PN-3-3633-AD1

APCO Project 25 Half-Rate Vocoder Addendum

APIC Vocoder Task Group

TIA TR-8.4 Vocoder subcommittee

TIA-102.BABA-1

Version 1.0.5

27 April 2009

TELECOMMUNICATIONS INDUSTRY ASSOCIATION

NOTICE OF DISCLAIMER AND LIMITATION OF LIABILITY

The document to which this Notice is affixed (the “Document”) has been prepared by one or more Engineering Committees or Formulating Groups of the Telecommunications Industry Association (“TIA”). TIA is not the author of the Document contents, but publishes and claims copyright to the Document pursuant to licenses and permission granted by the authors of the contents.

TIA Engineering Committees and Formulating Groups are expected to conduct their affairs in accordance with the TIA Engineering Manual (“Manual”), the current and predecessor versions of which are available at http://www.tiaonline.org/standards/procedures/manuals/engineering.cfm. TIA’s function is to administer the process, but not the content, of document preparation in accordance with the Manual and, when appropriate, the policies and procedures of the American National Standards Institute (“ANSI”). TIA does not evaluate, test, verify or investigate the information, accuracy, soundness, or credibility of the contents of the Document. In publishing the Document, TIA disclaims any undertaking to perform any duty owed to or for anyone.

If the Document is identified or marked as a project number (PN) document, or as a standards proposal (SP) document, persons or parties reading or in any way interested in the Document are cautioned that: (a) the Document is a proposal; (b) there is no assurance that the Document will be approved by any Committee of TIA or any other body in its present or any other form; (c) the Document may be amended, modified or changed in the standards development or any editing process.

The use or practice of contents of this Document may involve the use of intellectual property rights (“IPR”), including pending or issued patents, or copyrights, owned by one or more parties. TIA makes no search or investigation for IPR. When IPR consisting of patents and published pending patent applications are claimed and called to TIA’s attention, a statement from the holder thereof is requested, all in accordance with the Manual. TIA takes no position with reference to, and disclaims any obligation to investigate or inquire into, the scope or validity of any claims of IPR. TIA will neither be a party to discussions of any licensing terms or conditions, which are instead left to the parties involved, nor will TIA opine or judge whether proposed licensing terms or conditions are reasonable or non-discriminatory. TIA does not warrant or represent that procedures or practices suggested or provided in the Manual have been complied with as respects the Document or its contents.

If the Document contains one or more Normative References to a document published by another organization (“other SSO”) engaged in the formulation, development or publication of standards (whether designated as a standard, specification, recommendation or otherwise), whether such reference consists of mandatory, alternate or optional elements (as defined in the TIA Engineering Manual, 4th edition) then (i) TIA disclaims any duty or obligation to search or investigate the records of any other SSO for IPR or letters of assurance relating to any such Normative Reference; (ii) TIA’s policy of encouragement of voluntary disclosure (see Engineering Manual Section 6.5.1) of Essential Patent(s) and published pending patent applications shall apply; and (iii) Information as to claims of IPR in the records or publications of the other SSO shall not constitute identification to TIA of a claim of Essential Patent(s) or published pending patent applications.

TIA does not enforce or monitor compliance with the contents of the Document. TIA does not certify, inspect, test or otherwise investigate products, designs or services or any claims of compliance with the contents of the Document.

TIA AND THE AUTHOR OF THIS DOCUMENT DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY AND ALL WARRANTIES CONCERNING THE ACCURACY OF THE CONTENTS, ITS FITNESS OR APPROPRIATENESS FOR A PARTICULAR PURPOSE OR USE, ITS MERCHANTABILITY AND ITS NONINFRINGEMENT OF ANY THIRD PARTY’S INTELLECTUAL PROPERTY RIGHTS. TIA AND THE AUTHOR OF THIS DOCUMENT EXPRESSLY DISCLAIM ANY AND

ALL RESPONSIBILITIES FOR THE ACCURACY OF THE CONTENTS AND MAKES NO REPRESENTATIONS OR WARRANTIES REGARDING THE CONTENT'S COMPLIANCE WITH ANY APPLICABLE STATUTE, RULE OR REGULATION, OR THE SAFETY OR HEALTH EFFECTS OF THE CONTENTS OR ANY PRODUCT OR SERVICE REFERRED TO IN THE DOCUMENT OR PRODUCED OR RENDERED TO COMPLY WITH THE CONTENTS.

TIA AND THE AUTHOR OF THIS DOCUMENT SHALL NOT BE LIABLE FOR ANY AND ALL DAMAGES, DIRECT OR INDIRECT, ARISING FROM OR RELATING TO ANY USE OF THE CONTENTS CONTAINED HEREIN, INCLUDING WITHOUT LIMITATION ANY AND ALL INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS, LOSS OF PROFITS, LITIGATION, OR THE LIKE), WHETHER BASED UPON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING NEGATION OF DAMAGES IS A FUNDAMENTAL ELEMENT OF THE USE OF THE CONTENTS HEREOF, AND THESE CONTENTS WOULD NOT BE PUBLISHED BY TIA WITHOUT SUCH LIMITATIONS.

TITLE:	APCO Project 25 Half-Rate Vocoder Addendum
PROJECT:	PN–3-3633-AD1
VERSION:	Version 1.0.5
DATE:	27 April 2009
SOURCE:	**Digital Voice Systems, Inc.** 234 Littleton Road Westford, MA 01886 USA Phone: (978) 392-0002 Fax: (978) 392-8866 email: info@dvsinc.com web: www.dvsinc.com
COPYRIGHT STATEMENT:	**©Copyright, Digital Voice Systems, Inc., 2008** Digital Voice Systems Inc. (DVSI) claims certain rights, including patent rights under U.S. Patents 6,199,037, 5,870,405, 5,754,974, 5,664,051, 5,630,011, 5,517,511, 5,491,772, 5,247,579, 5,226,108, 5,226,084, 5,216,747 and 5,081,681 and under other U.S. and foreign patents and patents pending, in the Half-Rate vocoder technology described in this document and elsewhere. Any use of this technology requires a separate written license grant from DVSI. DVSI agrees that a license under these rights on reasonable and nondiscriminatory terms and conditions will be made available for use in APCO Project 25 equipment. Any enquiries directed to DVSI should be sent to the address shown below. IMBE is a trademark of Digital Voice Systems, Inc.

Foreword

The foreword is not a normative part of this document.

The original 7200 bps Project 25 vocoder standard was selected in 1992, and subsequently standardized by TIA as TIA-102.BABA. Since that time, various improvements have been developed in the industry, and the original suite of Project 25 standards has diversified into Phase 1 and Phase 2. During the development of Phase 2, a need was identified for a lower rate vocoder to facilitate higher spectral efficiency. This document describes a 3600 bps "Half-Rate" Vocoder intended to meet this need.

Revision History

Date	Revision	Description
10 October 2007	1.0.0	Initial Half-Rate Vocoder Addendum released.
10 March 2008	1.0.1	Revised Half-Rate Vocoder Addendum released.
28 March 2008	1.0.2	Typographical corrections made to Section 7.2.
1 July 2008	1.0.3	Revised Scope to reference Dual-Rate vocoder.
16 April 2009	1.0.4	Edits based on TIA ballot comments.
27 April 2009	1.0.5	Additional typographical edits based on TIA ballot comments.

Contents

List of Figures

List of Tables

1 Scope

This document is structured as an addendum to the APCO Project 25 Vocoder Description which has been published previously as TIA-102BABA [1]. The combination of these two documents describe a “Dual-Rate” Vocoder, where this addendum describes the 3600 bps “Half-Rate” mode of operation and the previously published APCO Project 25 Vocoder Description describes the 7200 bps “Full-Rate” mode of operation. The two documents together serve as the interoperability specification for the Dual-Rate vocoder employed within the APCO Project 25 System and Standard suite.

The focus of this document is to describe the interoperability requirements for transmitting and receiving voice through a Half-Rate Vocoder using a net bit rate of 2450 bps for voice data plus 1150 bps for Forward Error Correction (FEC) information, resulting in a a total channel bit rate of 3600 bps. As an addendum to TIA-102BABA, this document relies on its parent document for description of certain aspects (including the Multi-Band Excitation Speech Model, the Speech Input/Output Characteristics, Speech Analysis, and Speech Synthesis) held in common between the Half-Rate Vocoder described in this document and the Full-Rate Vocoder described in TIA-102BABA. The focus of this document is a technical description of the quantization and FEC elements of the Half-Rate vocoder necessary for interoperability with the Standard. Implementation considerations and improvements/enhancements not necessary for interoperability are not discussed.

The Half Rate Vocoder as described herein is intended for use over a radio air interface providing 3600 bps for voice data plus FEC. The reader is referred to the Project 25 Phase 2 Common Air Interface [8] for more information on the radio air interface.

2 Abbreviation Definitions

For the purposes of this Standard, the following definitions apply.

Table 1: Abbreviation Definitions

Abbreviation	Definition
DVSI	Digital Voice Systems, Inc.
FEC	Forward Error Correction
MBE	Multi-Band Excitation
DCT	Discrete Cosine Transform
LSB	Least Significant Bit
MSB	Most Significant Bit
PRBA	Predictive Residual Block
HOC	Higher Order Coefficient
IMBE	Improved Multi-Band Excitation
KNOX	Dual Tone
DTMF	Dual Tone Multi-Frequency
bps	Bits per Second
V/UV	Voiced Unvoiced

3 Introduction

The DVSI Half-Rate vocoder described in this document is based on the robust Multi-Band Excitation (MBE) speech model [2]. It divides a digital speech input signal into overlapping speech segments (or frames) spaced 20 ms apart. Each segment of speech is analyzed in the context of the underlying MBE speech model, and a set of model parameters are estimated for that particular subframe. The encoder quantizes each frame of model parameters, adds redundant FEC information, and transmits a bit stream at 3600 bps. The decoder receives this bit stream, applies FEC decoding to correct/detect bit errors and then reconstructs the MBE model parameters, and uses these model parameters to generate a synthetic speech signal. This synthesized speech signal is the output of the Half-Rate as shown in Figure 1.

Note that the Half-Rate Vocoder defined in this document is a digital-to-digital function. In practice a suitable A-to-D and D-to-A converter is usually needed to interface to typical analog elements such as a microphone and speaker. The reader is referred to Chapter 4 of [1] for more information on the recommended analog-to-digital interface and the definition of levels into and out of the vocoder.

The vocoder described in this document is compatible with many other signal processing functions which may be combined with the vocoder to improve performance in certain circumstances. Examples of optional processing functions include echo cancellation [7], noise removal [5], level adjustment, and others. These functions are outside the scope of this document, however additional information can be found in the aforementioned references and elsewhere in the open literature.

One defining characteristic of the Half-Rate vocoder is that it is an Improved Multi-Band Excitation IMBE^{TM} model-based voice coder, which does not try to reproduce the input speech signal on a sample by sample basis. Instead the vocoder constructs a synthetic speech signal which contains the same perceptual information as the original speech signal. By using a robust speech model and sophisticated parameter estimation algorithms, the vocoder is able to achieve a low data rate while maintaining most of the quality, intelligibility and speaker recognizability found in the original speech signal. This ability is what differentiates the MBE vocoder familty from many older model-based systems such as LPC vocoders, homomorphic vocoders, and channel vocoders which have not been as successful in reproducing high quality speech. Since the IMBE^{TM} vocoder represents speech as a sequence of parameters estimated according to an underlying speech model, its behavior may be unpredictable for non-speech input signals. For example, music, modulated data signals, single or multi-frequency tones or other test and measurement signals may not be reproduced faithfully if passed through the vocoder. In general the Half-Rate vocoder is designed to encode and decode human speech, but is not intended for other signal types such as music.

The Half-Rate encoder inputs digital speech and estimates the MBE speech model parameters using the method described in Chapter 5 of [1]. The result is an estimated fundamental frequency, an estimated set of voiced/unvoiced (V/UV) decisions and an estimated set of spectral amplitudes for each 20 ms segment of speech. Section 4 of this document describes the quantization of these MBE speech model parameters into a set of 49 bits for each voice frame, resulting in a net voice data rate of 2450 bps.

Section 5 of this Addendum describes the bit manipulations, including FEC encoding, used by the Half-Rate vocoder. Each frame of 49 voice bits is FEC encoded into 72 channel bits which are used to form a 3600 bps bit stream that is suitable for transmission over a radio air interface.

The Half-Rate decoder reverses the processes applied in the encoder. It first inputs a 3600

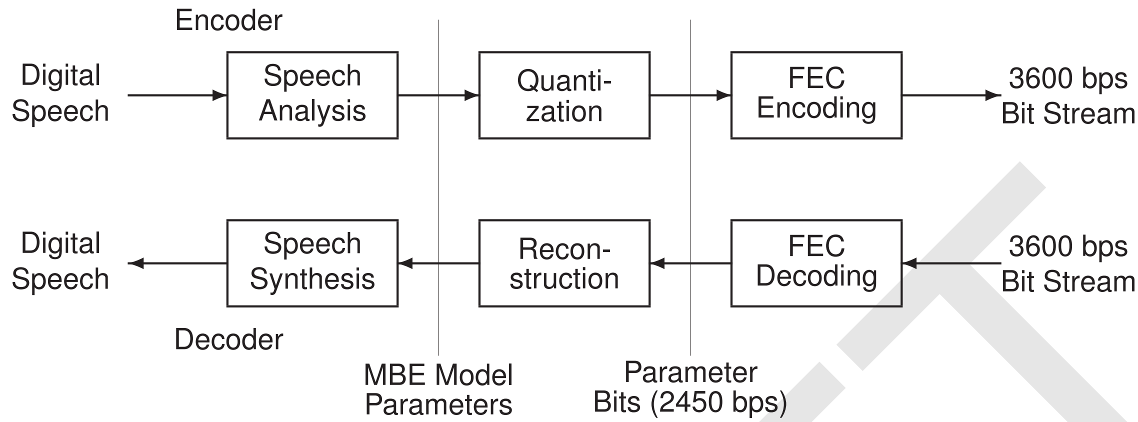

Figure 1: Half-Rate Vocoder

bps bit stream that may be received over a radio air interface. This input bit stream is similar to the bit stream produced by the encoder, except it may include bit errors due to channel noise. The decoder applies FEC decoding to try to correct and/or detect any bit errors that are present. The FEC decoder outputs decoded voice bits at a rate of 49 bits per frame. The FEC decoder also outputs side information indicating channel quality as described in Clause 5.5. This side information is used in the error mitigation described in Clauses 5.6 and 5.7. The data output by the FEC decoder is then used to reconstruct the MBE model parameters as described in Section 4 of this document. The reconstructed MBE model parameters are then used to produce a synthetic digital speech signal as described in Chapter 11 of [1]. The synthetic digital speech signal output by the decoder is suitable for playback through a D-to-A converter and a loudspeaker.

Note that while DVSI has attempted to make this document accurate and complete, it may still contain certain errors or omissions. DVSI may occassionaly update this document to correct such problems. Errors or omissions which are discovered should be communicated to DVSI using the contact information provided on the title page.

Table 2: Half-Rate Voice/Silence Frame Bit Allocation

Quantizer Value	Model Parameter	Bits Per Frame
b_0	Fundamental Frequency	7
b_1	V/UV Decisions	5
b_2	Gain	5
b_3	PRBA24 Vector	9
b_4	PRBA58 Vector	7
b_5	HOC1 Vector	5
b_6	HOC2 Vector	4
b_7	HOC3 Vector	4
b_8	HOC4 Vector	3
Total		49

4 Half-Rate Vocoder Quantization

This section describes the final parameter estimation and quantization used by the Half-Rate Vocoder for each voice frame.

The analysis of each speech frame described in Chapter 5 of [1] generates a set of MBE model parameters consisting of the fundamental frequency, $\hat{\omega}_0$, the V/UV decisions, $\hat{v}_k$ for $1 \leq k \leq \hat{K}$, and the spectral amplitudes, $\hat{M}_l$ for $1 \leq l \leq \hat{L}$. The number of V/UV decisions, $\hat{K}$, and the number of spectral magnitudes, $\hat{L}$, varies from frame-to-frame as a function of the fundamental frequency. Since the Half-Rate vocoder is designed to operate at 3.6 kbps with a 20 ms frame length, 72 bits per frame are available for encoding the model parameters. Of these 72 bits, 23 are reserved for error control as is discussed in Clause 5.2 of this document, and the remaining 49 bits are divided among the model parameters as shown in Table 2.

This section describes the manner in which these bits are used to quantize, encode, decode and reconstruct the model parameters. In Clause 4.1 the encoding and decoding of the fundamental frequency is discussed, while Clause 4.2 discusses the encoding and decoding of the V/UV decisions. Clause 4.3 discusses the quantization and encoding of the spectral amplitudes, and Clause 4.4 discusses the decoding and reconstruction of the spectral amplitudes. Reference [3] provides general information on many of the techniques used in this section.

At the encoder, the output of the Half-Rate quantizer is a set of nine quantizer values, $\hat{b}_0, \hat{b}_1, ...\hat{b}_8$ which contain the 49 bits used to quantize the frame parameters. At the decoder, the corresponding received quantizer values, $\tilde{b}_0, \tilde{b}_1, ...\tilde{b}_8$ contain the 49 input bits used to recontruct the MBE parameters for the frame. The number of bits assigned to each quantizer value are shown in Table 2, where b_i refers to both $\hat{b}_i$ in the encoder as well as $\tilde{b}_i$ in the decoder. This table applies to both voice frames and silence frames. Note that for tone frames, this table does not apply, and intead tone frames are formatted as described in Section 7.

4.1 Fundamental Frequency Encoding and Decoding

When voice is present, the fundamental frequency is estimated in the interval $\frac{2\pi}{123.125} \leq \hat{\omega}_0 \leq \frac{2\pi}{19.875}$ and is encoded by setting $\hat{b}_0$ equal to the 7-bit index corresponding to the entry in Annex A that is closest to $\hat{\omega}_0$. Note that values of $\hat{b}_0 >= 120$ are reserved for non voice frames as shown in

Table 3: Seven Bit Binary Representation

value	*bits*
0	000 0000
1	000 0001
2	000 0010
.	.
.	.
.	.
127	111 1111

Table 4: Vocoder Frame Types

Frame Type	b_0
voice	0 - 119
erasure	120 - 123
silence	124 - 125
tone	126 - 127

Table 4. If the frame is a silence frame, then $\hat{b}_0 = 124$.

The selected quantizer value $\hat{b}_0$ is represented with 7 bits using the unsigned binary representation shown in Table 3. This representation is used throughout the vocoder to convert quantized values into a specific bit pattern.

At the receiver, the frame type is decoded first by checking $\tilde{b}_0$ against Table 4 to determine whether the current frame is a voice frame, erasure frame, silence frame or tone frame. If the frame is a voice frame, then the fundamental frequency is decoded by setting $\tilde{\omega}_0$ equal to the value shown in Annex A corresponding to the 7-bit index $n = \tilde{b}_0$. In addition $\tilde{b}_0$ is used to calculate $\tilde{L}$, the number of spectral amplitudes, as shown in Annex A.

If the frame is determined to be a silence frame, then the values of $\tilde{\omega}_0$, $\tilde{L}$, and $\tilde{v}_l$ are set as shown in the following equations:

$$\tilde{\omega}_0 = \frac{2\pi}{32} \quad (1)$$

$$\tilde{L} = 14 \quad (2)$$

$$\tilde{v}_l = 0 \quad (3)$$

The spectral amplitudes are then decoded using the same procedure as for voice frames (see Clause 4.4).

If the frame is determined to be a tone frame, then a set of MBE speech model parameters are not decoded for the frame. Instead a set of tone parameters are decoded from the tone frame as described in Clause 7.2.

If the frame is determined to be an erasure frame, then the frame is considered invalid (i.e. erased) and a set of MBE speech model parameters is not decoded for the frame. Instead a frame repeat is performed as described in Clause 5.6. Note that for erasure frames, only the frame type needs to be decoded, and the remaining bits in frame are ignored.

4.2 Voiced/Unvoiced Decision Encoding and Decoding

The V/UV decisions $\hat{v}_k$, for $1 \leq k \leq \hat{K}$, are binary values that classify each of the K frequency bands as either voiced or unvoiced. These values are encoded into the 5-bit quantizer value $\hat{b}_1$ using the vector table shown in Annex B. Each of the first 17 vectors, $\nu(n)$ for $0 \leq n < 17$ are evaluated using distance Equation 4 and for voice frames $\hat{b}_1$ is set equal to the index n of the vector that yields the minimum distance.

$$E_n = \sum_{l=1}^{\hat{L}} |\hat{M}_l|^2 |\hat{v}_{k_l} - \nu_{j_l}(n)|^2 \tag{4}$$

where $\hat{v}_k$ are the estimated voicing decisions, and where the indices k_l and j_l are computed as follows:

$$j_l = \lfloor \frac{16\hat{\omega}_0}{2\pi} \rfloor \tag{5}$$

$$k_l = \begin{cases} \lfloor \frac{l+2}{3} \rfloor & \text{if } l \leq 36 \\ 12 & \text{othewise} \end{cases} \tag{6}$$

For silence frames $\hat{b}_1 = 0$.

For voice frames, the decoder reconstructs the V/UV decisions using the received 5-bit quantizer value $\tilde{b}_1$. This value is used as an index $n = \tilde{b}_1$ to select the corresponding 8 element vector $\nu(n)$ from Annex B. The voicing decisions, $\tilde{v}_l$ for $1 \leq l \leq \tilde{L}$, are then reconstructed from the selected vector using the following relationship:

$$\tilde{v}_l = \nu_{j_l}(n) \tag{7}$$

where j_l is the index computed according to Equation 5. For silence frames, the voicing decisions are reconstructed according to Equation 3.

Note that the decoder reconstructs the V/UV decisions for each spectral amplitude $1 \leq l \leq L$, rather than for each voicing band $1 \leq k \leq K$. This is a departure from the V/UV convention used by the encoder, which used a single V/UV decision to represent an entire frequency band. Instead the decoder assigns a separate V/UV decision for each spectral amplitude. This same procedure (i.e. computing the voicing decisions corresponding to each spectral amplitude) is also done by the encoder for use in computing the log spectral magnitudes according to Equation 8 in Clause 4.3.

4.3 Spectral Amplitudes Encoding

The process described in this Clause and the corresponding decoding Clause 4.4 is used to encode and decode, respectively, the spectral amplitudes for both voice frames and silence frames.

For the purpose of the following discussion $\hat{L}(0)$ or $\hat{L}$ refer to the number of harmonics in the current frame, while $\hat{L}(-1)$ refers to the number of harmonics in the previous frame. Similarly, $\hat{\Lambda}_l(0)$ (or $\hat{\Lambda}_l$) for $1 \leq l \leq \hat{L}(0)$ refers to the unquantized spectral amplitudes of the current frame, while $\tilde{\Lambda}_l(-1)$ for $1 \leq l \leq \hat{L}(-1)$ refers to the quantized spectral amplitudes of the last voice frame.

The spectral amplitudes $\hat{M}_l$, for $1 \leq l \leq \hat{L}$, are real values which must be quantized prior to encoding. This is accomplished by first computing the log spectral amplitudes, $\hat{\Lambda}_l$, for $1 \leq l \leq \hat{L}$, as follows:

$$\hat{\Lambda}_l = \begin{cases} \log_2 \hat{M}_l + 0.5 \log_2 \hat{L} & \text{if } \tilde{v}_l = 1 \\ \log_2 \hat{M}_l + 0.5 \log_2 (\tilde{\omega}_0 \cdot \hat{L}) + 2.289 & \text{otherwise} \end{cases} \tag{8}$$

where $\tilde{v}_l$ is the voicing decision corresponding to the $l'th$ spectral amplitude that is computed by the encoder. The gain is then computed from the log spectral parameters as follows:

$$\hat{\gamma}(0) = \frac{1}{\hat{L}} \sum_{l=1}^{\hat{L}} \hat{\Lambda}_l \tag{9}$$

and the differential gain is then computed as:

$$\hat{\Delta}_\gamma = \hat{\gamma}(0) - 0.5\tilde{\gamma}(-1) \tag{10}$$

where $\hat{\gamma}(0)$ is the computed gain for the current frame while $\tilde{\gamma}(-1)$ is the reconstructed gain from the previous frame. The differential gain $\hat{\Delta}_\gamma$ is then quantized using the 5 bit quantizer shown in Annex D. The quantizer value $\hat{b}_2$ is determined as the 5 bit index n corresponding to the quantizer level from the table in Annex D that is closest to $\hat{\Delta}_\gamma$. The value of $\tilde{\gamma}(-1)$ should be initialized to 0.

Once the gain has been computed and quantized, the log spectral amplitudes, $\hat{\Lambda}_l$ for $1 \leq l \leq \hat{L}$, are then used to compute the prediction residuals $\hat{T}_l$ for $1 \leq l \leq \hat{L}$, according to Equations (11) through (15).

$$\hat{k}_l = \frac{\hat{L}(-1)}{\hat{L}(0)} l \tag{11}$$

$$\hat{\delta}_l = \hat{k}_l - \lfloor \hat{k}_l \rfloor \tag{12}$$

$$\begin{aligned} \hat{T}_l = \hat{\Lambda}_l(0) \; & - \; 0.65\,(1 - \hat{\delta}_l)\,\tilde{\Lambda}_{\lfloor \hat{k}_l \rfloor}(-1) \\ & - \; 0.65\,\hat{\delta}_l\,\tilde{\Lambda}_{\lfloor \hat{k}_l \rfloor + 1}(-1) \\ & + \; \frac{0.65}{\hat{L}(0)} \sum_{\lambda=1}^{\hat{L}(0)} (1 - \hat{\delta}_\lambda)\,\tilde{\Lambda}_{\lfloor \hat{k}_\lambda \rfloor}(-1) + \hat{\delta}_\lambda\,\tilde{\Lambda}_{\lfloor \hat{k}_\lambda \rfloor + 1}(-1) \end{aligned} \tag{13}$$

In order to form $\hat{T}_l$ using Equations (11) through (13), the following assumptions are made:

$$\tilde{\Lambda}_0(-1) = \tilde{\Lambda}_1(-1) \tag{14}$$

$$\tilde{\Lambda}_l(-1) = \tilde{\Lambda}_{\tilde{L}(-1)}(-1) \qquad \text{for } l > \tilde{L}(-1) \tag{15}$$

Upon initialization $\tilde{\Lambda}_l(-1)$ should be set equal to 1.0 for all l, and $\hat{L}(-1) = 15$.

Once the $\hat{L}$ prediction residuals have been computed, they are divided into 4 blocks. The length of each block, denoted $\hat{J}_i$ for $1 \leq i \leq 4$, is determined as a function of $\hat{L}$ in accordance with Annex C.

The first or lowest frequency block is denoted by $\hat{c}_{1,j}$ for $1 \leq j \leq \hat{J}_1$, and it consists of the first $\hat{J}_1$ consecutive elements of $\hat{T}_l$ (i.e. $1 \leq l \leq \hat{J}_1$). The second block is denoted by $\hat{c}_{2,j}$ for $1 \leq j \leq \hat{J}_2$, and it consists of the next $\hat{J}_2$ consecutive elements of $\hat{T}_l$ (i.e. $\hat{J}_1 + 1 \leq l \leq \hat{J}_1 + \hat{J}_2$). This continues through the fourth or highest frequency block, which is denoted by $\hat{c}_{4,j}$ for $1 \leq j \leq \hat{J}_4$. It consists of the last $\hat{J}_4$ consecutive elements of $\hat{T}_l$ (i.e. $\hat{L} + 1 - \hat{J}_4 \leq l \leq \hat{L}$).

Each of the four blocks is transformed using a Discrete Cosine Transform (DCT), which is discussed in [3]. The length of the DCT for the i'th block is equal to $\hat{J}_i$. The DCT coefficients are denoted by $\hat{C}_{i,k}$, where $1 \leq i \leq 4$ refers to the block number, and $1 \leq k \leq \hat{J}_i$ refers to the particular coefficient within each block. The formula for the computation of these DCT coefficients is as follows:

$$\hat{C}_{i,k} = \frac{1}{\hat{J}_i} \sum_{j=1}^{\hat{J}_i} \hat{c}_{i,j} \cos[\frac{\pi(k-1)(j-\frac{1}{2})}{\hat{J}_i}] \qquad \text{for } 1 \leq k \leq \hat{J}_i \tag{16}$$

The DCT coefficients from each of the four blocks are then divided into two groups. The first group consists of the first two DCT coefficients from each of the four blocks. These first coefficients are used to form an eight element Predictive Residual Block Average (PRBA) vector, $\hat{R}_i$ for $1 \leq i \leq 8$, as follows:

$$\hat{R}_1 = \hat{C}_{1,1} + \sqrt{2}\hat{C}_{1,2} \tag{17}$$
$$\hat{R}_2 = \hat{C}_{1,1} - \sqrt{2}\hat{C}_{1,2} \tag{18}$$
$$\hat{R}_3 = \hat{C}_{2,1} + \sqrt{2}\hat{C}_{2,2} \tag{19}$$
$$\hat{R}_4 = \hat{C}_{2,1} - \sqrt{2}\hat{C}_{2,2} \tag{20}$$
$$\hat{R}_5 = \hat{C}_{3,1} + \sqrt{2}\hat{C}_{3,2} \tag{21}$$
$$\hat{R}_6 = \hat{C}_{3,1} - \sqrt{2}\hat{C}_{3,2} \tag{22}$$
$$\hat{R}_7 = \hat{C}_{4,1} + \sqrt{2}\hat{C}_{4,2} \tag{23}$$
$$\hat{R}_8 = \hat{C}_{4,1} - \sqrt{2}\hat{C}_{4,2} \tag{24}$$

The quantization of the PRBA vector is discussed in Clause 4.3.1. The second group consists of the remaining higher order DCT coefficients from each block. These coefficients correspond to $\hat{C}_{i,j}$, for $1 \leq i \leq 4$ and $3 \leq j \leq \hat{J}_i$. Note that if $\hat{J}_i = 2$, then there are no higher order DCT coefficients in the i'th block. The quantization of the higher order DCT coefficients is discussed in Clause 4.3.2.

One important feature of the spectral amplitude encoding algorithm, is that the spectral amplitude information is transmitted differentially. Specifically, a prediction residual is transmitted which measures the change in the spectral envelope between the current frame and the previous frame. In order for a differential scheme of this type to work properly, the encoder must simulate the operation of the decoder and normally use the reconstructed spectral amplitudes from the previous frame to predict the spectral amplitudes of the current frame. The Half-Rate spectral amplitude encoder simulates the spectral amplitude decoder by setting $\tilde{L} = \hat{L}$ and then reconstructing the spectral amplitudes as discussed in Clause 4.4 and the corresondnig voicing decisions for each spectral amplitude as discussed in Clause 4.2. These reconstructed spectral amplitudes are then saved by the encoder for use in quantizing the spectral amplitudes of the next frame. An exception to this reconstruct and update process occurs for silence frames, tone frames and erasure frames in which case the saved reconstructed spectral amplitudes are not updated with the reconstructed spectral amplitudes from the current frame. This ensures that only voice frames (i.e. not silence frames, tone frames or erasure frames) are used in predicting future specral amplitudes.

4.3.1 Encoding the PRBA Vector

The quantization of the PRBA vector begins with an eight point DCT of $\hat{R}_i$ for $1 \leq i \leq 8$ as shown in the following equation:

$$\hat{G}_m = \frac{1}{8} \sum_{i=1}^{8} \hat{R}_i \cos[\frac{\pi(m-1)(i-\frac{1}{2})}{8}] \qquad \text{for } 1 \leq m \leq 8 \tag{25}$$

The resulting vector, denoted by $\hat{G}_m$ for $1 \leq m \leq 8$, is vector quantized in two parts. The first element, $\hat{G}_1$, is discarded. The next three elements of the PRBA vector (i.e. $[\hat{G}_2, \hat{G}_3, \hat{G}_4]$ are jointly quantized using the 9-bit vector quantizer shown in Annex E. The quantizer value $\hat{b}_3$ is determined as the 9-bit index n corresponding to the quantizer vector from the table in Annex E which is closest (i.e. has the minimum mean-squared error) to the three element vector $[\hat{G}_2, \hat{G}_3, \hat{G}_4]$. The final four elements of the PRBA vector (i.e. $[\hat{G}_5, \hat{G}_6, \hat{G}_7, \hat{G}_8]$ are jointly quantized using the 7-bit vector quantizer shown in Annex F. The quantizer value $\hat{b}_4$ is determined as the 7-bit index n corresponding to the quantizer vector from the table in Annex F which is closest (i.e. has the minimum mean-squared error) to the three element vector $[\hat{G}_5, \hat{G}_6, \hat{G}_7, \hat{G}_8]$.

4.3.2 Encoding the Higher Order DCT Coefficients

Once the gain and the PRBA vector has been quantized, the higher order DCT coefficients from each of the four blocks are quantized to form the four quantizer values $\hat{b}_5$, $\hat{b}_6$, $\hat{b}_7$, and $\hat{b}_8$. The number of bits for each of these quantizer values is specified in Table 2

The quantizer value $\hat{b}_{i+4}$, for $1 \leq i \leq 4$, is the quantizer output for the $i'th$ block. These quantizer values are determined by first forming a set of Higher Order Coefficient (HOC) vectors $\hat{H}_{i,j} = \hat{C}_{i,j+2}$, for $1 \leq i \leq 4$ and $1 \leq j \leq \hat{J}_i - 2$. However, if $\hat{J}_i \leq 2$ for any i, then an HOC vector is not formed for that block and the corresponding quantizer value $\hat{b}_{i+4} = 0$.

Once the 4 HOC vectors are formed in this manner, then each HOC vector is vector quantized using the specified vector quantization table shown in Annex G. A 5-bit (32 vector) quantization table is used to determine $\hat{b}_5$ from $\hat{H}_{1,j}$; a 4-bit (16 vector) quantization table is used to determine $\hat{b}_6$ from $\hat{H}_{2,j}$; another 4-bit (16 vector) quantization table is used to determine $\hat{b}_7$ from $\hat{H}_{3,j}$; and a 3-bit (8 vector) quantization table is used to determine $\hat{b}_8$ from $\hat{H}_{4,j}$. Each of these vector quantization tables is listed seperately in Annex G. The quantizer value for each HOC vector, is determined by setting $\hat{b}_{i+4}$ to the index of the quantizer vector from the specified table in Annex F which is closest (i.e. has the minimum mean-squared error) to the HOC vector $\hat{H}_{i,j}$ for $1 \leq i \leq 4$. Note that each quantizaton vector shown in Annex G is 4 elements long, while the length of the HOC vectors can vary from 1 to nearly 20 (ignoring the zero length HOC vectors). If an HOC vector has a non-zero length less than 4, then only the first $\hat{J}_i - 2$ elements of each quantization vector are used in computing the minimum mean squared error and the remaining elements from each quantization vector are ignored. Similarly, if an HOC vector has a length greater than 4, then only the first 4 elements of that HOC vector are used in computing the minimum mean squared error, and the remaining elements in the HOC vector are ignored.

4.4 Spectral Amplitudes Decoding

In order for the decoder to reconstruct the spectral amplitudes, the parameter $\tilde{L}$ must first be computed from $\tilde{b}_0$ using Annex A, and the V/UV decisions $\tilde{v}_l$ for $1 \leq l \leq \tilde{L}$ must be computed from $\tilde{b}_1$ using Equation 7. Next the spectral amplitudes are decoded and reconstructed by inverting the quantization and encoding procedure described above.

Reconstruction of the spectral amplitudes is accomplished by dividing them into 4 DCT blocks, where the length of each block, $\tilde{J}_i$ for $1 \leq i \leq 4$ is set depending on $\tilde{L}$ according to Annex C.

4.4.1 Decoding the Gain

The first spectral parameter to be decoding is the differential gain, $\tilde{\Delta}_\gamma$ using the quantizer value $\tilde{b}_2$. This is accomplished by setting $\tilde{\Delta}_\gamma$ equal to the quantization level from Annex D corresponding to the index $n = \tilde{b}_2$. The decoded gain for the current frame, $\tilde{\gamma}(0)$, is then reconstructed as follows:

$$\tilde{\gamma}(0) = \tilde{\Delta}_\gamma + 0.5\tilde{\gamma}(-1) \tag{26}$$

where $\tilde{\gamma}(-1)$ is the reconstucted gain from the last valid voice frame, not including any past tone, silence or erasure frames, and where $\tilde{\gamma}(-1)$ is initialized to 0.

4.4.2 Decoding the PRBA Vector

Once the gain has been decoded, the transformed PRBA vector, denoted by $\tilde{G}_m$ for $1 \leq m \leq 8$ is decoded using the quantizer values $\tilde{b}_3$ and $\tilde{b}_4$. The first element of the transformed PRBA vector is set equal to 0 (i.e. $\tilde{G}_1 = 0$). The next three elements of the transformed PRBA vector, $[\tilde{G}_2, \tilde{G}_3, \tilde{G}_4]$ are set equal to the quantization vector from the table in Annex E corresponding to index $n = \tilde{b}_3$. The final four elements of the transformed PRBA vector, $[\tilde{G}_5, \tilde{G}_6, \tilde{G}_7, \tilde{G}_8]$ are set equal to the quantization vector from the table in Annex F corresponding to index $n = \tilde{b}_4$.

Once the transformed PRBA vector has been reconstructed in this manner, then the PRBA vector $\tilde{R}_m$ for $1 \leq m \leq 8$ is computed through and inverse DCT of $\tilde{G}_m$ as shown in the following equation:

$$\tilde{R}_i = \sum_{m=1}^{8} \alpha(m)\,\tilde{G}_m \cos[\frac{\pi(m-1)(i-\frac{1}{2})}{8}] \qquad \text{for } 1 \leq i \leq 8 \tag{27}$$

$$\alpha(m) = \begin{cases} 1 & \text{if } m = 1 \\ 2 & \text{otherwise} \end{cases} \tag{28}$$

The first two elements of each of the four DCT blocks, denoted by $\tilde{C}_{i,k}$ for $1 \leq i \leq 4$ and $1 \leq k \leq 2$ are then computed from the PRBA vector as follows:

$$\tilde{C}_{1,1} = \frac{1}{2}(\tilde{R}_1 + \tilde{R}_2) \tag{29}$$

$$\tilde{C}_{1,2} = \frac{1}{2\sqrt{2}}(\tilde{R}_1 - \tilde{R}_2) \tag{30}$$

$$\tilde{C}_{2,1} = \frac{1}{2}(\tilde{R}_3 + \tilde{R}_4) \tag{31}$$

$$\tilde{C}_{2,2} = \frac{1}{2\sqrt{2}}(\tilde{R}_3 - \tilde{R}_4) \tag{32}$$

$$\tilde{C}_{3,1} = \frac{1}{2}(\tilde{R}_5 + \tilde{R}_6) \tag{33}$$

$$\tilde{C}_{3,2} = \frac{1}{2\sqrt{2}}(\tilde{R}_5 - \tilde{R}_6) \tag{34}$$

$$\tilde{C}_{4,1} = \frac{1}{2}(\tilde{R}_7 + \tilde{R}_8) \tag{35}$$

$$\tilde{C}_{4,2} = \frac{1}{2\sqrt{2}}(\tilde{R}_7 - \tilde{R}_8) \tag{36}$$

Reconstruction of the remaining elements of $\tilde{C}_{i,k}$ is discussed in the Clause 4.4.3.

4.4.3 Decoding the Higher Order DCT Coefficients

The higher order DCT coefficients, which are denoted by $\tilde{C}_{i,k}$ for $1 \leq i \leq 4$ and $2 \leq k \leq \tilde{J}_i$, are reconstructed from the quantizer values $\tilde{b}_5$, $\tilde{b}_6$, $\tilde{b}_7$, and $\tilde{b}_8$. For the $i'th$ block, an HOC vector, denoted by $\tilde{H}_{i,j}$ is first reconstructed as the quantization vector from the specified table in Annex G corresponding to the index $n = \tilde{b}_{i+4}$.

The higher order DCT coefficients, $\tilde{C}_{i,k}$ for $1 \leq i \leq 4$ and $2 \leq k \leq \tilde{J}_i$, are then reconstructed from the HOC vectors according to the following equation:

$$\tilde{C}_{i,k} = \begin{cases} \tilde{H}_{i,k-2} & \text{for } 2 \leq k \leq \tilde{J}_i \text{ and } k \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

Note that if $\tilde{J}_i \leq 2$, then there are no higher order DCT coefficients to reconstruct, and this step is skipped for the $i'th$ block.

Once all the DCT coefficients $\tilde{C}_{i,k}$ have been reconstructed, an inverse DCT is computed on each of the four blocks to form the vectors $\tilde{c}_{i,j}$. This is done using the following equations for $1 \leq i \leq 4$.

$$\tilde{c}_{i,j} = \sum_{k=1}^{\tilde{J}_i} \alpha(k)\tilde{C}_{i,k} \cos[\frac{\pi(k-1)(j-\frac{1}{2})}{\tilde{J}_i}] \qquad \text{for } 1 \leq j \leq \tilde{J}_i \tag{38}$$

$$\alpha(k) = \begin{cases} 1 & \text{if } k = 1 \\ 2 & \text{otherwise} \end{cases} \tag{39}$$

The four inverse transformed blocks $\tilde{c}_{i,j}$ are then joined to form a single vector of length $\tilde{L}$, which is denoted $\tilde{T}_l$ for $1 \leq l \leq \tilde{L}$. The vector $\tilde{T}_l$ corresponds to the reconstructed spectral amplitude prediction residuals. The adopted convention is that the first $\tilde{J}_1$ elements of $\tilde{T}_l$ are equal to $\tilde{c}_{1,j}$ for $1 \leq j \leq \tilde{J}_1$. The next $\tilde{J}_2$ elements of $\tilde{T}_l$ are equal to $\tilde{c}_{2,j}$ for $1 \leq j \leq \tilde{J}_2$. This continues until the last $\tilde{J}_4$ elements of $\tilde{T}_l$ are equal to $\tilde{c}_{4,j}$ for $1 \leq j \leq \tilde{J}_4$.

Once the vector $\tilde{T}_l$ has been reconstructed in this manner, then the reconstructed log spectral amplitudes for the current frame, denoted by $\tilde{\Lambda}_l(0)$ for $1 \leq l \leq \tilde{L}$, are computed using the following equations:

$$\tilde{k}_l = \frac{\tilde{L}(-1)}{\tilde{L}(0)} l \tag{40}$$

$$\tilde{\delta}_l = \tilde{k}_l - \lfloor \tilde{k}_l \rfloor \tag{41}$$

$$\tilde{\Gamma} = \tilde{\gamma}(0) - 0.5 \log_2 \tilde{L} - \frac{1}{\tilde{L}} \sum_{\lambda=1}^{\tilde{L}} \tilde{T}_\lambda \tag{42}$$

$$\begin{aligned} \tilde{\Lambda}_l(0) = \tilde{T}_l & + 0.65\,(1-\tilde{\delta}_l)\,\tilde{\Lambda}_{\lfloor \tilde{k}_l \rfloor}(-1) \\ & + 0.65\,\tilde{\delta}_l\,\tilde{\Lambda}_{\lfloor \tilde{k}_l \rfloor + 1}(-1) \\ & - \frac{0.65}{\tilde{L}(0)} \sum_{\lambda=1}^{\tilde{L}(0)} (1-\tilde{\delta}_\lambda)\,\tilde{\Lambda}_{\lfloor \tilde{k}_\lambda \rfloor}(-1) + \tilde{\delta}_\lambda\,\tilde{\Lambda}_{\lfloor \tilde{k}_\lambda \rfloor + 1}(-1) \\ & + \tilde{\Gamma} \end{aligned} \tag{43}$$

where $\tilde{\Lambda}_l(-1)$ denotes the reconstructed log spectral magnitudes from the last voice frame, not including any tone, silence, erasure or invalid frames. In applying Equations (40) through (43), the following assumptions are made:

$$\tilde{\Lambda}_0(-1) = \tilde{\Lambda}_1(-1) \tag{44}$$

$$\tilde{\Lambda}_l(-1) = \tilde{\Lambda}_{\tilde{L}(-1)}(-1) \qquad \text{for } l > \tilde{L}(-1) \tag{45}$$

In addition, it is assumed that upon initialization $\tilde{\Lambda}_l(-1) = 1$ for all l, and $\tilde{L}(-1) = 15$.

Finally, the spectral amplitudes for the current frame, denoted by $\tilde{M}_l$, for $1 \leq l \leq \tilde{L}$ are reconstructed from $\tilde{\Lambda}_l$ using the following equation:

$$\tilde{M}_l = \begin{cases} \exp\left(0.693\, \tilde{\Lambda}_l\right) & \text{if } \tilde{v}_l = 1 \\ \frac{0.2046}{\sqrt{\tilde{\omega}_0}} \exp\left(0.693\, \tilde{\Lambda}_l\right) & \text{otherwise} \end{cases} \tag{46}$$

where $\tilde{v}_l$ is the voicing decision corresponding to the $l'th$ spectral amplitude that is computed by the decoder. The reconstructed spectral amplitudes $\tilde{M}_l$ are then used by the synthesis algorithm, as described in Chapter 11 of [1], to produce the synthetic digital speech output from the Half-Rate decoder.

One final note is that the Half-Rate encoder uses the value $\hat{L}$ to determine the four DCT block sizes, $\hat{J}_i$, while at the decoder the corresponding block sizes, denoted by $\tilde{J}_i$ are determined by the value $\tilde{L}$. In order to ensure proper operation it is necessary that these two values be equal (i.e. $\hat{L} = \tilde{L}$). The encoder and decoder are designed to ensure this property except in the presence of a very large number of bit errors. In addition, the decoder is designed to detect frames where a large number of bit errors may prevent the generation of the correct bit allocation and quantizer step sizes. In this case, the decoder discards the bits for the current frame and repeats the parameters from the previous frame. This is discussed in more detail in later clauses of this document.

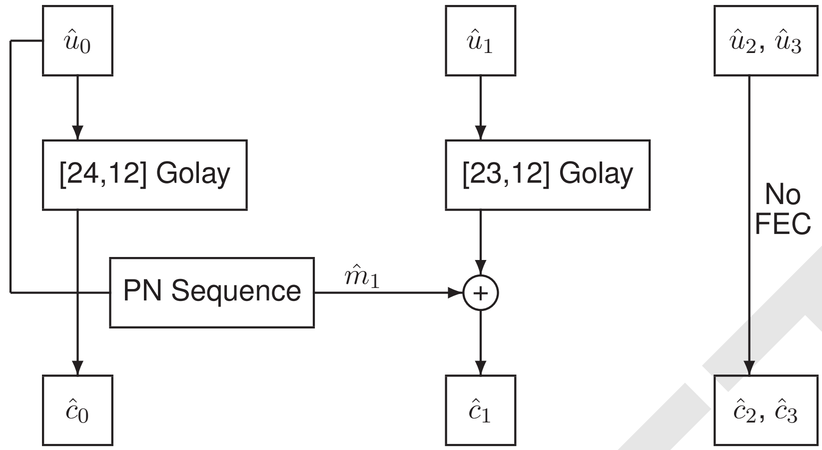

Figure 2: Half-Rate Code Vector Construction

5 Bit Manipulations

The Half-Rate vocoder uses a number of different bit manipulations in order to increase its robustness to channel degradations. The quantizer values, $\hat{b}_0, \ldots, \hat{b}_8$, are first prioritized into a set of four bit vectors, denoted by $\hat{u}_0, \ldots, \hat{u}_3$. These vectors are protected with error control codes, consisting of one [24,12] Golay code and one [23,12] Golay codes, and modulated to produce a set of code vectors denoted by $c_0, \ldots, c_3$. The error control codes add redundancy by increasing the number of bits per voice frame from 49 (2450 bps) to 72 (3600 bps). The construction of the code vectors the from the quantizer values is depicted in Figure 2. Intra-frame bit interleaving is applied to these code vectors for form the transmitted bit stream for the Half-Rate Vocoder.

The Half-Rate decoder reverses the bit manipulations performed by the encoder. First the decoder de-interleaves each frame of 72 bits to obtain the four code vectors $\tilde{c}_0, \ldots, \tilde{c}_3$. The decoder then demodulates and error control decodes these code vectors to produce the decoded bit vectors $\tilde{u}_0, \ldots, \tilde{u}_3$. In order to ensure sufficient performance it is necessary that the decoder decode all error control codes up to their maximum error correction/detection capability, and soft-decision decoding is recommended to further improve the robustness to bit errors (soft-decision decoding was used in the MOS evaluation of the Half-Rate Vocoder performed by TIA in 2003). Note that there are a number of well established methods for decoding of the Golay codes, and specific methods are not presented in this document.

Once the decoder has performed error correction and detection, it must rearrange the bit vectors to reconstruct the quantizer values, denoted by $\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_8$. These values are then used to reconstruct the MBE model parameters as described in Section 4, and the resulting parameters are finally used to synthesize the current frame of speech.

One should note that the Half-Rate decoder employs a number of different mechanisms to improve performance in the presence of bit errors. These mechanisms consist first of error control codes, which are able to remove a significant number of errors. In addition, the Half-Rate decoder uses bit modulation combined with frame repeats and frame mutes to detect and discard highly corrupted frames. Finally, the Half-Rate decoder uses adaptive smoothing to reduce the perceived effect of any remaining errors. These mechanisms are all discussed in the following clauses of this description.

Table 5: Construction of Bit Vector $\hat{u}_0$

11 8	7 4	3 0
$\hat{b}_0(6,5,4,3)$	$\hat{b}_1(4,3,2,1)$	$\hat{b}_2(4,3,2,1)$

Table 6: Construction of Bit Vector $\hat{u}_1$

11 4	3 0
$\hat{b}_3(8,7,6,5,4,3,2,1)$	$\hat{b}_4(6,5,4,3)$

5.1 Bit Prioritization

The first bit manipulation performed by the Half-Rate encoder is a rearrangement of the quantizer values $\hat{b}_0$, $\hat{b}_1$, ..., $\hat{b}_8$ into a set of 4 prioritized bit vectors denoted by $\hat{u}_0$, $\hat{u}_1$, ..., $\hat{u}_3$. The bit vectors $\hat{u}_0$ and $\hat{u}_1$ are both 12 bits long, while the bit vector $\hat{u}_2$ is 11 bits long, and the bit vector $\hat{u}_3$ is 14 bits long. Throughout this clause the convention has been adopted that $\hat{b}(N-1)$, where N is the vector length, represents bit $N-1$ which is the most significant bit (MSB), and $\hat{b}(0)$ represents bit 0 which is the least signficant bit (LSB).

The prioritization of the quantizer values into the set of bit vectors begins with $\hat{u}_0$. The four most significant bits of $\hat{u}_0$ (i.e. bits 11 through 8) are set equal to the four most significant bits of $\hat{b}_0$ (i.e. bits 6 through 3). The next four most significant bits of $\hat{u}_0$ (i.e. bits 7 through 4) are set equal to the four most significant bits of $\hat{b}_1$ (i.e. bits 4 through 1). The remaining four bits of $\hat{u}_0$ (i.e. bits 3 through 0) are generated from the four most significant bits of $\hat{b}_2$ (i.e. bits (4 through 1). The construction of bit vector $\hat{u}_0$ is further depicted in Table 5, where the top row indicates the bit position in $\hat{u}_0$ (bit 11 is the MSB), and the bottom row represents the corresponding quantizer bits.

The prioritzation of the quantizer values continues with bit vector $\hat{u}_1$. The eight most significant bits of $\hat{u}_1$ (i.e. bits 11 through 4) are set equal to the eight most significant bits of $\hat{b}_3$ (i.e. bits 8 through 1). The four least significant bits of $\hat{u}_1$ (i.e. bits 3 through 0) are set equal to the four most significant bits of $\hat{b}_4$ (i.e. bits 6 through 3). The construction of bit vector $\hat{u}_1$ is further depicted in Table 6.

Next bit vector $\hat{u}_2$ in constructed starting with the four most significant bits of $\hat{b}_5$ (i.e. bits 4 through 1), followed by the three most significant bits of $\hat{b}_6$ (i.e. bits 3 through 1), then followed by the three most significant bits of $\hat{b}_7$ (i.e. bits 3 through 1), and finishing with the most significant bit of $\hat{b}_8$ (i.e. bit 2). The construction of bit vector $\hat{u}_2$ is further depicted in Table 7.

Finally, bit vector $\hat{u}_3$ is constructed from the remaining bits starting with the least significant bit of $\hat{b}_1$ (i.e. bit 0), and continuing as shown in Table 8.

Table 7: Construction of Bit Vector $\hat{u}_2$

10 7	6 4	3 1	0
$\hat{b}_5(4,3,2,1)$	$\hat{b}_6(3,2,1)$	$\hat{b}_7(3,2,1)$	$\hat{b}_8(2)$

Table 8: Construction of Bit Vector $\hat{u}_3$

13	12	11 9	8	7 5	4	3	2	1 0
$\hat{b}_1(0)$	$\hat{b}_2(0)$	$\hat{b}_0(2,1,0)$	$\hat{b}_3(0)$	$\hat{b}_4(2,1,0)$	$\hat{b}_5(0)$	$\hat{b}_6(0)$	$\hat{b}_7(0)$	$\hat{b}_8(1,0)$

5.2 Error Control Coding

At 3.6 kbps with a 20 ms frame size, 72 bits per frame are available for transmitting voice over the air interface. The Half-Rate vocoder uses 49 of these bits to quantize the model parameters as previously described. The remaining 23 bits are used for forward error correction. These 23 error control bits are divided between one [24,12] Golay code and one [23,12] Golay code. These two codes are identical except the [24,12] Golay code includes one additional parity bit in the LSB position. The reader is referred to references [4, 6] for more information on the encoding and decoding of Golay codes.

The generation of the four code vectors $\hat{c}_i$ for $0 \leq i \leq 3$ is performed using the following set of equations to map each bit vector, $\hat{u}_i$, to a corresponding code vector, $\hat{c}_i$:

$$\begin{aligned}
\hat{c}_0 &= \hat{u}_0 G_{24,12} \quad (47)\\
\hat{c}_1 &= [\hat{u}_1 G_{23,12}] + \hat{m}_1 \quad (48)\\
\hat{c}_2 &= \hat{u}_2 \quad (49)\\
\hat{c}_3 &= \hat{u}_3 \quad (50)\\
& \quad (51)
\end{aligned}$$

where the $G_{24,12}$ and $G_{23,12}$ are the generator matrices for the extended and regular Golay codes shown below, and the modulation vector $\hat{m}_1$ applied in Equation 48 is described in the following clause. Note all operations are modulo 2. The construction of the code vectors is further depicted in Figure 2.

$$G_{24,12} = \left[\begin{array}{cccccccccccccccccccccccc}
1&0&&&&&&&&&&&1&1&0&0&0&1&1&1&0&1&0&1\\
0&1&0&&&&&&&&&&0&1&1&0&0&0&1&1&1&0&1&1\\
&0&1&0&&&&&&&&&1&1&1&1&0&1&1&0&1&0&0&0\\
&&0&1&0&&&&&&&&0&1&1&1&1&0&1&1&0&1&0&0\\
&&&0&1&0&&&&&&&0&0&1&1&1&1&0&1&1&0&1&0\\
&&&&0&1&0&&&&&&1&1&0&1&1&0&0&1&1&0&0&1\\
&&&&&0&1&0&&&&&0&1&1&0&1&1&0&0&1&1&0&1\\
&&&&&&0&1&0&&&&0&0&1&1&0&1&1&0&0&1&1&1\\
&&&&&&&0&1&0&&&1&1&0&1&1&1&0&0&0&1&1&0\\
&&&&&&&&0&1&0&&1&0&1&0&1&0&0&1&0&1&1&1\\
&&&&&&&&&0&1&0&1&0&0&1&0&0&1&1&1&1&1&0\\
&&&&&&&&&&0&1&1&0&0&0&1&1&1&0&1&0&1&1
\end{array}\right]$$

$$G_{23,12} = \left[\begin{array}{ccccccccccccccccccccccc}
1 & 0 & & & & & & & & & & & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & & & & & & & & & & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
 & 0 & 1 & 0 & & & & & & & & & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & & 0 & 1 & 0 & & & & & & & & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
 & & & 0 & 1 & 0 & & & & & & & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
 & & & & 0 & 1 & 0 & & & & & & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 & & & & & 0 & 1 & 0 & & & & & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
 & & & & & & 0 & 1 & 0 & & & & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 & & & & & & & 0 & 1 & 0 & & & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & & & & & & & & 0 & 1 & 0 & & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
 & & & & & & & & & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & & & & & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1
\end{array}\right]$$

Note that in these generator matrices absent entries are assumed to equal 0, and the input and output vectors are assumed to be row vectors, where the “left” most bit is the MSB. This convention is used throughout this clause.

5.3 Bit Modulation

The Half-Rate vocoder uses a psuedo-random bit pattern, keyed off the bit vector $\hat{u}_0$ to compute a modulation vector $\hat{m}_1$ that is applied in the computation of $\hat{c}_1$ according to Equation 48. This modulation step provides a mechanism for detecting errors in $\hat{u}_0$ beyond the error correction/detection capability of the [24,12] Golay code. Note that the term bit modulation in the context of this document refers to the presented method for multiplying (or modulating) a code vector by a data dependent pseudo-random sequence. The first step in this procedure is to generate the modulation vector $\hat{m}_1$ from a pseudo-random sequence whose seed is derived from $\hat{u}_0$. Specifically, the sequence defined in the following equations is used,

$$p_r(0) = 16\hat{u}_0 \quad (52)$$

$$p_r(n) = 173p_r(n-1) + 13849 - 65536\lfloor\frac{173p_r(n-1)+13849}{65536}\rfloor \quad (53)$$

where the bit vector $\hat{u}_0$ is interpreted as an unsigned 12 bit number in the range [0, 4095]. Equation (53) is then used to recursively compute the pseudo-random sequence, $p_r(n)$, over the range $1 \leq n \leq 23$. Each element of this sequence can be interpreted as a 16 bit random number which is uniformly distributed over the interval [0, 65535]. Using this interpretation, a binary modulation vector $\hat{m}_1$ is generated from this sequence as shown below.

$$\hat{m}_1 = \left[\lfloor\frac{p_r(1)}{32768}\rfloor, \lfloor\frac{p_r(2)}{32768}\rfloor, \ldots, \lfloor\frac{p_r(23)}{32768}\rfloor\right] \quad (54)$$

The modulation vector $\hat{m}_1$ is then added (modulo 2) to the result of the [23,12] Golay code to produce $\hat{c}_1$ as described in Equation 48

One should note that the bit modulation performed by the Half-Rate encoder can be inverted by the decoder if $\tilde{c}_0$ does not contain any uncorrectable bit errors. In this case, Golay decoding $\tilde{c}_0$ will yield the correct value of $\tilde{u}_0$. The decoder can then use $\tilde{u}_0$ to reconstruct the pseudo-random sequence and the modulation vector $\tilde{m}_1$. Subtracting this modulation vector (modulo 2) from $\tilde{c}_1$

and Golay decoding the result will then yield the decoded bit vector $\tilde{u}_1$. The remaining bit vectors are recovered using the relationships $\tilde{u}_2 = \tilde{c}_2$ and $\tilde{u}_3 = \tilde{c}_3$ without any FEC decoding.

One should also note that in the other case, where $\tilde{c}_0$ contains uncorrectable bit errors, the modulation cannot generally be inverted by the decoder. In this case the likely result of Golay decoding $\tilde{c}_0$ will be some $\tilde{u}_0$ which does not equal $\hat{u}_0$. Consequently, the decoder will initialize the pseudo-random sequence incorrectly, and the modulation vector computed by the decoder will be uncorrelated with the modulation vectors used by the encoder. Using these incorrect modulation vectors to reconstruct the code vectors is essentially the same as passing $\tilde{c}_1$ through a 50 percent bit error rate (BER) channel. The Half-Rate decoder exploits the fact that, statistically, a 50 percent BER causes the [23,12] Golay decoder to correct a number of errors which is near the maximum capability of the code (i.e. near 3). By counting the total number of errors which are corrected in all of these code vectors, the decoder is able to detect many frames in which $\tilde{c}_0$ is likely to contain uncorrectable bit errors. The decoder performs frame repeats during these frames in order to reduce the perceived degradation in the presence of bit errors. This is explained more fully in Clauses 5.5 and 5.6.

5.4 Bit Interleaving

Intra-frame bit interleaving is used to spread short bursts of errors among several code vectors (i.e. between $\tilde{u}_0$ and $\tilde{u}_1$). The division of each frame of 72 bits into 36 dibit symbols is shown in Annex H, where Position 0 is the first bit of each voice frame (in time) to be transmitted and received. Note this annex uses the notation $c_j(n)$ to designate the $n'th$ bit of the modulated code vector $\hat{c}_j$ (or the demodulated code vector $\tilde{c}_j$), where bit $N-1$ (assuming a vector length of N) is the MSB of each vector and bit 0 is the LSB.

5.5 Error Estimation

The Half-Rate speech decoder estimates the number of errors in each received data frame by computing the number of errors corrected by the two Golay codes. The number of bit errors that were corrected or detected during FEC decoding of the $i'th$ code is denoted ϵ_i and ranges over $0 \le i \le 1$. From these two error values two other error parameters are computed as shown below.

$$\epsilon_T = \epsilon_0 + \epsilon_1 \quad (55)$$

$$\epsilon_R(0) = 0.95\epsilon_R(-1) + 0.001064\epsilon_T \quad (56)$$

The parameter $\epsilon_R(0)$ is the estimate of the error rate for the current frame, while $\epsilon_R(-1)$ is the estimate of the error rate for the previous frame. Note that $\epsilon_R(-1) = 0$ upon initialization of the decoder. These error parameters are used to control the frame repeat process described below.

5.6 Frame Repeats

The Half-Rate decoder examines each received data frame in order to detect and discard frames which are highly corrupted. A number of different fault conditions are checked and if any of these conditions indicate the current frame is invalid, then a frame repeat is performed.

The Half-Rate speech encoder uses values of $\hat{b}_0$ to denote different frame types as shown in Table 4. If a value of $\tilde{b}_0$ is received in the range $120 \le \hat{b}_0 \le 123$ then the decoder interprets the frame as an erasure frame and performs a frame repeat. Similarly, a frame repeat is performed by the decoder if $\epsilon_0 \geq 4$, or if both of the following two equations are true.

$$\epsilon_0 \geq 2 \tag{57}$$
$$\epsilon_T \geq 6 \tag{58}$$

These two equations serve to detect the incorrect bit demodulation which results if there are uncorrectable bit errors in $\tilde{c}_0$.

The decoder performs a frame repeat by taking the following steps:

1) The current received voice frame is marked as invalid and subsequently ignored during future processing steps.

2) The MBE speech model parameters for the current frame are set equal to the decoded model parameters for the previous frame. Specifically, for voice frames the following update expressions are computed.

$$\tilde{\omega}_0(0) = \tilde{\omega}_0(-1) \tag{59}$$
$$\tilde{L}(0) = \tilde{L}(-1) \tag{60}$$
$$\tilde{K}(0) = \tilde{K}(-1) \tag{61}$$
$$\tilde{v}_l(0) = \tilde{v}_l(-1) \text{ for } 1 \leq l \leq \tilde{L}(-1) \tag{62}$$
$$\tilde{M}_l(0) = \tilde{M}_l(-1) \text{ for } 1 \leq l \leq \tilde{L}(-1) \tag{63}$$
$$\bar{M}_l(0) = \bar{M}_l(-1) \text{ for } 1 \leq l \leq \tilde{L}(-1) \tag{64}$$

3) The repeated model parameters are used in all future processing wherever the current model parameters are required. This includes the synthesis of the current segment of speech as is described in Chapter 11 of [1].

5.7 Frame Muting

The Half-Rate decoder is required to mute in severe bit error environments for which $\epsilon_R > .096$ or if 4 consecutive voice frames are determined to be invalid (muting would occur instead of performing the 4'th consecutive frame repeat). This procedure causes the Half-Rate decoder to squelch its output if reliable communications cannot be supported.

The recommended muting method is to first compute the update equations as listed in step (2) of the frame repeat process (see Clause 5.6). The decoder should then bypass the speech synthesis procedure described in Chapter 11 of [1], and aternately set the synthetic speech signal, $\tilde{s}(n)$ to random noise which is uniformly distributed over the interval [-5,5]. This technique provides for a small amount of "comfort noise".

6 Spectral Amplitude Enhancement

The Half-Rate speech decoder attempts to improve the perceived quality of the synthesized speech by enhancing the spectral amplitudes. These enhanced spectral amplitudes are used throughout speech synthesis. The spectral amplitude enhancement is accomplished using the procedure described in Chapter 8 of [1].

7 Tone Frames

The Half-Rate vocoder supports the transmission of tone frames to allow transmission and regeneration of a tone signal through the vocoder. Supported tones include Dual Tone Multi-Frequency (DTMF) and KNOX tones, as well as certain single-frequency and call progress tones. The supported tone signals are shown in Table 9

In order to pass tones (single or DTMF), the voice encoder must repeatedly detect whether the incoming signal $s(n)$ is either voice or a tone signal. If voice is detected, then the voice signal is encoded and transmitted as specified in the Section 4 of this document. However, if a tone is detected, then a special "Tone Frame" bit sequence is transmitted which specifies the Tone Index I_D from Table 9 and amplitude of the tone which was detected.

Similarly, the voice decoder must check whether the received data for the frame contains a special Tone Frame. This is done by decoding the first six bits or $\tilde{u}_0$. If these 6 bits equal 63 as shown in the first line of Table 9, then the frame is a Tone Frame. Note this is equivalent to decoding the the frame type from quantizer value $\tilde{b}_0$ and checking the frame type is set to "tone" (i.e. $[126, 127]$) as described in Clause 4.1. If the frame type is determined to be a Tone Trame, then the decoder decodes the tone index and amplitude and synthesizes a tone signal with the correct amplitude and frequency as its output.

Note that the format of the bit vectors is different for a Tone Frame compared to a Voice or Silence Frame. Consequently, for Tone Frames, the frame format is determined by Table 10. Table 2 and Tables 5 through Table 8 should be ignored.

The remaining clauses of this document discuss further details of the Half-Rate vocoders operation in the presence of tones. Clause 7.1 addresses tone detection, Clause 7.2 discusses the bit format used to transmit Tone Frames and Clause 7.3 discusses tone regeneration.

7.1 Tone Detection

The Half-Rate Vocoder divides an input signal $s(n)$ into 20 ms segments and then analyzes each segment to extract the MBE model parameters. In order to support the required tone features, a separate tone detection stage must be added to the voice encoder. Note: tone detection is an optional feature and not required for interoperability.

The tone detection algorithm must analyze the input signal every 20 ms and determine whether the input signal for the current frame corresponds to voice or a tone. This is typically performed by analyzing the input signal $s(n)$ every 20 ms. Many standard techniques for tone detection can be found in the literature and the reader is referred elsewhere for more information on this subject. Regardless of the method, the algorithm used to perform tone detection must determine whether the input signal is voice or a supported tone signal during each 20 ms voice frame. If a tone is detected it must identify which tone is present according to the Tone Index shown in Table 9 and estimate the amplitude of the tone. Furthermore, if a single tone is detected it must estimate the center frequency of the tone.

7.2 Tone Transmission

The Half-Rate Vocoder supports transmission of tone information through the use of dedicated Tone Frame, where the bit format of a tone frame is shown in Table 10. Note that the first 6 bits of the bit vector $\hat{u}_0$ is always set equal to 63 for a Tone Frame. This equates to a value of $\hat{b}_0$ in the range $120 \leq \hat{b}_0 \leq 127$

Table 9: Half-Rate Vocoder: Supported Tone Signals

Tone Index I_D	Tone Type	Frequency 1 f_1	Frequency 2 f_2
0 - 4	Invalid	N/A	N/A
5	Single Frequency	156.25	N/A
6	Single Frequency	187.5	N/A
7 - 122	Single Frequency	$31.25 I_D$	N/A
123-127	Invalid	N/A	N/A
128	DTMF "0"	1336	941
129	DTMF "1"	1209	697
130	DTMF "2"	1336	697
131	DTMF "3"	1477	697
132	DTMF "4"	1209	770
133	DTMF "5"	1336	770
134	DTMF "6"	1477	770
135	DTMF "7"	1209	852
136	DTMF "8"	1336	852
137	DTMF "9"	1477	852
138	DTMF "A"	1633	697
139	DTMF "B"	1633	770
140	DTMF "C"	1633	852
141	DTMF "D"	1633	941
142	DTMF "*"	1209	941
143	DTMF "#"	1477	941
144	KNOX "0"	1162	820
145	KNOX "1"	1052	606
146	KNOX "2"	1162	606
147	KNOX "3"	1279	606
148	KNOX "4"	1052	672
149	KNOX "5"	1162	672
150	KNOX "6"	1279	672
151	KNOX "7"	1052	743
152	KNOX "8"	1162	743
153	KNOX "9"	1279	743
154	KNOX "A"	1430	606
155	KNOX "B"	1430	672
156	KNOX "C"	1430	743
157	KNOX "D"	1430	820
158	KNOX "*"	1052	820
159	KNOX "#"	1279	820
160	Call Progress	440	350
161	Call Progress	480	440
162	Call Progress	620	480
163	Call Progress	490	350
164-254	Invalid	N/A	N/A
255	Zero Amplitude	N/A	N/A

Table 10: Tone Frame Format

Bit Vector	Value
$\hat{u}_0(11,10,9,8,7,6)$	63
$\hat{u}_0(5,4,3,2,1,0)$	$A_D(6,5,4,3,2,1)$
$\hat{u}_1(11,10,9,8,7,6,5,4)$	$I_D(7,6,5,4,3,2,1,0)$
$\hat{u}_1(3,2,1,0)$	$I_D(7,6,5,4)$
$\hat{u}_2(10,9,8,7)$	$I_D(3,2,1,0)$
$\hat{u}_2(6,5,4,3,2,1,0)$	$I_D(7,6,5,4,3,2,1)$
$\hat{u}_3(13)$	$I_D(0)$
$\hat{u}_3(12,11,10,9,8,7,6,5)$	$I_D(7,6,5,4,3,2,1,0)$
$\hat{u}_3(4)$	$A_D(0)$
$\hat{u}_3(3,2,1,0)$	0

and forces the two MSB's of $\hat{b}_1$ to 1. The result is that this allows the frame type to be positively identifed as a Tone Frame at the decoder.

The amplitude of the tone is represented through the log amplitude parameter A_D, which is a 7-bit parameter that spans the range $0 \leq A_D \leq 127$. This parameter is scaled such that $A_D = 127$ for a tone at the maximum sinusoidal input level of the A-to-D (+3.17 dBm0), and scaled such that $A_D = 0$ for a tone at -87.13 dBm0. This equates to an amplitude step size of 0.711 dB. The 6 MSBs of A_D are transmitted in the last 6 bits of $\hat{u}_0$, and the LSB of A_D is transmitted in bit 5 of $\hat{u}_3$.

The tone index I_D is repeated 4 times within the bit vectors $\hat{u}_1, \hat{u}_2, \hat{u}_3$ to provide an extra measure of redundancy. The last 4 bits of $\hat{u}_3$ are set to 0 for all Tone Frames.

Tone Frames are transmitted using the same FEC, bit modulation and interleaving as voice frames. Consequently, the encoder and decoder apply the same methods described in Clauses 5.2, 5.3, and 5.4 for all Tone Frames.

7.3 Tone Regeneration

In the event the decoder receives a Tone Frame with a valid tone index, then it must synthesize a tone signal for the current frame using the decoded tone amplitude and frequency parameters. This synthesized tone signal is output by the decoder for the current frame instead of the voice signal described in Chapter 11 of [1]. If the decoder receives a Tone Frame with $I_D = 255$, then the amplitude of the synthesized tone signal is set to zero, regardless of the received value of A_D.

If the decoder receives a Tone Frame with an invalid tone index, then the frame is considered an erasure frame and a frame repeat is performed as indicated in Clause 5.6.

Table 11: Frame Type Conversion

Half-Rate Frame Type	Full-Rate Frame Type
voice	voice
erasure	invalid
silence	voice
tone	voice

8 Parametric Rate Conversion

The Half-Rate vocoder described in this Addendum, and the Full-Rate vocoder described in [1] share the same MBE model parameters for every 20 ms frame. As a result parametric rate conversion can be used to convert between Full-Rate and Half-Rate bit streams.

Parametric rate conversion is accomplished by decoding and reconstructing the MBE model parameters at the input rate and then quantizing and re-encoding these MBE parameters at the output rate. A major advantage of this method of performing rate conversion is that the signal is not converted back to speech as part of the conversion process (i.e. it does not use tandeming). Instead the MBE model parameters are used as the intermediate stage between bit rate, and the synthesis of speech from the MBE model parameters, and the analysis (i.e. extraction) of the model parameters from speech is not performed. This approach avoids the added delay and much of the distortion normally associated with tandeming.

The details of the method are straightforward. The input bit stream is de-interleaved, FEC decoding and the quantizer values $\tilde{b}_n$ are formed for each frame. The quantizer values are then used to reconstruct the MBE model parameters: $\tilde{L}$, $\tilde{\omega}_0$, $\tilde{v}_l$, and $\tilde{M}_l$. These model parameters are then requantized into new quantizer values $\hat{b}_m$ at the output rate. These quantizer values are used to form the new bit vectors, which are FEC encoded and interleaved to produce the output bit stream.

As shown in Table 4, the Half-Rate Vocoder identifies four separate frame types which may be received. In contrast the Full-Rate vocoder only supports voice frames and invalid frames ($\tilde{b}_0 \geq 208$). Conversion of Half-Rate frames to Full-Rate frames works as shown in Table 11. Conversion from Full-Rate frames to Half-Rate frames works in the same manner except only the first two lines of Table 11 are used.

To convert a Half-Rate erasure frame to a Full-Rate invalid frame, the value of $\tilde{b}_0$ in the Full-Rate frame is set to 240 and all other bit vectors can be set to zero since they are ignored at the receiver. Similarly to convert a Full-Rate invalid frame to a Half-Rate erasure frame, the value of $\tilde{b}_0$ in the Half-Rate frame is set to 120 and all other bit vectors can be set to zero.

To convert a Half-Rate silence frame to a Full-Rate voice frame, the reconstructed the MBE model parameters: $\tilde{L}$, $\tilde{\omega}_0$, $\tilde{v}_l$, and $\tilde{M}_l$ are just quantized using the Full-Rate quantizer in the same manner as if the frame was a voice frame.

To convert a Half-Rate tone frame to a Full-Rate voice frame, an approximate set of MBE model parameters is first generated for the Tone Frame to provide a reasonable fit between the frequency components of the tone and the generated MBE spectral parameters. Annex J lists the tone frame parameters, f_0, l_1, and l_2 corresponding to each possible value of the received tone index I_D. Using the value of f_0, the MBE parameters $\tilde{\omega}_0$ and $\tilde{L}$ are generated as follows:

$$\tilde{\omega}_0 = \frac{2\pi}{8000} f_0 \tag{65}$$

$$\tilde{L} = \lfloor \frac{3812.5}{f_0} \rfloor \tag{66}$$

The tone frame parameters l_1 and l_2 from Annex J are then used to determine the voicing decisions $\tilde{v}_l$ and and spectral amplitudes $\tilde{M}_l$ according to Equations 67 and 68, respectively.

$$\tilde{v}_l = \begin{cases} 1 & \text{if } l = l_1 \text{ or } l = l_2 \\ 0 & \text{otherwise} \end{cases} \tag{67}$$

$$\tilde{M}_l = \begin{cases} 16384 \cdot 10^{[0.03555(A_D - 127)]} & \text{if } l = l_1 \text{ or } l = l_2 \\ 0 & \text{otherwise} \end{cases} \tag{68}$$

Once the approximate set of MBE model parameters have been generated in this manner, then the frame is quantized using the Full-Rate quantizer in the same manner as if the frame was a voice frame.

Annex A (Normative) Fundamental Frequency Quantization Table

b_0	L	ω_0	b_0	L	ω_0	b_0	L	ω_0
0	9	0.049971	40	17	0.027122	80	31	0.014721
1	9	0.049215	41	17	0.026712	81	31	0.014496
2	9	0.048471	42	17	0.026304	82	32	0.014277
3	9	0.047739	43	17	0.025906	83	32	0.014061
4	9	0.047010	44	18	0.025515	84	33	0.013847
5	9	0.046299	45	18	0.025129	85	33	0.013636
6	10	0.045601	46	18	0.024746	86	34	0.013430
7	10	0.044905	47	18	0.024372	87	34	0.013227
8	10	0.044226	48	19	0.024002	88	35	0.013025
9	10	0.043558	49	19	0.023636	89	36	0.012829
10	10	0.042900	50	19	0.023279	90	36	0.012634
11	10	0.042246	51	20	0.022926	91	37	0.012444
12	11	0.041609	52	20	0.022581	92	37	0.012253
13	11	0.040979	53	20	0.022236	93	38	0.012068
14	11	0.040356	54	21	0.021900	94	38	0.011887
15	11	0.039747	55	21	0.021570	95	39	0.011703
16	11	0.039148	56	21	0.021240	96	40	0.011528
17	11	0.038559	57	22	0.020920	97	40	0.011353
18	12	0.037971	58	22	0.020605	98	41	0.011183
19	12	0.037399	59	22	0.020294	99	42	0.011011
20	12	0.036839	60	23	0.019983	100	42	0.010845
21	12	0.036278	61	23	0.019684	101	43	0.010681
22	12	0.035732	62	23	0.019386	102	43	0.010517
23	13	0.035198	63	24	0.019094	103	44	0.010359
24	13	0.034672	64	24	0.018805	104	45	0.010202
25	13	0.034145	65	24	0.018520	105	46	0.010050
26	13	0.033636	66	25	0.018242	106	46	0.009895
27	13	0.033133	67	25	0.017965	107	47	0.009747
28	14	0.032635	68	26	0.017696	108	48	0.009600
29	14	0.032148	69	26	0.017431	109	48	0.009453
30	14	0.031670	70	26	0.017170	110	49	0.009312
31	14	0.031122	71	27	0.016911	111	50	0.009172
32	15	0.030647	72	27	0.016657	112	51	0.009033
33	15	0.030184	73	28	0.016409	113	52	0.008896
34	15	0.029728	74	28	0.016163	114	52	0.008762
35	15	0.029272	75	29	0.015923	115	53	0.008633
36	16	0.028831	76	29	0.015686	116	54	0.008501
37	16	0.028395	77	30	0.015411	117	55	0.008375
38	16	0.027966	78	30	0.015177	118	56	0.008249
39	16	0.027538	79	30	0.014946	119	56	0.008125

Annex B (Normative) V/UV Quantization Vectors

b_1	ν_0	ν_1	ν_2	ν_3	ν_4	ν_5	ν_6	ν_7
0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	0
5	1	1	0	1	1	1	1	1
6	1	1	1	0	1	1	1	1
7	1	1	1	1	1	0	1	1
8	1	1	1	1	0	0	0	0
9	1	1	1	1	1	0	0	0
10	1	1	1	0	0	0	0	0
11	1	1	1	0	0	0	0	1
12	1	1	0	0	0	0	0	0
13	1	1	1	0	0	0	0	0
14	1	0	0	0	0	0	0	0
15	1	1	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0

Annex C (Normative) Log Magnitude Prediction Residual Block Lengths

L	J_1	J_2	J_3	J_4
9	2	2	2	3
10	2	2	3	3
11	2	3	3	3
12	2	3	3	4
13	3	3	3	4
14	3	3	4	4
15	3	3	4	5
16	3	4	4	5
17	3	4	5	5
18	4	4	5	5
19	4	4	5	6
20	4	4	6	6
21	4	5	6	6
22	4	5	6	7
23	5	5	6	7
24	5	5	7	7
25	5	6	7	7
26	5	6	7	8
27	5	6	8	8
28	6	6	8	8
29	6	6	8	9
30	6	7	8	9
31	6	7	9	9
32	6	7	9	10
33	7	7	9	10
34	7	8	9	10
35	7	8	10	10
36	7	8	10	11
37	8	8	10	11
38	8	9	10	11
39	8	9	11	11
40	8	9	11	12
41	8	9	11	13
42	8	9	12	13
43	8	10	12	13
44	9	10	12	13
45	9	10	12	14
46	9	10	13	14
47	9	11	13	14
48	10	11	13	14
49	10	11	13	15
50	10	11	14	15
51	10	12	14	15
52	10	12	14	16
53	11	12	14	16
54	11	12	15	16
55	11	12	15	17
56	11	13	15	17

Annex D (Normative) Gain Quantizer Levels

b_2	Δ_γ
0	-2.00000
1	-0.67000
2	0.297941
3	0.663728
4	1.036829
5	1.438136
6	1.890077
7	2.227970
8	2.478289
9	2.667544
10	2.793619
11	2.893261
12	3.020630
13	3.138586
14	3.237579
15	3.322570
16	3.432367
17	3.571863
18	3.696650
19	3.814917
20	3.920932
21	4.022503
22	4.123569
23	4.228291
24	4.370569
25	4.543700
26	4.707695
27	4.848879
28	5.056757
29	5.326468
30	5.777581
31	6.874496

Annex E (Normative) PRBA24 Vector Quantizer Levels

Quantization Vectors for b_3

b_3	G_2	G_3	G_4	b_3	G_2	G_3	G_4
0	0.526055	−0.328567	−0.304727	50	0.286428	−0.210542	−0.029587
1	0.441044	−0.303127	−0.201114	51	0.257656	−0.261837	−0.056566
2	1.030896	−0.324730	−0.397204	52	−0.235852	−0.310760	−0.165147
3	0.839696	−0.351933	−0.224909	53	−0.334949	−0.385870	−0.197362
4	0.272958	−0.176118	−0.098893	54	0.094870	−0.241144	0.059122
5	0.221466	−0.160045	−0.061026	55	0.060177	−0.225884	0.031140
6	0.496555	−0.211499	0.047305	56	−0.301184	−0.306545	−0.446189
7	0.424376	−0.223752	0.069911	57	−0.293528	−0.504146	−0.429844
8	0.264531	−0.353355	−0.330505	58	−0.055084	−0.379015	−0.125887
9	0.273650	−0.253004	−0.250241	59	−0.115434	−0.375008	−0.059939
10	0.484531	−0.297627	−0.071051	60	−0.777425	−0.592163	−0.107585
11	0.410814	−0.224961	−0.084998	61	−0.950500	−0.893847	−0.181762
12	0.039519	−0.252904	−0.115128	62	−0.259402	−0.396726	0.010357
13	0.017423	−0.296519	−0.045921	63	−0.368905	−0.449026	0.038299
14	0.225113	−0.224371	0.037882	64	0.279719	−0.063196	−0.184628
15	0.183424	−0.260492	0.050491	65	0.255265	−0.067248	−0.121124
16	0.308704	−0.073205	−0.405880	66	0.458433	−0.103777	0.010074
17	0.213125	−0.101632	−0.333208	67	0.437231	−0.092496	−0.031028
18	0.617735	−0.137299	−0.213670	68	0.082265	−0.028050	−0.041262
19	0.514382	−0.126485	−0.170204	69	0.045920	−0.051719	−0.030155
20	0.130009	−0.076955	−0.229303	70	0.271149	−0.043613	0.112085
21	0.061740	−0.108259	−0.203887	71	0.246881	−0.065274	0.105436
22	0.244473	−0.110094	−0.051689	72	0.056590	−0.117773	−0.142283
23	0.230452	−0.076147	−0.028190	73	0.058824	−0.104418	−0.099608
24	0.059837	−0.254595	−0.562704	74	0.213781	−0.111974	0.031269
25	0.011630	−0.135223	−0.432791	75	0.187554	−0.070340	0.011834
26	0.207077	−0.152248	−0.148391	76	−0.185701	−0.081106	−0.073803
27	0.158078	−0.128800	−0.122150	77	−0.266112	−0.074133	−0.085370
28	−0.265982	−0.144742	−0.199894	78	−0.029368	−0.046490	0.124679
29	−0.356479	−0.204740	−0.156465	79	−0.017378	−0.102882	0.140482
30	0.000324	−0.139549	−0.066471	80	0.114700	0.092738	−0.244271
31	0.001888	−0.170557	−0.025025	81	0.072922	0.007863	−0.231476
32	0.402913	−0.581478	−0.274626	82	0.270022	0.031819	−0.094208
33	0.191289	−0.540335	−0.193040	83	0.254403	0.024805	−0.050389
34	0.632914	−0.401410	−0.006636	84	−0.182905	0.021629	−0.168481
35	0.471086	−0.463144	0.061489	85	−0.225864	−0.010109	−0.130374
36	0.044829	−0.438487	0.033433	86	0.040089	0.013969	0.016028
37	0.015513	−0.539475	−0.006719	87	0.001442	0.010551	0.032942
38	0.336218	−0.351311	0.214087	88	−0.287472	−0.036130	−0.296798
39	0.239967	−0.380836	0.157681	89	−0.332344	−0.108862	−0.342196
40	0.347609	−0.901619	−0.688432	90	0.012700	0.022917	−0.052501
41	0.064067	−0.826753	−0.492089	91	−0.040681	−0.001805	−0.050548
42	0.303089	−0.396757	−0.108446	92	−0.718522	−0.061234	−0.278820
43	0.235590	−0.446122	0.006437	93	−0.879205	−0.213588	−0.303508
44	−0.236964	−0.652532	−0.135520	94	−0.234102	−0.065407	0.013686
45	−0.418285	−0.793014	−0.034730	95	−0.281223	−0.076139	0.046830
46	−0.038262	−0.516984	0.273681	96	0.141967	−0.193679	−0.055697
47	−0.037419	−0.958198	0.214749	97	0.100318	−0.161222	−0.063062
48	0.061624	−0.238233	−0.237184	98	0.265859	−0.132747	0.078209
49	−0.013944	−0.235704	−0.204811	99	0.244805	−0.139776	0.122123

Quantization Vectors for b_3 (continued)

b_3	G_2	G_3	G_4	b_3	G_2	G_3	G_4
100	−0.121802	−0.179976	0.031732	150	0.359915	0.101273	−0.052997
101	−0.185318	−0.214011	0.018117	151	0.318117	0.125888	−0.003486
102	0.047014	−0.153961	0.218068	152	0.150452	0.050219	−0.409155
103	0.047305	−0.187402	0.282114	153	0.188753	0.091894	−0.325733
104	−0.027533	−0.415868	−0.333841	154	0.334922	0.029098	−0.098587
105	−0.125886	−0.334492	−0.290317	155	0.324508	0.015809	−0.135408
106	−0.030602	−0.190918	0.097454	156	−0.042506	0.038667	−0.208535
107	−0.054936	−0.209948	0.158977	157	−0.083003	0.094758	−0.174054
108	−0.507223	−0.295876	−0.217183	158	0.094773	0.102653	−0.025701
109	−0.581733	−0.403194	−0.208936	159	0.063284	0.118703	−0.000071
110	−0.299719	−0.289679	0.297101	160	0.355965	−0.139239	−0.191705
111	−0.363169	−0.362718	0.436529	161	0.392742	−0.105496	−0.132103
112	−0.124627	−0.042100	−0.157011	162	0.663678	−0.204627	−0.031242
113	−0.161571	−0.092846	−0.183636	163	0.609381	−0.146914	0.079610
114	0.084520	−0.100217	−0.000901	164	0.151855	−0.132843	−0.007125
115	0.055655	−0.136381	0.032764	165	0.146404	−0.161917	0.024842
116	−0.545087	−0.197713	−0.026888	166	0.400524	−0.135221	0.232289
117	−0.662772	−0.179815	0.026419	167	0.324931	−0.116605	0.253458
118	−0.165583	−0.148913	0.090382	168	0.169066	−0.215132	−0.185604
119	−0.240772	−0.182830	0.105474	169	0.128681	−0.189394	−0.160279
120	−0.576315	−0.359473	−0.456844	170	0.356194	−0.116992	−0.038381
121	−0.713430	−0.554156	−0.476739	171	0.342866	−0.144687	0.020265
122	−0.275628	−0.223640	−0.051584	172	−0.065545	−0.202593	−0.043688
123	−0.359501	−0.230758	−0.027006	173	−0.124296	−0.260225	−0.035370
124	−1.282559	−0.284807	−0.233743	174	0.083224	−0.235149	0.153301
125	−1.060476	−0.399911	−0.562698	175	0.046256	−0.309608	0.190944
126	−0.871952	−0.272197	0.016126	176	0.187385	−0.008168	−0.198575
127	−0.747922	−0.329404	0.276696	177	0.190401	−0.018699	−0.136858
128	0.643086	0.046175	−0.660078	178	0.398009	−0.025700	−0.007458
129	0.738204	−0.127844	−0.433708	179	0.346948	−0.022258	−0.020905
130	1.158072	0.025571	−0.177856	180	−0.047064	−0.085629	−0.080677
131	0.974840	−0.009417	−0.112337	181	−0.067523	−0.128972	−0.119538
132	0.418014	0.032741	−0.124545	182	0.186086	−0.016828	0.070014
133	0.381422	−0.001557	−0.085504	183	0.187364	0.017133	0.075949
134	0.768280	0.056085	0.095375	184	−0.112669	−0.037433	−0.298944
135	0.680004	0.052035	0.152318	185	−0.068276	−0.114504	−0.265795
136	0.473182	0.012560	−0.264221	186	0.147510	−0.040616	−0.013687
137	0.345153	0.036627	−0.248756	187	0.133084	−0.062849	−0.032637
138	0.746238	−0.025880	−0.106050	188	−0.416571	−0.041544	−0.125088
139	0.644319	−0.058256	−0.095133	189	−0.505337	−0.044193	−0.157651
140	0.185924	−0.022230	−0.070540	190	−0.154132	−0.075106	0.050466
141	0.146068	−0.009550	−0.057871	191	−0.148036	−0.059719	0.121516
142	0.338488	0.013022	0.069961	192	0.490555	0.157659	−0.222208
143	0.298969	0.047403	0.052598	193	0.436700	0.120500	−0.205869
144	0.346002	0.256253	−0.380261	194	0.754525	0.269323	0.045810
145	0.313092	0.163821	−0.314004	195	0.645077	0.271923	0.013942
146	0.719154	0.103108	−0.252648	196	0.237023	0.115337	−0.026429
147	0.621429	0.172423	−0.265180	197	0.204895	0.121020	−0.008541
148	0.240461	0.104684	−0.202582	198	0.383999	0.153963	0.171763
149	0.206946	0.139642	−0.138016	199	0.385026	0.222074	0.239731

Quantization Vectors for b_3 (continued)

b_3	G_2	G_3	G_4
200	0.198232	0.072972	−0.108179
201	0.147882	0.074743	−0.123341
202	0.390929	0.075205	0.081828
203	0.341623	0.089405	0.069389
204	−0.003381	0.159694	−0.016026
205	−0.043653	0.206860	−0.040729
206	0.135515	0.107824	0.179310
207	0.081086	0.119673	0.174282
208	0.192637	0.400335	−0.341906
209	0.171196	0.284921	−0.221516
210	0.377807	0.359087	−0.151523
211	0.411052	0.297925	−0.099774
212	−0.010060	0.261887	−0.149567
213	−0.107877	0.287756	−0.116982
214	0.158003	0.209727	0.077988
215	0.109710	0.232272	0.088135
216	0.000698	0.209353	−0.395208
217	−0.094015	0.230322	−0.279928
218	0.137355	0.230881	−0.124115
219	0.103058	0.166855	−0.100386
220	−0.305058	0.305422	−0.176026
221	−0.422049	0.337137	−0.293297
222	−0.121744	0.185124	0.048115
223	−0.171052	0.200312	0.052812
224	0.224091	−0.010673	−0.019727
225	0.200266	−0.020167	0.001798
226	0.382742	0.032362	0.161665
227	0.345631	−0.019705	0.164451
228	0.029431	0.045010	0.071518
229	0.031940	0.010876	0.087037
230	0.181935	0.039112	0.202316
231	0.181810	0.033189	0.253435
232	−0.008677	−0.066679	−0.144737
233	−0.021768	−0.021288	−0.125903
234	0.136766	0.000100	0.059449
235	0.135405	−0.020446	0.103793
236	−0.289115	0.039747	−0.012256
237	−0.338683	0.025909	−0.034058
238	−0.016515	0.048584	0.197981
239	−0.046790	0.011816	0.199964
240	0.094214	0.127422	−0.169936
241	0.048279	0.096189	−0.148153
242	0.217391	0.081732	0.013677
243	0.179656	0.084671	0.031434
244	−0.227367	0.118176	−0.039803
245	−0.327096	0.159747	−0.018931
246	0.000834	0.113118	0.125325
247	−0.014617	0.128924	0.163776
248	−0.254570	0.154329	−0.232018
249	−0.353068	0.124341	−0.174409
250	−0.061004	0.107744	0.037257
251	−0.100991	0.080302	0.062701
252	−0.927022	0.285660	−0.240549
253	−1.153224	0.277232	−0.322538
254	−0.569012	0.108135	0.172634
255	−0.555273	0.131461	0.325930
256	0.518847	0.065683	−0.132877
257	0.501324	−0.006585	−0.094884
258	1.066190	−0.150380	0.201791
259	0.858377	−0.166415	0.081686
260	0.320584	−0.031499	0.039534
261	0.311442	−0.075120	0.026013
262	0.625829	−0.019856	0.346041
263	0.525271	−0.003948	0.284868
264	0.312594	−0.075673	−0.066642
265	0.295732	−0.057895	−0.042207
266	0.550446	−0.029110	0.046850
267	0.465467	−0.068987	0.096167
268	0.122669	−0.051786	0.044283
269	0.079669	−0.044145	0.045805
270	0.238778	−0.031835	0.171694
271	0.200734	−0.072619	0.178726
272	0.342512	0.131270	−0.163021
273	0.294028	0.111759	−0.125793
274	0.589523	0.121808	−0.049372
275	0.550506	0.132318	0.017485
276	0.164280	0.047560	−0.058383
277	0.120110	0.049242	−0.052403
278	0.269181	0.035000	0.103494
279	0.297466	0.038517	0.139289
280	0.094549	−0.030880	−0.153376
281	0.080363	0.024359	−0.127578
282	0.281351	0.055178	0.000155
283	0.234900	0.039477	0.013957
284	−0.118161	0.011976	−0.034270
285	−0.157654	0.027765	−0.005010
286	0.102631	0.027283	0.099723
287	0.077285	0.052532	0.115583
288	0.329398	−0.278552	0.016316
289	0.305993	−0.267896	0.094952
290	0.775270	−0.394995	0.290748
291	0.583180	−0.252159	0.285391
292	0.192226	−0.182242	0.126859
293	0.185908	−0.245779	0.159940
294	0.346293	−0.250404	0.355682
295	0.354160	−0.364521	0.472337
296	0.134942	−0.313666	−0.115181
297	0.126077	−0.286568	−0.039927
298	0.405618	−0.211792	0.199095
299	0.312099	−0.213642	0.190972

Quantization Vectors for b_3 (continued)

b_3	G_2	G_3	G_4
300	−0.071392	−0.297366	0.081426
301	−0.165839	−0.301986	0.160640
302	0.147808	−0.290712	0.298198
303	0.063302	−0.310149	0.396302
304	0.141444	−0.081377	−0.076621
305	0.115936	−0.104440	−0.039885
306	0.367023	−0.087281	0.096390
307	0.330038	−0.117958	0.127050
308	0.002897	−0.062454	0.025151
309	−0.052404	−0.082200	0.041975
310	0.181553	−0.137004	0.230489
311	0.140768	−0.094604	0.265928
312	−0.101763	−0.209566	−0.135964
313	−0.159056	−0.191005	−0.095509
314	0.045016	−0.081562	0.075942
315	0.016808	−0.112482	0.068593
316	−0.408578	−0.132377	0.079163
317	−0.431534	−0.214646	0.157714
318	−0.096931	−0.101938	0.200304
319	−0.167867	−0.114851	0.262964
320	0.393882	0.086002	0.008961
321	0.338747	0.048405	−0.004187
322	0.877844	0.374373	0.171008
323	0.740790	0.324525	0.242248
324	0.200218	0.070150	0.085891
325	0.171760	0.090531	0.102579
326	0.314263	0.126417	0.322833
327	0.313523	0.065445	0.403855
328	0.164261	0.057745	−0.005490
329	0.122141	0.024122	0.009190
330	0.308248	0.078401	0.180577
331	0.251222	0.073868	0.160457
332	−0.047526	0.023725	0.086336
333	−0.091643	0.005539	0.093179
334	0.079339	0.044135	0.206697
335	0.104213	0.011277	0.240060
336	0.226607	0.186234	−0.056881
337	0.173281	0.158131	−0.059413
338	0.339400	0.214501	0.052905
339	0.309166	0.188181	0.058028
340	0.014442	0.194715	0.048945
341	−0.028793	0.194766	0.089078
342	0.069564	0.206743	0.193568
343	0.091532	0.202786	0.269680
344	−0.071196	0.135604	−0.103744
345	−0.118288	0.152837	−0.060151
346	0.146856	0.143174	0.061789
347	0.104379	0.143672	0.056797
348	−0.541832	0.250034	−0.017602
349	−0.641583	0.278411	−0.111909
350	−0.094447	0.159393	0.164848
351	−0.113612	0.120702	0.221656
352	0.204918	−0.078894	0.075524
353	0.161232	−0.090256	0.088701
354	0.378460	−0.033687	0.309964
355	0.311701	−0.049984	0.316881
356	0.019311	−0.050048	0.212387
357	0.002473	−0.062855	0.278462
358	0.151448	−0.090652	0.410031
359	0.162778	−0.071291	0.531252
360	−0.083704	−0.076839	−0.020798
361	−0.092832	−0.043492	0.029202
362	0.136844	−0.077791	0.186493
363	0.089536	−0.086826	0.184711
364	−0.270255	−0.058858	0.173048
365	−0.350416	−0.009219	0.273260
366	−0.105248	−0.205534	0.425159
367	−0.135030	−0.197464	0.623550
368	−0.051717	0.069756	−0.043829
369	−0.081050	0.056947	−0.000205
370	0.190388	0.016366	0.145922
371	0.142662	0.002575	0.159182
372	−0.352890	0.011117	0.091040
373	−0.367374	0.056547	0.147209
374	−0.003179	0.026570	0.282541
375	−0.069934	−0.005171	0.337678
376	−0.496181	0.026464	0.019432
377	−0.690384	0.069313	−0.004175
378	−0.146138	0.046372	0.161839
379	−0.197581	0.034093	0.241003
380	−0.989567	0.040993	0.049384
381	−1.151075	0.210556	0.237374
382	−0.335366	−0.058208	0.480168
383	−0.502419	−0.093761	0.675240
384	0.862548	0.264137	−0.294905
385	0.782668	0.251324	−0.122108
386	1.597797	0.463818	−0.133153
387	1.615756	0.060653	0.084764
388	0.435588	0.209832	0.095050
389	0.431013	0.165328	0.047909
390	1.248164	0.265923	0.488086
391	1.009933	0.345440	0.473702
392	0.477017	0.194237	−0.058012
393	0.401362	0.186915	−0.054137
394	1.202158	0.284782	−0.066531
395	1.064907	0.203766	0.046383
396	0.255848	0.133398	0.046049
397	0.218680	0.128833	0.065326
398	0.490817	0.182041	0.286583
399	0.440714	0.106576	0.301120

Quantization Vectors for b_3 (continued)

b_3	G_2	G_3	G_4	b_3	G_2	G_3	G_4
400	0.604263	0.522925	−0.238629	450	1.259194	0.901494	0.256085
401	0.526329	0.377577	−0.198100	451	1.296139	0.607949	0.302184
402	1.038632	0.606242	−0.121253	452	0.319619	0.307231	0.099647
403	0.995283	0.552202	0.110700	453	0.287232	0.359355	0.186844
404	0.262232	0.313664	−0.086909	454	0.751306	0.676688	0.499386
405	0.230835	0.273385	−0.054268	455	0.479609	0.553030	0.560447
406	0.548466	0.490721	0.278201	456	0.276377	0.214032	−0.003661
407	0.466984	0.355859	0.289160	457	0.238146	0.223595	0.028806
408	0.367137	0.236160	−0.228114	458	0.542688	0.266205	0.171393
409	0.309359	0.233843	−0.171325	459	0.460188	0.283979	0.158288
410	0.465268	0.276569	0.010951	460	0.057385	0.309853	0.144517
411	0.378124	0.250237	0.011131	461	−0.006881	0.348152	0.097310
412	0.061885	0.296810	−0.011420	462	0.244434	0.247298	0.322601
413	0.000125	0.350029	−0.011277	463	0.253992	0.335420	0.402241
414	0.163815	0.261191	0.175863	464	0.354006	0.579776	−0.130176
415	0.165132	0.308797	0.227800	465	0.267043	0.461976	−0.058178
416	0.461418	0.052075	−0.016543	466	0.534049	0.626549	0.046747
417	0.472372	0.046962	0.045746	467	0.441835	0.468260	0.057556
418	0.856406	0.136415	0.245074	468	0.110477	0.628795	0.102950
419	0.834616	0.003254	0.372643	469	0.031409	0.489068	0.090605
420	0.337869	0.036994	0.232513	470	0.229564	0.525640	0.325454
421	0.267414	0.027593	0.252779	471	0.105570	0.582151	0.509738
422	0.584983	0.113046	0.583119	472	0.005690	0.521474	−0.157885
423	0.475406	−0.024234	0.655070	473	0.104463	0.424022	−0.080647
424	0.264823	−0.029292	0.004270	474	0.223784	0.389860	0.060904
425	0.246071	−0.019109	0.030048	475	0.159806	0.340571	0.062061
426	0.477401	0.021039	0.155448	476	−0.173976	0.573425	0.027383
427	0.458453	−0.043959	0.187850	477	−0.376008	0.587868	0.133042
428	0.067059	−0.061227	0.126904	478	−0.051773	0.348339	0.231923
429	0.044608	−0.034575	0.150205	479	−0.122571	0.473049	0.251159
430	0.191304	−0.003810	0.316776	480	0.324321	0.148510	0.116006
431	0.153078	0.029915	0.361303	481	0.282263	0.121730	0.114016
432	0.320704	0.178950	−0.088835	482	0.690108	0.256346	0.418128
433	0.300866	0.137645	−0.056893	483	0.542523	0.294427	0.461973
434	0.553442	0.162339	0.131987	484	0.056944	0.107667	0.281797
435	0.490083	0.123682	0.146163	485	0.027844	0.106858	0.355071
436	0.118950	0.083109	0.034052	486	0.160456	0.177656	0.528819
437	0.099344	0.066212	0.054329	487	0.227537	0.177976	0.689465
438	0.228325	0.122445	0.309219	488	0.111585	0.097896	0.109244
439	0.172093	0.135754	0.323361	489	0.083994	0.133245	0.115789
440	0.064213	0.063405	−0.058243	490	0.208740	0.142084	0.208953
441	0.011906	0.088795	−0.069678	491	0.156072	0.143303	0.231368
442	0.194232	0.129185	0.125708	492	−0.185830	0.214347	0.309774
443	0.155182	0.174013	0.144099	493	−0.311053	0.240517	0.328512
444	−0.217068	0.112731	0.093497	494	−0.041749	0.090901	0.511373
445	−0.307590	0.171146	0.110735	495	−0.156164	0.098486	0.478020
446	−0.014897	0.138094	0.232455	496	0.151543	0.263073	−0.033471
447	−0.036936	0.170135	0.279166	497	0.126322	0.213004	−0.007014
448	0.681886	0.437121	0.078458	498	0.245313	0.217564	0.120210
449	0.548559	0.376914	0.092485	499	0.259136	0.225542	0.176601

Quantization Vectors for b_3 (concluded)

b_3	G_2	G_3	G_4
500	-0.190632	0.260214	0.141755
501	-0.189271	0.331768	0.170606
502	0.054763	0.294766	0.357775
503	-0.033724	0.257645	0.365069
504	-0.184971	0.396532	0.057728
505	-0.293313	0.400259	0.001123
506	-0.015219	0.232287	0.177913
507	-0.022524	0.244724	0.240753
508	-0.520342	0.347950	0.249265
509	-0.671997	0.410782	0.153434
510	-0.253089	0.412356	0.489854
511	-0.410922	0.562454	0.543891

Annex F (Normative) PRBA58 Vector Quantizer Levels

Quantization Vectors for b_4

b_4	G_5	G_6	G_7	G_8
0	−0.103660	0.094597	−0.013149	0.081501
1	−0.170709	0.129958	−0.057316	0.112324
2	−0.095113	0.080892	−0.027554	0.003371
3	−0.154153	0.113437	−0.074522	0.003446
4	−0.109553	0.153519	0.006858	0.040930
5	−0.181931	0.217882	−0.019042	0.040049
6	−0.096246	0.144191	−0.024147	−0.035120
7	−0.174811	0.193357	−0.054261	−0.071700
8	−0.183241	−0.052840	0.117923	0.030960
9	−0.242634	0.009075	0.098007	0.091643
10	−0.143847	−0.028529	0.040171	−0.002812
11	−0.198809	0.006990	0.020668	0.026641
12	−0.233172	−0.028793	0.140130	−0.071927
13	−0.309313	0.056873	0.108262	−0.018930
14	−0.172782	−0.002037	0.048755	−0.087065
15	−0.242901	0.036076	0.015064	−0.064366
16	0.077107	0.172685	0.159939	0.097456
17	0.024820	0.209676	0.087347	0.105204
18	0.085113	0.151639	0.084272	0.022747
19	0.047975	0.196695	0.038770	0.029953
20	0.113925	0.236813	0.176121	0.016635
21	0.009708	0.267969	0.127660	0.015872
22	0.114044	0.202311	0.096892	−0.043071
23	0.047219	0.260395	0.050952	−0.046996
24	−0.055095	0.034041	0.200464	0.039050
25	−0.061582	0.069566	0.113048	0.027511
26	−0.025469	0.040440	0.132777	−0.039098
27	−0.031388	0.064010	0.067559	−0.017117
28	−0.074386	0.086579	0.228232	−0.055461
29	−0.107352	0.120874	0.137364	−0.030252
30	−0.036897	0.089972	0.155831	−0.128475
31	−0.059070	0.097879	0.084489	−0.075821
32	−0.050865	−0.025167	−0.086636	0.011256
33	−0.051426	0.013301	−0.144665	0.038541
34	−0.073831	−0.028917	−0.142416	−0.025268
35	−0.083910	0.015004	−0.227113	−0.002808
36	−0.030840	−0.009326	−0.070517	−0.041304
37	−0.022018	0.029381	−0.124961	−0.031624
38	−0.064222	−0.014640	−0.108798	−0.092342
39	−0.038801	0.038133	−0.188992	−0.094221
40	−0.154059	−0.183932	−0.019894	0.082105
41	−0.188022	−0.113072	−0.117380	0.090911
42	−0.243301	−0.207086	−0.053735	−0.001975
43	−0.275931	−0.121035	−0.161261	0.004231
44	−0.118142	−0.157537	−0.036594	−0.008679
45	−0.153627	−0.111372	−0.103095	−0.009460
46	−0.173458	−0.180158	−0.057130	−0.103198
47	−0.208509	−0.127679	−0.149336	−0.109289
48	0.096310	0.047927	−0.024094	−0.057018
49	0.044289	0.075486	−0.008505	−0.067635

Quantization Vectors for b_4 (continued)

b_4	G_5	G_6	G_7	G_8
50	0.076751	0.025560	−0.066428	−0.102991
51	0.025215	0.090417	−0.058616	−0.114284
52	0.125980	0.070078	0.016282	−0.112355
53	0.070859	0.118988	0.001180	−0.116359
54	0.097520	0.059219	−0.026821	−0.172850
55	0.048226	0.145459	−0.050093	−0.188853
56	0.007242	−0.135796	0.147832	−0.034080
57	0.012843	−0.069616	0.077139	−0.047909
58	−0.050911	−0.116323	0.082521	−0.056362
59	−0.039630	−0.055678	0.036066	−0.067992
60	0.042694	−0.091527	0.150940	−0.124225
61	0.029225	−0.039401	0.071664	−0.113665
62	−0.025085	−0.099013	0.074622	−0.138674
63	−0.031220	−0.035717	0.020870	−0.143376
64	0.040638	0.087903	−0.049500	0.094607
65	0.026860	0.125924	−0.103449	0.140882
66	0.075166	0.110186	−0.115173	0.067330
67	0.036642	0.163193	−0.188762	0.103724
68	0.028179	0.095124	−0.053258	0.028900
69	0.002307	0.148211	−0.096037	0.046189
70	0.072227	0.137595	−0.095629	0.001339
71	0.033308	0.221480	−0.152201	0.012125
72	0.003458	−0.085112	0.041850	0.113836
73	−0.040610	−0.044880	0.029732	0.177011
74	0.011404	−0.054324	−0.012426	0.077815
75	−0.042413	−0.030930	−0.034844	0.122946
76	−0.002206	−0.045698	0.050651	0.054886
77	−0.041729	−0.016110	0.048005	0.102125
78	0.013963	−0.022204	0.001613	0.028997
79	−0.030218	−0.002052	−0.004365	0.065343
80	0.299049	0.046260	0.076320	0.070784
81	0.250160	0.098440	0.012590	0.137479
82	0.254170	0.095310	0.018749	0.004288
83	0.218892	0.145554	−0.035161	0.069784
84	0.303486	0.101424	0.135996	−0.013096
85	0.262919	0.165133	0.077237	0.071721
86	0.319358	0.170283	0.054554	−0.072210
87	0.272983	0.231181	−0.014471	0.011689
88	0.134116	−0.026693	0.161400	0.110292
89	0.100379	0.026517	0.086236	0.130478
90	0.144718	−0.000895	0.093767	0.044514
91	0.114943	0.022145	0.035871	0.069193
92	0.122051	0.011043	0.192803	0.022796
93	0.079482	0.026156	0.117725	0.056565
94	0.124641	0.027387	0.122956	−0.025369
95	0.090708	0.027357	0.064450	0.013058
96	0.159781	−0.055202	−0.090597	0.151598
97	0.084577	−0.037203	−0.126698	0.119739
98	0.192484	−0.100195	−0.162066	0.104148
99	0.114579	−0.046270	−0.219547	0.100067

Quantization Vectors for b_4 (concluded)

b_4	G_5	G_6	G_7	G_8
100	0.153083	−0.010127	−0.086266	0.068648
101	0.088202	−0.010515	−0.102196	0.046281
102	0.164494	−0.057325	−0.132860	0.024093
103	0.109419	−0.013999	−0.169596	0.020412
104	0.039180	−0.209168	−0.035872	0.087949
105	0.012790	−0.177723	−0.129986	0.073364
106	0.045261	−0.256694	−0.088186	0.004212
107	−0.005314	−0.231202	−0.191671	−0.002628
108	0.037963	−0.153227	−0.045364	0.003322
109	0.030800	−0.126452	−0.114266	−0.010414
110	0.044125	−0.184146	−0.081400	−0.077341
111	0.029204	−0.157393	−0.172017	−0.089814
112	0.393519	−0.043228	−0.111365	−0.000740
113	0.289581	0.018928	−0.123140	0.000713
114	0.311229	−0.059735	−0.198982	−0.081664
115	0.258659	0.052505	−0.211913	−0.034928
116	0.300693	0.011381	−0.083545	−0.086683
117	0.214523	0.053878	−0.101199	−0.061018
118	0.253422	0.028496	−0.156752	−0.163342
119	0.199123	0.113877	−0.166220	−0.102584
120	0.249134	−0.165135	0.028917	0.051838
121	0.156434	−0.123708	0.017053	0.043043
122	0.214763	−0.101243	−0.005581	−0.020703
123	0.140554	−0.072067	−0.015063	−0.011165
124	0.241791	−0.152048	0.106403	−0.046857
125	0.142316	−0.131899	0.054076	−0.026485
126	0.206535	−0.086116	0.046640	−0.097615
127	0.129759	−0.081874	0.004693	−0.073169

Annex G (Normative) Quantization Tables for Higher Order Coefficients

Quantization table for b_5

b_5	$H_{1,1}$	$H_{1,2}$	$H_{1,3}$	$H_{1,4}$
0	0.264108	0.045976	−0.200999	−0.122344
1	0.479006	0.227924	−0.016114	−0.006835
2	0.077297	0.080775	−0.068936	0.041733
3	0.185486	0.231840	0.182410	0.101613
4	−0.012442	0.223718	−0.277803	−0.034370
5	−0.059507	0.139621	−0.024708	−0.104205
6	−0.248676	0.255502	−0.134894	−0.058338
7	−0.055122	0.427253	0.025059	−0.045051
8	−0.058898	−0.061945	0.028030	−0.022242
9	0.084153	0.025327	0.066780	−0.180839
10	−0.193125	−0.082632	0.140899	−0.089559
11	0.000000	0.033758	0.276623	0.002493
12	−0.396582	−0.049543	−0.118100	−0.208305
13	−0.287112	0.096620	0.049650	−0.079312
14	−0.543760	0.171107	−0.062173	−0.010483
15	−0.353572	0.227440	0.230128	−0.032089
16	0.248579	−0.279824	−0.209589	0.070903
17	0.377604	−0.119639	0.008463	−0.005589
18	0.102127	−0.093666	−0.061325	0.052082
19	0.154134	−0.105724	0.099317	0.187972
20	−0.139232	−0.091146	−0.275479	−0.038435
21	−0.144169	0.034314	−0.030840	0.022207
22	−0.143985	0.079414	−0.194701	0.175312
23	−0.195329	0.087467	0.067711	0.186783
24	−0.123515	−0.377873	−0.209929	−0.212677
25	0.068698	−0.255933	0.120463	−0.095629
26	−0.106810	−0.319964	−0.089322	0.106947
27	−0.158605	−0.309606	0.190900	0.089340
28	−0.489162	−0.432784	−0.151215	−0.005786
29	−0.370883	−0.154342	−0.022545	0.114054
30	−0.742866	−0.204364	−0.123865	−0.038888
31	−0.573077	−0.115287	0.208879	−0.027698

Quantization table for b_6

b_6	$H_{2,1}$	$H_{2,2}$	$H_{2,3}$	$H_{2,4}$
0	−0.143886	0.235528	−0.116707	0.025541
1	−0.170182	−0.063822	−0.096934	0.109704
2	0.232915	0.269793	0.047064	−0.032761
3	0.153458	0.068130	−0.033513	0.126553
4	−0.440712	0.132952	0.081378	−0.013210
5	−0.480433	−0.249687	−0.012280	0.007112
6	−0.088001	0.167609	0.148323	−0.119892
7	−0.104628	0.102639	0.183560	0.121674
8	0.047408	−0.000908	−0.214196	−0.109372
9	0.113418	−0.240340	−0.121420	0.041117
10	0.385609	0.042913	−0.184584	−0.017851
11	0.453830	−0.180745	0.050455	0.030984
12	−0.155984	−0.144212	0.018226	−0.146356
13	−0.104028	−0.260377	0.146472	0.101389
14	0.012376	−0.000267	0.006657	−0.013941
15	0.165852	−0.103467	0.119713	−0.075455

Quantization table for b_7

b_7	$H_{3,1}$	$H_{3,2}$	$H_{3,3}$	$H_{3,4}$
0	0.182478	0.271794	−0.057639	0.026115
1	0.110795	0.092854	0.078125	−0.082726
2	0.057964	0.000833	0.176048	0.135404
3	−0.027315	0.098668	−0.065801	0.116421
4	−0.222796	0.062967	0.201740	−0.089975
5	−0.193571	0.309225	−0.014101	−0.034574
6	−0.389053	−0.181476	0.107682	0.050169
7	−0.345604	0.064900	−0.065014	0.065642
8	0.319393	−0.055491	−0.220727	−0.067499
9	0.460572	0.084686	0.048453	−0.011050
10	0.201623	−0.068994	−0.067101	0.108320
11	0.227528	−0.173900	0.092417	−0.066515
12	−0.016927	0.047757	−0.177686	−0.102163
13	−0.052553	−0.065689	0.019328	−0.033060
14	−0.144910	−0.238617	−0.195206	−0.063917
15	−0.024159	−0.338822	0.003581	0.060995

Quantization table for b_8

b_8	$H_{4,1}$	$H_{4,2}$	$H_{4,3}$	$H_{4,4}$
0	0.323968	0.008964	−0.063117	0.027909
1	0.010900	−0.004030	−0.125016	−0.080818
2	0.109969	0.256272	0.042470	0.000749
3	−0.135446	0.201769	−0.083426	0.093888
4	−0.441995	0.038159	0.022784	0.003943
5	−0.155951	0.032467	0.145309	−0.041725
6	−0.149182	−0.223356	−0.065793	0.075016
7	0.096949	−0.096400	0.083194	0.049306

Annex H (Normative) Bit Frame Format

Symbol	*Bit 1*	*Bit 0*
0	$c_0(23)$	$c_0(5)$
1	$c_1(10)$	$c_2(3)$
2	$c_0(22)$	$c_0(4)$
3	$c_1(9)$	$c_2(2)$
4	$c_0(21)$	$c_0(3)$
5	$c_1(8)$	$c_2(1)$
6	$c_0(20)$	$c_0(2)$
7	$c_1(7)$	$c_2(0)$
8	$c_0(19)$	$c_0(1)$
9	$c_1(6)$	$c_3(13)$
10	$c_0(18)$	$c_0(0)$
11	$c_1(5)$	$c_3(12)$
12	$c_0(17)$	$c_1(22)$
13	$c_1(4)$	$c_3(11)$
14	$c_0(16)$	$c_1(21)$
15	$c_1(3)$	$c_3(10)$
16	$c_0(15)$	$c_1(20)$
17	$c_1(2)$	$c_3(9)$
18	$c_0(14)$	$c_1(19)$
19	$c_1(1)$	$c_3(8)$
20	$c_0(13)$	$c_1(18)$
21	$c_1(0)$	$c_3(7)$
22	$c_0(12)$	$c_1(17)$
23	$c_2(10)$	$c_3(6)$
24	$c_0(11)$	$c_1(16)$
25	$c_2(9)$	$c_3(5)$
26	$c_0(10)$	$c_1(15)$
27	$c_2(8)$	$c_3(4)$
28	$c_0(9)$	$c_1(14)$
29	$c_2(7)$	$c_3(3)$
30	$c_0(8)$	$c_1(13)$
31	$c_2(6)$	$c_3(2)$
32	$c_0(7)$	$c_1(12)$
33	$c_2(5)$	$c_3(1)$
34	$c_0(6)$	$c_1(11)$
35	$c_2(4)$	$c_3(0)$

Annex J (Normative) Tone Frame Parameters

I_D	f_0	l_1	l_2
0 - 4	N/A	N/A	N/A
5 - 12	$31.250 I_D$	1	1
13 - 25	$15.625 I_D$	2	2
26 - 38	$10.417 I_D$	3	3
39 - 51	$7.8125 I_D$	4	4
52 - 64	$6.2500 I_D$	5	5
65 - 76	$5.2803 I_D$	6	6
77 - 89	$4.4643 I_D$	7	7
90 - 102	$3.9063 I_D$	8	8
103 - 115	$3.4722 I_D$	9	9
116 - 122	$3.1250 I_D$	10	10
123 - 127	N/A	N/A	N/A
128	78.5	12	17
129	173.48	4	7
130	70.0	10	19
131	87.0	8	17
132	109.95	7	11
133	191.68	4	7
134	70.17	11	21
135	71.06	12	17
136	121.58	7	11
137	212.0	4	7
138	116.41	6	14
139	96.15	8	17
140	71.0	12	23
141	234.26	4	7
142	134.38	7	9
143	134.35	7	11
144	68.33	12	17
145	150.89	4	7
146	67.82	9	17
147	86.5	7	15
148	95.79	7	11
149	166.92	4	7
150	67.7	10	19
151	74.74	10	14
152	105.90	7	11
153	92.78	8	14
154	101.55	6	14
155	84.02	8	17
156	67.83	11	21
157	102.3	8	14
158	117.0	7	9
159	117.49	7	11
160	87.78	4	5
161	70.83	6	7
162	122.0	4	5
163	70.0	5	7
164 - 254	N/A	N/A	N/A
255	250.0	0	0

References

[1] Telecommunications Industry Association. *APCO Project 25 Vocoder Description (TIA-102BABA)*. November 1995. Reaffirmed December, 2008.

[2] D. W. Griffin and J. S. Lim. Multiband Excitation Vocoder. *IEEE Transactions on ASSP*, 36(8), August 1988.

[3] N. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[4] A. Levesque and A. Michelson. *Error-Control Techniques for Digital Communication*. Wiley, 1985.

[5] J. S. Lim and A. V. Oppenheim. Enhancement and Bandwidth Compression of Noisy Speech. *Proceedings of the IEEE*, 67(12):1586–1604, December 1979.

[6] Lin and Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[7] M. S. Mueller and J. Salz. A Unified Theory of Data Aided Equalization. *BELL SYSTEM TECHNICAL JOURNAL*, 60(9):2023–2038, November 1981.

[8] Telecommunications Industry Association. *APCO Project 25 Phase 2 Common Air Interface*.