+17n)$	$I_E(8+17n)$	$I_E(9+17n)$	$I_E(10+17n)$
0	7.357492E-2	-1.760500E-1	6.238571E-1	6.238571E-1	-1.760500E-1	7.357492E-2
1	8.809371E-2	-1.788657E-1	4.934066E-1	7.373307E-1	-1.480059E-1	4.678921E-2
2	9.025505E-2	-1.604327E-1	3.555172E-1	8.252803E-1	-9.275004E-2	9.433596E-3
3	8.139432E-2	-1.261773E-1	2.198452E-1	8.809472E-1	-1.067283E-2	-3.518170E-2
4	6.395375E-2	-8.231784E-2	9.530775E-2	9.000000E-1	9.530775E-2	-8.231784E-2
5	4.107321E-2	-3.518170E-2	-1.067283E-2	8.809472E-1	2.198452E-1	-1.261773E-1
6	1.614589E-2	9.433596E-3	-9.275004E-2	8.252803E-1	3.555172E-1	-1.604327E-1
7	-7.606618E-3	4.678921E-2	-1.480059E-1	7.373307E-1	4.934066E-1	-1.788657E-1

Table 8-11c. Interpolation Filter Coefficients, Cutoff=0.9 (3/3)

n	$I_E(11+17n)$	$I_E(12+17n)$	$I_E(13+17n)$	$I_E(14+17n)$	$I_E(15+17n)$	$I_E(16+17n)$
0	-2.749339E-2	5.505681E-3	3.032594E-3	-4.447694E-3	3.333057E-3	-2.669328E-3
1	-7.606618E-3	-7.363276E-3	1.024916E-2	-7.873206E-3	4.671966E-3	-3.038022E-3
2	1.614589E-2	-2.113197E-2	1.723493E-2	-1.082686E-2	5.612335E-3	-3.074976E-3
3	4.107321E-2	-3.401131E-2	2.294780E-2	-1.278129E-2	5.939437E-3	-2.749624E-3
4	6.395375E-2	-4.408088E-2	2.636338E-2	-1.325900E-2	5.477784E-3	-2.054155E-3
5	8.139432E-2	-4.954024E-2	2.662804E-2	-1.191523E-2	4.128520E-3	-1.011791E-3
6	9.025505E-2	-4.897606E-2	2.320955E-2	-8.614938E-3	1.903512E-3	3.143855E-4
7	8.809371E-2	-4.161043E-2	1.602235E-2	-3.490370E-3	-1.051010E-3	1.814939E-3

Table 8-12. Fixed Codebook Gain Quantization, Rate 1

k	$g_{ccb}(k)$	k	$g_{ccb}(k)$
0	1.2840254E+0	16	7.0105412E+1
1	1.6487213E+0	17	9.0017131E+1
2	2.1170000E+0	18	1.1558429E+2
3	2.7182818E+0	19	1.4841316E+2
4	3.4903430E+0	20	1.9056627E+2
5	4.4816891E+0	21	2.4469193E+2
6	5.7546027E+0	22	3.1419066E+2
7	7.3890561E+0	23	4.0342879E+2
8	9.4877358E+0	24	5.1801283E+2
9	1.2182494E+1	25	6.6514163E+2
10	1.5642632E+1	26	8.5405876E+2
11	2.0085537E+1	27	1.0966332E+3
12	2.5790340E+1	28	1.4081049E+3

13	3.3115452E+1	29	1.8080424E+3
14	4.2521082E+1	30	2.3215724E+3
15	5.4598150E+1	31	2.9809580E+3

Table 8-13. Fixed Codebook Gain Quantization, Rate 1/2

k	$G_{ccb}(k)$	k	$G_{ccb}(k)$
0	1.6487213E+0	8	9.0017131E+1
1	2.7182818E+0	9	1.4841316E+2
2	4.4816891E+0	10	2.4469193E+2
3	7.3890561E+0	11	4.0342879E+2
4	1.2182494E+1	12	6.6514163E+2
5	2.0085537E+1	13	1.0966332E+3
6	3.3115452E+1	14	1.8080424E+3
7	5.4598150E+1	15	2.9809580E+3

Table 8-14. Residual Shift Interpolation Filter Coefficients

j	$I_f(-1, j)$	$I_f(0, j)$	$I_f(1, j)$
0	3.750000E-01	7.500000E-01	-1.250000E-01
1	2.578125E-01	8.593750E-01	-1.171875E-01
2	1.562500E-01	9.375000E-01	-9.375000E-02
3	7.031250E-02	9.843750E-01	-5.468750E-02
4	0.000000E+00	1.000000E+00	0.000000E+00
5	-5.468750E-02	9.843750E-01	7.031250E-02
6	-9.375000E-02	9.375000E-01	1.562500E-01
7	-1.171875E-01	8.593750E-01	2.578125E-01

Table 8-15. Rate 1/8 Frame Energy Quantization

k	$q_{log}(0, k)$	$q_{log}(1, k)$	$q_{log}(2, k)$	k	$q_{log}(0, k)$	$q_{log}(1, k)$	$q_{log}(2, k)$
0	-2.464E-02	-4.005E-03	-1.107E-01	128	-4.208E-02	-1.491E-01	-7.639E-02
1	8.734E-01	1.004E+00	9.930E-01	129	1.046E+00	9.598E-01	9.176E-01
2	4.222E-01	3.894E-01	5.020E-01	130	4.478E-01	4.605E-01	5.111E-01
3	1.450E+00	1.328E+00	1.278E+00	131	1.521E+00	1.292E+00	1.342E+00
4	1.957E-01	2.169E-01	2.735E-01	132	2.220E-01	2.549E-01	2.510E-01
5	1.142E+00	1.240E+00	1.157E+00	133	1.186E+00	1.254E+00	1.171E+00
6	7.881E-01	6.778E-01	4.185E-01	134	8.999E-01	4.960E-01	4.943E-01
7	1.504E+00	1.468E+00	1.534E+00	135	1.423E+00	1.484E+00	1.620E+00
8	3.173E-01	2.693E-01	-9.526E-02	136	2.796E-01	2.778E-01	-2.820E-01
9	1.141E+00	1.154E+00	1.044E+00	137	1.170E+00	1.181E+00	1.076E+00
10	5.147E-01	5.784E-01	8.802E-01	138	4.068E-01	8.541E-01	9.352E-01
11	1.502E+00	1.407E+00	1.409E+00	139	1.584E+00	1.416E+00	1.387E+00
12	3.163E-01	3.592E-01	2.830E-01	140	3.325E-01	3.655E-01	3.340E-01
13	1.217E+00	1.213E+00	1.216E+00	141	1.224E+00	1.257E+00	1.245E+00
14	1.023E+00	1.139E+00	-9.526E-02	142	1.061E+00	1.138E+00	-9.526E-02
15	1.619E+00	1.655E+00	1.642E+00	143	1.681E+00	1.704E+00	1.673E+00
16	1.437E-01	1.505E-01	6.838E-02	144	1.932E-01	1.489E-01	1.258E-01
17	9.794E-01	1.021E+00	1.117E+00	145	1.023E+00	1.088E+00	1.145E+00
18	4.701E-01	6.426E-01	5.519E-01	146	5.190E-01	6.873E-01	5.172E-01
19	1.366E+00	1.397E+00	1.406E+00	147	1.380E+00	1.405E+00	1.474E+00
20	2.918E-01	3.022E-01	2.420E-01	148	3.393E-01	3.100E-01	2.231E-01

21	1.309E+00	1.241E+00	1.220E+00	149	1.354E+00	1.249E+00	1.270E+00
22	7.989E-01	7.654E-01	7.391E-01	150	7.363E-01	8.508E-01	8.247E-01
23	1.612E+00	1.502E+00	1.447E+00	151	1.612E+00	1.537E+00	1.509E+00
24	2.594E-01	1.948E-01	2.555E-01	152	2.952E-01	2.053E-01	2.590E-01
25	1.091E+00	1.150E+00	1.272E+00	153	1.138E+00	1.219E+00	1.262E+00
26	3.423E-01	4.150E-01	1.294E+00	154	1.345E+00	1.289E+00	1.338E+00
27	1.729E+00	1.377E+00	1.065E+00	155	1.437E+00	1.360E+00	1.442E+00
28	4.103E-01	3.287E-01	3.228E-01	156	4.826E-01	3.298E-01	3.842E-01
29	1.144E+00	1.281E+00	1.416E+00	157	1.219E+00	1.311E+00	1.413E+00
30	1.047E+00	1.117E+00	6.188E-01	158	1.212E+00	1.186E+00	6.357E-01
31	1.914E+00	1.777E+00	1.516E+00	159	1.873E+00	1.939E+00	1.674E+00
32	-2.117E-02	2.159E-01	2.351E-01	160	1.260E+00	1.306E+00	1.368E+00
33	1.093E+00	1.088E+00	1.026E+00	161	1.146E+00	1.077E+00	1.025E+00
34	5.567E-01	5.092E-01	4.654E-01	162	6.029E-01	5.039E-01	5.781E-01
35	1.510E+00	1.449E+00	1.201E+00	163	1.514E+00	1.420E+00	1.324E+00
36	2.362E-01	3.426E-01	2.549E-01	164	2.652E-01	3.192E-01	3.042E-01
37	1.340E+00	1.225E+00	1.117E+00	165	1.368E+00	1.198E+00	1.200E+00
38	1.203E+00	3.819E-01	2.269E-01	166	1.234E+00	4.910E-01	3.464E-02
39	1.373E+00	1.404E+00	1.830E+00	167	1.347E+00	1.560E+00	1.861E+00
40	2.570E-01	2.668E-01	1.636E-01	168	2.766E-01	2.887E-01	2.029E-01
41	1.219E+00	1.098E+00	1.122E+00	169	1.257E+00	1.105E+00	1.145E+00
42	6.985E-01	8.456E-01	1.069E+00	170	1.351E+00	1.353E+00	1.406E+00
43	1.550E+00	1.501E+00	1.388E+00	171	1.506E+00	1.580E+00	1.362E+00
44	2.870E-01	3.060E-01	3.599E-01	172	2.794E-01	3.868E-01	4.277E-01
45	1.178E+00	1.345E+00	1.302E+00	173	1.234E+00	1.334E+00	1.336E+00
46	1.270E+00	1.215E+00	1.812E-01	174	1.280E+00	1.252E+00	1.805E-01
47	1.725E+00	1.777E+00	1.693E+00	175	1.387E+00	1.396E+00	1.434E+00
48	2.074E-01	2.104E-01	1.539E-01	176	2.902E-01	1.170E-01	1.698E-01
49	1.105E+00	1.034E+00	1.104E+00	177	1.134E+00	1.077E+00	1.117E+00
50	6.683E-01	6.646E-01	6.639E-01	178	6.986E-01	7.177E-01	7.366E-01
51	1.403E+00	1.462E+00	1.435E+00	179	1.370E+00	1.491E+00	1.495E+00
52	3.389E-01	3.754E-01	2.150E-01	180	4.031E-01	5.144E-01	1.751E-01
53	1.288E+00	1.325E+00	1.257E+00	181	1.333E+00	1.377E+00	1.257E+00
54	8.933E-01	8.253E-01	8.133E-01	182	9.212E-01	8.934E-01	8.897E-01
55	1.555E+00	1.579E+00	1.565E+00	183	1.589E+00	1.614E+00	1.523E+00
56	3.264E-01	2.434E-01	2.852E-01	184	3.152E-01	2.164E-01	3.230E-01
57	1.242E+00	1.180E+00	1.202E+00	185	1.300E+00	1.145E+00	1.212E+00
58	1.314E-01	1.698E-01	1.646E+00	186	1.269E+00	1.245E+00	1.497E+00
59	1.797E+00	1.597E+00	1.241E+00	187	1.763E+00	1.716E+00	1.311E+00
60	4.721E-01	5.346E-01	3.066E-01	188	4.702E-01	5.422E-01	4.306E-01
61	1.274E+00	1.401E+00	1.351E+00	189	1.342E+00	1.433E+00	1.423E+00
62	1.455E+00	1.386E+00	6.430E-01	190	1.472E+00	1.404E+00	8.371E-01
63	1.828E+00	1.867E+00	1.825E+00	191	1.936E+00	1.883E+00	1.838E+00
64	-3.265E-01	-2.956E-01	-2.462E-01	192	1.266E+00	1.295E+00	1.302E+00
65	1.035E+00	1.020E+00	1.003E+00	193	1.074E+00	1.002E+00	1.023E+00
66	3.702E-01	4.307E-01	7.072E-01	194	5.206E-01	4.045E-01	6.549E-01
67	1.424E+00	1.345E+00	1.352E+00	195	1.457E+00	1.378E+00	1.363E+00
68	2.267E-01	2.680E-01	3.037E-01	196	2.715E-01	2.629E-01	2.841E-01
69	1.235E+00	1.249E+00	1.146E+00	197	1.264E+00	1.271E+00	1.175E+00

70	9.944E-01	6.485E-01	5.248E-01	198	1.337E+00	1.305E+00	1.306E+00
71	1.539E+00	1.492E+00	1.612E+00	199	1.555E+00	1.571E+00	1.657E+00
72	3.815E-01	3.360E-01	-9.526E-02	200	3.341E-01	4.147E-01	-3.648E-01
73	1.163E+00	1.144E+00	1.117E+00	201	1.188E+00	1.185E+00	1.161E+00
74	6.734E-01	7.656E-01	1.014E+00	202	6.198E-01	7.208E-01	1.157E+00
75	1.568E+00	1.438E+00	1.455E+00	203	1.582E+00	1.465E+00	1.513E+00
76	3.409E-01	3.317E-01	3.856E-01	204	3.839E-01	3.651E-01	3.814E-01
77	1.180E+00	1.284E+00	1.284E+00	205	1.214E+00	1.256E+00	1.292E+00
78	1.244E+00	1.214E+00	-9.526E-02	206	1.361E+00	1.363E+00	1.312E+00
79	1.753E+00	1.598E+00	1.744E+00	207	1.793E+00	1.693E+00	1.669E+00
80	1.548E-01	1.388E-01	2.020E-01	208	1.889E-01	1.275E-01	2.534E-01
81	1.027E+00	1.133E+00	1.093E+00	209	1.066E+00	1.174E+00	1.133E+00
82	3.906E-01	7.505E-01	5.705E-01	210	4.999E-01	8.207E-01	5.813E-01
83	1.420E+00	1.357E+00	1.543E+00	211	1.478E+00	1.416E+00	1.497E+00
84	3.252E-01	3.136E-01	2.804E-01	212	3.814E-01	3.138E-01	2.889E-01
85	1.351E+00	1.309E+00	1.224E+00	213	1.396E+00	1.265E+00	1.233E+00
86	8.781E-01	8.095E-01	7.109E-01	214	9.458E-01	9.161E-01	5.875E-01
87	1.614E+00	1.580E+00	1.433E+00	215	1.672E+00	1.632E+00	1.553E+00
88	3.222E-01	2.298E-01	2.157E-01	216	3.505E-01	2.525E-01	2.364E-01
89	1.216E+00	1.077E+00	1.247E+00	217	1.211E+00	1.138E+00	1.235E+00
90	1.363E+00	1.280E+00	1.317E+00	218	1.391E+00	1.231E+00	1.355E+00
91	1.751E+00	1.457E+00	1.182E+00	219	1.783E+00	1.510E+00	1.199E+00
92	4.428E-01	4.082E-01	3.181E-01	220	4.227E-01	4.548E-01	3.671E-01
93	1.157E+00	1.227E+00	1.604E+00	221	1.281E+00	1.254E+00	1.661E+00
94	1.286E+00	1.268E+00	8.167E-01	222	1.338E+00	1.379E+00	9.531E-01
95	1.994E+00	2.018E+00	1.307E+00	223	2.148E+00	1.965E+00	1.584E+00
96	2.671E-02	2.594E-01	3.397E-01	224	9.324E-02	3.575E-01	3.522E-01
97	1.164E+00	1.080E+00	9.321E-01	225	1.212E+00	1.086E+00	1.044E+00
98	5.998E-01	6.076E-01	5.081E-01	226	6.128E-01	6.136E-01	6.060E-01
99	1.442E+00	1.442E+00	1.375E+00	227	1.484E+00	1.507E+00	1.396E+00
100	2.390E-01	3.554E-01	3.426E-01	228	2.820E-01	3.848E-01	3.156E-01
101	1.287E+00	1.307E+00	1.144E+00	229	1.368E+00	1.287E+00	1.128E+00
102	1.200E+00	7.495E-01	3.967E-01	230	1.369E+00	1.352E+00	1.358E+00
103	1.561E+00	1.517E+00	1.898E+00	231	1.381E+00	1.765E+00	2.113E+00
104	3.598E-01	3.463E-01	1.200E-01	232	1.314E+00	1.345E+00	1.334E+00
105	1.298E+00	1.125E+00	1.062E+00	233	1.290E+00	1.172E+00	1.119E+00
106	7.577E-01	1.013E+00	1.194E+00	234	1.304E+00	1.377E+00	1.427E+00
107	1.537E+00	1.513E+00	1.464E+00	235	1.490E+00	1.540E+00	1.536E+00
108	4.041E-01	4.038E-01	3.897E-01	236	3.994E-01	4.402E-01	4.173E-01
109	1.293E+00	1.219E+00	1.378E+00	237	1.323E+00	1.307E+00	1.392E+00
110	1.250E+00	1.391E+00	2.451E-01	238	1.400E+00	1.388E+00	1.369E+00
111	1.558E+00	1.764E+00	1.728E+00	239	1.669E+00	1.818E+00	1.834E+00
112	2.700E-01	1.894E-01	1.924E-01	240	2.742E-01	2.235E-01	1.986E-01
113	1.111E+00	1.112E+00	1.173E+00	241	1.137E+00	1.139E+00	1.201E+00
114	7.579E-01	8.342E-01	4.781E-01	242	1.324E+00	1.385E+00	1.349E+00
115	1.464E+00	1.477E+00	1.469E+00	243	1.455E+00	1.574E+00	1.454E+00
116	4.001E-01	3.104E-01	2.217E-01	244	5.019E-01	3.255E-01	2.555E-01
117	1.346E+00	1.421E+00	1.312E+00	245	1.388E+00	1.438E+00	1.300E+00

118	1.071E+00	8.967E-01	7.511E-01	246	1.394E+00	1.349E+00	1.411E+00
119	1.616E+00	1.551E+00	1.574E+00	247	1.639E+00	1.580E+00	1.681E+00
120	3.329E-01	2.785E-01	3.140E-01	248	3.920E-01	2.498E-01	3.523E-01
121	1.281E+00	1.209E+00	1.239E+00	249	1.301E+00	1.221E+00	1.285E+00
122	2.805E-01	2.687E-01	1.646E+00	250	1.318E+00	1.342E+00	1.494E+00
123	1.814E+00	1.514E+00	1.510E+00	251	1.910E+00	1.680E+00	1.470E+00
124	6.231E-01	4.200E-01	3.701E-01	252	6.082E-01	5.270E-01	4.173E-01
125	1.255E+00	1.429E+00	1.454E+00	253	1.255E+00	1.477E+00	1.503E+00
126	1.642E+00	1.581E+00	7.112E-01	254	1.807E+00	1.742E+00	6.553E-01
127	1.844E+00	1.963E+00	1.895E+00	255	2.000E+00	2.072E+00	2.051E+00

9 APPENDIX C. INFORMATIVE REFERENCES

—Books:

1. Rabiner, L. R. and Schafer, R. W., *Digital Processing of Speech Signals,* New Jersey, Prentice-Hall Inc., 1978.
2. Crochiere, R. E. and Rabiner L. R., *Multirate Digital Signal Processing*, New Jersey, Prentice-Hall Inc., 1983.
3. Oppenheim, A. V. and Schafer, R. W., *Digital Signal Processing*, New Jersey, Prentice-Hall Inc., 1975.
4. Proakis, J. G. and Manolakis, D. G., *Introduction to Digital Signal Processing*, New York, Macmillan, 1988.
5. Dellar, J. R., Proakis, J. G., Hansen, J. H. L., *Discrete-Time Processing of Speech Signals*, New York, Macmillan, 1993.
6. Alejandro, A., *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Boston, Kluwer Academic Publishers, 1993.

—Other Technical References:

7. Kleijn, W. B., Kroon, P., and Nahumi, D., “The RCELP Speech-Coding Algorithm”, European Transactions on Telecommunications, Vol 5, Number 5. Sept/Oct 1994, pp 573-582.
8. Nahumi, D and Kleijn, W. B., “An Improved 8 kb/s RCELP Coder”, IEEE Workshop on Speech Coding, 1995.
9. Atal, B. S. and Schroeder, M. R., “Stochastic coding of speech at very low bit rates”, Proc Int. Conf. Comm., Amsterdam, 1984, pp 1610-1613.
10. Laflamme, C., Adoul, J-P., Salami, R., Morissette, S., and Mabilleau, P., “16 kbps wideband speech coding technique based on algebraic CELP”, Proc. ICASSP'91, pp. 13-16.
11. Salami, R., Laflamme, C., Adoul, J-P., et.al, “A Toll Quality 8 kbits Speech coder for Personal Communication Systems (PCS)”, IEEE Trans. on Vehicle Technology, 1994.

10 APPENDIX D. CHANGE HISTORY FOR ANSI-127 EVRC

Introduction

The following is catalog of changes that have been adopted for the EVRC ANSI standard.

The fixes presented here have been previously presented in [1,2].

Bug Fixes and Sanity Checks

Bottom Guard for Offset

- Text Changes: Modify Sections 4.5.6.2.1-3
- Floating Point C Sim: cshift.c
- Fixed Point C Sim: cshift.c
- Test Vector: ansi_offset.pcm

The offset computed in cshiftframe() is used in maxeloc() to index elements in the array residual[]. Under certain circumstances, offset value causes maxeloc() to access elements below residual[0]. A fix constrains the offset so that maxeloc() does not attempt to access residual[] below &residual[0].

All-Zeros Packet

- Text: Create Section 1.4.3
- Floating Point C Sim: fer.c
- Fixed Point C Sim: fer.c
- Test Vector: ansi_zeros.pkt

Detect of all-zeros full rate and half rate packets. If such a packet is detected, the frame is erased. This avoids clicks and pops in the decoded speech.

All-Ones Packet

- Text: Modify Section 1.4.2
- Floating Point C Sim: d_globs.c, d_globs.h, decode.c, fer.c
- Fixed Point C Sim: d_globs.c, d_rate_1.c, fer.c
- Test Vector: ansi_ones.pkt

Mute after more than 2 all-ones Rate 1/8 packets are received. This fixed an end-of-call buzz problem.

Gaussian Noise Fix

- Text: No change
- Floating Point C Sim: decode.c
- Fixed Point C Sim: d_fer.c
- Test Vector: ansi_fer.pkt

This fix has the seed value of the erasure noise random number generator initialized only once at the beginning of a set of consecutive erasure frames. Otherwise, the seed is reloaded for each sub frame generating a "noise envelope" that repeats with the period of a subframe length. It sounds vaguely like the noise from a bug's wings, hence, it has been referred to as the "wing" sound.

Delta Delay Check

- Text: New Section 5.1.3
- Floating Point C Sim: decode.c
- Fixed Point C Sim: d_no_fer.c
- Test Vector: ansi_ddelay.pkt

This change increases the integrity checks on the incoming frame to make sure the DDELAY value would yield a valid pitch value. In some cases, out of range pitch values would drive the adaptive post filter berserk.

Bandwidth Expansion of Erasure LSPs

- Text: Modify Section 5.2.1
- Floating Point C Sim: fer.c
- Fixed Point C Sim: d_fer.c , d_rate_1.c
- Test Vector: ansi_fer.pkt

This improvement addresses the "buzz" sound. In some cases, valid frames may generate LSP/LPC values that cause the synthesis filter to go unstable. Typically, these values are used for only one frame, but if these frames are followed by a series of erasure frames, the old LSP/LPC values are "reused" causing "rail-to-rail" oscillations of the synthesis filter. This fix interpolates the old LSP values, eventually causing the LSP to go "full spread", which yields LPC coefficients that make the synthesis filter to go to unity gain, passing the erasure noise only.

TTY Changes

This section is to serve as a reminder that the TTY library for ANSI 127 EVRC and ANSI 733 13K have been changed to support both 45.45 baud and 50 baud Baudot code. The library is based on the TTY library for SMV, which uses the 40-bit DSP math library. The major differences are:

1. Supports both 45.45 baud and 50 baud Baudot code.
2. Makes the half-duplex solution mandatory for mitigating the effects of echo, on both the mobile and the infrastructure.
3. Applies a minimum input energy threshold to the input signal so that very low level tones (below –50 dBm) are not detected and regenerated.
4. Uses the 40-bit DSP math library, based on SMV.

References

[1] TR45.5.1.1/02.02.04.04, *Proposed Changes to EVRC Algorithm for ANSI Standard*, Lucent Technologies, February 2002.

[2] 3GPP2 TSG-C11-20020708-022, *Proposed Changes to EVRC Algorithm for ANSI Standard*, Lucent Technologies, August 2002.