NATO OTAN

NATO STANDARDIZATION AGENCY
AGENCE OTAN DE NORMALISATION

NSA

3 October 2008 NSA/1025(2008)-C3/4591

STANAG 4591 C3 (EDITION 1) – THE 600 BIT/S, 1200 BIT/S AND 2400 BIT/S NATO INTEROPERABLE NARROW BAND VOICE CODER

Reference:

a. AC/322(SC/6)N(2006)0008, dated 25 January 2006 (NU) - Ratification Request

1. The enclosed NATO Standardization Agreement, which has been ratified by nations as reflected in the **NATO Standardization Document Database (NSDD)**, is promulgated herewith.

2. The reference listed above is to be destroyed in accordance with local document destruction procedures.

ACTION BY NATIONAL STAFFS

3. National staffs are requested to examine their ratification status of the STANAG and, if they have not already done so, advise the NHQC3S, through their national delegation as appropriate of their intention regarding its ratification and implementation.

Juan A. MORENO
Vice Admiral, ESP(N)
Director, NATO Standardization Agency

Enclosure:
STANAG 4591 (Edition 1)

NATO Standardization Agency – Agence OTAN de Normalisation
B-1110 Brussels, Belgium Tel 32.2.707.4309 – Fax 32.2.707.5709
E-mail: c3s@hq.nato.int – Internet site: http://nsa.nato.int

STANAG 4591
(Edition 1)

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION AGENCY (NSA)

STANDARDIZATION AGREEMENT (STANAG)

SUBJECT: THE 600 BIT/S, 1200 BIT/S AND 2400 BIT/S NATO INTEROPERABLE NARROW BAND VOICE CODER

Promulgated on 3 October 2008

Juan A. MORENO
Vice Admiral, ESP(N)
Director, NATO Standardization Agency

RECORD OF AMENDMENTS

No.	Reference/date of Amendment	Date Entered	Signature

EXPLANATORY NOTES

AGREEMENT

1. This NATO Standardization Agreement (STANAG) is promulgated by the Director NATO Standardization Agency under the authority vested in him by the NATO Standardization Organisation Charter.

2. No departure may be made from the agreement without informing the tasking authority in the form of a reservation. Nations may propose changes at any time to the tasking authority where they will be processed in the same manner as the original agreement.

3. Ratifying nations have agreed that national orders, manuals and instructions implementing this STANAG will include a reference to the STANAG number for purposes of identification.

RATIFICATION, IMPLEMENTATION AND RESERVATIONS

4. Ratification, implementation and reservation details are available on request or through the NSA websites (internet http://nsa.nato.int; NATO Secure WAN http://nsa.hq.nato.int).

FEEDBACK

5. Any comments concerning this publication should be directed to NATO/NSA – Bvd Leopold III - 1110 Brussels - BE.

STANAG 4591
(Edition 1)

NATO STANDARDISATION AGREEMENT
(STANAG)

THE 600 BIT/S, 1200 BIT/S AND 2400 BIT/S NATO INTEROPERABLE NARROW BAND VOICE CODER

Annexes:

A. Description of the STANAG 4591 MELPe Algorithm
B. Performance Verification Requirements for 2400 bit/s and 1200 bit/s STANAG 4591 Implementations
C. Codebooks used by STANAG 4591
D. Description of the1200 bit/s MELPe Variation
E. Description of the Noise Preprocessor
F. Definitions and Acronyms
G. Fixed Point C Source Code
H. Floating Point C Source Code
I. Test Vectors for 2400 bit/s STANAG 4591
J. Test Vectors for 1200 bit/s STANAG 4591
K. Description of 1200 bit/s STANAG 4591 to 2400 bit/s STANAG 4591 Transcoder (Optional)
L. Description of 2400 bit/s STANAG 4591 to 2400 bit/s STANAG 4198 Transcoder (Optional)
M. MELPe Variation for 600 Bit/s NATO Narrow Band Voice Coder
N. Description of 600 bit/s to 2400 bit/s and 2400 bit/s to 600 bit/s MELPe Transcoders
O. Description of MELPe Frame Synchronization
P. Test Vectors for 600 bit/s STANAG 4591

Related Documents

A. NATO Documents

STANAG 4198 Parameters and Coding Characteristics That Must Be Common To Assure Interoperability of 2400 bps Linear Predictive Encoded Digital Speech

STANAG 4209 The NATO Multi-Channel Tactical Digital Gateway -- Standards for Analogue to Digital Conversion of Speech Samples

B. ITU Documents

See Annex B (Section B.2.2)

C. United States Government Federal Standards

FED-STD-1016 Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 Bit/Second Code Excited Linear Prediction (CELP)

FED-STD-1037 Glossary of Telecommunications Terms

FIPSPUB-137 Telecommunications: Analog to Digital Conversion of Voice by 2,400 Bit/Second Linear Predictive Coding

D. United States Government Military Standards

MIL-STD-188-113 Interoperability and Performance Standards for Analog-to-Digital Conversion Techniques

MIL-STD-3005 Analog-To-Digital Conversion of Voice by 2,400 Bit/Second Mixed Excitation Linear Prediction (MELP)

E. Order of precedence

In the event of a conflict between the text of this standard and the references cited herein, the text of this standard shall take precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

1 AIM

The aim of this agreement is to achieve a minimum standard of voice performance and interoperability for low rate voice coding across the communications infrastructure. To achieve this standard a common tri rate voice-coding algorithm at 600 bit/s, 1200 bit/s and 2400 bit/s is defined. This is intended to enable seamless end-to-end interoperability among and between the strategic and tactical NATO and National communication domains.

The aim is accomplished by the specification of the voice digitizer characteristics, voice coder performance, the coding tables and the bit stream definition requirements.

2 AGREEMENT

Participating nations agree to use the specifications in this STANAG as an interoperable method of transmitting and receiving 600 bit/s, 1200 bit/s and 2400 bit/s digital representations of a voice signal. This agreement applies for 2-party and multi-party communications, and it applies for clear and encrypted voice communication.

3 DEFINITIONS

See Annex F

4 GENERAL

4.1 Contents of the STANAG

This STANAG contains design requirements for digital coding of voice by Enhanced Mixed Excitation Linear Prediction (MELPe). STANAG-4591 is defined by the bit streams generated by the 600 bit/s MELPe, 1200 bit/s MELPe and the 2400 bit/s MELPe voice-coding algorithm. In addition, the STANAG contains the design requirements for a mandatory noise pre-processor (NPP). Note that the fixed point C code is included in this STANAG as Annex G to provide the authoritative example of how this bit stream can be correctly derived. This fixed point C code will be available in electronic format for distribution.

4.2 Structure of the STANAG

Section 5 of the STANAG contains the detailed requirements necessary to achieve interoperability among 2400 bit/s implementations of MELPe. The details of the algorithms, coding tables, and verification requirements are contained in Annexes as follows:

Annex A - Description of the STANAG 4591 MELPe Algorithm
Annex B - Performance Verification Requirements for 2400 bit/s and 1200 bit/s STANAG 4591 Implementations
Annex C - Codebooks used by STANAG 4591
Annex D - Description of the 1200 bit/s MELPe Variation.
Annex E - Description of the Noise Preprocessor
Annex F - Definitions and Acronyms
Annex G - Fixed Point C Source Code
Annex H - Floating Point C Source Code
Annex I - Test Vectors for 2400 bit/s STANAG 4591
Annex J - Test Vectors for 1200 bit/s STANAG 4591
Annex K - Description of 1200bit/s STANAG 4591 to 2400 bit/s STANAG 4591 Transcoder
Annex L - Description of 2400bit/s STANAG 4591 to 2400 bit/s STANAG 4198 Transcoder
Annex M - Description of the MELPe Variation for 600 Bit/s NATO Narrow Band Voice Code
Annex N - Description of 600 bit/sec to 2400 bit/sec and 2400 bit/sec to 600 bit/sec MELPe Tr
Annex O - Description of MELPe Frame Synchronization
Annex P - Test Vectors for 600 bit/s STANAG 4591

5 DETAILS OF THE AGREEMENT

5.1 General

The Enhanced Mixed Excitation Linear Prediction coder is based on the traditional Linear Prediction Coder (LPC) parametric model, but also includes five additional features. They are mixed excitation, aperiodic pulses, adaptive spectral enhancement, pulse dispersion, and Fourier magnitude modeling. A MELPe frame interval is 22.5 ms ± 0.01 percent in duration which corresponds to 180 voice samples (8000 samples/s).

5.2 Analog specification

The recommended analog requirements for the MELPe coder are for a nominal bandwidth ranging from 100 Hz to 3800 Hz. Although the MELPe coder will operate with a more band limited signal, performance degradation will result. To ensure proper

operation of the MELPe coder, the A to D subsystem should have a performance at least equivalent to a 16 bit linear A-D conversion process producing integer values within the range –32768 to 32767 with an effective sampling rate of 8kHz. Additionally the coder / decoder combination should have unity gain, which means that the output speech level should match that of the input speech. The analogue input circuit to the MELPe coder shall take precautions to limit signals outside the band of 100 Hz to 3800 Hz by means of filtering or a combination of oversampling and filtering. It shall also limit maximum signal amplitude excursions such that the worst combination of an in, or out-of-band, frequency and an amplitude overload shall not cause the A to D converter to produce an anomalous digital output. The anti-alias filter should attenuate at 4kHz by at least 20dB and at 8kHz and above by a least 40dB. Input DC offsets shall be controlled so as not to effect the operation of the coder.

5.3 Parameter quantization and encoding

The MELPe parameters which are quantized and transmitted are the final pitch (P_3); the bandpass voicing strengths (Vbp_i, i = 1, 2, …, 5); the two gain values (G_1 and G_2); the line spectral frequency (LSF) representation of the linear prediction coefficients (a_i, i= 1, 2, …, 10); the Fourier magnitudes; and the aperiodic flag. The use of the following quantization procedures is required for interoperability among various implementations.

Pitch and overall voicing

The final pitch (P_3), and the low band voicing strength (Vbp_1), are quantized jointly using 7 bits, as follows. If $Vbp_1 \leq 0.6$, then the frame is unvoiced and the all-zero code is sent. Otherwise, the log of P_3 is quantized with a 99-level uniform scalar quantizer (see 5.3.7) ranging from log20 to log160. The resulting index (range 0 to 98) is then mapped to the transmitted 7-bit codeword using the encode/decode values in Table 1. All 28 codes with Hamming weight of 1 or 2 are reserved for error protection. This table is also used in decoding the 7-bit pitch code to determine if a frame is voiced, unvoiced, or whether a frame erasure is indicated. A frame is determined unvoiced if the pitch code is all zero or has only one bit set. If two bits are set, then a frame erasure is indicated. Otherwise, the voiced mode is used and the pitch index is determined from the received code according to Table 1.

Bandpass voicing

When $Vbp_1 > 0.6$, the remaining bandpass voicing strengths are quantized to 1 if their value exceeds 0.6, and quantized to 0 otherwise. There is one exception. If the quantized values of Vbp_i, i = 2, 3, 4, 5 are 0001, respectively, then Vbp_5 is quantized to 0. The quantized values are transmitted using 4 bits. When $Vbp_1 \leq 0.6$, the bandpass voicing bits are replaced with FEC parity bits.

Gain

Two gain parameters, G_1 and G_2, are transmitted each frame. G_2 is quantized to 5 bits using a 32-level uniform quantizer ranging from 10.0 to 77.0 dB. The quantizer index is the transmitted codeword. G_1 is quantized to 3 bits using the following adaptive algorithm. This algorithm determines if the frame is a steady state frame or a transition frame. If G_2, for the current frame, is within 5 dB of G_2 for the previous frame, and G_1 is within 3 dB of the average of G_2 values for the current and previous frames, then the frame is steady-state and a special code (all zero) is sent to indicate that the decoder should set G_1 to the mean of the G_2 values for the current and previous frames. Otherwise, the frame represents a transition and G_1 is quantized with a 7-level uniform quantizer ranging from 6 dB below the minimum of the G_2 values for the current and previous frames to 6 dB above the maximum of those G_2 values. In this case, the

quantizer index plus 1 is the transmitted codeword. See 5.3.7 for details on the uniform quantizer.

Table 1. Encode / decode table for pitch and overall voicing parameter.

Code	Index	Code	Index	Code	Index	Code	Index
0x0	UNVOIC	0x2	UNVOICE	0x4	UNVOICE	0x6	ERASUR
0x0	UNVOIC	0x2	ERASURE	0x4	ERASURE	0x6	68
0x0	UNVOIC	0x2	ERASURE	0x4	ERASURE	0x6	69
0x0	ERASUR	0x2	16	0x4	42	0x6	70
0x0	UNVOIC	0x2	ERASURE	0x4	ERASURE	0x6	71
0x0	ERASUR	0x2	17	0x4	43	0x6	72
0x0	ERASUR	0x2	18	0x4	44	0x6	73
0x0	0	0x2	19	0x4	45	0x6	74
0x0	UNVOIC	0x2	ERASURE	0x4	ERASURE	0x6	75
0x0	ERASUR	0x2	20	0x4	46	0x6	76
0x0	ERASUR	0x2	21	0x4	47	0x6	77
0x0	1	0x2	22	0x4	48	0x6	78
0x0	ERASUR	0x2	23	0x4	49	0x6	79
0x0	2	0x2	24	0x4	50	0x6	80
0x0	3	0x2	25	0x4	51	0x6	81
0x0	4	0x2	26	0x4	52	0x6	82
0x1	UNVOIC	0x3	ERASURE	0x5	ERASURE	0x7	83
0x1	ERASUR	0x3	27	0x5	53	0x7	84
0x1	ERASUR	0x3	28	0x5	54	0x7	85
0x1	5	0x3	29	0x5	55	0x7	86
0x1	ERASUR	0x3	30	0x5	56	0x7	87
0x1	6	0x3	31	0x5	57	0x7	88
0x1	7	0x3	32	0x5	58	0x7	89
0x1	8	0x3	33	0x5	59	0x7	90
0x1	ERASUR	0x3	34	0x5	60	0x7	91
0x1	9	0x3	35	0x5	61	0x7	92
0x1	10	0x3	36	0x5	62	0x7	93
0x1	11	0x3	37	0x5	63	0x7	94
0x1	12	0x3	38	0x5	64	0x7	95
0x1	13	0x3	39	0x5	65	0x7	96
0x1	14	0x3	40	0x5	66	0x7	97
0x1	15	0x3	41	0x5	67	0x7	98

Linear prediction coefficients

The linear prediction coefficients are converted into line spectrum frequencies (LSF) and the resulting LSF vector is checked for monotonicity. If the vector is not monotonic it is adjusted accordingly. The LSF vector is also checked for minimum separation of 50 Hz and adjusted accordingly. The resulting LSF vector is then quantized by a multi-stage vector quantizer (MSVQ). The MSVQ codebook consists of four stages whose indices have 7, 6, 6, and 6 bits, respectively. The quantized LSF vector, $\hat{f}$, is the sum of the vectors selected by the search process, with one vector selected from each stage. The MSVQ search finds the codebook vector which minimizes the square of the weighted Euclidean distance, d^2, between the unquantized and quantized LSF vectors:

$$d^2(f,\hat{f}) = \sum_{i=1}^{10} w_i (f_i - \hat{f}_i)^2 \quad \text{, where } w_i = \begin{cases} P(f_i)^{0.3}, 1 \le i \le 8 \\ 0.64P(f_i)^{0.3}, i = 9 \\ 0.16P(f_i)^{0.3}, i = 10 \end{cases}, \qquad \textbf{EQUATION 1}$$

f_i is the i^{th} component of the unquantized LSF vector, and $P(f_i)$ is the inverse prediction filter power spectrum evaluated at frequency f_i. The indices of the four vectors are transmitted. The code vectors and corresponding indices are provided in Tables C-1 to C-4.

Fourier magnitudes

The ten Fourier magnitudes are coded with an 8-bit vector quantizer. The index of the code vector, which minimizes the weighted Euclidean distance between the input and code vectors, is transmitted. The weights are fixed and are given by:

$$w_i = [117/(25 + 75(1 + 1.4(F_i/1000)^2)^{0.69})]^2, \; i = 1, 2, \ldots, 10, \qquad \textbf{EQUATION 2}$$

where F_i = 8000*i/60 is the frequency in Hz corresponding to the i^{th} harmonic for a default pitch period of 60 samples. The code vectors and corresponding indices are given in Table C-5.

Aperiodic flag

The aperiodic flag is a single bit, transmitted as is. The aperiodic flag is set to 1 if Vbp_1 < 0.5 and set to 0 otherwise. When set, this flag tells the decoder that the pulse component of the excitation should be aperiodic, rather than periodic.

Uniform quantization

The pitch and gain quantization processes employ uniform quantizers which operate as

follows. The stepsize for an n-level quantizer ranging from x_1 to x_2 is $s = (x_2 - x_1)/(n-1)$. The n quantizer output values are $x_1 + i \cdot s$, i = 0, 1, …, n-1. The threshold values between levels i and i+1 are $x_1 + (0.5 + i)s$, i = 0, 1, …, n-2 . The quantizer produces n indices, 0, 1, …, n-1, which correspond to an increasing value of the parameter being quantized. For example, let $x_1 = 1$, $x_2 = 7$, and n = 7. This gives s = 1, levels of 1, 2, …, 7, and thresholds of 1.5, 2.5, …, 6.5. Index 0 is assigned to input values x, for which x < 1.5; index 1 is assigned to input values for which
1.5 $\leq$ x < 2.5; etc.

5.4 Error protection

The internal MELPe Forward Error Correction (FEC) is implemented in the unvoiced mode only, when the Fourier magnitudes, bandpass voicing, and jitter bits need not be transmitted. FEC replaces those 13 bits with the parity bits of three Hamming (7,4) codes and one Hamming (8,4) code. These codes protect the first stage LSF index (7 bits) and both gain indices (8 bits); there is one spare information bit, set to 0.

The protected bits are placed into a column vector, u, which post-multiplies the parity generator matrix to produce the n-bit parity vector, $p = [p_0 \;\; p_1 \;\; \ldots \;\; p_{n-1}]^T$, where n is 3 or 4.

The parity generator matrix for the Hamming (7,4) code is: $G_{7,4} = \begin{bmatrix} 1101 \\ 1011 \\ 0111 \end{bmatrix}$.

The parity generator matrix for the Hamming (8,4) code is: $G_{8,4} = \begin{bmatrix} 1101 \\ 1011 \\ 0111 \\ 1110 \end{bmatrix}$.

The 4 most significant bits (MSBs) of the first stage LSF index (u = [b6 b5 b4 b3]T) are protected by the (8,4) code, with the 4 parity bits written to the LSBs of the bandpass voicing index (p0 p1 p2 p3). The remaining 3 bits of the first stage index and the spare bit (u = [b2 b1 b0 0]T) are protected with 3 parity bits written to the MSBs of the Fourier magnitude index (p0 p1 p2). The Gain parameters are defined as follows, the second gain parameter is G2 = [$g2_4$, $g2_3$, $g2_2$, $g2_1$, $g2_0$] and the first gain parameter is G1 = [$g1_2$, $g1_1$, $g1_0$]. The 4 MSBs of the second gain index (u = [$g2_4$, $g2_3$, $g2_2$, $g2_1$]^T) are protected with 3 parity bits written to the next 3 bits of the Fourier magnitude index (p0 p1 p2). The LSB of the second gain index and the 3-bit first gain index (u = [$g2_0$, $g1_2$, $g1_1$, $g1_0$]^T) are protected with 3 parity bits written to the 2 LSBs of the Fourier magnitude

index (p0 p1) and the aperiodic flag (p2). The parenthesized groups of parity bits show their placement in the given index, with the right-most bit having the least significance.

5.5 Transmission format

This section provides information on the transmission rate for the coder, the number of bits allocated for each MELPe frame and the transmission order for the bits in each MELPe frame.

Transmission rate

The transmission rate shall be 2400 bit/s ± 0.01 percent. Since all frames contain 54 bits, the frame length is 22.5 ms ± 0.01 percent.

Bit allocation

Table 2 shows how the 54 bits in a 2400 bit/s MELPe frame are allocated among the parameters.

Table 2. 2400 bit/s MELPe bit allocation.

Parameters	Voiced	Unvoiced
LSFs	25	25
Fourier Magnitudes	8	-
Gain (2 per frame)	8	8
Pitch, overall voicing	7	7
Bandpass Voicing	4	-
Aperiodic Flag	1	-
Error Protection	-	13
Sync Bit	1	1
Total Bits / 22.5 ms Frame	54	54

Bit transmission order

Table 3 shows the transmission order for the 54 bits in each MELPe frame for both voiced and unvoiced frames. The sync bit alternates between 0 and 1 from frame to frame. Note that bit number 1 is transmitted first and bit 54 is transmitted last, with all other bits transmitted in sequence.

Table 3. 2400 bit/s MELPe bit transmission order.

Bit	Voiced	Unvoiced	Bit	Voiced	Unvoiced	Bit	Voiced	Unvoiced
1	$g2_0$	$g2_0$	19	$LSF1_6$	$LSF1_6$	37	$g1_0$	$g1_0$
2	BP_0	$FEC1_0$	20	$LSF4_5$	$LSF4_5$	38	BP_2	$FEC1_2$
3	P_0	P_0	21	P_3	P_3	39	BP_1	$FEC1_1$
4	$LSF2_0$	$LSF2_0$	22	$LSF1_5$	$LSF1_5$	40	$LSF2_1$	$LSF2_1$
5	$LSF3_0$	$LSF3_0$	23	$LSF1_4$	$LSF1_4$	41	$LSF3_3$	$LSF3_3$
6	$g2_3$	$g2_3$	24	$LSF2_5$	$LSF2_5$	42	$LSF2_2$	$LSF2_2$
7	$g2_4$	$g2_4$	25	BP_3	$FEC1_3$	43	$LSF3_2$	$LSF3_2$
8	$LSF3_5$	$LSF3_5$	26	$LSF1_3$	$LSF1_3$	44	$LSF3_1$	$LSF3_1$
9	$g2_1$	$g2_1$	27	$LSF1_2$	$LSF1_2$	45	$LSF4_3$	$LSF4_3$
10	$g2_2$	$g2_2$	28	$LSF2_4$	$LSF2_4$	46	$LSF4_2$	$LSF4_2$
11	P_4	P_4	29	$LSF4_4$	$LSF4_4$	47	AF	$FEC4_2$
12	$LSF3_4$	$LSF3_4$	30	FM_0	$FEC4_0$	48	$LSF4_1$	$LSF4_1$
13	P_5	P_5	31	$LSF1_1$	$LSF1_1$	49	FM_4	$FEC3_2$
14	P_1	P_1	32	$LSF2_3$	$LSF2_3$	50	FM_3	$FEC3_1$
15	P_2	P_2	33	FM_7	$FEC2_2$	51	FM_2	$FEC3_0$
16	$LSF4_0$	$LSF4_0$	34	FM_6	$FEC2_1$	52	FM_1	$FEC4_1$
17	P_6	P_6	35	FM_5	$FEC2_0$	53	$g1_2$	$g1_2$
18	$LSF1_0$	$LSF1_0$	36	$g1_1$	$g1_1$	54	SYNC	SYNC

NOTES: g = Gain BP = Bandpass Voicing
P = Pitch/Voicing LSF = Line Spectral Frequencies
FEC = Forward Error Correction Parity Bits FM = Fourier Magnitudes
Bit 1 = least significant bit of data set AF = Aperiodic Flag
Highlighted Bits = 24 Most Significant MELPe Bits

Bit format for packet transmission

To keep interoperability between packetized voice systems the MELPe frames are octet aligned. The order of the bits in the octet is least significant bit first. Bit 1 of the MELPe frame is the least significant bit of the first byte (Table 4). Seven octets are required for a frame at 2400 bit/s rate and 11 for the 1200 bit/s. The MELPe frames do not align exactly to octet boundaries. This leaves extra bits in the most significant bits of the last octet of the frame. The 2400 bit/s rate has 2 reserved bits (seven 8-bit octets minus 54 MELPe information bits) and there are 7 reserved bits (eleven 8-bit octets minus 81 MELPe information bits) for the 1200 bit/s rate. The reserved bits are set to zero on sending and ignored on receiving.

Table 4. Bit assignment for sending 2400 bit/s MELPe in a packet.

	MSB							**LSB**
Octet	8	7	6	5	4	3	2	1
1	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
2	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
3	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17
4	Bit 32	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25
5	Bit 40	Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33
6	Bit 48	Bit 47	Bit 46	Bit 45	Bit 44	Bit 43	Bit 42	Bit 41
7	Res 2	Res 1	Bit 54	Bit 53	Bit 52	Bit 51	Bit 50	Bit 49
Bit X – Bit number in MELPe frame (Table 3) Res X – reserved bits								

Table 5. Bit assignment for sending 1200 bit/s MELPe in a packet.

	MSB							**LSB**
Octet	8	7	6	5	4	3	2	1
1	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
2	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
3	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17
4	Bit 32	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25
5	Bit 40	Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33
6	Bit 48	Bit 47	Bit 46	Bit 45	Bit 44	Bit 43	Bit 42	Bit 41
7	Bit 56	Bit 55	Bit 54	Bit 53	Bit 52	Bit 51	Bit 50	Bit 49
8	Bit 64	Bit 63	Bit 62	Bit 61	Bit 60	Bit 59	Bit 58	Bit 57
9	Bit 72	Bit 71	Bit 70	Bit 69	Bit 68	Bit 67	Bit 66	Bit 65
10	Bit 80	Bit 79	Bit 78	Bit 77	Bit 76	Bit 75	Bit 74	Bit 73
11	Res 7	Res 6	Res 5	Res 4	Res 3	Res 2	Res 1	Bit 81
Bit X – Bit number in MELPe frame (Table D-9a and Table D-9b)) Res X – reserved bits								

6 PROTECTION OF PROPRIETARY RIGHTS

6.1 Proprietary rights

The following proprietary rights have been indicated as involved in this STANAG by the United States because this algorithm was developed under US government contract.

The United States Government has acquired irrevocable rights in the MELPe software, documentation, and algorithm, through various contract vehicles and hereby provides NATO and Partnership for Peace governments an irrevocable license to:

a. use, modify, reproduce, release or perform the MELPe software, in whole or in part, in any manner and for any purpose whatsoever, and to have or authorize others to do so provided that any modification, reproduction, or release, of the MELPe source code must contain a clearly visible notice, preceding the source code, advising of the existence of Texas Instrument's intellectual property rights in the MELPe algorithm and providing the following Texas Instrument contact information for commercial and nongovernmental use: Director, Government Contracts, Texas Instruments Incorporated, Semiconductor Group, Telephone +1 (972) 480-7442;

b. use, modify, reproduce or release the MELPe documentation, in whole or in part, in any manner and for any purpose whatsoever , and to have or authorize others to do so; and

c. practice the MELPe algorithm or have the MELPe algorithm practiced for the governments of NATO and Partnership for Peace nations for non-commercial government purposes.

7 IMPLEMENTATION OF THE AGREEMENT

This agreement is implemented by a nation when this nation has issued instructions that relevant future communication systems using 1200 or 2400 bit/s voice coders for its forces will be procured, manufactured and placed in service in accordance with the specifications detailed in the body of this STANAG and the relevant annexes.

ANNEX A

Description of the STANAG 4591 MELPe Algorithm

A.1 SCOPE

A.1.1 Scope

This annex provides a complete description of a MELPe algorithm. This annex is not a mandatory part of this standard and is included for guidance only.

A.2 APPLICABLE DOCUMENTS

A.2.1 Government documents

The Related Documents section in the body of this STANAG contains documents that apply to this annex.

A.2.2 Other publications

The following documents form a part of this annex to the extent specified.

"A Mixed Excitation LPC Vocoder Model for Low Bit Rate Speech Coding"
by A.V. McCree and T.P. Barnwell III, IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 4, July 1995, 242-250

"A 2.4 kbits/s MELP Coder Candidate for the New U.S. Federal Standard"
by A. McCree, K. Truong, E.B. George, T.P. Barnwell III, and V. Viswanathan, Proceedings of IEEE ICASSP 1996, pp.200-203

"Super Resolution Pitch Determination of Speech Signals"
by Y. Medan, E. Yair, and D. Chazan, IEEE Transactions on Signal Processing, Vol. 39, No. 1, January 1991, pp. 40-48

"The Computation of Line Spectral Frequencies Using Chebyshev Polynomials"
by P. Kabal and R.P. Ramachandran, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No.6, December 1986, pp.1419-1426

"Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 kb/s Speech Coding" by W.P. LeBlanc, B. Bhattacharya, S.A. Mahmoud, and V. Cuperman, IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 4, October 1993, pp. 373-385

"New Methods for Adaptive Noise Suppression"
by L. Arslan, A. McCree, and V. Viswanathan, Proceedings of IEEE ICASSP 1995, pp. 812-815

"MELP: The New Federal Standard at 2400 BPS"
by L. Supplee, R. Cohn, J. Collura, A. McCree, Proceedings of IEEE ICASSP 1997, pp. 1591-1594

(Applications for copies should be addressed to IEEE Customer Service, 445 Hoes Lane, P.O. Box 1331 Piscataway, New Jersey 08855-1331, USA)

A.2.3 Order of precedence

In the event of a conflict between the text of this standard and the references stated herein, the text of this standard shall take precedence.

A.3 DEFINITIONS

A.3.1 Terms

The definitions in Annex F of this standard apply to this annex.

A.3.2 Acronyms

The acronyms used is this annex are defined in Annex F or as follows:

DC - Direct Current
DFT - Discrete Fourier Transform
FFT - Fast Fourier Transform
FIR - Finite Impulse Response
RMS - Root Mean Square
VQ - Vector Quantization

A.4 GENERAL REQUIREMENTS

Not applicable.

A.5 DETAILED REQUIREMENTS

A.5.1 General

The Enhanced Mixed Excitation Linear Prediction coder is based on the traditional Linear Prediction Coding (LPC) parametric model, but also includes five additional

features. These are: mixed excitation, aperiodic pulses, adaptive spectral enhancement, pulse dispersion, and Fourier magnitude modeling. These features are illustrated in the MELPe decoder block diagram shown in Figure A-1b. The block diagram in Figure A-1a illustrates the MELPe coder processing sequence.

The mixed excitation is implemented using a multi-band mixing model. The excitation signal is generated by mixing periodic and noise excitations in frequency domain according to the frequency-dependent voicing strength. The primary effect of this mixed excitation is to reduce the buzz usually associated with LPC vocoders, especially in broadband acoustic noise.

When the input speech is voiced, the MELPe coder can synthesize using either periodic or aperiodic pulses. Aperiodic pulses are used most often during transition regions between voiced and unvoiced segments of the speech signal. This feature enables the decoder to reproduce erratic glottal pulses without introducing tonal sounds.

The adaptive spectral enhancement filter is based on the poles of the linear prediction synthesis filter. Its use enhances the formant structure of the synthetic speech and improves the match between the synthetic and natural bandpass waveforms. It also gives the synthetic speech a more natural quality.

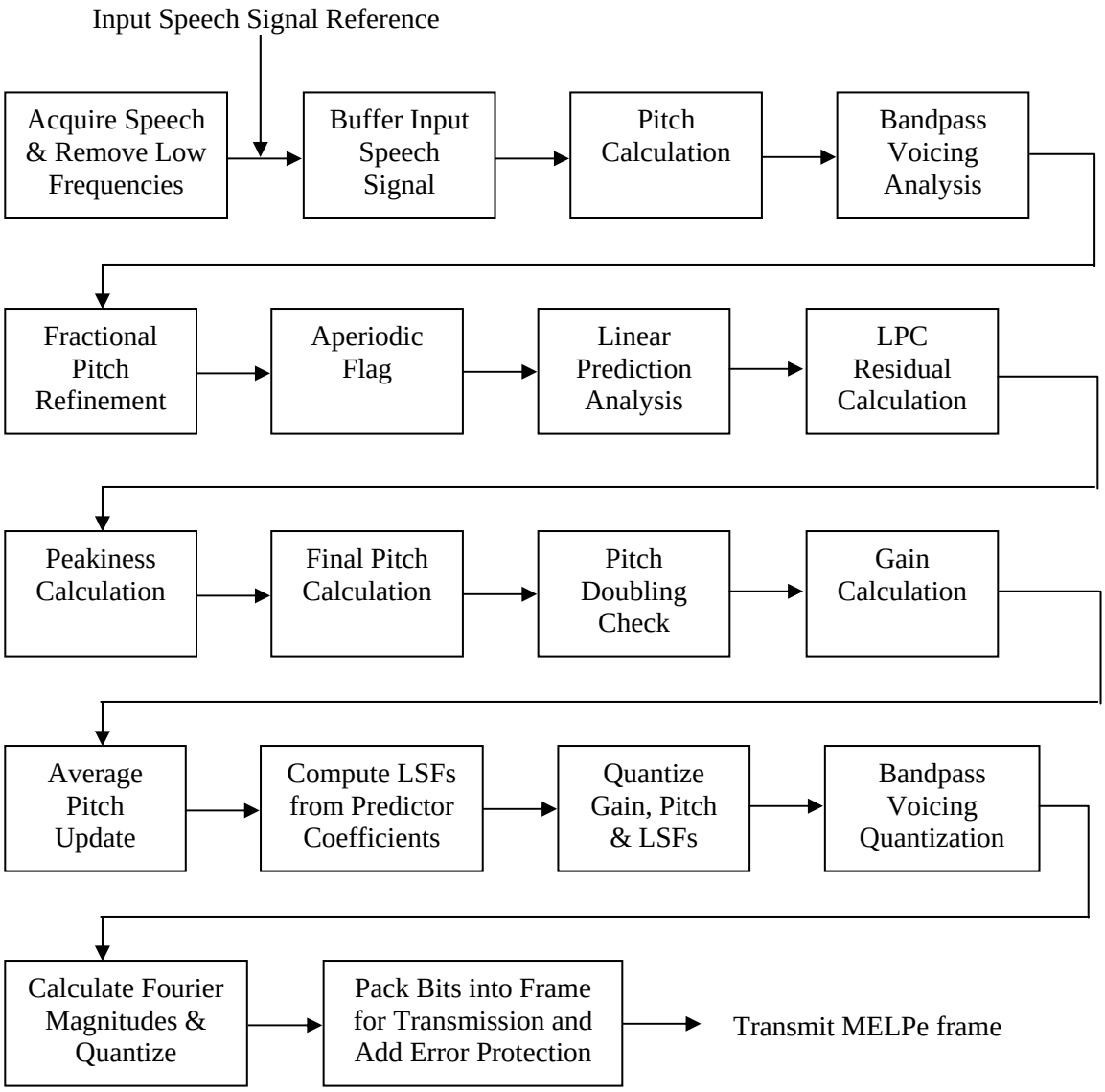


Figure A-1a. MELPe Coder Block Diagram

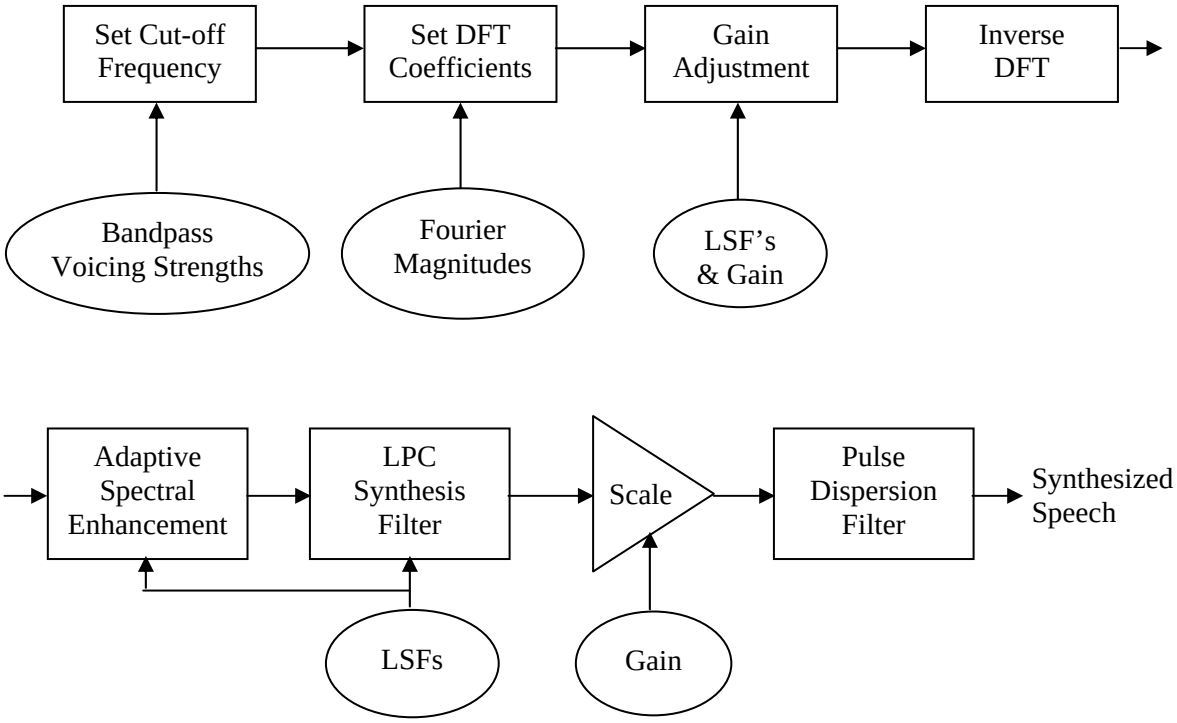


Figure A-1b. MELPe Decoder Block Diagram

Pulse dispersion is implemented using a fixed filter based on a spectrally-flattened triangle pulse. This filter spreads the excitation energy within a pitch period, reducing some of the harsh quality of the synthetic speech.

The first ten Fourier magnitudes are determined from the peaks of the Fourier transform of the prediction residual signal. The information in these coefficients improves the accuracy of the speech production model at the perceptually-important lower frequencies. This increases the quality of the synthetic speech, particularly for male speakers and when background noise is present.

A.5.2 Encoder

Digital input speech, defined as 8000 samples per second, as described in section 5.2 of this document is encoded by performing the following steps in the order given.

A.5.2.1 Low frequency removal

The first step in the encoding process is to remove any low frequency energy which may be present in the input signal. This is accomplished with a 4th order Chebychev type II highpass filter, having a cutoff frequency of 60 Hz and a stopband rejection of 30 dB. The filter output is referred to as the input speech signal throughout the following encoder description.

A.5.2.2 Input sample buffering of encoder

A buffer containing the most recent samples of the input speech signal is maintained in the encoder. One of these samples is designated the last sample in the current frame. The buffer extends beyond this sample into the past and future to contain the samples needed for the encoding process. The last sample in the current frame serves as a reference point for many of the encoder calculations.

A.5.2.3 Integer pitch calculation

For this pitch calculation, the input speech signal is first processed with a 1 kHz, 6th order Butterworth lowpass filter. The integer pitch value, P_1, is the value of , τ, $\tau = 40,41,...,160$, for which the normalized autocorrelation function, $r(\tau)$, is maximized. This function is defined by:

$$r(\tau) = \frac{c_\tau(0,\tau)}{\sqrt{c_\tau(0,0)c_\tau(\tau,\tau)}} \qquad \textbf{EQUATION A-1}$$

where

$$c_\tau(m,n) = \sum_{k=-\lfloor\tau/2\rfloor-80}^{-\lfloor\tau/2\rfloor+79} s_{k+m}s_{k+n} \qquad \textbf{EQUATION A-2}$$

and $\lfloor\tau/2\rfloor$ represents truncation to an integer value. The center of the pitch analysis window is at sample s_0 in equation A-2. For the integer pitch calculation, this window is centered on the last sample in the current frame. The lowpass filter output is sample s_0 when its input is the last sample in the current frame. The time index k in the autocorrelation preserves the pitch analysis window alignment around its center point; the normalization compensates for changing signal amplitudes. The final pitch calculation (see A.5.2.10) extends the pitch range to a lag of 20 samples.

A.5.2.4 Bandpass voicing analysis

This portion of the encoder determines the five bandpass voicing strengths, $Vbp_i, i = 1,2,...,5$. It also refines the integer pitch measurement and the corresponding normalized autocorrelation value. The bandpass voicing analysis begins by filtering the input speech signal into five frequency bands. These filters are 6th order Butterworth, with passbands of 0-500, 500-1000, 1000-2000, 2000-3000, and 3000-4000 Hz.

A refined pitch measurement is made using the 0-500 Hz filter output signal. This measurement is centered on the filter output produced when its input is the last sample in the current frame. Two pitch candidates are considered in this refinement, namely the integer pitch values, P_1, from the current and previous frames. For each candidate, equation A-1 is used to perform an integer pitch search over lags from 5 samples shorter to 5 samples longer than the candidate, and a fractional pitch refinement (see A.5.2.5) is performed around the optimum integer pitch lag. This produces two fractional pitch candidates and their corresponding normalized autocorrelation values. The candidate having the higher normalized autocorrelation is selected as the fractional pitch, P_2. The corresponding normalized autocorrelation, $r(P_2)$, is saved as the lowest band voicing strength, Vbp_1. P_2 is saved for use in determining the voicing strength for the remaining frequency bands. It is also used in the final pitch calculation (see A.5.2.10) and gain calculation (see A.5.2.12).

For each remaining band, the bandpass voicing strength is the larger of $r(P_2)$ as determined by the fractional pitch procedure for the bandpass signal and the time envelope of the bandpass signal, where $r(P_2)$ for the time envelope is first decremented by 0.1 to compensate for an experimentally observed bias (due to the smoothness of the time envelope signals). The envelopes are calculated by full-wave rectification followed by a smoothing filter. This filter consists of a zero at DC in cascade with a complex pole pair at 150 Hz with a radius of 0.97. For each calculation of $r(P_2)$, the analysis window is centered on the last sample in the current frame, as was the case for the first band.

A.5.2.5 Fractional pitch refinement

This procedure, which is used at several places in the encoding process, utilizes an interpolation formula to increase the accuracy of an input pitch value. This value is first rounded to the nearest integer. Assume that this integer has a value of T samples. The interpolation formula presumes that $r(\tau)$ is a continuous function with a maximum between lags of T-1 and T+1. Hence, $c_T(0, T-1)$ and $c_T(0, T+1)$ are computed and

compared to determine if the maximum is more likely to fall between *T* and *T*+1 or between T-1 and T. If $c_T(0,T-1) > c_T(0,T+1)$, then the maximum probably falls between T-1 and T and the pitch, T, is decremented by one prior to interpolation. The fractional offset, Δ, is then computed by the interpolation equation:

$$\Delta = \frac{c_T(0,T+1)c_T(T,T) - c_T(0,T)c_T(T,T+1)}{c_T(0,T+1)[c_T(T,T) - c_T(T,T+1)] + c_T(0,T)[c_T(T+1,T+1) - c_T(T,T+1)]} \quad \textbf{EQUATION A-3}$$

where $c_T(m,n)$ is defined by equation A-2. In some cases, this formula produces an offset outside the range of 0.0 to 1.0, so the offset is clamped between -1 and 2. The fractional pitch is $T+\Delta$ and is clamped between 20 and 160.

The normalized autocorrelation at the fractional pitch value is given by:

$$r(T+\Delta) = \frac{(1-\Delta)c_T(0,T) + \Delta c_T(0,T+1)}{\sqrt{c_T(0,0)[(1-\Delta)^2 c_T(T,T) + 2\Delta(1-\Delta)c_T(T,T+1) + \Delta^2 c_T(T+1,T+1)]}} \quad \textbf{EQUATION A-4}$$

Equations A-3 and A-4 produce the fractional offset and corresponding normalized autocorrelation which would be obtained if the input signal had been linearly interpolated to obtain values between the actual sampling times.

A.5.2.6 Aperiodic flag

The aperiodic flag is set to 1 if $Vbp_1 < 0.5$ and set to 0 otherwise. The Vbp_1 value determined by bandpass voicing analysis (see A.5.2.4) is used for this comparison. When set, this flag tells the decoder that the pulse component of the excitation should be aperiodic, rather than periodic. A.5.3.1 describes the use of the aperiodic flag.

A.5.2.7 Linear prediction analysis

A 10^{th} order linear prediction analysis is performed on the input speech signal using a 200 sample (25 ms) Hamming window centered on the last sample in the current frame. The traditional autocorrelation analysis procedure is implemented using the Levinson-Durbin recursion. In addition, a bandwidth expansion coefficient of 0.994 (15 Hz) is applied to the prediction coefficients, a_i, $i = 1,2,...,10$, where each coefficient is multiplied by 0.994^i.

A.5.2.8 Linear prediction residual calculation

The linear prediction residual signal is calculated by filtering the input speech signal with the prediction filter whose coefficients were determined by linear prediction analysis

(see A.5.2.7). The residual window is centered on the last sample in the current frame, and is made wide enough for use by the final pitch calculation (see A.5.2.10).

A.5.2.9 Peakiness calculation

The peakiness of the residual signal is calculated over a 160 sample window centered on the last sample in the current frame. The peakiness value is the ratio of the L2 norm to the L1 norm of the residual signal, r_n, in the window:

$$\text{peakiness} = \frac{\sqrt{\frac{1}{160}\sum_{n=1}^{160} r_n^2}}{\frac{1}{160}\sum_{n=1}^{160} |r_n|} \qquad \textbf{EQUATION A-5}$$

If the peakiness exceeds 1.34, then voicing the lowest band voicing strength, Vbp_1, is forced to 1.0. If the peakiness exceeds 1.6, then the lowest three band strengths, $Vbp_i, i = 1,2,3,$ are all forced to 1.0. This is the only use of the peakiness measure.

A.5.2.10 Final pitch calculation

The final pitch measurement uses the lowpass filtered residual signal, where the filter is a 6th order Butterworth, with a 1 kHz cutoff. Equation A-1 is used to perform an integer pitch search over lags from 5 samples shorter to 5 samples longer than P_2, rounded to the nearest integer. This measurement is centered on the filter output produced when its input is the last residual sample in the current frame. A fractional pitch refinement (see A.5.2.5) is then made around the optimum integer pitch lag. This produces tentative values for the final pitch, P_3, and for the corresponding normalized autocorrelation, $r(P_3)$.

If $r(P_3) \geq 0.6$, the pitch doubling check procedure (see A.5.2.11) is performed on the filtered residual, using P_3 as the candidate pitch, and doubling threshold $D_{th} = 0.75$ if $P_3 \leq 100$, or $D_{th} = 0.5$ otherwise. The doubling check procedure may produce new values for P_3 and $r(P_3)$.

The else action for the preceding if is as follows. A fractional pitch refinement around P_2 is performed using the input speech signal. This measurement is centered on the last sample in the current frame and produces new values for P_3 and $r(P_3)$. If $r(P_3) < 0.55$, then P_3 is replaced by P_{avg}, the long-term average pitch (see A.5.2.13). Otherwise, the pitch doubling check procedure is performed on the input speech signal, using P_3 as the

candidate pitch, and doubling threshold $D_{th} = 0.9$ if $P_3 \leq 100$, or $D_{th} = 0.7$ otherwise. The doubling check procedure may produce new values for P_3 and $r(P_3)$.

Finally, if $r(P_3) < 0.55$, then P_3 is replaced by P_{avg}.

The following pseudo code shows the final pitch algorithm:

```
inputs: the input speech signal; the residual signal; P2; Pavg
outputs: P3, cor_P3

fresid buffer = filter the residual with a 1 kHz Butterworth
P3 = best integer pitch on fresid over the range P2-5 to P2+5
P3, cor_P3 = frac_pitch(fresid, P3)
if (cor_P3 >= 0.6)
    Dth = 0.5
    if (P3 <= 100) Dth = 0.75
    P3, cor_P3 = double_ck(fresid, P3, Dth)
else
    P3, cor_P3 = frac_pitch(input, P2)
    if (cor_P3 < 0.55)
        P3 = Pavg
    else
        Dth = 0.7
        if (P3 <= 100) Dth = 0.9
        P3, cor_P3 = double_ck(input, P3, Dth)
    endif
endif
if (cor_P3 < 0.55) P3 = Pavg
```

A.5.2.11 Pitch doubling check

The pitch doubling check procedure looks for and corrects pitch values which are multiples of the actual pitch. This procedure takes a signal, a candidate pitch P, and a doubling threshold D_{th}, and returns the checked pitch P_c, and the corresponding correlation, $r(P_c)$. All fractional pitch calculations are made using the signal given to the doubling check procedure.

This procedure begins with a fractional pitch refinement around P. This produces tentative values for P_c and $r(P_c)$. Next, the largest value of k is found for which $r(P_c / k) > D_{th} r(P_c)$, where $(P_c / k) \geq 20$ and $k = 8,7,...,2$. $r(P_c / k)$ is calculated in two steps: 1) a fractional pitch refinement around P_c / k, producing P_k; and 2) a double verification, if $P_k < 30$. If such a k is found, then a fractional pitch refinement around P_k is performed, producing new values for P_c and $r(P_c)$.

Finally, if P_c is less than 30 samples, then double verification is performed.

The following pseudo code shows the pitch double check procedure:

```
inputs: signal; P; Dth
outputs: Pc, cor_Pc

Pc, cor_Pc = frac_pitch(signal, P)
for (k=8; k>=2; k--)
    Pk = Pc/k
    if (Pk >= 20)
        Pk, cor_Pk = frac_pitch(signal, Pk)
        if (Pk < 30) cor_Pk = double_ver(Pk, cor_Pk)
        if (cor_Pk > Dth * cor_Pc)
            Pc, cor_Pc = frac_pitch(signal, Pk)
            break
        endif
    endif
endfor
if (Pc < 30) cor_Pc = double_ver(Pc, cor_Pc)
```

For inputs P and $r(P)$, the double verification procedure returns the smaller of $r(P)$ and $r(2P)$, where $r(2P)$ is determined by the fractional pitch procedure around $2P$. The use of double verification in the double check procedure provides robustness against spurious short pitch values.

A.5.2.12 Gain calculation

The input speech signal gain is measured twice per frame using a pitch-adaptive window length. This length is identical for both gain measurements and is determined as follows. When $\mathrm{Vbp}_1 > 0.6$, the window length is the shortest multiple of P_2 which is longer than 120 samples. If this length exceeds 320 samples, it is divided by 2. When $\mathrm{Vbp}_1 \le 0.6$, the window length is 120 samples. The gain calculation for the first window produces G_1 and is centered 90 samples before the last sample in the current frame. The calculation for the second window produces G_2 and is centered on the last sample in the current frame. The gain is the RMS value, measured in dB, of the signal in the window, s_n:

$$\mathrm{G}_\mathrm{i} = 10\log_{10}\left(0.01 + \frac{1}{\mathrm{L}}\sum_{\mathrm{n}=1}^{\mathrm{L}} \mathrm{s}_\mathrm{n}^2\right) \qquad \textbf{EQUATION A-6}$$

where L is the window length. The 0.01 term prevents the log argument from going too close to zero. If a gain measurement is less than 0.0, it is clamped to 0.0. The gain measurement assumes that the input signal range is -32768 to 32767 (see 5.2).

A.5.2.13 Average pitch update

The long-term average pitch, $\mathrm{P}_{\mathrm{avg}}$, is updated with a simple smoothing procedure. If $\mathrm{r}(\mathrm{P}_3) > 0.8$ and $\mathrm{G}_2 > 30\mathrm{dB}$, then P_3 is placed into a buffer containing the three most recent strong pitch values, p_i, $\mathrm{i} = 1,2,3$. Otherwise, all three pitch values in the buffer are moved toward a default pitch, $\mathrm{P}_{\mathrm{default}} = 50$ samples, according to:

$$\mathrm{p}_\mathrm{i} = 0.95\mathrm{p}_\mathrm{i} + 0.05\mathrm{P}_{\mathrm{default}}, \mathrm{i} = 1,2,3 \qquad \textbf{EQUATION A-7}$$

The average pitch is then updated as the median of the three values in the buffer. $\mathrm{P}_{\mathrm{avg}}$ is used in the final pitch calculation (see A.5.2.10).

A.5.2.14 Quantization of prediction coefficients

First, the linear prediction coefficients a_i, $\mathrm{i} = 1,2,...,10$, are converted into Line Spectral Frequencies (LSFs). Next, a process which forces the LSF components to be in ascending order with a minimum separation of 50 Hz is performed. This process begins by checking all adjacent pairs of the LSF components and swapping any pair not in ascending order. This step is repeated as many as ten times, if necessary. The

minimum separation criterion is then applied by correcting each pair, f_i and f_{i+1}, for which $d = f_{i+1} - f_i$ is less than 50 Hz, Δ_{min}, as shown in the following pseudo code. The LSF components and frequency-related constants are in Hertz; scaling in other implementations may differ. The minimum separation process is repeated ten times.

```
Dmin = 50
for (i=1;i<10; i++)
    d = f[i+1] - f[i]
    if (d < dmin)
        s1 = s2 = (dmin-d)/2
        if (i == 1 and f[i] < dmin) s1 = f[i]/2
        else if (i > 1)
            tmp = f[i] - f[i-1]
            if     (tmp < dmin)   s1 = 0
            else if (tmp < 2*dmin) s1 = (tmp-dmin)/2
        endif
        if (i == 9 and f[i+1] > 4000-dmin) s2 = (4000-f[i+1])/2
        else if (i < 9)
            tmp = f[i+2] - f[i+1]
            if     (tmp < dmin)   s2 = 0
            else if (tmp < 2*dmin) s2 = (tmp-dmin)/2
        endif
        f[i+1] = f[i+1] + s2
    endif
endfor
```

The resulting LSF vector, f, is then quantized using a MSVQ. The MSVQ codebook consists of four stages of 128, 64, 64, and 64 levels respectively. The quantized vector, $\hat{f}$, is the sum of the vectors selected by the search process, with one vector selected from each stage. The MSVQ search finds the codebook vector which minimizes the square of the weighted Euclidean distance, d^2, between the unquantized and quantized LSF vectors:

$$d^2\left(f, \hat{f}\right) = \sum_{i=1}^{10} w_i \left(f_i - \hat{f}_i\right)^2, \text{ where } w_i = \begin{cases} P(f_i)^{0.3}, 1 \le i \le 8 \\ 0.64P(f_i)^{0.3}, i = 9 \\ 0.16P(f_i)^{0.3}, i = 10 \end{cases}, \qquad \textbf{EQUATION A-8}$$

f_i is the i^{th} component of the unquantized LSF vector, and $P(f_i)$ is the inverse prediction filter power spectrum evaluated at frequency f_i. The search procedure is an M-best approximation to a full search, in which the M=8 best code vectors from each stage are

saved for use with the next stage. The process to ensure ascending order and minimum separation (described in the first part of this section) is then applied to the quantized LSF vector. The resulting vector is used in the Fourier magnitude calculation (see A.5.2.18).

A.5.2.15 Pitch quantization

The final pitch value, P_3, is quantized on a base 10 logarithmic scale with a 99-level uniform quantizer ranging from 20 to 160 samples. These pitch values are then mapped to a 7-bit codeword using a look-up table, as shown in section 5.3.1. The all-zero codeword represents the unvoiced state, and is sent if $Vbp_1 \leq 0.6$. All 28 codewords with Hamming weight of 1 or 2 are reserved for error protection. The uniform quantizer details are described in 5.3.7.

A.5.2.16 Gain quantization

The two gain values are quantized as follows. First, G_2 is quantized with a 5-bit uniform quantizer ranging from 10 to 77 dB. Then, G_1 is quantized to 3 bits using the following adaptive algorithm. If G_2 for the current frame is within 5 dB of G_2 for the previous frame, and G_1 is within 3 dB of the average of the G_2 values for the current and previous frames, then the frame is steady-state and a special code (all zero) is sent to indicate that the decoder should set G_1 to the mean of the G_2 values for the current and previous frames. Otherwise, the frame represents a transition and G_1 is quantized with a 7-level uniform quantizer ranging from 6 dB below the minimum of the G_2 values for the current and previous frames to 6 dB above the maximum of those G_2 values. The quantizer range is clamped to 10 and 77 dB. The uniform quantizer details are described in 5.3.7.

Pseudo code for the adaptive quantization of G_1 is shown below.

```
If (|G2 - G2p| < 5.0 and |G1 - 0.5 *(G2 + G2p)| < 3.0)
    quantizer_index = 0
else
    gain_max = max(G2p, G2) + 6.0
    gain_min = min(G2p, G2) - 6.0
    if (gain_min < 10.0) gain_min = 10.0
    if (gain_max > 77.0) gain_max = 77.0
    quantizer_index values 1 to 7 are determined by quantizing G1 with a 7-level,
    uniform quantizer ranging from gain_min to gain_max
endif
```

A.5.2.17 Bandpass voicing quantization

When $\mathrm{Vbp}_1 \le 0.6$ (unvoiced), the remaining voicing strengths, $\mathrm{Vbp}_i, i = 2,3,4,5$, are quantized to 0. When $\mathrm{Vbp}_1 > 0.6$, the remaining voicing strengths are quantized to 1 if their value exceeds 0.6, and quantized to 0 otherwise. There is one exception. If the quantized values of $\mathrm{Vbp}_i, i = 2,3,4,5$ are 0001, respectively, then Vbp_5 is quantized to 0.

A.5.2.18 Fourier magnitude calculation and quantization

This analysis measures the Fourier magnitudes of the first 10 pitch harmonics of the prediction residual generated by the quantized prediction coefficients. It uses a 512-point Fast Fourier Transform (FFT) of a 200-sample window centered at the end of the frame. First, a set of quantized predictor coefficients is calculated from the quantized LSF vector (see A.5.2.14). Then the residual signal or samples are generated using the quantized prediction coefficients. Next, a 200-sample Hamming window is applied, the signal is zero-padded to 512 points, and the complex FFT is performed. Finally, the complex FFT output is transformed into magnitudes, and the harmonics are found with a spectral peak-picking algorithm.

The peak-picker finds the maximum within a width of $512/\hat{\mathrm{P}}_3$ frequency samples centered around the initial estimate for each pitch harmonic, where $\hat{\mathrm{P}}_3$ is the quantized pitch. This width is truncated to an integer. The initial estimate for the location of the i^{th} harmonic is $512i/\hat{\mathrm{P}}_3$. The number of harmonic magnitudes searched for is limited to the smaller of 10 or $\hat{\mathrm{P}}_3/4$. These magnitudes are then normalized to have an RMS value of 1.0. If fewer than 10 harmonics are found, the remaining magnitudes are set to 1.0.

The 10 magnitudes are quantized with an 8-bit vector quantizer. The codebook is searched using a perceptually weighted Euclidean distance, with fixed weights that emphasize low frequencies over higher frequencies. The weights are given by:

$$w_i = \left[\frac{117}{25 + 75\left(1 + 1.4\left(\frac{f_i}{1000}\right)^2\right)^{0.69}} \right]^2, i = 1,2,...,10, \qquad \textbf{EQUATION A-9}$$

where $f_i = 8000i/60$ is the frequency in Hz corresponding to the i^{th} harmonic for a default pitch period of 60 samples. The weights are applied to the squared difference between the input Fourier magnitudes and the codebook values.

A.5.2.19 Error protection and bit packing

Table 2 in 5.5.2 shows the bit allocation for the MELPe coder. To improve performance in channel errors, the unused coder parameters for the unvoiced mode are replaced with forward error correction. Three Hamming (7,4) codes and one Hamming (8,4) code are used. The (7,4) code corrects single bit-errors, while the (8,4) code in addition detects double bit-errors. The (8,4) code is applied to the 4 most significant bits (MSBs) of the first MSVQ index, and the 4 parity bits are written over the bandpass voicing. The remaining 3 bits of the first MSVQ index along with a reserved bit (set to zero), are covered by a (7,4) code with the resulting 3 parity bits written to the MSBs of the Fourier series VQ index. The 4 MSBs of the G_2 codeword are protected with 3 parity bits that are written to the next 3 bits of the Fourier magnitudes. Finally, the LSB of the second gain index and the 3-bit G_1 codeword are protected with 3 parity bits written to the 2 LSBs of the Fourier magnitudes and the aperiodic flag.

The bit transmission order is given in 5.5.3.

A.5.3 Decoder

The channel data is decoded by performing the following steps in the order given.

A.5.3.1 Bit unpacking and error correction

The received bits are unpacked from the channel and assembled into the parameter codewords. Parameter decoding is different for voiced and unvoiced modes. The pitch is decoded first, since it contains the mode information. If the pitch code is all-zero or has only one bit set, then the unvoiced mode is used. If two bits are set, a frame erasure is indicated. Otherwise, the pitch value is decoded and the voiced mode is used.

In the unvoiced mode, the (8,4) Hamming code is decoded to correct single bit errors and detect double errors. If an uncorrectable error is detected, a frame erasure is indicated. Otherwise, the (7,4) Hamming codes are decoded, correcting single errors but without double error detection.

If any erasure is detected in the current frame, by the Hamming code, by the pitch code, or directly signaled from the channel, then a frame repeat mechanism is implemented. All of the parameters for the current frame are replaced with the parameters from the previous frame. In addition, the first gain term is set equal to the second gain term so that no gain transitions are allowed.

If an erasure is not indicated, the remaining parameters are decoded. The LSFs are checked for ascending order and minimum separation as described in A.5.2.14. In the

unvoiced mode, default parameter values are used for the pitch, jitter, bandpass voicing, and Fourier magnitudes. The pitch value is set to 50 samples, the jitter is set to 25%, all of the bandpass voicing strengths are set to 0, and the Fourier magnitudes are set to 1. In the voiced mode, Vbp_1 is set to 1; jitter is set to 25% if the aperiodic flag is a 1; otherwise jitter is set to 0%. The bandpass voicing strength for the upper four bands is set to 1 if the corresponding bit is a 1; otherwise the voicing strength is set to 0. There is one exception. If 0001 is received for $Vbp_i, i = 2,3,4,5$, respectively, then Vbp_5 is set to 0. When the special all-zero code for the first gain parameter, G_1, is received, some errors in the second gain parameter, G_2, can be detected and corrected. This correction process provides improved performance in channel errors. The decoding for the two gain parameters is shown in the following pseudo code.

```
Inputs: G1_index, G2_index          quantized to 3 bits and 5 bits respectively
outputs: G1, G2
internal: G2p, G2p_error            initialized to “0” and “FALSE” respectively
                                    prior to first use

G2 = decode(G2_index)               32 levels; range: 10 to 77 dB
if (G1_index == 0)                  special G1 code: use mean of G2 and G2p
    if (|G2 - G2p| > 5)             G2_index probably in error
        if (G2p_error == FALSE)     i.e., if G2p is correct, then
            G2 = G2p                replace the erroneous G2 with past value
        endif
        G2p_error = TRUE
    else                            G2_index probably correct
        G2p_error = FALSE
    endif
    G1 = 0.5 * (G2 + G2p)           mean of G2 and G2p
else
    G1 = decode(G1_index)           7 levels; range: min(G2,G2p)-6 to max(G2,G2p)+6
    G2p_error = FALSE               (above range is clamped to 10 to 77 dB)
endif
G2p = G2                            save for use as past value
```

A.5.3.2 Noise attenuation

For quiet input signals, a small amount of gain attenuation is applied to both decoded gain parameters using a power subtraction rule. This attenuation is a simplified, frequency invariant case of the Smoothed Spectral Subtraction noise suppression method.

Before determining the attenuation for the first gain term, G_1, a background noise estimate, G_n, is updated as follows. If $G_1 > G_n + C_{up}$ then $G_n = G_n + C_{up}$. If $G_1 < G_n - C_{down}$ then $G_n = G_n - C_{down}$. Otherwise, $G_n = G_1$, $C_{up} = 0.0337435$ and $C_{down} = 0.135418$, so that the noise estimator moves up by 3 dB per second and down by 12 dB per second for the gain update rate of 88.9 updates per second. The noise estimate is clamped between 10 and 80. Noise estimation is disabled for repeated frames to prevent repeated attenuation. The background noise estimate is also used in the adaptive spectral enhancement calculation (see A.5.3.5).

Gain G_1 is then modified by subtracting a (positive) correction term, G_{att}, given in dB by

$$G_{att} = -10\log_{10}\left(1 - 10^{0.1[G_n + 3 - G_1]}\right), \qquad \textbf{EQUATION A-10}$$

where G_n is the background noise estimate (in dB), and G_1 is the first gain term (in dB). The correction is clamped to a maximum value of 6 dB to avoid fluctuations and signal distortion. To ensure that the attenuation is applied only to quiet signals, the G_n value as used in equation A-10 is clamped at an upper limit of 20 dB.

The noise estimation and gain modification steps are then repeated for the second gain term, G_2. Noise estimation and gain attenuation are disabled for repeated frames.

A.5.3.3 Parameter interpolation

All MELPe synthesis parameters are interpolated pitch-synchronously for each synthesized pitch period. The interpolated parameters are the gain (in dB), LSFs, pitch, jitter, Fourier magnitudes, pulse and noise coefficients for mixed excitation, and spectral tilt coefficient for the adaptive spectral enhancement filter. If the starting point, t_0 {where t_0 = 0,1,…,179}, of the new pitch period is less than 90 samples, gain is linearly interpolated between the second gain of the prior frame, $G_{2p,}$ and the first gain of the current frame, G_1, otherwise, gain is interpolated between G_1 and G_2. Normally, the other parameters are linearly interpolated between the past and current frame values. The interpolation factor, int, for these parameters is based on the starting point of the new pitch period:

$$\text{int} = t_0 / 180, \qquad \textbf{EQUATION A-11}$$

There are two exceptions to this interpolation procedure. First, if there is an onset with a high pitch frequency, pitch interpolation is disabled and the new pitch is immediately used. This condition is met when G_1 is more than 6 dB greater than G_{2p} and the current frame's pitch period is less than half the prior frame's pitch period. The second exception also involves a gain onset. If G_2 differs from G_{2p} by more than 6 dB, then the LSFs, spectral tilt, and pitch are interpolated using the interpolated gain trajectory as a

basis, since the gain is transmitted twice per frame and has a more accurate interpolation path. In this case, the interpolation factor is given by

$$\text{int} = \frac{G_{\text{int}} - G_{2p}}{G_2 - G_{2p}}, \qquad \textbf{EQUATION A-12}$$

where G_{int} is the interpolated gain. This interpolation factor is then clamped between 0 and 1.

A.5.3.4 Mixed excitation generation

The periodic and the noise excitations are combined in the frequency domain and then converted to a time-domain signal of one pitch period in length using inverse Discrete Fourier Transform. The excitation spectrum is generated based on two parameters, the cutoff frequency *F* and the Fourier magnitude vector $M(k), k = 1,2,...,L$. The cutoff frequency *F* is obtained from the quantized bandpass voicing strengths, $Vbp_i, i = 1,2,...,5,$ and then interpolated for each pitch cycle. *F* is set to be zero if the overall voicing Vbp_1 of the frame is set to be unvoiced. Otherwise the quantized voicing strengths $Vbp_i, i = 2,3,4,5$ are mapped into a cutoff frequency *F* according to the table below.

Table A-1. Cutoff Frequencies for Voiced Frames

Cutoff frequency (Hz)	**500**	**1000**	**2000**	**4000**
Voicing patterns of $Vbp_i, i = 2,3,4,5$	0000 0001 0010 0011 0100 0101 0110	1000 1001 1010	1100	0111 1011 1101 1110 1111

The pitch period *T*, is the interpolated pitch value plus the jitter times the pitch, where the jitter is the interpolated jitter strength times the output of a uniform random number generator between -1 and 1. This pitch period is rounded to the nearest integer and clamped between 20 and 160. The corresponding fundamental frequency of the pitch period is $f_0 = 2\pi/T$. The Fourier magnitude vector length is $L = \lfloor T/2 \rfloor$. Two transition frequencies F_H and F_L are determined according to the cutoff frequency *F* employing an empirically derived algorithm. The transition frequencies are in the range $F_L \in [0.85F, 0.98F]$, $F_H \in [F, 1.05F]$. These transition frequencies are equivalent to two DFT frequency component indices V_L and V_H. A voiced model is used for all the frequency samples below V_L, a mixed model is used for frequency samples between V_L and V_H, and an unvoiced model is used for frequency samples above V_H. To define the mixed mode, a gain factor *g* is selected with the value depending on the cutoff frequency (the higher is the cutoff frequency *F*, the smaller is the gain factor). The frequency components of the excitation are determined as follows,

$$|X(k)| = \begin{cases} M(k) & k < V_L \\ \dfrac{k - V_L}{V_H - V_L} \cdot g \cdot M(k) + \dfrac{V_H - k}{V_H - V_L} \cdot M(k) & V_L \le k \le V_H \\ g \cdot M(k) & k > V_H \end{cases}$$ **EQUATION A-13**

$$\angle X(k) = \begin{cases} k\phi_0 & k < V_L \\ k\phi_0 - \dfrac{k - V_L}{V_H - V_L} \cdot \phi_{RND} & V_L \le k \le V_H \\ \phi_{RND} & k > V_H \end{cases}$$ **EQUATION A-14**

where ϕ_0 is a constant selected such as to avoid a pitch pulse at the pitch cycle boundary. The ϕ_{RND} is a random number between 2π and -2π. The frequency samples of the mixed excitation are then converted to time domain excitation using an inverse Discrete Fourier Transform.

A.5.3.5 Adaptive spectral enhancement

The adaptive spectral enhancement filter is applied to the mixed excitation signal. This filter is a tenth order pole/zero filter, with an additional first-order tilt compensation. Its coefficients are generated by bandwidth expansion of the linear prediction filter transfer function, $\mathrm{A(z)}$, corresponding to the interpolated LSFs. The transfer function of the enhancement filter, $\mathrm{H_{ase}(z)}$, is given by:

$$\mathrm{H_{ase}(z)} = \frac{\mathrm{A}\left(\alpha z^{-1}\right)}{\mathrm{A}\left(\beta z^{-1}\right)} * \left(1 + \mu z^{-1}\right), \quad \text{where} \quad \begin{matrix} \alpha = 0.5p \\ \beta = 0.8p \end{matrix},$$ **EQUATION A-15**

and tilt the coefficient μ is first calculated as max(0.5k_1,0), then interpolated, then multiplied by p, the signal probability. The first reflection coefficient, k_1, is calculated from the decoded LSFs. By the MELPe predictor coefficient sign convention, k_1, is usually negative for voiced spectra. The signal probability p is estimated by comparing the current interpolated gain, G_{int}, to the background noise estimate G_n using the formula:

$$\mathrm{p} = \frac{\mathrm{G_{int}} - \mathrm{G_n} - 12}{18},$$ **EQUATION A-16**.

This signal probability is clamped between 0 and 1.

A.5.3.6 Linear prediction synthesis

The synthesis uses a direct form filter, with the coefficients corresponding to the interpolated LSFs.

A.5.3.7 Gain adjustment

Since the excitation is generated at an arbitrary level, the speech gain must be introduced to the synthesized speech. The correct scaling factor, S_{gain}, is computed for each synthesized pitch period of length T by dividing the desired RMS value (G_{int} must be converted from dB) by the RMS value of the unscaled synthetic speech signal $\hat{s}_n$:

$$S_{gain} = \frac{10^{G_{int}/20}}{\sqrt{\frac{1}{T}\sum_{n=1}^{T}\hat{s}_n^2}},$$ **EQUATION A-19**.

To prevent discontinuities in the synthesized speech, this scale factor is linearly interpolated between the previous and current values for the first ten samples of the pitch period.

A.5.3.8 Pulse dispersion

The pulse dispersion filter is a 65th order FIR filter derived from a spectrally-flattened triangle pulse. The coefficients are listed in.

Table A-2. Filter coefficients for the pulse dispersion filter

Samples 1-13	Samples 14-26	Samples 27-39	Samples 40-52	Samples 53-65
-0.17304259	0.24325127	0.07343483	0.02968464	0.00019707
-0.01405709	-0.01767043	-0.00518645	-0.01247640	-0.02825247
0.01224406	-0.00018612	0.01298488	0.01854666	0.01720989
0.11364226	0.05869485	0.02928440	0.00076184	-0.06004292
0.00198199	-0.00327456	-0.01989405	-0.07749640	-0.07076744
0.00000658	0.00607395	0.01216758	0.01244697	0.00914347
0.04529633	0.02753924	0.01180979	-0.02721777	0.06082730
-0.00092027	-0.03351673	-0.38924775	0.07266098	0.01805528
-0.00103078	0.00602189	0.00720325	0.00472008	-0.00318634
0.02552787	0.01436539	-0.01154561	0.03526439	0.03444110
-0.06339257	0.82854582	0.08426287	0.02674603	0.00026302
-0.00122031	0.00033165	-0.00355720	-0.00744038	-0.01053809
0.01412525	-0.00360180	0.02151233	0.02582623	0.02165922

A.5.3.9 Synthesis loop control

After processing each pitch period, the decoder updates t_0 by adding T, the number of samples in the period just synthesized. If $t_0 < 180$, synthesis for the current frame continues from the parameter interpolation step (see A.5.3.3). Otherwise, the decoder buffers the remainder of the current period which extends beyond the end of the current frame and subtracts 180 from t_0 to produce its initial value next frame.

A.5.3.10 Optional Postfilter

Once a frame of synthesized speech is generated, an optional postfilter may be applied. The frame is divided into four equal sized subframes. The LSFs are linearly interpolated for each subframe, then converted to LPC coefficients, $a_i, i = 1,2,...,10$, and reflection coefficients, $r_i, i = 1,2,...,10$. The transfer function of the postfilter is given by

$$H_{pf}(z) = \frac{\sum_{i=0}^{10} a_i \gamma^i z^{-i}}{\sum_{i=0}^{10} a_i \beta^i z^{-i}} (1 - \mu z^{-1}) \qquad \textbf{EQUATION A-20}$$

where $\gamma = 0.56$, $\beta = 0.75$. μ is determined by the reflection coefficients; if $t = \prod_{i=0}^{10} (1 - r_i^2)$, is larger than 0.3, $\mu = 0.0$, otherwise $\mu = 0.2$.

For each subframe, a gain adjustment factor *g* is computed as follows:

$$e_{in} = \sum_{n=0}^{N-1} |s(n)|, \quad e_{out} = \sum_{n=0}^{N-1} |s'_{pf}(n)| \qquad \textbf{EQUATION A-21}$$

$$g = \begin{cases} \frac{e_{in}}{e_{out}} & e_{out} >= 64 \\ g_{prev} & \text{else} \end{cases} \qquad \textbf{EQUATION A-22}$$

where *N* = 45 is the subframe size, $s(n)$ is the synthesized speech signal before applying the postfilter, and $s'_{pf}(n)$ is the output of the filter $H_{pf}(z)$. g_{prev} is the gain adjustment factor of the previous subframe. Then the gain-scaled signal is given by

$$s_{pf}(n) = \left[\left(1 - \frac{n}{N}\right) g_{prev} + \frac{n}{N} g \right] s'_{pf}(n) \qquad n = 0,1,...,N-1 \qquad \textbf{EQUATION A-23}$$

Finally $s_{pf}(n)$ is passed through a low-pass filter and a high-pass filter to generate the final output signal. Both filters are 2nd order Butterworth filters with cutoff frequencies at 3800 Hz and 60 Hz respectively.

ANNEX B - MANDATORY

Performance Verification Requirements for 2400 and 1200 bit/s STANAG 4591 Implementations

B.1 SCOPE

B.1.1 Scope

In order to assure interoperability at a minimum acceptable performance level between NATO nations, this annex is a mandatory part of this standard. The information contained herein is intended for compliance. All new implementations of the STANAG 4591 coder must be tested to verify that their performance meets or exceeds that of the STANAG 4591 reference coder (Annex G, henceforth referred to as the reference coder). This annex provides guidelines for verifying the performance of a STANAG 4591 implementation. Two independent methods of verification are presented.

B.2 APPLICABLE DOCUMENTS

B.2.1 Government documents

Not applicable.

B.2.2 Other publications

The following documents form a part of this annex to the extent specified.

ANSI Standard

S3.2-1989	American National Standard Method for Measuring the Intelligibility of Speech over Communications Systems (Applications for copies should be addressed to ANSI Customer Service, 11 West 42nd Street, New York, New York 10036, USA)

ITU –T Recommendations

P.800 Methods for subjective determination of transmission quality

P.830 Subjective performance assessment of telephone-band and wideband digital codecs

Handbook of Telephonometry
ITU, Geneva, 1992
ISBN 92-61-04911-7

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE)

"IEEE Recommended Practice for Speech Quality Measurements" by IEEE Subcommittee on Subjective Measurements, Transactions on Audio and
Electroacoustics, Vol. 17, 1969, pp. 227-246

(Applications for copies should be addressed to
IEEE Customer Service
445 Hoes Lane, P.O. Box 1331
Piscataway
New Jersey 08855-1331, USA)

B.2.3 Order of precedence

In the event of a conflict between the text of this standard and the references stated herein, the text of this standard shall take precedence.

B.3 DEFINITIONS

B.3.1 Terms

Terms used in this annex are defined in Annex F of this standard or as follows.

B.3.1.1 A/B test

An A/B test is a subjective listening method using direct paired forced choice comparisons to assess the relative preference of one voice coder against another voice coder.

B.3.2 Acronyms used in this annex

Acronyms used in this section are either defined in Annex F of this standard or as follows.

ANSI - American National Standards Institute
CELP - Code Excited Linear Prediction
CVSD - Continuously Variable Slope Delta Modulation
DRT - Diagnostic Rhyme Test
HMMWV - High Mobility Multipurpose Wheeled Vehicle
MCE - Mobile Command Enclosure
PCM - Pulse Code Modulation

B.4 GENERAL REQUIREMENTS

B.4.1 General

New implementations of STANAG 4591 must be verified to assure that their performance meets or exceeds the performance of the STANAG 4591 reference coder. A new implementation must also meet the same performance standards when interoperating with the STANAG 4591 reference coder. Testing is accomplished through formal evaluation of intelligibility and quality or by showing bit exactness between the new implementation and a verified STANAG 4591 implementation. Both verification methods evaluate an implementation over a selected set of conditions.

B.5 DETAILED REQUIREMENTS

B.5.1 Formal evaluation

Subjective evaluation of the intelligibility and quality performance of new implementations shall be conducted using standardized test methodologies. These evaluations shall be conducted over a representative set of talkers (minimum 3 male + 3 female) and shall include the following conditions:

- Quiet for intelligibility and quality,
- Motorized field vehicle (HMMWV) for intelligibility and quality,
- A mobile field communication enclosure (MCE) for intelligibility and quality,
- Military fast jet aircraft (F15) for intelligibility and quality,
- Military helicopter (UH-60) for intelligibility,
- Staff vehicle (Volvo) for quality,
- Modern office environment for quality.

The source material used to represent these environments will represent operational noise levels and shall include the effects of the normal resident microphone used in the various environments. The test material must meet the requirements of the selected standardized test methodology. The number of test subjects used by a test method must be sufficient to provide results with standard errors less than or equal to 1% of the test method's scale range, where:

$$S.E. = S.D. / SQRT (N); N= \#Talkers * \#Listeners.$$

Validation of the new implementation will be achieved if the performance results for each condition, as averaged over talkers and subjects, are not less than that of the standard STANAG 4591 reference implementation at a 95% confidence interval. For each condition, this validation shall be conducted at both coder rates (2400 bit/s and 1200 bit/s). The validation must also be conducted for all coder configurations.

Table B-1 summarizes the coder configurations which are evaluated both for intelligibility and quality. In Table B-1, the "Implementation →STANAG 4591 Reference" and "STANAG 4591 Reference → Implementation" cases are "cross-coder" configurations that test interoperability.

Table B-1. Tested coder configurations

Encoder → Decoder
Implementation →Implementation
Implementation → STANAG 4591 Reference
STANAG 4591 Reference → Implementation
STANAG 4591 Reference → STANAG 4591 Reference

For some test methodologies the STANAG 4591 Reference performance results can be made available from prior evaluations. In some cases where either the implementation encoder or decoder have been shown to be bit exact to the reference coder or other validated implementation, this table can be reduced. For example if the implementation decoder is bit exact to the reference coder, Table B-1 becomes Table B-2. As can be seen, this greatly reduces the required number of tests.

Table B-2. Special Case

Encoder → Decoder
Implementation → STANAG 4591 Reference
Implementation → STANAG 4591 Reference
STANAG 4591 Reference → STANAG 4591 Reference
STANAG 4591 Reference → STANAG 4591 Reference

A recommended validation procedure using standardized test methodologies is presented in the following subsections.

B.5.1.1 Example of a recommended validation procedure

The intelligibility of a new implementation is evaluated using the Diagnostic Rhyme Test (DRT). DRT performance thresholds have been set using the STANAG 4591 Reference with postfilter during the Phase II selection process. Quality is evaluated using a direct paired comparison, forced choice, test, i.e., an A/B test. Performance thresholds for the quality tests have also been set and are based on the percent preference for the new implementation over the STANAG 4591 reference coder. The US Federal Standard 1016 CELP coder is also included in the quality test to broaden the context of the test and to include a known difference from STANAG 4591 [comparison to reference STANAG 4591 only]. Table B-3 shows the intelligibility and quality test conditions. Six talkers (3 male, 3 female) are used for each condition. Source material for the DRT is a standardized word list. Source material for the paired comparison test is Harvard sentences. All source material has been generated with acoustically mixed noise simulations at operational noise levels with resident microphones. All tests are conducted in North American English.

Table B-3. Intelligibility and quality test conditions

Intelligibility Condition (microphone)	Quality Condition (microphone)
Quiet (Dynamic)	Quiet (Dynamic)
HMMWV (H250)	HMMWV (H250)
MCE Field Shelter (M87)	MCE Field Shelter (M87)
F-15 jet (M101)	F-15 jet (M101)
Blackhawk (M87)	
	Office (STU III)
	Car (STU III)
5 Intelligibility Conditions	**6 Quality Conditions**

Recommended intelligibility tests

The DRT will be performed in accordance with ANSI standard S3.2-1989 and will be scored with eight listeners from a trained listener crew of minimum size 10 listeners. The combined talker score determined by the test lab must meet or exceed the corresponding threshold score for each condition. Table B-4 shows the threshold score for each condition. The threshold score for each condition is based on a one-tail 95% confidence interval.

Table B-4. Threshold performance for intelligibility conditions

Intelligibility condition (microphone)	Threshold score 2400 bit/s	Threshold score 1200 bit/s
Quiet (Dynamic)	91.75	90.32
HMMWV (H250)	73.87	70.55
MCE Field Shelter (M87)	91.14	87.27
Black Hawk (M87)	77.00	72.20
F-15 jet (M101)	78.61	74.98

Recommended quality test

The quality of the three coder configurations involving the new implementation (see Table B-1) is compared with the quality of the Reference coder using a direct paired forced choice comparison (A/B) test performed in accordance with the ITU Handbook of Telephonometry. Quality is measured in each condition by the percent preference averaged over all talkers and listeners. The A/B test is conducted with 32 subjects. For the case of six conditions, six talkers, 3 coders (1200, 2400, CELP), two presentation orders (A/B + B/A) and one repeat there are 432 comparisons presented to each subject. Block randomization procedures are used to assure presentation balance and to remove order effects. In order to accommodate the repeat measure, single sentence samples are used for each talker. This provides the additional benefit of directly adjacent A/B comparisons during presentation. The repeat measure is made using a unique second sentence per talker.

For each coder configuration involving the new implementation, the percent preferred must meet or exceed the threshold for each condition. The threshold for individual conditions is 45.84%, i.e., each coder configuration involving the new implementation must have a preference percentage of 45.84% or more in each condition. The threshold is based on a one-tail 95% confidence interval for a binary distribution.

B.5.1.2 Alternate validation procedure

Alternate intelligibility tests

Any standardized intelligibility test methodology can be used in any language for the validation of a STANAG 4591 implementation as long as the requirements of Section B.5.1 are fulfilled. The performance requirements for validation will be based on the performance of the reference STANAG 4591 coder, averaged over talkers and listeners, for each condition as measured by the test method. A threshold score for each condition will be based on a one-tail 95% confidence interval.

Alternate quality tests

An A/B paired comparison with forced choice test methodology should be implementable in any language. This test methodology and the percent preference threshold of 45.84% is the only procedure acceptable for the validation of a coder implementation in quality.

B.5.2 Reference coder bit exactness

A lower cost alternative for implementation verification is accomplished by demonstrating that the new implementation is bit exact as compared to the Reference coder or any previously verified STANAG 4591 implementation. Bit exactness means that given the same digital input, the new implementation's encoder produces the same bitstream as produced by the encoder of the Reference coder or the verified STANAG 4591 implementation. Also given the same bitstream, the new implementation's decoder produces the same 16 bit linear PCM samples as produced by the decoder of the Reference coder or the verified STANAG 4591 implementation. A test vector database for bit exactness is available from the NC3A and is described in annexes I and J of this STANAG document. This database is separated into three distinct parts: the input speech vectors, the coded bit stream vectors and the output speech vectors. The exactness must be shown over the coded and output test vectors, as generated with and without the noise processor active. Errors in the least significant bit will be allowed. A flow diagram illustrating the process required to validate an implementation is provided in Figure B-1.

For information only, the above procedure can be used to check an implementation without the noise pre-processor active. This would be done using versions of the coded and output vectors generated from the reference coder, again without the noise pre-processor active.

B.5.3 Test Material

The source test material used for either intelligibility or quality validation must fulfill the requirements of Section B.5.1. DRT and Paired Comparison input material in North American English is available from NC3A. The Reference coder in fixed point C source code is available in Annex G. The code is also available in electronic form from NC3A via a secure web site.

The Input Speech Test Vector Data Base, the Coded Speech Test Vector Data Base, and the Output Synthesized Speech Test Vector Data Base shown in Figure B-1 for use in the bit exactness test method of Section B.5.2 are available in electronic form from NC3A at the same web site.

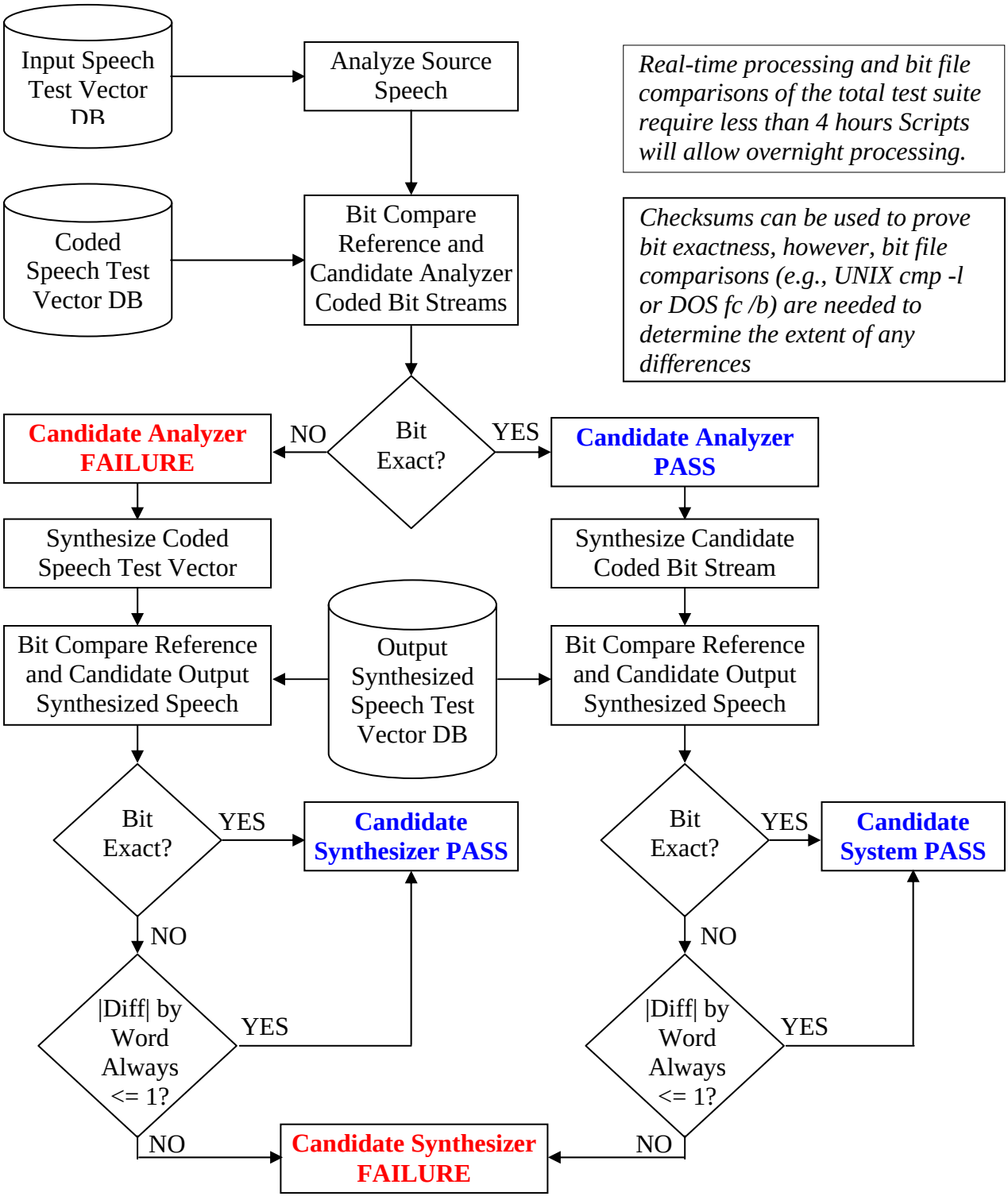


Figure B-1. Test Process Flow for STANAG 4591 Voice Coders

ANNEX C

Codebooks Used by STANAG 4591

The codebooks used by STANAG 4591 are available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX D

Description of the 1200 bit/s MELPe Variation

D.1 INTRODUCTION

D.1.1 Objective

This document describes the MELPe 1200 bit/s speech coder based on the STANAG 4591 MELPe coder at 2400 bit/s. The coder has both 2400 bit/s and 1200 bit/s modes. A Noise Pre-Processor is integrated into the coder. Since the 1200 bit/s coder is developed based on the 2400 bit/s MELPe coder, this document only describes the routines that are modified or newly added to the 2400 bit/s MELPe coder. Please refer to the original 2400 bit/s MELPe descriptions if more information about the MELPe algorithm is needed.

D.1.2 Algorithm Overview

The coder has analysis modules similar to the original 2400 bit/s MELPe coder except the changes described below. A block or "superframe" structure consisting of three consecutive frames is adopted to quantize the transmitted parameters more efficiently for the 1.2 kbit/s mode.

The frame size of the 1200 bit/s coder is 22.5 ms, the same as for the original MELPe coder. The buffer structure of the original 2400 bit/s MELPe is shown in Figure D-1. In order to avoid big pitch errors, the length of lookahead is increased in the 1200 bit/s coder by 129 samples. A pitch smoother was introduced in the 1200 bit/s coder. The buffer structure of the coder is shown in Figure D-2. The algorithmic delay for the coder in the 1200 bit/s mode is 103.75 ms.

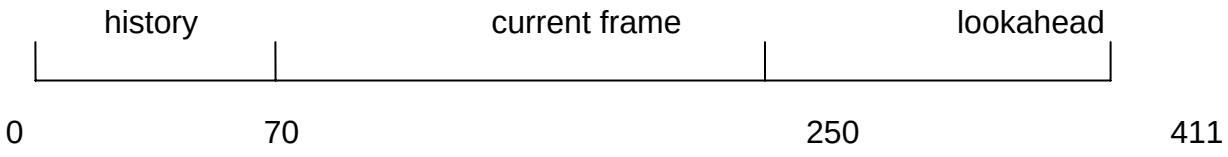


Figure D-1. The buffer structure of standard 2400 bit/s MELPe coder

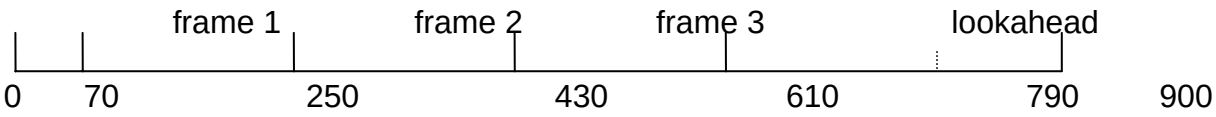


Figure D-2. The buffer structure of the 1200 bit/s coder

The transmitted parameters for the 1200 bit/s coder are the same as for the 2400 bit/s MELPe coder.

The 1200 bit/s coder has several modes that use different quantization schemes. Mode selection is done according to the UV patterns of the superframe. Techniques to reduce the effect of mode mismatch due to the channel errors have been developed and integrated into the decoder.

D.1.3 Bit Allocation

The bit allocation schemes for both 2400 bit/s and 1200 bit/s modes are shown in Table D-1.

Table D-1. Bit Allocation of both 2400 bit/s and 1200 bit/s modes

Parameters	Bits for quantization of three frames(540 samples)						
	2400 bit/s Voiced	2400 bit/s Unvoiced	1200 bit/s Mode1	1200 bit/s Mode2	1200 bit/s Mode3	1200 bit/s Mode4	1200 bit/s Mode5
Pitch & Global UV Decisions	7*3	7*3	12	12	12	12	12
Parity	0	0	1	1	1	1	1
LSFs	25*3	25*3	42	42	39	42	27
Gains	8*3	8*3	10	10	10	10	10
Bandpass Voicing	4*3	0	6	4	4	2	0
Fourier Magnitudes	8*3	0	8	8	8	8	0
Jitter	1*3	0	1	1	1	1	0
Synchronization	1*3	1*3	1	1	1	1	1
Error Protection	0	13*3	0	2	5	4	30
Total	162	162	81	81	81	81	81

*Note: 1200 bit/s Mode1: All three frames are voiced.
1200 bit/s Mode2: One of the first two frames is unvoiced, other frames are voiced.
1200 bit/s Mode3: The 1st and 2nd frames are voiced. The 3rd frame is unvoiced.
1200 bit/s Mode4: One of the three frames is voiced, other two frames are unvoiced.

1200 bit/s Mode5: All three frames are unvoiced.

D.1.4 Organization

This document is organized as follows. The speech analysis modules are described in section D-2. Section D-3 gives the quantization method for each transmitted parameter. The decoder description is included in Section D-4.

D.2 SPEECH ANALYSIS

D.2.1 Overview

The basic structure of the encoder remains the same as for the 2400 bit/s MELPe coder except a new pitch and bandpass-voicing smoother modules are added. The coder extracts the feature parameters of three successive frames using the same algorithm as in the MELPe standard. Then the pitch and bandpass voicing parameters are smoothed to take the benefit of the long block size.

D.2.2 Pitch Smoother

The pitch smoother takes the pitch estimates from the MELPe analysis module as the starting point. A set of new parameters is computed every 11.25 ms, i. e., each half frame. The nine computation positions in the current superframe are illustrated in Figure D-3.

In the 1200 bit/s model, we introduced frame classification into onset and offset frames in order to guide the pitch and class smoothing process. The new parameters introduced for onset/offset classification include:

(1) Energy

(2) Zero-crossing rate.

(3) Peakiness measurement in speech domain.

(4) Maximum correlation coefficient in pitch search range.

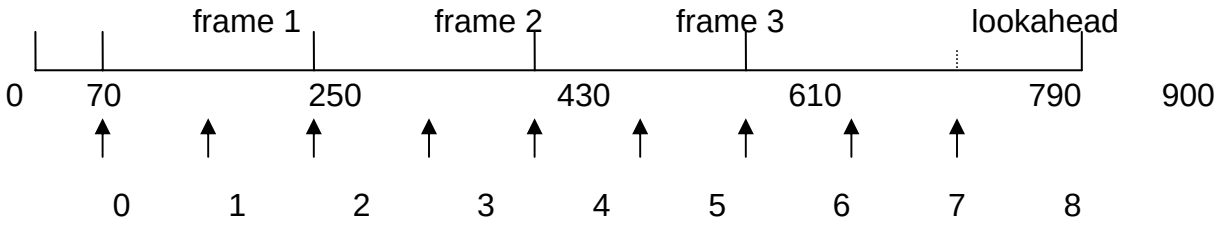


Figure D-3. The computation positions of parameters for pitch smoother

These new parameters are used to make rough U/V decisions for each half frame. The U/V decisions are used to classify the onset and offset frames. This classification is internal to the encoder and not transmitted. For each current frame, first the possibility of an offset is checked. An offset frame is selected if the current voiced frame is followed by a sequence of unvoiced frames, or the energy declines at least 8 dB within one frame or 12 dB within one and half frame. The pitch of an offset frame is not smoothed.

If the current frame is the first voiced frame, or the energy increases by at least 8 dB within one frame or 12 dB within one and a half frame, the current frame is classified as an onset frame. For the onset frames, a look-ahead pitch candidate is estimated from the maxima of autocorrelation function of the next frame. The look-ahead pitch candidate is selected as current pitch if the difference between the original pitch estimate and the look-ahead pitch is larger than 15%.

If the current frame is neither offset nor onset, the pitch variation is checked. If a pitch jump is detected, the pitch of the current frame is smoothed using interpolation between the pitch of the previous frame and the pitch of the next frame. For the third frame in the superframe, the pitch of the next frame is not available. A predictive pitch that is obtained from maxima of autocorrelation functions of the next frame is used instead of the pitch of the next frame.

The above pitch smoother can detect some of the gross pitch errors. In both DRT and DAM tests, the pitch smoother showed significant quality improvement.

D.2.3 Bandpass Voicing Smoother

First, the bandpass voicing information derived by the MELPe analysis module is mapped into one single cutoff frequency. The voicing information of the lowest band out of the total of 5 bands is determined from the U/V decision. The voicing information of the remaining 4 bands is valid only for voiced frames. The binary voicing decisions (1 for voiced and 0 for unvoiced) of the remaining 4 bands are mapped into 4 codewords using the 2-bit codebook shown below. The entries of the codebook are equivalent to four cutoff frequencies: 500 Hz, 1000 Hz, 2000 Hz and 4000 Hz.

Table D-2. Bandpass voicing index mapping

Codeword**:**	**0000**	**1000**	**1100**	**1111**
Voicing patterns assigned to the codeword.	0000 0001 0010 0011 0100 0101 0110	1000 1001 1010	1100	0111 1011 1101 1110 1111

For example, when the bandpass voicing pattern for a voiced frame is 1001, this index is then mapped into 1000 which corresponds to a cutoff frequency of 1000 Hz.

For the first two frames of the current block, the cutoff frequency is smoothed according to the bandpass voicing information of the previous frame and the next frame. There are three smoothing cases to be considered:

(1) If the cutoff frequencies of the previous frame and the next frame are both above 2000 Hz and the cutoff frequency of the current frame is lower than 2000 Hz, the cutoff frequency of the current frame may be set to be 1000 Hz or 2000 Hz depending on the energy and the bandpass voicing strengths.

(2) If the cutoff frequencies of the previous frame and the next frame are both above 1000 Hz and the cutoff for the current frame is below 1000 Hz, the cutoff frequency of the current frame may be set to be 1000Hz depending on the energy and the bandpass voicing strengths.

(3) If the cutoff frequencies of the previous frame and the next frame are all below 1000Hz, the cutoff frequency of current frame may be set to be 2000Hz according to the energy and the bandpass voicing strengths.

D.3 QUANTIZATION

D.3.1 Overview

The transmitted parameters of the 1200 bit/s coder are the same as those of the 2400 bit/s MELPe coder. The bit-allocation is shown in Table D-1. New quantization schemes are designed to take advantage of the long block size by using interpolation and VQ. The statistic properties of voiced and unvoiced speech are also taken into account. The same Fourier magnitude codebook of the 2400 bit/s coder is used in the 1200 bit/s coder in order to save memory and to make the transcoding easier.

D.3.2 Pitch Quantization Scheme

The pitch parameters are valid only for voiced frames. Different pitch quantization schemes are used for different U/V combinations across the three frames. The quantization schemes are summarized in Table D-3. For the superframes where the voicing patterns contain either 2 or 3 voiced frames, the pitch parameters are vector-quantized. For voicing patterns containing only one voiced frame, the same scalar quantizer as in the MELPe standard is applied. For the UUU voicing pattern, no bits are needed for pitch information.

Table D-3. Pitch quantization schemes

U/V pattern	Pitch quantization method
U U U	N/A
U U V	The pitch of the only voiced frame is scalar quantized using a 7-bit quantizer.
U V U	
V U U	
U V V	The pitches of the voiced frames are quantized using the same VQ as for the VVV case. A weighting function is applied which takes into account the U/V information.
V U V	
V V U	
V V V	Vector quantization of three pitches

Each pitch value, P, obtained from the pitch analysis of the 2400 bit/s standard is transformed to a logarithmic value, p = log P, before quantization.

For each superframe, a pitch vector is constructed with components equal to the log pitch value for each voiced frame and a zero value for each unvoiced frame. For voicing patterns with 2 or 3 voiced frames, the pitch vector is quantized using a VQ algorithm with a new distortion measure that takes into account the evolution of the pitch. This algorithm incorporates pitch differentials in the codebook search, which makes it possible to consider the time evolution of the pitch in selecting the VQ codebook entry.

The algorithm has three steps for obtaining the best index:

Step 1: Select the M-best candidates using the weighted squared Euclidean distance measure:

$$d = \sum_{i=1}^{3} w_i \left| p_i - \hat{p}_i \right|^2 \qquad \textbf{EQUATION D-1}$$

where $w_i = \begin{cases} 1, & \text{if the corresponding frame is voiced} \\ 0, & \text{if the corresponding frame is unvoiced.} \end{cases}$

and p_i is the unquantized log pitch, $\hat{p}_i$ is the quantized log pitch value. The above equation indicates that only voiced frames are taken into consideration in the codebook search.

Step 2: Calculate differentials of the unquantized log pitch values using

$$\Delta p_i = \begin{cases} p_i - p_{i-1} & \text{if i - th and (i - 1) - th frames are voiced} \\ 0 & \text{else} \end{cases} \qquad \textbf{EQUATION D-2}$$

for *i* = 1, 2, 3, where p_0 is the last log pitch value of the previous superframe. For the candidate log pitch values selected in step 1, calculate differentials of the candidates by replacing Δp_i and p_i by $\Delta \hat{p}_i$ and $\hat{p}_i$ respectively in equation D-2 where $\hat{p}_0$ is the quantized version of p_0.

Step 3: Select the index that minimizes

$$d' = \sum_{i=1}^{3} w_i \left| p_i - \hat{p}_i \right|^2 + \delta \sum_{i=1}^{3} \left| \Delta p_i - \Delta \hat{p}_i \right|^2 = d + \delta \sum_{i=1}^{3} \left| \Delta p_i - \Delta \hat{p}_i \right|^2 \qquad \textbf{EQUATION D-3}$$

where δ is a parameter to control the contribution of pitch differentials which is set to be 1.

For superframes containing only one voiced frame, scalar quantization of the pitch is performed. The pitch value is quantized on a logarithmic scale with a 99-level uniform quantizer ranging from 20 to 160 samples. The quantizer is the same as that in the 2.4 kbit/s standard.

D.3.3 Joint Quantization of Pitch and U/V Decisions

The U/V decisions and pitch parameters are jointly quantized using 12 bits. The joint quantization scheme is summarized in Table D-4. In this scheme, the 12-bit allocation is divided into 3 mode bits representing the U/V decisions and the remaining 9 bits used for pitch values. The scheme employs six types of 9-bit codebooks according to the bit patterns of the 3-bit codebook. Four codebooks are assigned to the VVV type superframes, which means that the pitch vectors in these superframes are quantized by a 2048-level vector quantizer. If the number of voiced frames in the superframe is no larger than one, the 3-bit mode pattern is set to 000 and the distinction between different modes is done in the 9-bit codebook. Note that the latter case includes the modes UUU, VUU, UVU, and UUV and the nine available bits can easily represent the mode information as well as the pitch value.

Table D-4. Joint quantization scheme of pitch and voicing decisions

<table>
<tr><th>U/V patterns</th><th>3-bit codebook</th><th>9-bit codebook</th></tr>
<tr><td>UUU</td><td rowspan="4">000</td><td rowspan="4">The pitch value is quantized with the same 99-level uniform quantizer as the 2.4kbit/s standard. The pitch value and U/V pattern are then mapped to this 9-bit codebook.</td></tr>
<tr><td>UUV</td></tr>
<tr><td>UVU</td></tr>
<tr><td>VUU</td></tr>
<tr><td>VVU</td><td>001</td><td rowspan="3">These U/V patterns share the same codebook containing 512 code vectors of the pitch triple.</td></tr>
<tr><td>VUV</td><td>010</td></tr>
<tr><td>UVV</td><td>100</td></tr>
<tr><td rowspan="4">VVV</td><td>011</td><td>512-level codebook A</td></tr>
<tr><td>101</td><td>512-level codebook B</td></tr>
<tr><td>110</td><td>512-level codebook C</td></tr>
<tr><td>111</td><td>512-level codebook D</td></tr>
</table>

D.3.4 Parity Bit

To improve robustness with transmission errors, a parity bit is computed and transmitted for the three mode bits defined at D.3.3.

D.3.5 LSF Quantization

The bit allocation for quantizing the line spectrum frequencies (LSFs) is shown in Table D-5. In the table, the original LSF vectors for the three frames are denoted by l_1, l_2, l_3.

For UUU, UUV, UVU and VUU modes, the LSF vectors of unvoiced frames are quantized using a 9-bit codebook, while the LSF vector of the voiced frame is quantized with a 24-bits MSVQ quantizer. The LSF vectors of other U/V patterns are encoded using a forward-backward interpolation scheme. The scheme works as follows. Denote the quantized LSF vector of the previous frame by $\hat{l}_p$. First the LSFs of the last frame in the current superframe, l_3, are directly quantized to $\hat{l}_3$ using the 9-bit codebook for unvoiced frames or the same MSVQ as the MELPe for voiced frames. Then, the predicted values of l_1 and l_2 are obtained by interpolating $\hat{l}_p$ and $\hat{l}_3$ using the following equations

$$\begin{aligned}\tilde{l}_1(j) &= a_1(j)\cdot\hat{l}_p(n) + [1-a_1(j)]\cdot\hat{l}_3(j)\\ \tilde{l}_2(j) &= a_2(j)\cdot\hat{l}_p(j) + [1-a_2(j)]\cdot\hat{l}_3(j) \qquad j=1,...,10\end{aligned}$$

EQUATION D-4

where $a_1(j)$ and $a_2(j)$ are the interpolation coefficients. The coefficients are stored in a codebook and the best coefficients are selected by minimizing the distortion measure:

$$E=\sum_{j=1}^{10} w_1(j)\left|l_1(j)-\tilde{l}_1(j)\right|^2+\sum_{j=1}^{10} w_2(j)\left|l_2(j)-\tilde{l}_2(j)\right|^2$$

EQUATION D-5

After obtaining the best interpolation coefficients, the residual LSP vector for the frames 1 and 2 is computed by

$$\begin{aligned}r_1(j) &= l_1(j)-\tilde{l}_1(j)\\ r_2(j) &= l_2(j)-\tilde{l}_2(j) \qquad j=1,...,10\end{aligned}$$

EQUATION D-6

The 20-dimension residual vector $\mathbf{R}=[r_1(1),r_1(2),...,r_1(10),r_2(1),r_2(2),...,r_2(10)]$ is then quantized using weighted multi-stage vector quantization.

Table D-5. Bit allocation for LSF quantization according to UV decisions

U/V pattern	LSF l_1	LSF l_2	LSF l_3	Interpolation	Residual of l_1 and l_2	Total
U U U	9	9	9	0	0	27
V U U	8+6+5+5	9	9	0	0	42
U V U	9	8+6+5+5	9	0	0	42

U U V	9	9	8+6+5+5	0	0	42
U V V V U V V V V	0	0	8+6+5+5	4	8+6	42
V V U	0	0	9	4	8+6+6+6	39

D.3.6 Gain Quantization

In the 1.2 kbit/s coder, two gain parameters are calculated per frame, i.e., 6 gains are obtained for each superframe. The 6 gain parameters are vector-quantized using a 10 bits vector quantizer with a MSE criterion defined in the logarithmic domain.

D.3.7 Bandpass Voicing Quantization

The voicing information of the lowest band out of the total of 5 bands is determined from the U/V decision. The voicing decisions of the remaining 4 bands are employed only for voiced frames. The binary voicing decisions (1 for voiced and 0 for unvoiced) of the 4 bands are quantized using the 2-bit codebook shown in Table D-2. This procedure results in two bits being used for voicing in each voiced frame. The bit allocation required in different coding modes for bandpass voicing quantization is shown in Table D-6.

Table D-6. Bit Allocation for bandpass voicing quantization

UV decisions pattern	VVV	VVU, VUV, UVV	VUU, UVU, UUV	UUU
Bits for bandpass voicing information	6	4	2	0

D.3.8 Quantization of Fourier Magnitudes

The Fourier magnitude vector is computed only for voiced frames. The quantization procedure for Fourier magnitudes is summarized in Table D-7. The unquantized Fourier magnitude vectors for the three frames in a superframe are denoted as $f_i, i = 1,2,3$. Denote by f_0 the Fourier magnitude vector of the last frame in the previous superframe

and by $\hat{f}_i$ the quantized vector f_i. Denote by Q(.) the quantizer function for the Fourier magnitude vector when using the same 8-bit codebook as the MELPe standard. The quantized Fourier magnitude vectors for the three frames in a superframe are obtained as shown in Table D-7.

Table D-7. Fourier magnitude vector quantization

U/V pattern for current superframe	U/V decision for the last frame of the previous superframe	
	U	**V**
UUU	N/A	
VUU	$\hat{f}_1 = Q(f_1)$	
UVU	$\hat{f}_2 = Q(f_2)$	
UUV	$\hat{f}_3 = Q(f_3)$	
UVV	$\hat{f}_3 = Q(f_3), \quad \hat{f}_2 = \hat{f}_3$	
VUV	$\hat{f}_3 = Q(f_3), \quad \hat{f}_1 = \hat{f}_3$	$\hat{f}_3 = Q(f_3), \quad \hat{f}_1 = \hat{f}_0$
VVU	$\hat{f}_2 = Q(f_2), \quad \hat{f}_1 = \hat{f}_2$	$\hat{f}_2 = Q(f_2), \quad \hat{f}_1 = \frac{\hat{f}_0 + \hat{f}_2}{2}$
VVV	$\hat{f}_2 = Q(f_2), \quad \hat{f}_1 = \hat{f}_2 = \hat{f}_3$	$\hat{f}_3 = Q(f_3),$ $\hat{f}_1 = \frac{2 \cdot \hat{f}_0 + \hat{f}_3}{3}, \hat{f}_2 = \frac{\hat{f}_0 + 2 \cdot \hat{f}_3}{3}$

D.3.9 Aperiodic Flag Quantization

The 1200 bit/s coder uses 1-bit per superframe for the quantization of the aperiodic flag. In the 2400 bit/s MELPe standard, the aperiodic flag requires one bit per frame, i.e., three bits per superframe. The compression to one bit per superframe is obtained using the quantization procedure shown in Table D-8. In the table, “J” and “-” mean that the aperiodic flag is set and not set, respectively.

Table D-8. Aperiodic flag quantization using 1 bit

U/V pattern	Quantization procedure	Quantization patterns	
		New flag = 0	**New flag=1**

U U U	N/A	J J J	J J J
U U V	If the voiced frame has aperiodic flag, set new flag.	J J -	J J J
U V U		J - J	J J J
V U U		- J J	J J J
U V V	If the second frame has aperiodic flag, set new flag.	J - -	J J -
V V U		- - J	- J J
V U V	N/A	- J -	- J -
V V V	If more than 1 frame has the aperiodic flag set, set new flag.	- - -	J J J

D.3.10 Error Protection

D.3.10.1 Mode Protection

Besides the parity bit, additional mode error protection techniques are applied to superframes by employing the spare bits that are available in all superframes, except the superframes in the VVV mode. The 1200 bit/s coder uses two bits for the quantization of the bandpass voicing for each voiced frame. Hence, in superframes that have one unvoiced frame, two bandpass voicing bits are spare and can be used for mode protection. In superframes that have two unvoiced frames, four bits can be used for mode protection. In addition, 4 bits of LSF quantization can be used for mode protection in the UUU and VVU modes. Table D-9 shows how these mode protection bits are used.

Table D-9. Mode protection schemes

U/V pattern	3-b codebook of joint quantization for pitch and U/V decisions	Bit pattern of bandpass voicing 1	Bit pattern of bandpass voicing 2	Bit pattern of LSF
U U U	000	00	00	0000
U U V		00	01	-
U V U		00	10	-
V U U		00	11	-
V V U	001	01	-	0101
V U V	010	10	-	-

U V V	100	11	-	-
V V V	011, 101, 110, 111	-	-	-

D.3.10.2 FEC for UUU Superframe

In the UUU mode, the first 8 MSBs of the gain index are divided into two groups of 4 bits and each group is protected by the Hamming (8,4) code. The remaining 2 bits of the gain index are protected with the Hamming (7,4) code. Note that the Hamming (7,4) code corrects single-bit errors, while the (8,4) code corrects single errors and in addition detects double-bit errors. The LSFs bits for each frame in the UUU superframes are protected by the CRC (13,9) code. The (13,9) code detects single and double-bit errors.

D.3.10.3 Bit transmission order

Table D-9a shows the transmission order for the 81 bits in each frame when all three of the underlying MELPe analysis frames are unvoiced (Mode 5). Table D-9b shows the bit transmission order for all other modes. The sync bit (SYN) alternates between 0 and 1 from frame to frame.

Table D-9a. 1200 bit/s MELPe bit transmission order – Mode 5, All Unvoiced

Byte No.	0 (LSB)	1	2	3	4	5
1	Syn	Pitch&UV0	Pitch&UV1	Pitch&UV2	Pitch&UV3	Pitch&UV4
2	Pitch&UV5	Pitch&UV6	Pitch&UV7	Pitch&UV8	Pitch&UV9	Pitch&UV10
3	Pitch&UV11	LSP0	LSP1	LSP2	LSP3	LSP4
4	LSP5	LSP6	LSP7	LSP8	LSP9	LSP10
5	LSP11	LSP12	LSP13	LSP14	LSP15	LSP16
6	LSP17	LSP18	LSP19	LSP20	LSP21	LSP22
7	LSP23	LSP24	LSP25	LSP26	Gain0	Gain1
8	Gain2	Gain3	Gain4	Gain5	Gain6	Gain7
9	Gain8	Gain9				
10						
11						
12						
13						
14						

Table D-9b. 1200 bit/s MELPe bit transmission order – Mode 1, 2, 3 ,4

Byte No.	0 (LSB)	1	2	3	4	5
1	Syn	Pitch&UV0	Pitch&UV1	Pitch&UV2	Pitch&UV3	Pitch&UV4
2	Pitch&UV5	Pitch&UV6	Pitch&UV7	Pitch&UV8	Pitch&UV9	Pitch&UV10
3	Pitch&UV11	LSP0	LSP1	LSP2	LSP3	LSP4
4	LSP5	LSP6	LSP7	LSP8	LSP9	LSP10
5	LSP11	LSP12	LSP13	LSP14	LSP15	LSP16
6	LSP17	LSP18	LSP19	LSP20	LSP21	LSP22
7	LSP23	LSP24	LSP25	LSP26	LSP27	LSP28
8	LSP29	LSP30	LSP31	LSP32	LSP33	LSP34
9	LSP35	LSP36	LSP37	LSP38	LSP39	LSP40
10	LSP41	LSP42	Gain0	Gain1	Gain2	Gain3
11	Gain4	Gain5	Gain6	Gain7	Gain8	Gain9
12	BP0	BP1	BP2	BP3	BP4	BP5
13	Jitter	FS0	FS1	FS2	FS3	FS4
14	FS5	FS6	FS7			

BP: Band pass voicing
FS: Fourier magnitudes

D.4 DECODER

D.4.1 Bit Unpacking and Error Correction

The received bits are unpacked from the channel and assembled into parameter codewords. Since the decoding procedures for most parameters depend on the mode (the U/V pattern), the 12 bits allocated for pitch and voicing (U/V) decisions are decoded first. For the bit pattern 000 in the 3-bit codebook, the 9-bit codeword specifies one of the UUU, UUV, UVU, and VUU modes. If the code of the 9-bit codebook is all-zeros or has one bit set, the UUU mode is used. If the code has two bits set or specifies an index unused for pitch, a frame erasure is indicated.

After decoding the voicing (U/V) pattern, the resulting mode information is checked using the parity bit and the mode protection bits. If an error is detected, a mode correction algorithm is performed. The algorithm tries to correct the mode error using the parity bits and mode protection bits. In the case that an uncorrectable error is detected, different decoding methods are applied for each parameter according to the mode error patterns. In addition, if a parity error is found, a parameter-smoothing flag is set. The correction procedures are described in Table D-10.

Table D-10. Parameter decoding schemes if a mode error is detected

U/V voicing pattern	Corrected U/V pattern	LSFs	Gain	Pitch	Bandpass voicing	Fourier Magnitude
UUU	UUU	Repeat LSFs of the last frame in the previous superframe	Decode and apply smoothing		Set to 0	Set to 1 all magnitudes
UUV						
UVU						
VUU						
VVU	VVV	Decode and apply smoothing	Decode and apply smoothing	Decode and apply smoothing	Set the first band to 1, others to 0	
VUV						
UVV						

In the UUU mode, assuming no errors were detected in the mode information, the two (8,4) Hamming codes representing the gain parameters are decoded to correct single bit errors and detect double errors. If an uncorrectable error is detected, a frame erasure is indicated. Otherwise, the (7,4) Hamming code for gain and the (13,9) CRC codes for LSFs are decoded to correct single errors and detect single and double errors, respectively. If an error is found in the CRC (13,9) codes, the incorrect LSFs are replaced by repeating previous LSFs or interpolating between the neighboring correct LSFs.

If a frame erasure is detected in the current superframe by the Hamming decoder, or an erasure is directly signaled from the channel, a frame repeat mechanism is implemented. All the parameters of the current superframe are replaced with the parameters form the last frame of the previous superframe.

For a superframe in which an erasure is not detected, the remaining parameters are decoded. If smoothing is needed, the parameter after smoothing is obtained by

$$x = 0.5\hat{x} + 0.5x' \qquad \textbf{EQUATION D-7}$$

where $\hat{x}$ and x' represent the decoded parameter of the current frame and the corresponding parameter of the previous frame, respectively.

D.4.2 Pitch Decoding

The pitch decoding is done as shown in Table D-4. For the unvoiced frames, the pitch value is set to 50 samples.

D.4.3 LSF Decoding

The LSFs are decoded as described in Section D.3.5 and Table D-5. The LSFs are checked for ascending order and minimum separation.

D.4.4 Gain Decoding

The gain index is used to retrieve a codeword containing six gain parameters from the 10-bit VQ gain codebook.

D.4.5 Decoding of Bandpass Voicing

In the unvoiced frames, all of the bandpass voicing strengths are set to zero. In the voiced frames, Vbp_1 is set to 1 and the remaining voicing patterns are decoded as shown in Table D-2.

D.4.6 Decoding of Fourier Magnitudes

The Fourier magnitudes of unvoiced frames are set to be equal to 1. For the last voiced frame of the current superframe, the Fourier magnitudes are decoded directly. The Fourier magnitudes of other voiced frames are generated by repetition or linear interpolation as shown in Table D-7.

D.4.7 Aperiodic Flag Decoding

The aperiodic flags are obtained from the new flag as shown in Table D-8. The jitter is set to 25% if the aperiodic flag is 1, otherwise the jitter is set to 0%.

D.4.8 MELPe Synthesis

The basic structure of the decoder is the same as in the 2400 bit/s MELPe except the new harmonic synthesis method that was introduced to generate the excitation signal for each pitch cycle. In the new harmonic synthesis algorithm, the mixed excitation is generated completely in the frequency domain and then an inverse Discrete Fourier Transform is used to convert it to the time domain.

The harmonic synthesis procedure generates the excitation in the frequency domain based on two parameters, the cutoff frequency F and the Fourier magnitude vector $A_l, l = 1,2,...,L$. The cutoff frequency is obtained from the bandpass voicing parameters and then interpolated for each pitch cycle. The Fourier magnitudes are interpolated the same way as in the 2400 bit/s MELP rate.

Denote the pitch length as N, then the corresponding fundamental frequency is $f_0 = 2\pi/N$. The Fourier magnitude vector length is $L = \lfloor N/2 \rfloor$. Two transition frequencies F_H and F_L are determined according to the cutoff frequency F employing an empirically derived algorithm. The transition frequencies are in the range $F_L \in [0.85F, 0.98F]$, $F_H \in [F, 1.05F]$. These transition frequencies are equivalent to two

DFT frequency component indices V_H and V_L. A voiced model is used for all the frequency samples below V_L, a mixed model is used for frequency samples between V_L and V_H, and an unvoiced model is used for frequency samples above V_H. To define the mixed mode, a gain factor *g* is selected with the value depending on the cutoff frequency (the higher is the cutoff frequency *F*, the smaller is the gain factor). The frequency components of the excitation are determined as follows,

$$|X(l)| = \begin{cases} A_l & l < V_L \\ \frac{l - V_L}{V_H - V_L} \cdot g \cdot A_l + \frac{V_H - l}{V_H - V_L} \cdot A_l & V_L \le l \le V_H \\ g \cdot A_l & l > V_H \end{cases}$$ **EQUATION D-8**

$$\angle X(l) = \begin{cases} l\phi_0 & l < V_L \\ l\phi_0 - \frac{l - V_L}{V_H - V_L} \cdot \phi_{RND} & V_L \le l \le V_H \\ \phi_{RND} & l > V_H \end{cases}$$ **EQUATION D-9**

where ϕ_0 is a constant selected such as to avoid a pitch pulse at the pitch cycle boundary. The ϕ_{RND} is a random number between 2π and -2π. The frequency samples of mixed excitation are then converted to time domain using an inverse Discrete Fourier Transform.

D.4.9 Optional Postfilter

Once a frame of synthesized speech is generated, an optional postfilter may be applied. The frame is divided into four equal sized subframes. The LSFs are linearly interpolated for each subframe, then converted to LPC coefficients, $a_i, i = 1,2,...,10$, and reflection coefficients, $r_i, i = 1,2,...,10$. The transfer function of the postfilter is given by

$$H_{pf}(z) = \frac{\sum_{i=0}^{10} a_i \gamma^i z^{-i}}{\sum_{i=0}^{10} a_i \beta^i z^{-i}} (1 - \mu z^{-1})$$ **EQUATION D-10**

where $\gamma = 0.56, \beta = 0.75$. μ is determined by the reflection coefficients; if $t = \prod_{i=0}^{10} (1 - r_i^2)$, is larger than 0.3, $\mu = 0.0$, otherwise $\mu = 0.2$.
For each subframe, a gain adjustment factor *g* is computed as follows:

$$e_{in} = \sum_{n=0}^{N-1} |s(n)|, \quad e_{out} = \sum_{n=0}^{N-1} |s'_{pf}(n)| \qquad \textbf{EQUATION D-11}$$

$$g = \begin{cases} \dfrac{e_{in}}{e_{out}} & e_{out} >= 64 \\ g_{prev} & \text{else} \end{cases} \qquad \textbf{EQUATION D-12}$$

where N = 45 is the subframe size, $s(n)$ is the synthesized speech signal before applying the postfilter, and $s'_{pf}(n)$ is the output of the filter $H_{pf}(z)$. g_{prev} is the gain adjustment factor of the previous subframe. Then the gain-scaled signal is given by

$$s_{pf}(n) = \left[\left(1 - \frac{n}{N}\right) g_{prev} + \frac{n}{N} g \right] s'_{pf}(n) \qquad n = 0,1,...,N-1 \qquad \textbf{EQUATION D-13}$$

Finally $s_{pf}(n)$ is passed through a low-pass filter and a high-pass filter to generate the final output signal. Both filters are 2nd order Butterworth filters with cutoff frequencies at 3800 Hz and 60 Hz respectively.

BLANK PAGE BLANCHE

ANNEX E

Description of the Noise Preprocessor

E.1 OVERVIEW

The noise preprocessing enhancement algorithm defined in this annex improves the estimation of spectral parameters by the MELPe coder. This noise preprocessor shall be used in conjunction with the voice coder in order to produce the optimum quality in the reconstructed speech signal.

The algorithm enhances the speech input signal by estimating the signal-to-noise ratio in the input signal, and modifying the input signal based on the estimates. The enhancement (or noise pre-processing) is done before the rest of the MELPe vocoder process is started. The noise pre-processor is an independent process that precedes the MELPe analysis process. The original input signal is, in effect, replaced by the enhanced speech signal produced by the enhancement algorithm, and the MELPe algorithm is applied to the replacement speech signal. As a result of the enhancement process, there is a delay between the original speech input signal and the enhanced speech signal provided to the vocoder. The delay is 9.5 ms for the MELPe algorithm. For information, a similar approach is discussed in reference [E2].

The estimates of the signal-to-noise ratio are made in the frequency domain after performing a Fourier transform on the input speech signal. Through the enhancement process that is defined in this annex, the power spectral density of the noise, and other related parameters are estimated and used to compute a spectral gain function that is applied multiplicatively to the real and imaginary components produced by the Fourier transform of the speech signal. The enhanced components of the Fourier transform are then processed by an inverse Fourier transform in order to construct the enhanced speech signal.

Successive groups of 180 speech samples corresponding to the number of samples used in a single frame of the 2400 bit/s MELPe coder are provided to the noise preprocessor. These are concatenated with the immediately preceding 76 speech samples in order to provide noise pre-processor analysis frames of 256 speech samples, as required by the use of a Fast Fourier Transform. A modified Tukey windowing function is applied to the input speech signal before the Fourier transform, and again to the enhanced speech signal after the inverse Fourier transform has been performed. The resulting 256 speech samples of the enhanced speech waveform are overlapped by 76 samples with the prior enhanced segment, and the overlapping 76 samples are added together. This results in 180 newly enhanced speech samples that are returned for use by the MELPe coder. The remaining, and most recent 76 additional samples are held in reserve for use with the next frame. The basic process is depicted in Figure E-1 below and will be described in depth in the subsequent sections

of the annex.

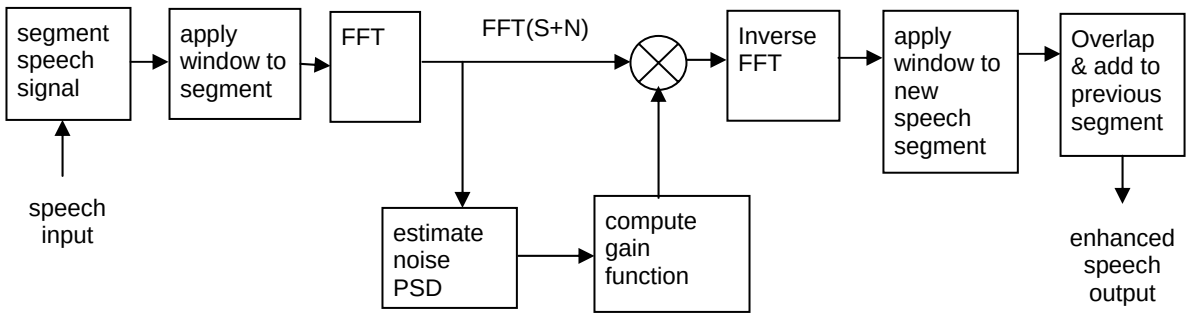
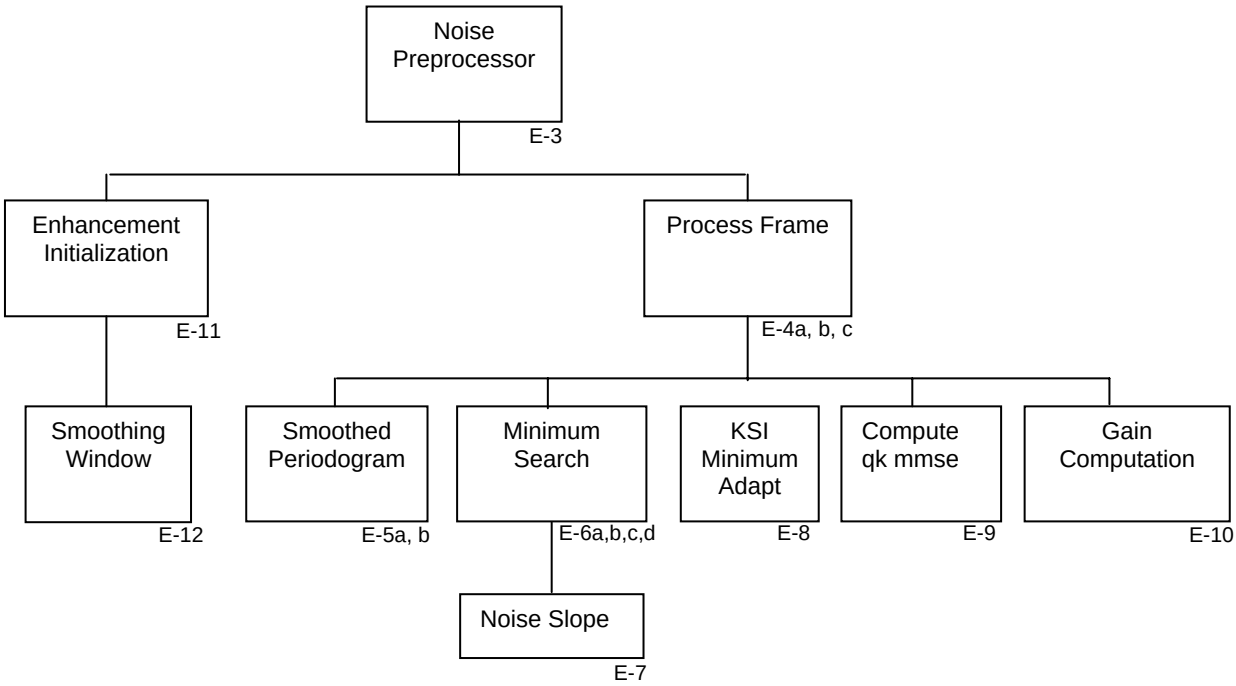


Figure E-1. Top Level View of Noise Preprocessor

E.2 DETAIL DESCRIPTION

The noise preprocessor algorithm will be described through a series of flow charts and accompanying text. Since the overall process builds up the estimates iteratively, it is necessary to initialize each part of the process the first time through. These initializations will be seen at the beginning of the overall process in the module shown as “Enhancement Initialization” that, in turn, invokes a process “Smoothing Window.” Following initialization, the Noise Preprocessor module assembles 256 input speech samples and applies a windowing function to the samples. These samples are then treated as a 256-sample frame for subsequent analysis, although as indicated in the introduction, only 180 of these samples are new, since each new frame of 256 samples processed includes 76 samples of the previous frame. Successive groups of 256 windowed speech samples are then subjected to a series of steps invoked by the module “Process Frame.” These steps are depicted in Figure E-2 with each step invoked once from left to right. As will be described with the module “Process Frame,” these steps are not immediately contiguous as computations are made on the results of each step in order to prepare the inputs for the next step. Each of these processes is described in turn in the figures indicated underneath each of the corresponding blocks. The module “Process Frame” is, for example described in detail in Figures E-4a, E-4b, and E-4c.

Figure E-2. Structure of the Noise Preprocessing Process and Description



E.3 FLOW CHARTS, PSEUDO CODE, AND FIXED POINT ARITHMETIC

By the nature of the noise preprocessor algorithm, the most precise description of the specified algorithm is through the use of flow charts and pseudo code. The pseudo code contained in the flow charts in this annex follows the convention of the C++ language, but should not be interpreted as specifying the use of C++ or the types of variables implied by the use of the C++ language for the pseudo code. In fact, this STANAG is specified for use in C fixed point arithmetic.

Corresponding to the use of fixed point arithmetic, some of the variables represented in the flow charts will be capable of being represented in 16-bit numbers, while to maintain required accuracy, other values will need to be represented in block floating point values, typically containing a value and an exponent or power of 2 for each entry. Thus, an implementation in fixed point arithmetic should interpret the flow charts and tables provided as defining the algorithm, leaving the task of implementing the algorithm in fixed point arithmetic to the software developer. The fixed point C software for this noise preprocessor is provided in Annex G of STANAG-4591. This description is a more precise depiction of the fixed point aspects of the processing, and will illustrate the means of translating these descriptive flow charts into fixed point arithmetic.

E.4 NOISE PREPROCESSOR TOP LEVEL FUNCTIONS

The diagram shown in Figure E-3 describes the top level of the noise preprocessing function. It is understood that during the overall implementation of the MELPe coder there will be mechanisms to acquire the speech input signal from the analog system through a sampling process. These samples are acquired outside of the procedures described in this Annex. Between the time the speech signals are received by the noise preprocessor and the time they are returned to the MELPe coder for further processing they are enhanced by the noise preprocessing process described in this annex. Thus, external to the process shown in Figure E-3, 180 speech samples, corresponding to 22.5 ms of speech are assembled for processing and presented to the noise preprocessor for enhancement.

The top level function of the noise preprocessor concatenates the 180 speech samples received with the last
76 samples from the previously received set of speech samples to assemble a frame of the most recent 256 speech samples for subsequent processing. These most recent 256 speech samples are provided to the function identified as “process_frame” which returns a set of 256 enhanced speech samples. The oldest 76 of these enhanced speech samples overlap the most recent 76 enhanced speech samples in the frame previously enhanced. These overlapping samples are added together in a process that minimizes the interframe distortions. Following this addition, the most recent 76 enhanced speech samples are saved for use with the next frame, and the 180 enhanced speech samples immediately preceding those saved are returned for use by the rest of the MELPe processing.

As a result of the enhancement process, there will be a delay of 76 samples, corresponding to the 76 enhanced speech samples saved for use with the next frame.

The noise preprocessing is identical for the 1200 bit/s and 2400 bit/s versions of MELPe except for a minor difference in the speech data used as input to the initial estimates made of the power spectral density (the array lambdaD), the noise power (noise_power), and long term signal-to-noise ratio (lonterm_snr) the first time the noise preprocessor is invoked. The difference in these initial estimates is the result of the ability of the 1200 bit/s version to use a full 256 samples of the speech waveform for the initial estimates since the 1200 bit/s version that must have three full frames of 180 speech samples available before it begins, while the 2400 bit/s version begins processing with the first 180 voice samples, and must pad these first 180 samples with zeroes to fill out the 256 sample buffer before the samples are provided as input to the windowing and FFT calculations. This difference appears in the flow charts in Figure E-3.

E.5 PROCESSING FRAMES

Within the functions defined in Figure E-4, the 256 speech samples received into the array “inspeech” are first weighted by multiplying each of the received samples by the corresponding entry in Table E-1. Each entry in Table E-1 represents the square root of the entries in a Tukey windowing function. These weighted speech samples are contained in the array “ybuf” and are used for the rest of the enhancement process.

When the enhancement process is complete and the new set of 256 enhanced speech samples are available, they are also weighted by the square root of the Tukey windowing function before they are returned back to the top level function of the noise preprocessor. This before and after weighting by the square root of the Tukey window permits the 76 overlapping speech samples to be added together in order to smooth across frame boundaries before the enhanced speech data is provided to the remainder of the MELPe process. This weighting process has been found to give “good performance when used as an analysis and synthesis window.

After the 256 speech samples are weighted, they are then processed by a 256-point Fast Fourier Transform in order to compute the real and imaginary components of the spectral representation of the speech samples. These spectral values contained in the array “sigbuf” are used to begin a sequence of processing steps that eventually provide a gain function used to adjust the real and imaginary spectral values. The modified spectral values are then processed by an inverse Fast Fourier Transform to produce the enhanced speech samples. Although the spectral values are used as the basis of the enhancement algorithm, they are also saved in the array “sigbuf” for use after the gain function has been computed.

The spectral values of the speech samples are used to estimate the noise contained in the speech signal in order to provide the gain function used to enhance the speech samples for subsequent processing. In order to perform the noise estimate, the real and imaginary components contained in the array “sigbuf” are used to compute the magnitude spectrum. This computation provides a 129-point magnitude spectrum contained in the array “ymag.” The spectral values are normalized by dividing each spectral magnitude sample by 256 -- the total number of speech samples used in the analysis window. The normalized spectral magnitudes are kept in the array “yy.” In Figure E-4, WIN_LEN representing the number of speech samples is equal to 256.

The average value of the normalized spectral magnitude is computed, and provided along with the normalized spectral magnitudes to the function “Smoothed_Periodogram.” This function computes the smoothed periodogram saved into the global array “smoothedspect” and the relative variance of the smoothed periodogram returned in the array “var_rel.” The smoothed periodogram and relative variance are used by the function “min_search” that searches for the minimal values of the smoothed periodogram in each frequency bin, and uses these minimum values to

estimate the power spectral density of the noise contained in the speech signal under the assumption that the minimum values occur when speech is not present, and thereby represent the noise contained in the signal. Details and refinements of the process for smoothing the spectral magnitudes and computing the variance (Figures E-5a,b) and searching for the minimum (Figures 6a,b,c,d) are essential and will be described in subsequent paragraphs. While the details will come later, the estimate of the power spectral density of the noise is contained in the globally available array "lambdaD." As a result of the processing by the function "min_search", the estimate of the noise contained in "lambdaD" is one that reflects the recent history of the speech samples, not just those contained in the current set of speech samples under analysis. In effect, the current set of speech samples is used to update the running estimate of the noise power spectral density.

The estimates of the power spectral density of the noise in each frequency bin are used to compute the signal-to-noise ratio for each frequency bin with the result contained in the array "gamaK." These signal-to-noise ratios are computed by dividing the normalized spectral magnitudes "yy" by the estimated noise spectral magnitudes "lambdaD." This division is executed on a frequency bin basis.

The average and maximum values of the signal-to-noise ratios across the frequency bins are computed and tested against pre-determined thresholds of 2/NOISE_BIAS and 50/NOISE_BIAS respectively, where NOISE_BIAS = SQRT (2.0). If both of the average and maximum signal to noise ratios are less than the pre-determined thresholds, then the preliminary indication is that the current set of samples contains only noise. This preliminary indication is overridden, however, if the signal-to-noise ratio in the current frame is greater than 3dB.

The signal-to-noise ratios computed in the array gamaK contain the a posteriori values of the signal-to-noise ratio in each frequency bin for the current set of samples. These a posteriori values of the signal-to-noise ratios will be used in conjunction with a similar set of a priori values to estimate the probabilities that the power spectral values in each frequency bin represent only noise.

The process estimates the a priori signal-to-noise ratio in each frequency bin for each new set of speech samples. The results are held in the array "ksi." An initial estimate of the a priori signal-to-noise ration is computed using the power spectral density for the enhanced samples from the previous frame and the running estimate of the power spectral density of the noise. These two factors are contained in the arrays "aga12" and "lambdaD" respectively. The power spectral density for the enhanced speech samples is computed after the enhancement gains are computed by multiplying the squared gain values by the unnormalized spectral magnitudes in the array "ymag." These are saved for use in the next frame in the array "aga12." The value for the a priori signal-to-noise ratio in each frequency bin is set as 0.93*(aga12/(lambdaD*256)). This weight of 0.93 is an experimentally determined value derived from a Rate Factor of 0.8 that is used in the computation [0.99 – ((Rate Factor)/16)*.12].

These initial estimates of the a priori signal-to-noise ratios are increased slightly when the power spectral density in the frequency bin is greater than a threshold of 1.0/NOISE_BIAS. In this case, the amount by which the power spectral density exceeds the threshold is weighted by 0.07 (=1.0-0.93) and added to the initial estimate.

These initial estimates of the a priori signal-to-noise rations are then adjusted so that none of the signal-to-noise ratios fall below a lower limit identified as the value "ksi_min_var." A new lower limit for the signal-to-noise ratios is determined for each frame of samples processed. The new minimum value for the current frame is composed of two components. The first component is the minimum value from the previous frame that is included with a weight of 90% to assure smooth transitions between frames. The remaining 10% of the new minimum depends on whether the frame is identified as containing noise only, or a combination of speech plus noise.

When the frame is thought to contain only noise, the second component is chosen as 10% of an experimentally determined value known as GM_MIN = 0.12 resulting in the new minimum computed as shown in Equation E-1.

When the frame is thought to contain speech and noise, however, an additional factor is included that takes into account the long-term value of the signal-to-noise ratio. In this case, the minimum threshold is computed as shown in Equation E-2.

$$ksi\_min\_var = 0.9 * ksi\_min\_var + 0.1 * 0.12 \quad \textbf{EQUATION E-1}$$

$$ksi\_min\_var = 0.9 * ksi\_min\_var + 0.1 * 0.12 * \exp\{-5\}(0.5 + longterm\_snr)^{0.65} \quad \textbf{EQUATION E-2}$$

A limit is set, however, on the amount that the long-term signal-to-noise-ratio can increase the threshold. This limit is chosen so that the factor [0.12 * exp{-5}(0.5 + longterm_snr)$^{0.65}$] is clamped at a maximum value of 0.25.

After the minimum value is computed, each entry in the array "ksi" representing the a priori signal-to-noise ratio estimates are compared to the lower limit. Any of those entries below the lower limit are raised to the lower limit.

After the a priori signal-to-noise ratios are available, frames containing speech are examined to compute a new long-term average of the magnitude spectrum, a new long-term average of the signal-to-noise ratio for use in the enhancement of the next speech frame, and a new set of probabilities "qk" that a frequency bin contains only noise. These latter probabilities had been set previously to 0.99 for use in the event that the frame was determined to contain only noise.

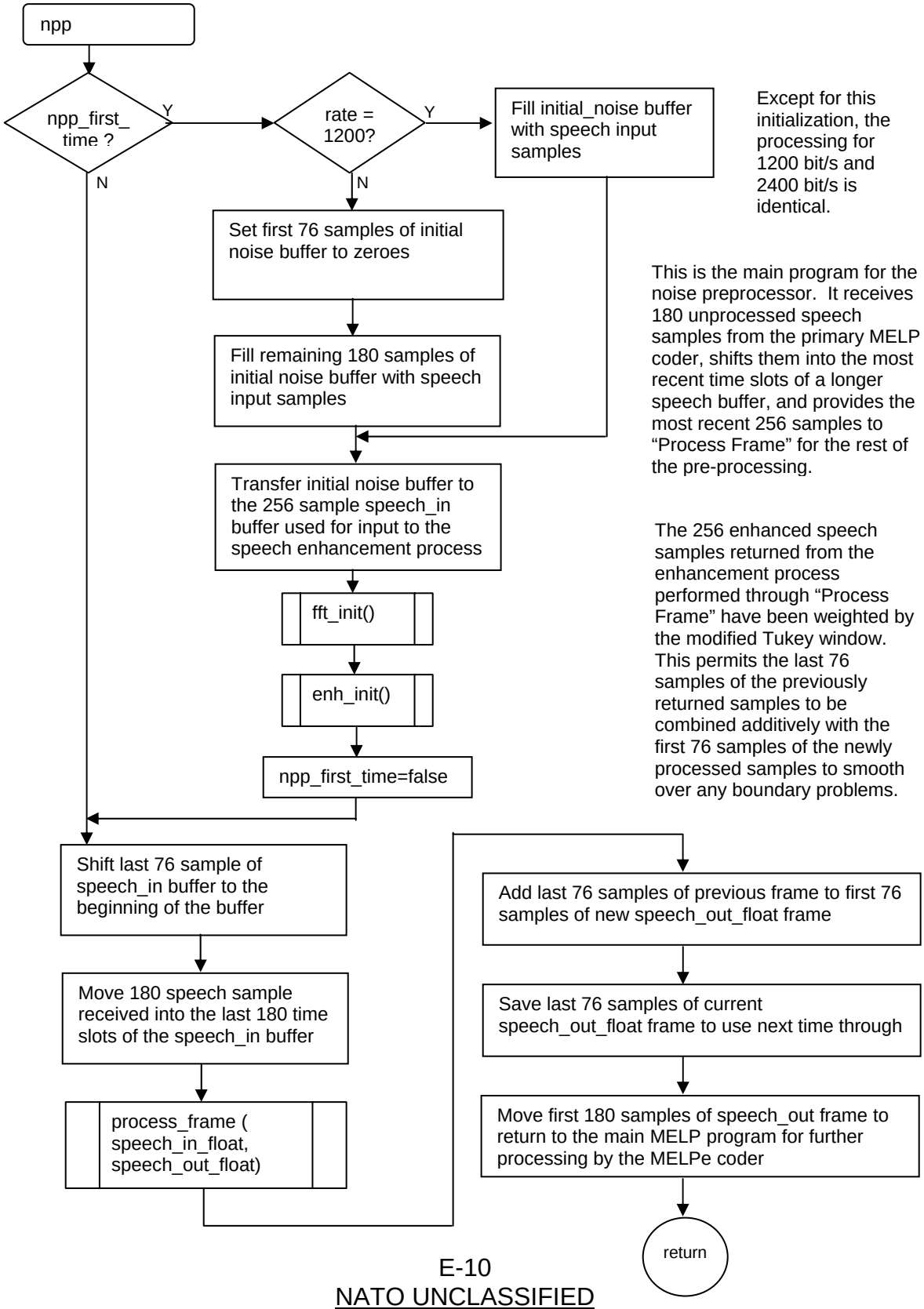
Finally, gain values used to enhance the speech spectral magnitudes are computed for each frequency bin. The gains are computed using the a priori and a posteriori signal-to-noise rations (ksi and gamaK) and the probabilities (gk) that the frequency bin

contains noise only. To cover the first frame processed, these gain values had previously been set to 0.12 for subsequent use.

The gain values are then applied to the original Fourier transform “sigbuf” of the speech samples to produce the enhanced Fourier transform that will be used to compute the enhanced speech samples. At the same time, the gain values are used to calculate the corresponding unnormalized power spectral density for the enhanced samples by applying the square of the enhancement gain values to the power spectral density of the original speech samples contained in the array “ymag.” The enhanced power spectral density is contained in the array “aga12.” The enhanced unnormalized power spectral density is computed for use in estimating the a priori signal-to-noise ratios in the next frame, while the modified Fourier transform samples in “sigbuf” and used as input to an inverse Fast Fourier Transform to compute the enhanced speech values. These enhanced speech samples are contained in the array “outspeech.” As indicated before, these enhanced speech samples are then weighted by the square root of the Tukey window so that the 76 overlapping samples can be combined later with the corresponding samples in the previous set of speech samples.

The processing of the frame is completed by updating the noise power for use in processing the next frame, and the 256 enhanced and weighted speech samples are returned to the top level function of the noise preprocessor.

Figure E-3. Flow diagram of Top Level of Noise Preprocessor



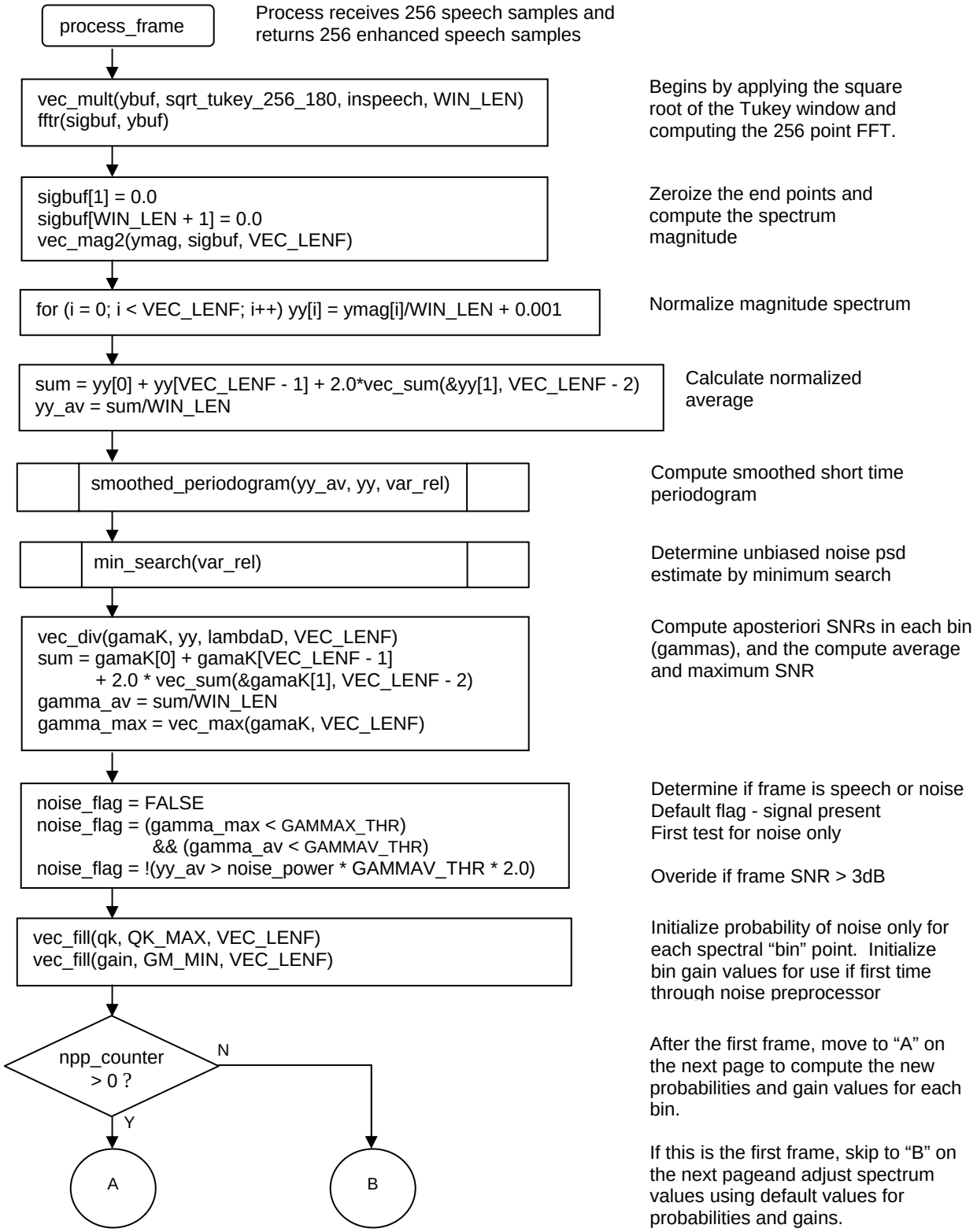


Figure E-4a. Process Frame (Part 1)

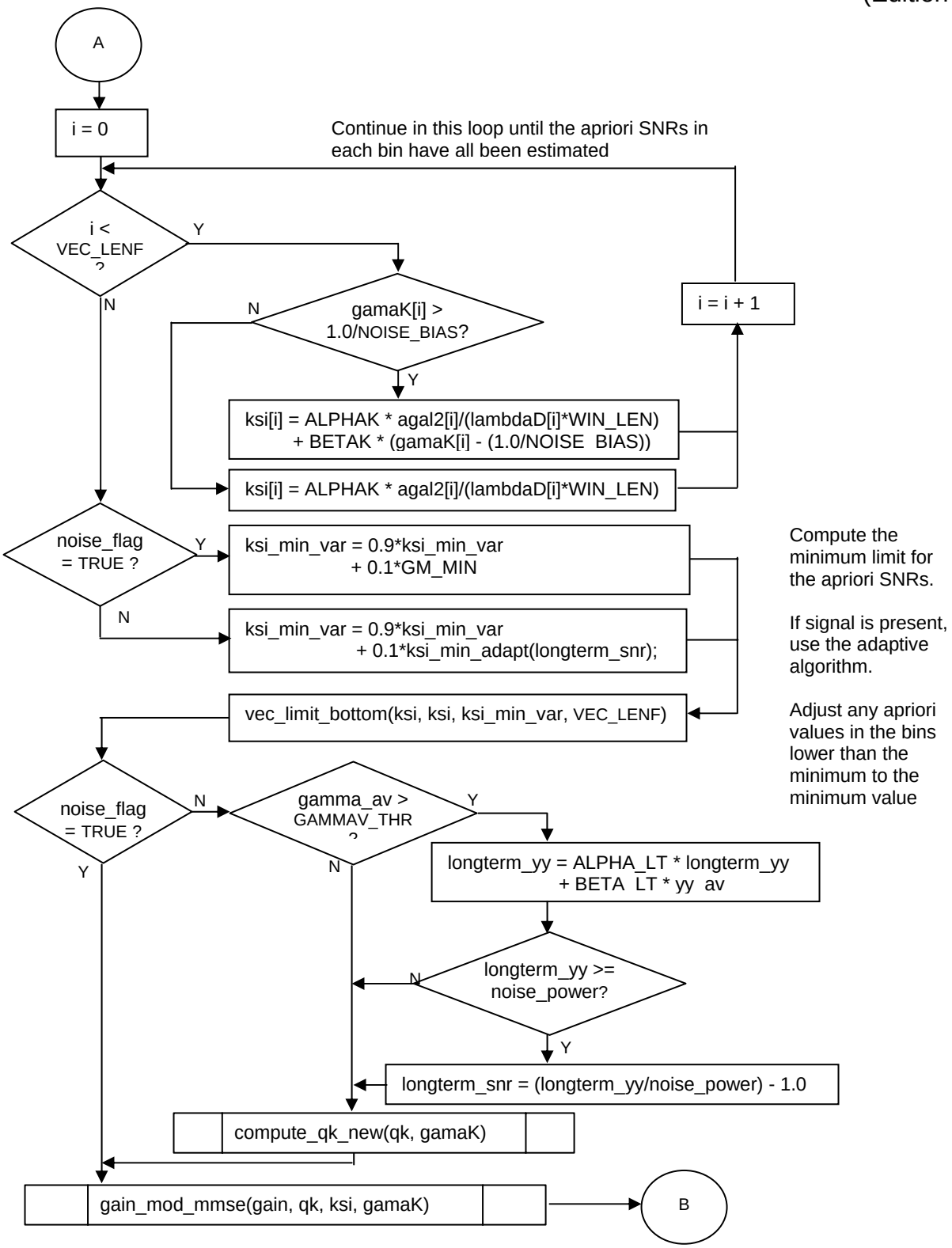


Figure E-4b. Process Frame (Part 2)

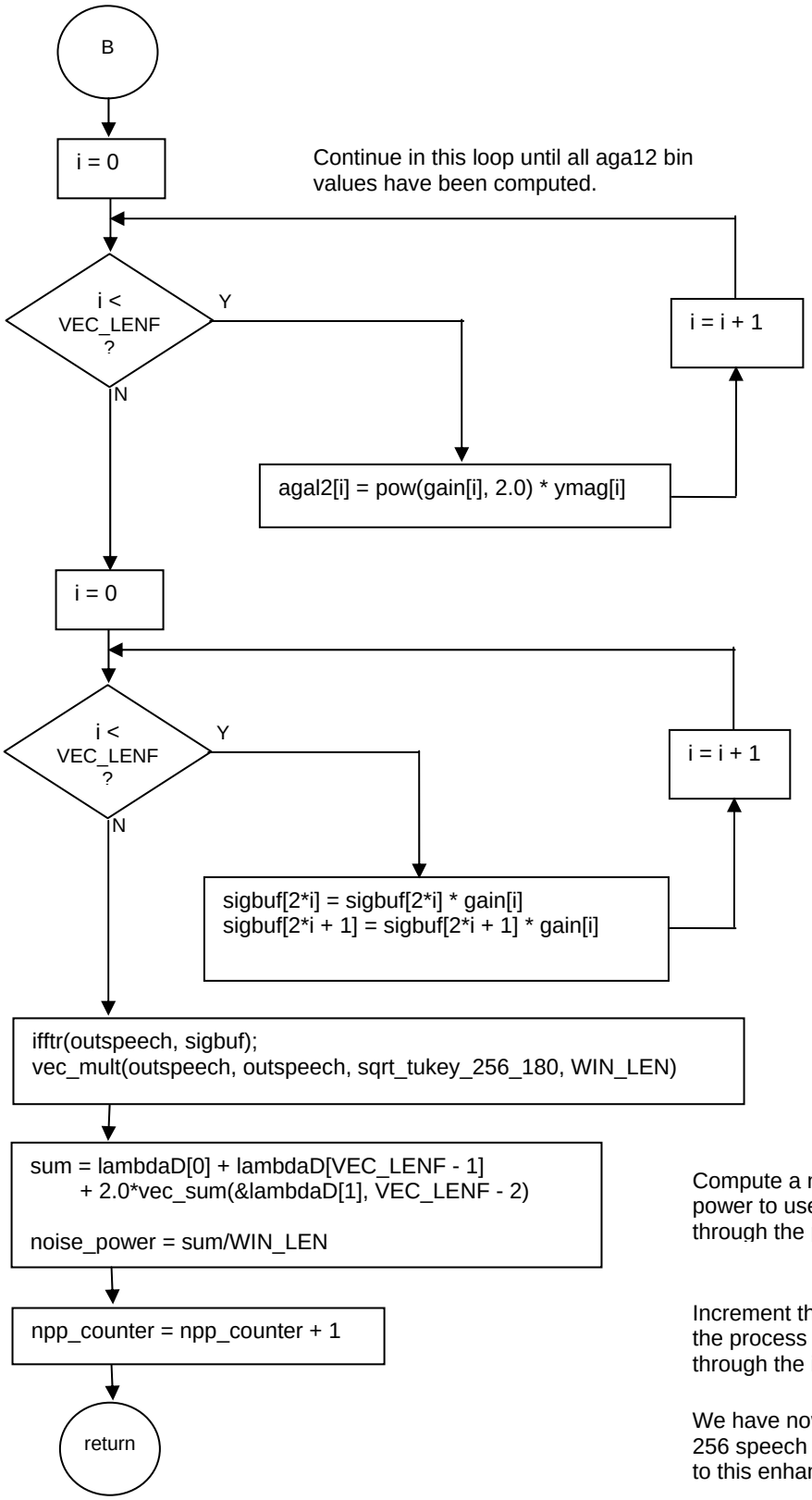


Figure E-4c. Process Frame (Part 3)

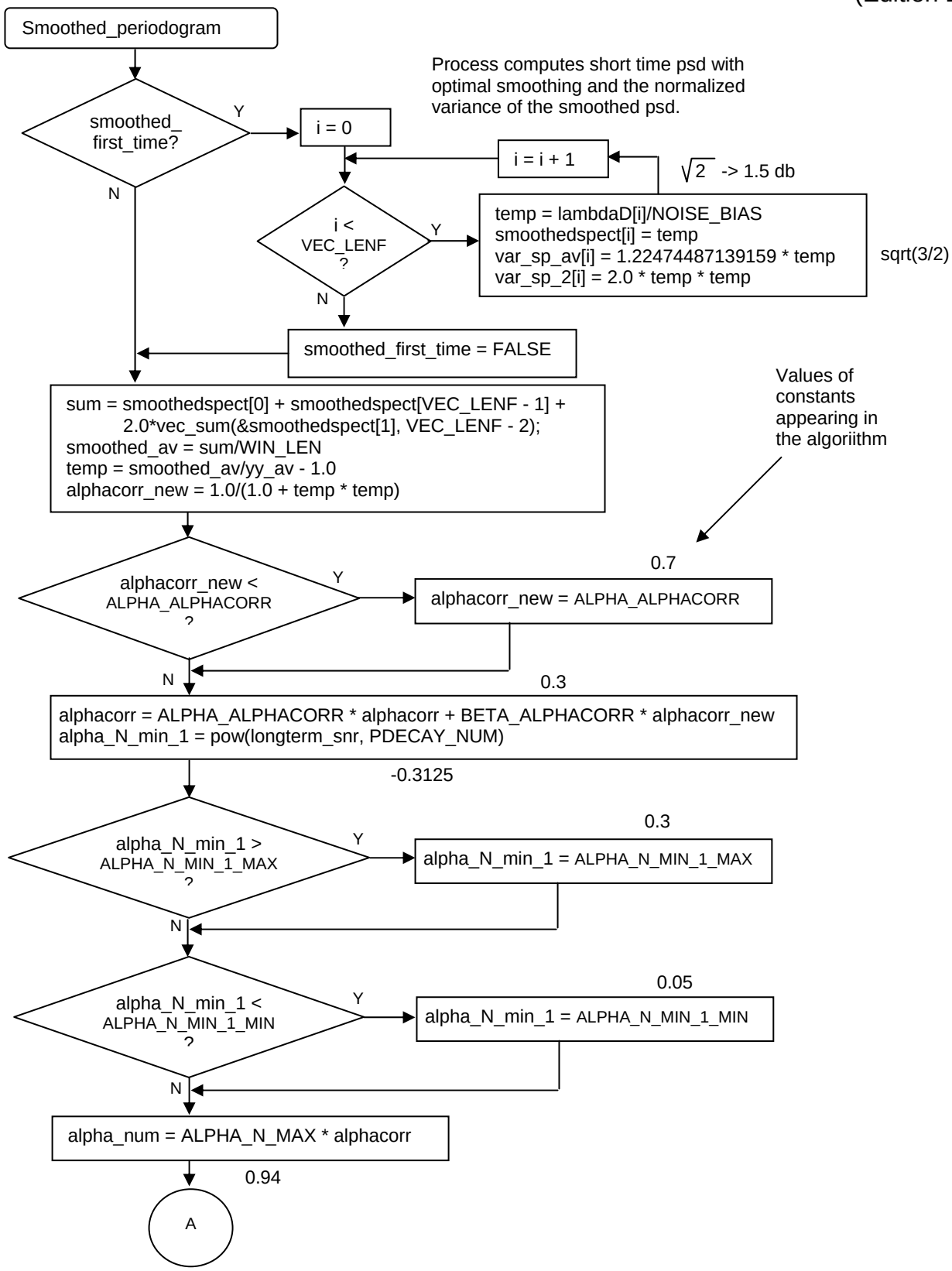


Figure E-5a. Processing for Smoothing Periodogram (Part 1)

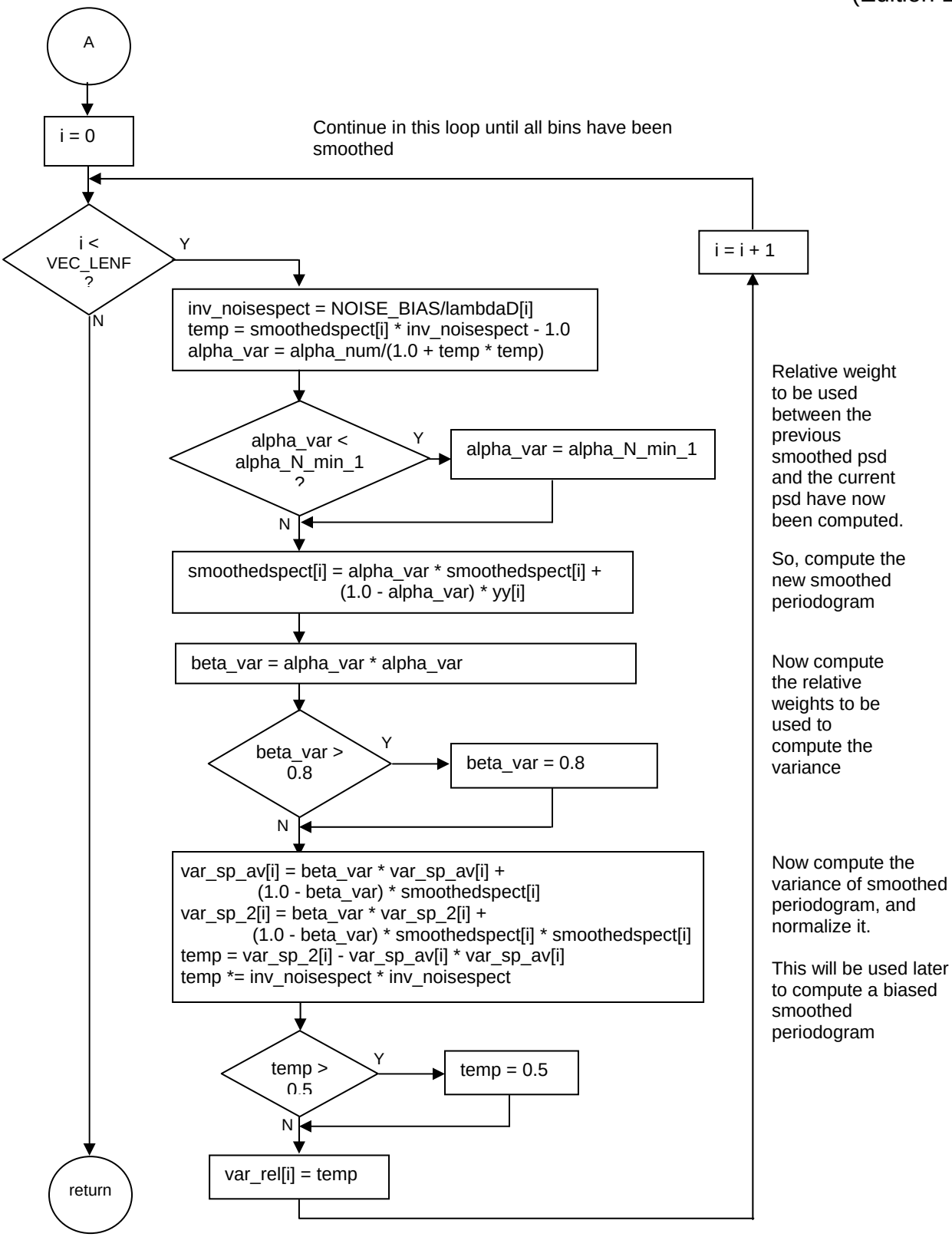


Figure E-5b. Processing for Smoothing Periodogram (Part 2)

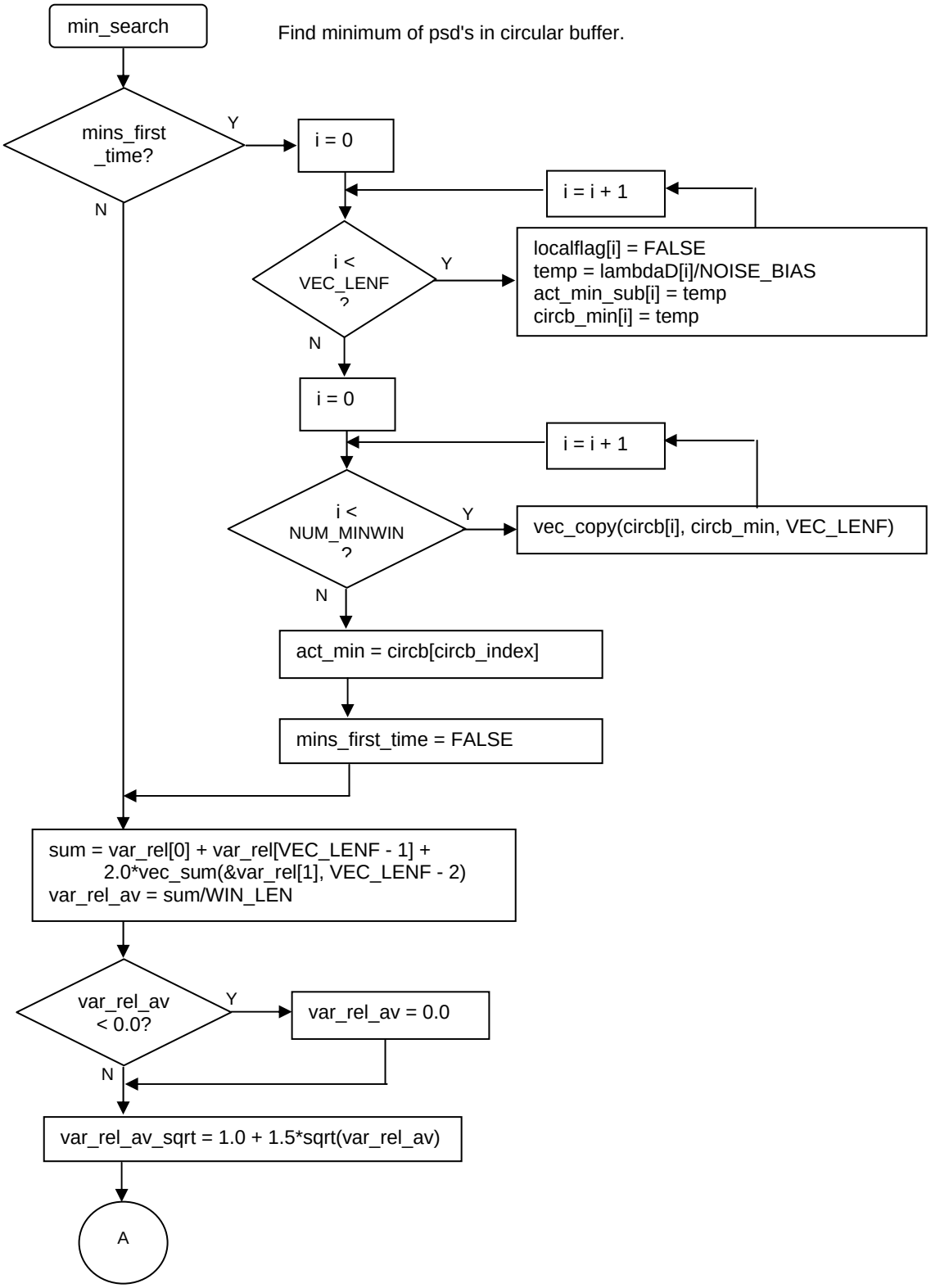


Figure E-6a. Processing to Search for Minimum PSD (Part 1)

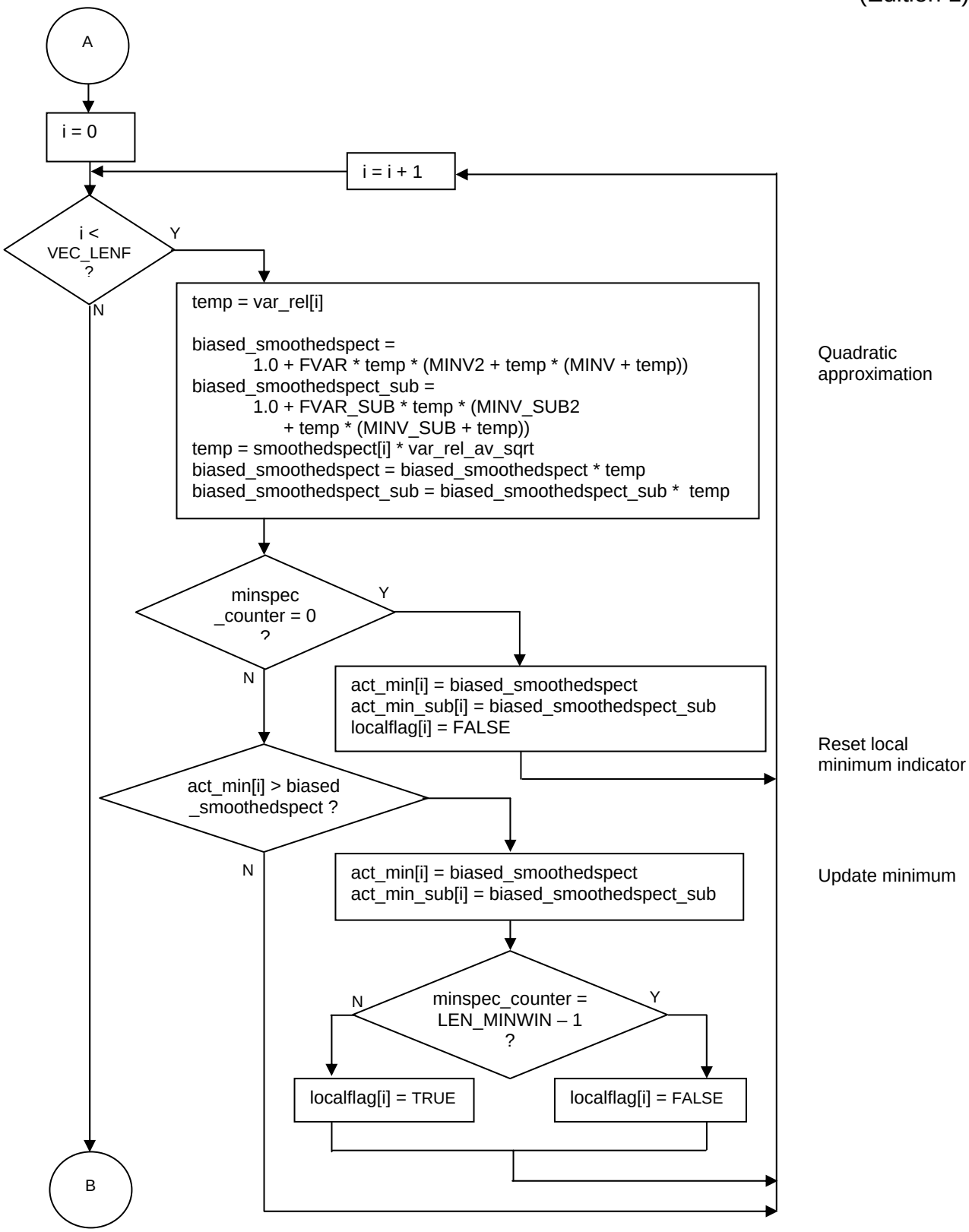


Figure E-6b. Processing to Search for Minimum PSD (Part 2)

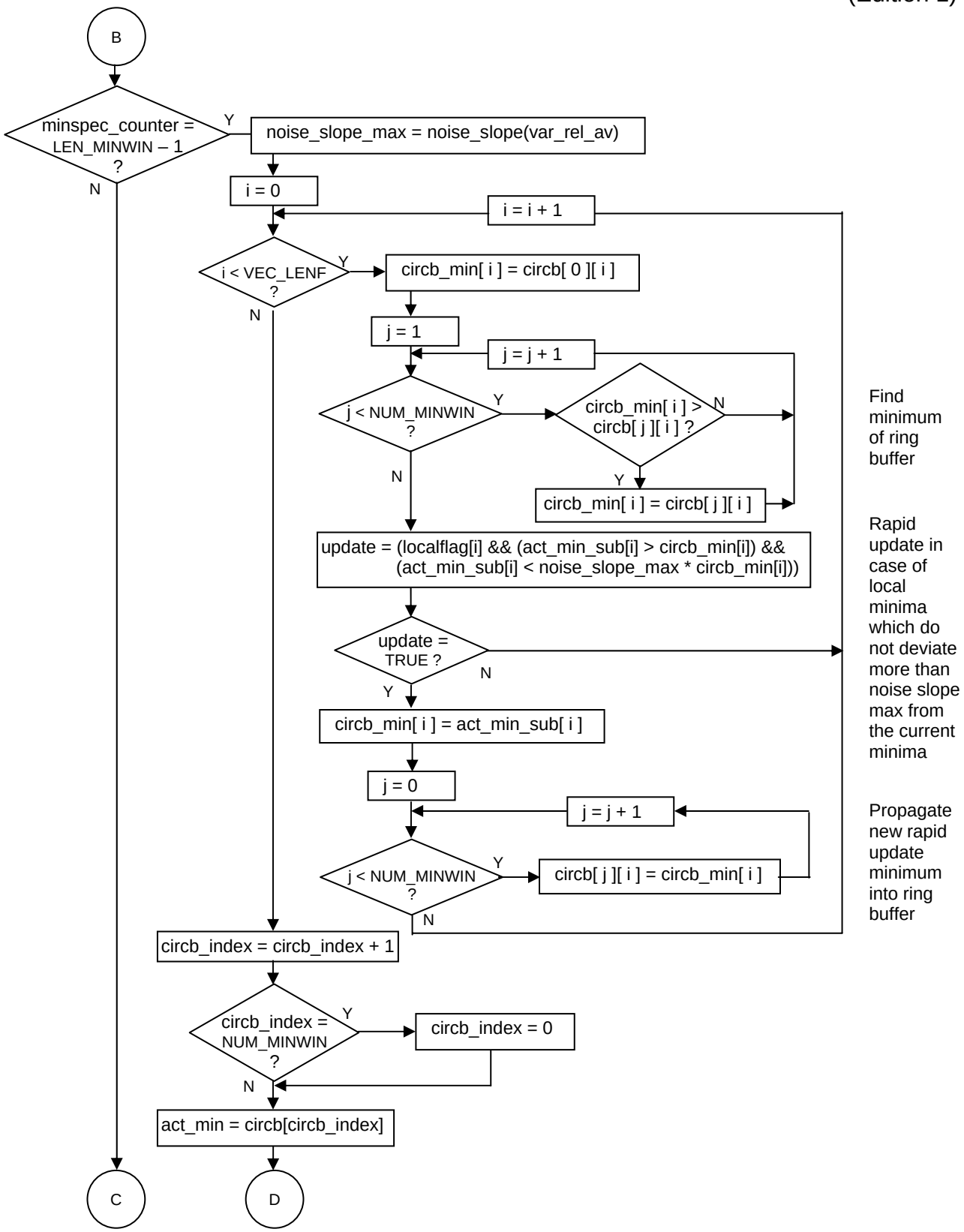


Figure E-6c. Processing to Search for Minimum PSD (Part 3)

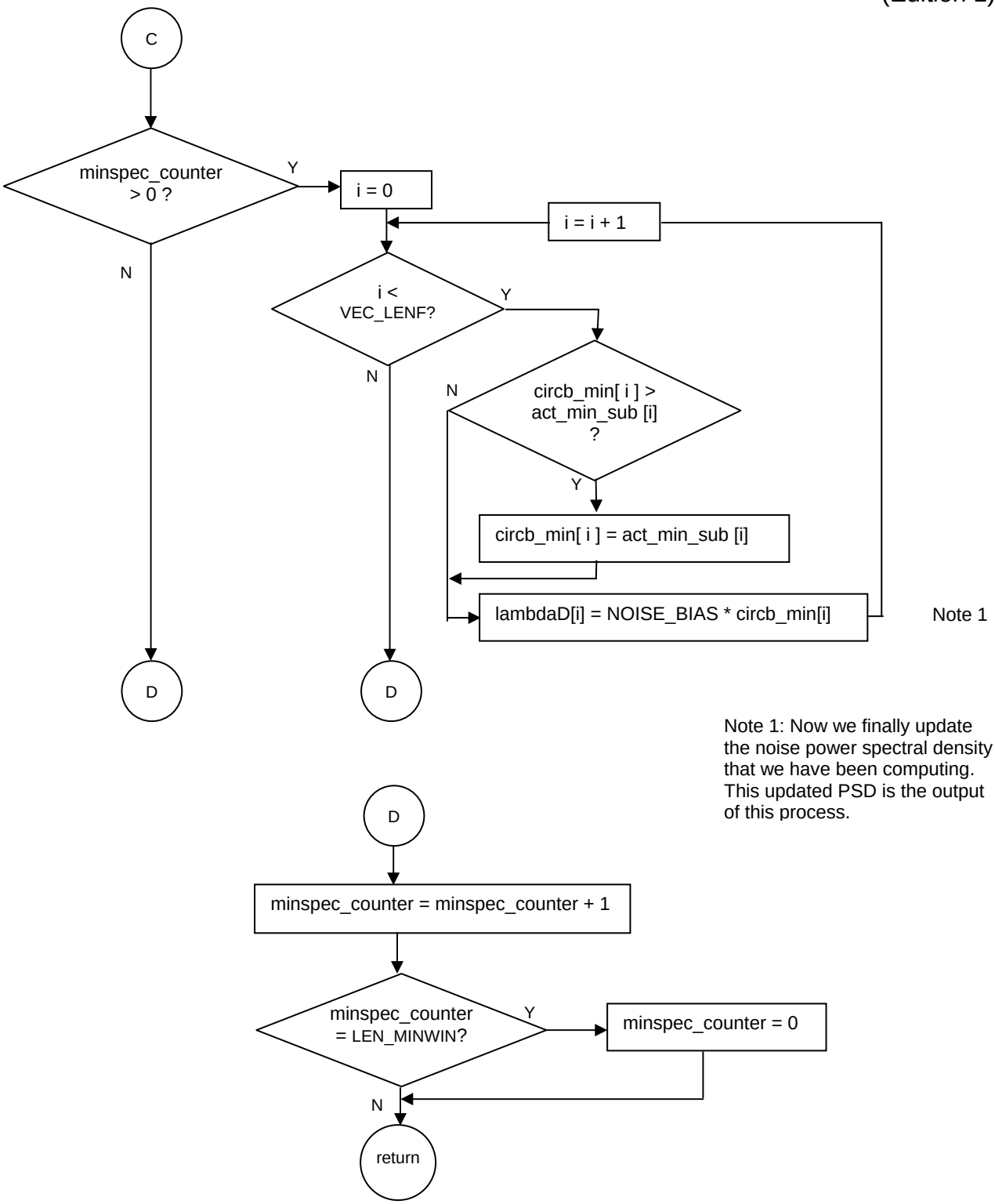
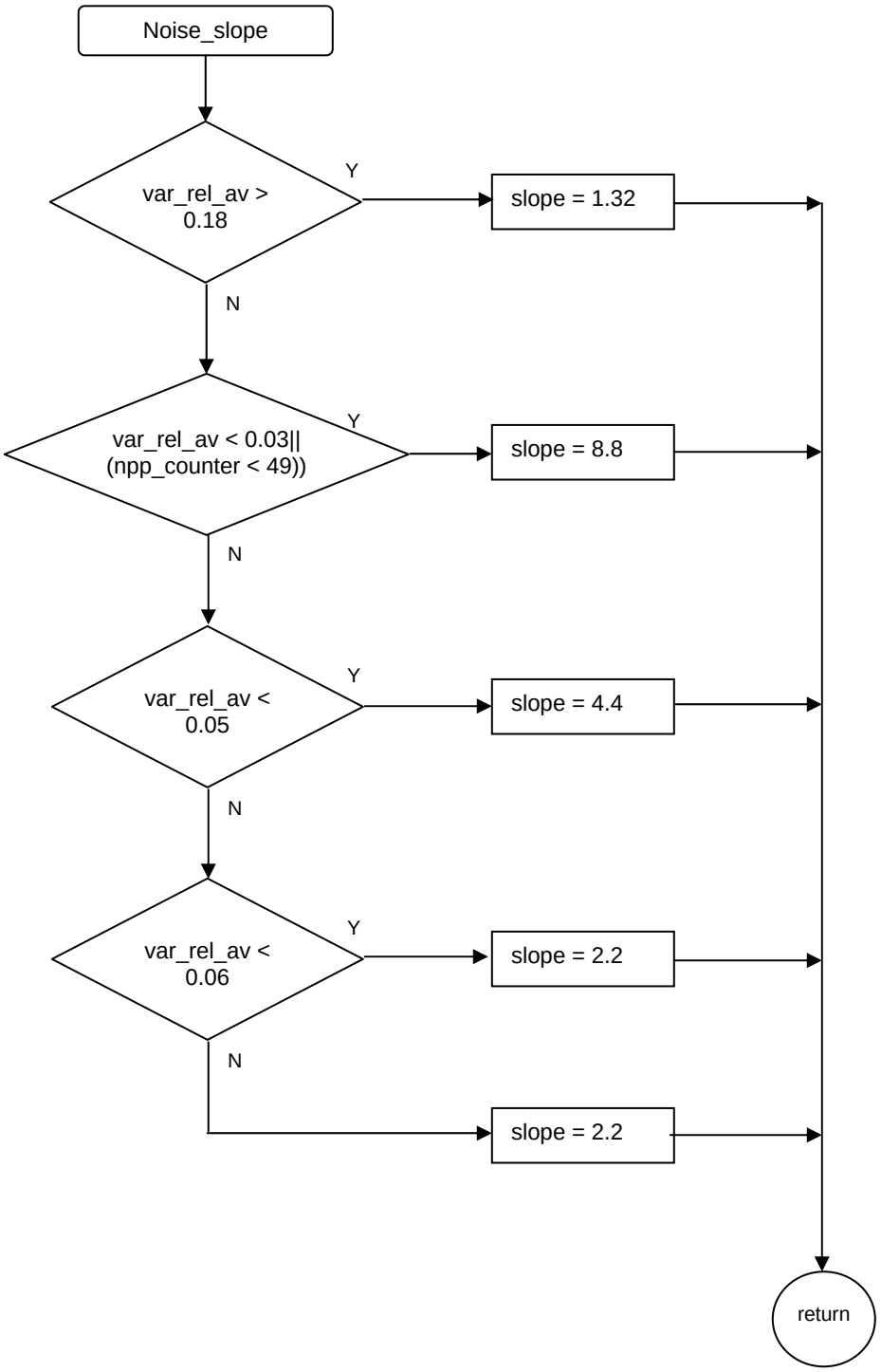


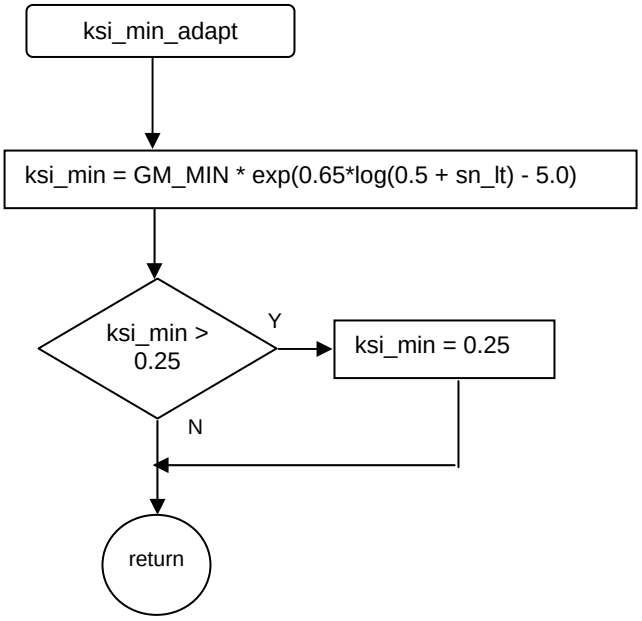
Figure E-6d. Processing to Search for Minimum PSD (Part 4)



This function computes the maximum of the permitted increase of the noise estimate as a function of the mean signal variance

Figure E-7. Computing the Noise Slope

Figure E-8. Compute adaptive minimum of ksi



This function computes the adaptive ksi_min

The function is supplied with the long term signal to noise ration (sn_lt) as its input, and uses the constant GM_MIN which is set to 0.12.

The computation is based on the Martin/Cox reference.

The maximum value allowed for ksi_min is 0.25

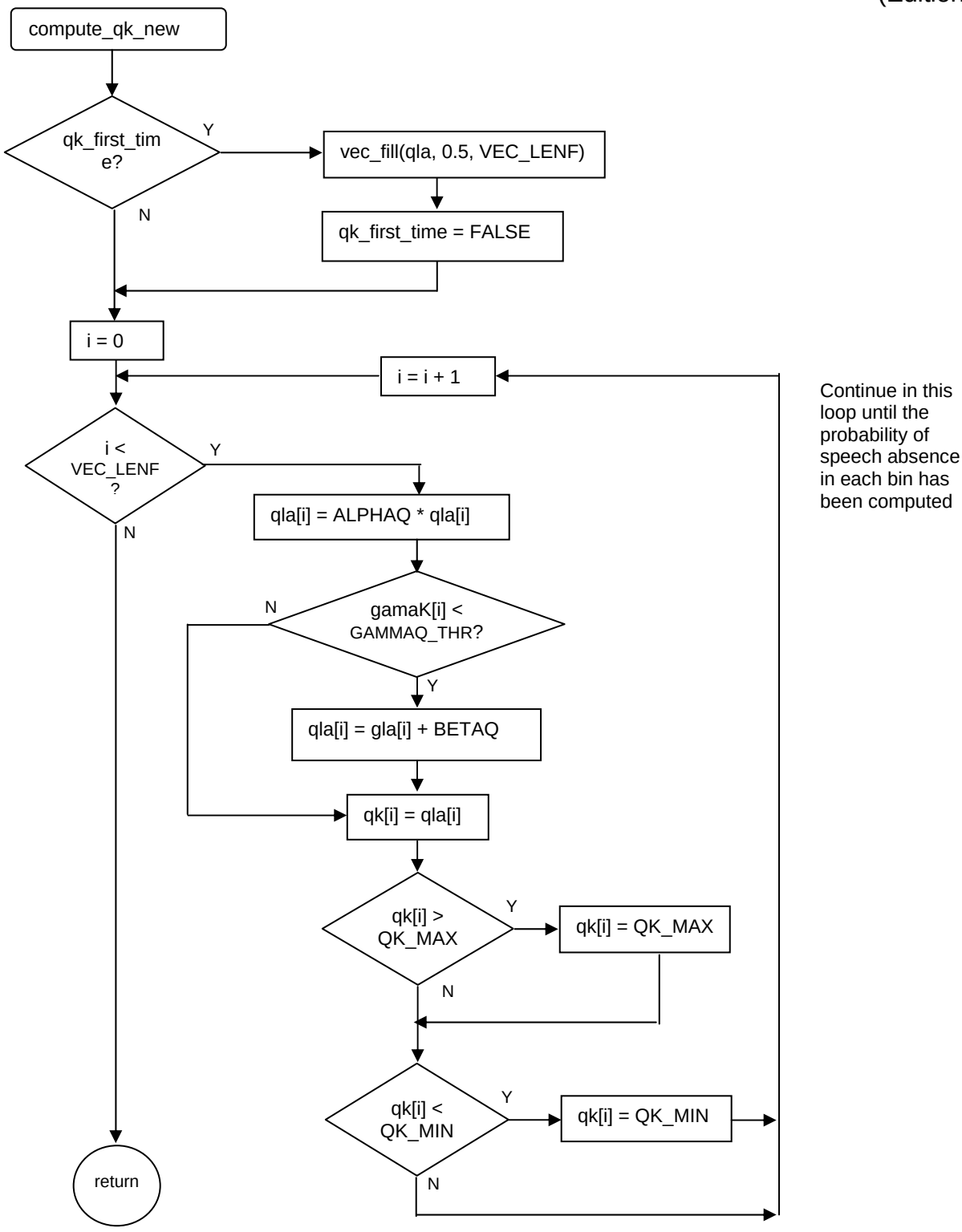


Figure E-9. Compute new qk

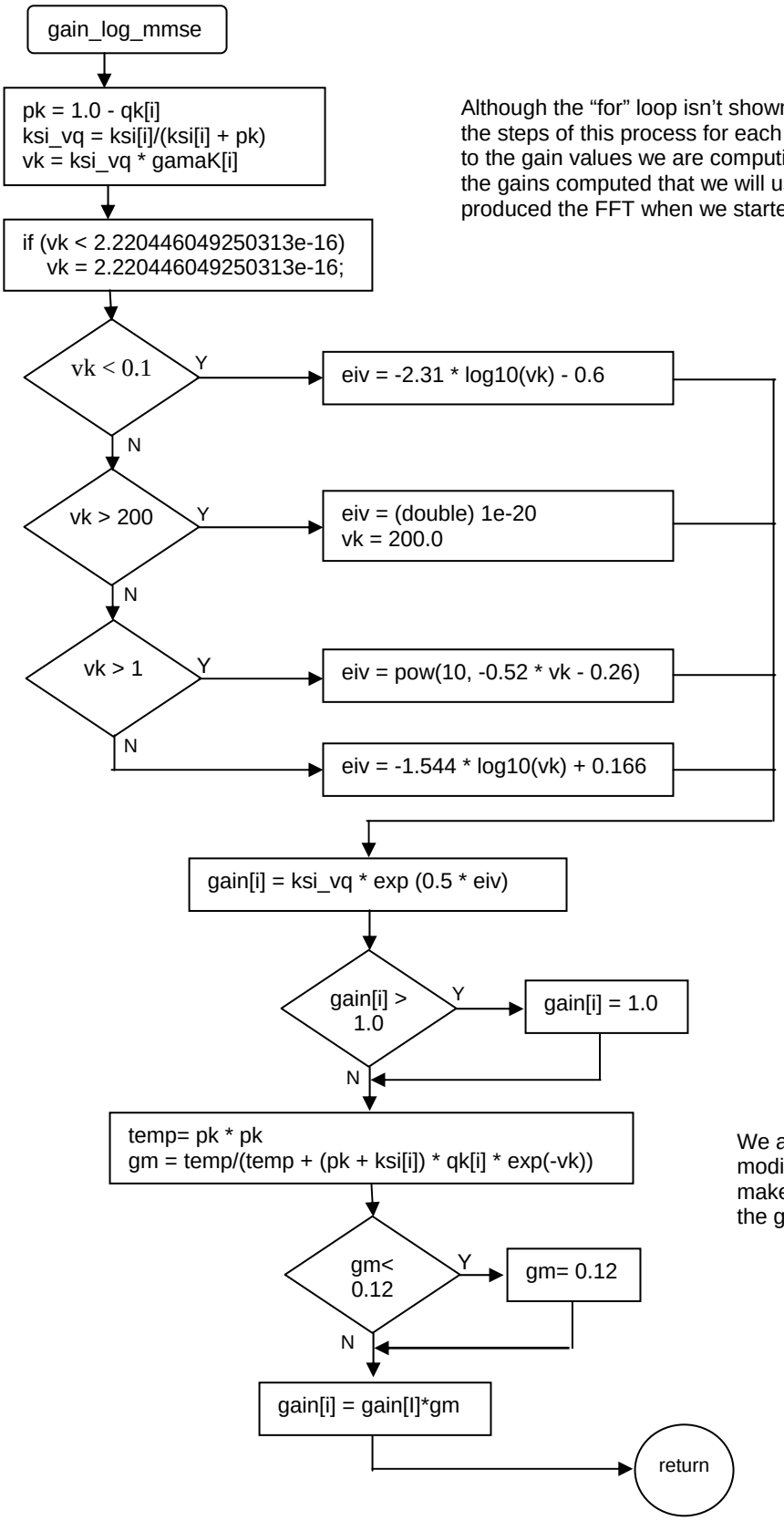


Figure E-10. Gain Computation

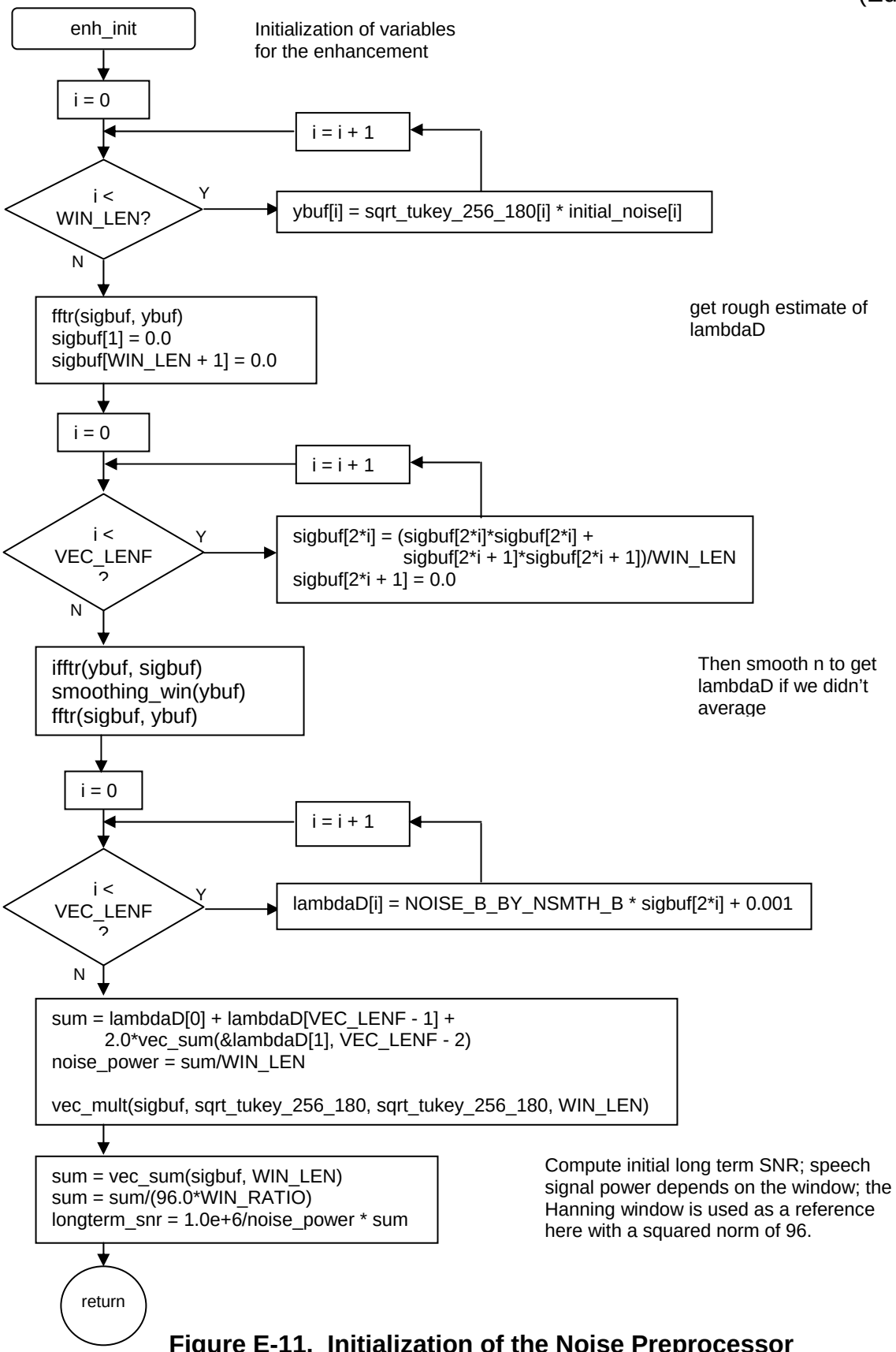


Figure E-11. Initialization of the Noise Preprocessor

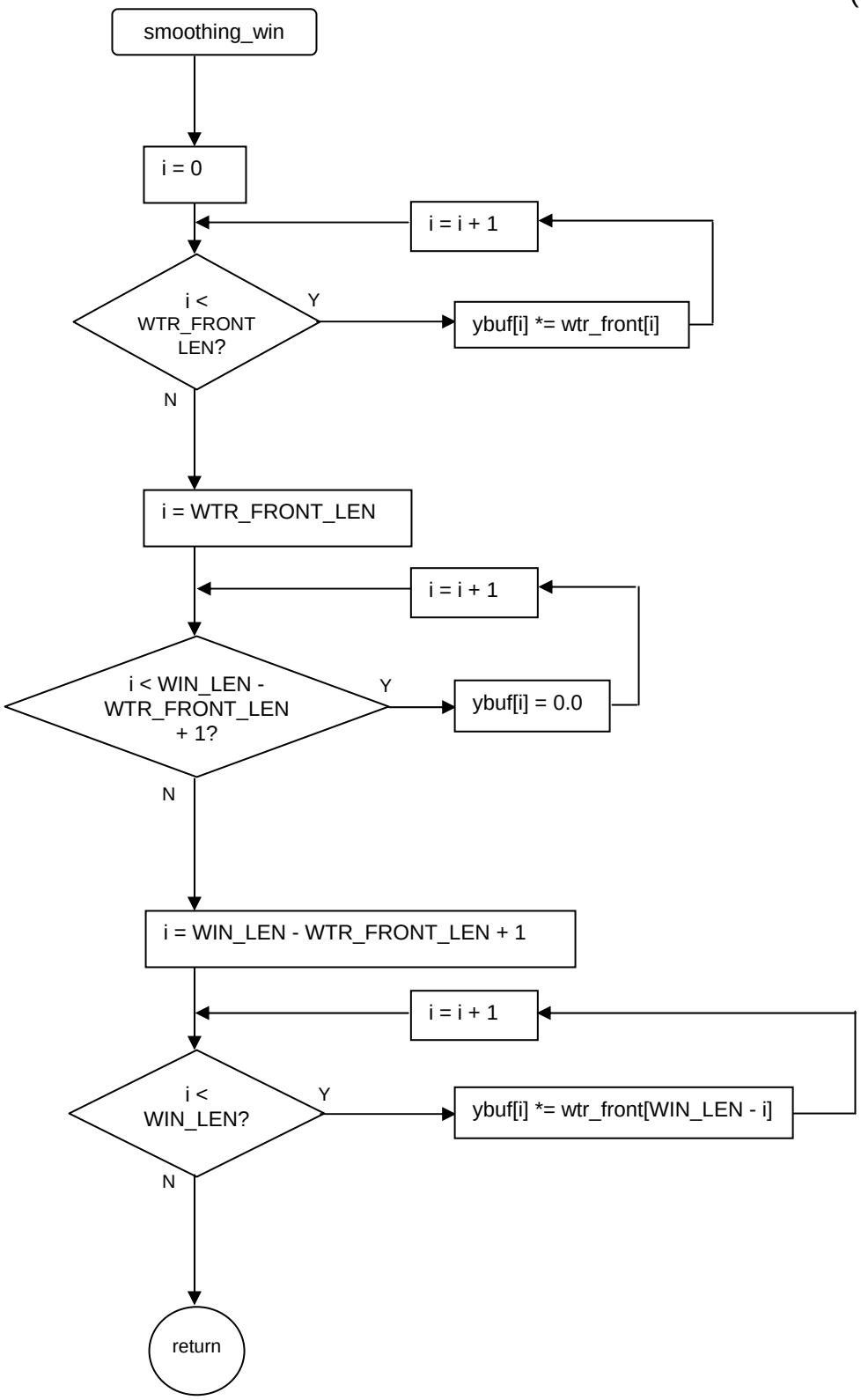


Figure E-12. Initialization of the Processing Buffer

Table E-1a. Square Root of Tukey Values for Indices 1 to 128

sqrt_tukey [1 to 32]	sqrt_tukey [33 to 64]	sqrt_tukey [65 to 96]	sqrt_tukey [97 to 128]
0.02066690122755	0.63039061612796	0.97426664263229	1.00000000000000
0.04132497424881	0.64629923786094	0.97871684532735	1.00000000000000
0.06196539462859	0.66193178225957	0.98274897304736	1.00000000000000
0.08257934547233	0.67728157162574	0.98636130340272	1.00000000000000
0.10315802119236	0.69234204904483	0.98955229332720	1.00000000000000
0.12369263126935	0.70710678118655	0.99232057973705	1.00000000000000
0.14417440400735	0.72156946105306	0.99466498011324	1.00000000000000
0.16459459028073	0.73572391067313	0.99658449300667	1.00000000000000
0.18494446727156	0.74956408374113	0.99807829846587	1.00000000000000
0.20521534219563	0.76308406819981	0.99914575838730	1.00000000000000
0.22539855601581	0.77627808876576	0.99978641678793	1.00000000000000
0.24548548714080	0.78914050939639	1.00000000000000	1.00000000000000
0.26546755510807	0.80166583569749	1.00000000000000	1.00000000000000
0.28533622424911	0.81384871727019	1.00000000000000	1.00000000000000
0.30508300733555	0.82568394999656	1.00000000000000	1.00000000000000
0.32469946920468	0.83716647826253	1.00000000000000	1.00000000000000
0.34417723036264	0.84829139711757	1.00000000000000	1.00000000000000
0.36350797056383	0.85905395436989	1.00000000000000	1.00000000000000
0.38268343236509	0.86944955261637	1.00000000000000	1.00000000000000
0.40169542465297	0.87947375120649	1.00000000000000	1.00000000000000
0.42053582614271	0.88912226813919	1.00000000000000	1.00000000000000
0.43919658884737	0.89839098189198	1.00000000000000	1.00000000000000
0.45766974151568	0.90727593318156	1.00000000000000	1.00000000000000
0.47594739303707	0.91577332665506	1.00000000000000	1.00000000000000
0.49402173581250	0.92387953251129	1.00000000000000	1.00000000000000
0.51188504908960	0.93159108805128	1.00000000000000	1.00000000000000
0.52952970226071	0.93890469915743	1.00000000000000	1.00000000000000
0.54694815812243	0.94581724170063	1.00000000000000	1.00000000000000
0.56413297609525	0.95232576287481	1.00000000000000	1.00000000000000
0.58107681540194	0.95842748245825	1.00000000000000	1.00000000000000
0.59777243820324	0.96411979400121	1.00000000000000	1.00000000000000
0.61421271268967	0.96940026593933	1.00000000000000	1.00000000000000

Table E-1b. Square Root of Tukey Values for Indices 129 to 256

sqrt_tukey [129 to 160]	sqrt_tukey [161 to 192]	sqrt_tukey [193 to 224]	sqrt_tukey [224 to 256]
1.00000000000000	1.00000000000000	0.96411979400121	0.59777243820324
1.00000000000000	1.00000000000000	0.95842748245825	0.58107681540194
1.00000000000000	1.00000000000000	0.95232576287481	0.56413297609525
1.00000000000000	1.00000000000000	0.94581724170063	0.54694815812243
1.00000000000000	1.00000000000000	0.93890469915743	0.52952970226071
1.00000000000000	1.00000000000000	0.93159108805128	0.51188504908960
1.00000000000000	1.00000000000000	0.92387953251129	0.49402173581250
1.00000000000000	1.00000000000000	0.91577332665506	0.47594739303707
1.00000000000000	1.00000000000000	0.90727593318156	0.45766974151568
1.00000000000000	1.00000000000000	0.89839098189198	0.43919658884737
1.00000000000000	1.00000000000000	0.88912226813919	0.42053582614271
1.00000000000000	1.00000000000000	0.87947375120649	0.40169542465297
1.00000000000000	1.00000000000000	0.86944955261637	0.38268343236509
1.00000000000000	1.00000000000000	0.85905395436989	0.36350797056383
1.00000000000000	1.00000000000000	0.84829139711757	0.34417723036264
1.00000000000000	1.00000000000000	0.83716647826253	0.32469946920468
1.00000000000000	1.00000000000000	0.82568394999656	0.30508300733555
1.00000000000000	1.00000000000000	0.81384871727019	0.28533622424911
1.00000000000000	1.00000000000000	0.80166583569749	0.26546755510807
1.00000000000000	1.00000000000000	0.78914050939639	0.24548548714080
1.00000000000000	0.99978641678793	0.77627808876576	0.22539855601581
1.00000000000000	0.99914575838730	0.76308406819981	0.20521534219563
1.00000000000000	0.99807829846587	0.74956408374113	0.18494446727156
1.00000000000000	0.99658449300667	0.73572391067313	0.16459459028073
1.00000000000000	0.99466498011324	0.72156946105306	0.14417440400735
1.00000000000000	0.99232057973705	0.70710678118655	0.12369263126935
1.00000000000000	0.98955229332720	0.69234204904483	0.10315802119236
1.00000000000000	0.98636130340272	0.67728157162574	0.08257934547233
1.00000000000000	0.98274897304736	0.66193178225957	0.06196539462859
1.00000000000000	0.97871684532735	0.64629923786094	0.04132497424881
1.00000000000000	0.97426664263229	0.63039061612796	0.02066690122755
1.00000000000000	0.96940026593933	0.61421271268967	0

Table E-2. Weights Used During Initialization of Noise Pre-Processor

Index	Weight	Index	Weight
1	1.00000000000000	17	0.25000000000000
2	0.99432373046875	18	0.20599365234375
3	0.97802734375000	19	0.16748046875000
4	0.95220947265625	20	0.13409423828125
5	0.91796875000000	21	0.10546875000000
6	0.87640380859375	22	0.08123779296875
7	0.82861328125000	23	0.06103515625000
8	0.77569580078125	24	0.04449462890625
9	0.71875000000000	25	0.03125000000000
10	0.65887451171875	26	0.02093505859375
11	0.59716796875000	27	0.01318359375000
12	0.53472900390625	28	0.00762939453125
13	0.47265625000000	29	0.00390625000000
14	0.41204833984375	30	0.00164794921875
15	0.35400390625000	31	0.00048828125000
16	0.29962158203125	32	0.00006103515625

E.6 REFERENCES FOR THE NOISE PREPROCESSOR

[E1] “New Speech Enhancement Techniques for Low Bit Rate Speech Coding” by R. Martin and R Cox, Proceedings IEEE Workshop on Speech Coding, 1999.

[E2] "Low Delay Analysis/Synthesis Schemes for Joint Speech Enhancement and Low Bit Rate Speech Coding”
by R. Martin, H. Kang, and R. Cox, Proceedings EUROSPEECH 1999, Volume 3, pp. 1463-66, Budapest, Hungary, 1999.

ANNEX F

Definitions and Acronyms

F.1 DEFINITIONS

F.1.1 Terms

Definitions of terms used in this standard shall be as specified in the current edition of FED-STD-1037. In addition, the following definitions are applicable for the purpose of this standard.

F.1.1.1 Adaptive spectral enhancement

This feature enhances the formant structure of the synthetic speech by use of an adaptive spectral enhancement filter that is applied to the mixed excitation.

F.1.1.2 Aperiodic pulses

Aperiodic pulses are used in the excitation model of the synthesizer when the aperiodic flag is set to 1. The aperiodic flag is set to one when the jittery voiced state is encountered during the voicing decision process. This feature is used to reduce the buzzy quality of the synthetic speech signal.

F.1.1.3 Fourier magnitude modeling

Fourier magnitude modeling involves determining the Fourier magnitudes of the first 10 pitch harmonics of the prediction residual and vector quantizing them with 8 bits for transmission. The use of this technique improves the accuracy of the speech production model at the perceptually important lower frequencies.

F.1.1.4 Hamming codes

A class of linear codes used for forward error correction. These codes are used only in the unvoiced mode.

F.1.1.5 Jitter

Random variations introduced into the duration of a signal.

F.1.1.6 Linear prediction coding

A method for approximating the current speech sample by using a linear combination of past and future speech samples. This method efficiently represents a speech signal and its spectrum characteristics with a very small number of parameters when combined with an appropriate excitation signal.

F.1.1.7 Mixed excitation

The combination of a periodic function (such as a pulse train) and random noise for use in the excitation model. This combination is applied to sub regions of the frequency domain of the excitation signal.

F.1.1.8 Prediction coefficients

A set of values that are calculated using a short segment of the input speech signal and provide an estimate of the spectral properties of that signal. These values are determined by performing linear prediction analysis on the input signal. The goal of the analysis is to produce values that minimize the short term mean-squared prediction error over the input segment.

F.1.1.9 Pulse dispersion

Uses a fixed filter to spread the excitation energy within a pitch period.

F.1.1.10 Uniform quantizer

A uniform quantizer uses levels and step sizes that are distributed uniformly.

F.1.1.11 Weighted Euclidean distance

The Euclidean distance is a distortion measure between two vectors. In this standard the Euclidean distance is determined by summing the squared difference between two vectors for a select number of samples. Normally the Euclidean distance is the square root of the measure described in the previous sentence.

F.1.1.12 Acronyms used in this standard

The acronyms used in this standard are defined as follows:

A-D - Analog to Digital
DoD - Department of Defense
DoDISS - Department of Defense Index of Specifications and Standards
DoDSSP - Department of Defense Single Stock Point
FEC - Forward Error Correction
LPC - Linear Prediction Coding
LSB - Least Significant Bit
LSF - Line Spectrum Frequency
MELP - Mixed Excitation Linear Predictions
MELPe – Enhanced MELP
MSB - Most Significant Bit
MSVQ - Multi-Stage Vector Quantizer
STANAG - Standardization Agreement

ANNEX G

Fixed Point C Source Code

Annex G containing the STANAG 4591 software listing is published as a separate file since it is impractically long to be included in a single document. The fixed point C code is included in this STANAG to provide the authoritative example of how the bit stream can be correctly derived.
This fixed point C code is available from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int) (http://s4591.nc3a.nato.int) in electronic format.

BLANK PAGE BLANCHE

ANNEX H

Floating Point C Source Code

Annex H containing a listing of the floating point version of the MELPe software is published as a separate file since it is impractically long to be included in a singe document. A compilable version of the software is available from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int) in electronic format.

BLANK PAGE BLANCHE

ANNEX I

Test Vectors for 2400 bit/s STANAG 4591

The Input Speech Test Vector Data Base, the Coded Speech Test Vector Data Base, and the Output Synthesized Speech Test Vector Data Base shown in Figure B-1 for use in the bit exactness test method of Section B.5.2 are available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX J

Test Vectors for 1200 bit/s STANAG 4591

The Input Speech Test Vector Data Base, the Coded Speech Test Vector Data Base, and the Output Synthesized Speech Test Vector Data Base shown in Figure B-1 for use in the bit exactness test method of Section B.5.2 is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX K

Description of 1200 bit/s STANAG 4591 to 2400 bit/s STANAG 4591 Transcoder (Optional)

K.1 TRANSCODER

K.1.1 Overview

This annex presents the symmetric parametric transcoder function to allow seamless interoperability between the two rates of the MELPe voice coding algorithm (1200 bit/s and 2400 bit/s). This transcoder function reads a bit stream representing the encoded speech of one rate (1200 or 2400 bit/s), translates it, and then writes out the transcoded bit stream for the alternate rate.

The quality of the transcoded speech is at best as good as that of the lowest rate scheme (for example the 1200 bit/s). However, without the transcoding software, the speech coder would have to encode the raw speech at one rate, decode it, and then encode the processed speech at the other rate. This is known as a tandem process. In most cases the degradation in speech performance introduced by the two analysis steps of the tandem connection result in performance that is significantly lower than that of the transcoded speech.

In the transcoder, we decode the incoming bit stream to recover speech parameters, and then we apply the quantization scheme corresponding to the other rate scheme to these parameters. This approach saves the processing for reconstructing the speech and analyzing it again for re-encoding, and eliminates the degradations introduced in these processes.

All the building blocks for the transcoder are directly acquired from those used for 1200 bit/s and 2400 bit/s quantization schemes. Some modifications in the other parts of the speech coder are necessary in order to implement the transcoder.

Certain variables need to be duplicated separately for each rate scheme, for instance, *bitNum,* the number of bits corresponding to each frame, which is used in the functions related to reading and writing bit-streams from the channel.

Parameter buffer sizes (including input speech) need to be adjusted to allow the software suite to handle 1200 bit/s, 2400 bit/s, and transcoding. This is necessary because one input frame encoded at 1200 bit/s is mapped by the transcoder into three frames at 2400 bit/s, so the algorithm needs buffers which can handle the larger of the following two requirements: one 1200 bit/s frame, or three 2400 bit/s frames.

K.1.2 1200 bit/s to 2400 bit/s Transcoder

Under the upward transcoding scheme, the algorithm reads the bit stream corresponding to one frame at 1200 bit/s, decodes it to extract speech parameters corresponding to 540 speech samples (three 2400 bit/s frames), and then steps into a loop repeated 3 times to re-quantize the parameters with the 2400 bit/s quantizers. The corresponding 2400 bit/s bit streams for three frames are written into the channel buffer.

K.1.3 2400 bit/s to 1200 bit/s Transcoder

Under the downward transcoding scheme, the algorithm reads the bit stream corresponding to three frames at 2400 bit/s, decodes it to extract speech parameters corresponding to 540 speech samples, and then quantizes the parameters with the 1200 bit/s quantizers. The resulting 1200 bit/s bit stream is transferred to the output channel buffer.

The downward transcoding is not executed unless we can read bit information corresponding to three complete frames at 2400 bit/s.

ANNEX L

Description of 2400 bit/s STANAG 4591 to 2400 bit/s STANAG 4198 Transcoder (Optional)

Annex L is a description of the two way 2400 bit/s STANAG 4591 to 2400 bit/s STANAG 4198 Transcoder developed by the U.S. Government.
This annex is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site.

BLANK PAGE BLANCHE

ANNEX M

Description of 600 bit/s STANAG 4591 algorithm details

Annex M is a description of the 600 bit/s STANAG 4591 algorithm developed by the Thales France.
This annex is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX N

Description of 600 bit/sec to 2400 bit/sec and 2400 bit/sec to 600 bit/sec MELPe Transcoders

Annex N is a description of the 600 bit/s to 2400 bit/s and 2400 bit/s to 600 bit/s STANAG 4591 transcoder algorithm.
This annex is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX O

Description of MELPe Frame Synchronization

Annex O is a description of the STANAG 4591 algorithm frame synchronization sequences. These sequences are used to identify that the MELPe algorithm is being used for the transmission of speech.
This annex is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE

ANNEX P

Test Vectors for 600 bit/s STANAG 4591

The Input Speech Test Vector Data Base, the Coded Speech Test Vector Data Base, and the Output Synthesized Speech Test Vector Data Base shown in Figure B-1 for use in the bit exactness test method of Section B.5.2.
This annex is available in electronic form from the CNSC website (www.nhqc3s..nato.int) and from the NC3A web site (http://s4591.nc3a.nato.int).

BLANK PAGE BLANCHE