DRAFT

EUROPEAN TELECOMMUNICATION STANDARD

pr **ETS 300 395-2**

September 1997

Second Edition

Source: EP TETRA

Reference: RE/TETRA-05032

ICS: 33.020

Key words: TETRA, codec

Terrestrial Trunked Radio (TETRA); Speech codec for full-rate traffic channel; Part 2: TETRA codec

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Contents

Foreword 7

1 Scope 9

2 Normative references 9

3 Abbreviations 9

4 Full rate codec 10
4.1 Structure of the codec 10
4.2 Functional description of the codec 12
4.2.1 Pre- and post-processing 12
4.2.2 Encoder 12
4.2.2.1 Short-term prediction 13
4.2.2.2 LP to LSP and LSP to LP conversion 14
4.2.2.3 Quantization and interpolation of LP parameters 16
4.2.2.4 Long-term prediction analysis 17
4.2.2.5 Algebraic codebook: structure and search 18
4.2.2.6 Quantization of the gains 21
4.2.2.7 Detailed bit allocation 23
4.2.3 Decoder 23
4.2.3.1 Decoding process 24
4.2.3.1.1 Decoding of LP filter parameters 24
4.2.3.1.2 Decoding of the adaptive codebook vector 24
4.2.3.1.3 Decoding of the innovation vector 25
4.2.3.1.4 Decoding of the adaptive and innovative codebook gains 25
4.2.3.1.5 Computation of the reconstructed speech 25
4.2.3.2 Error concealment 25

5 Channel coding for speech 26
5.1 General 26
5.2 Interfaces in the error control structure 26
5.3 Notations 28
5.4 Definition of sensitivity classes and error control codes 28
5.4.1 Sensitivity classes 28
5.4.2 CRC codes 28
5.4.3 16-state RCPC codes 30
5.4.3.1 Encoding by the 16-state mother code of rate 1/3 30
5.4.3.2 Puncturing of the mother code 30
5.5 Error control scheme for normal speech traffic channel 31
5.5.1 CRC code 31
5.5.2 RCPC codes 31
5.5.2.1 Puncturing scheme of the RCPC code of rate 8/12 (equal to 2/3) 31
5.5.2.2 Puncturing scheme of the RCPC code of rate 8/18 31
5.5.3 Matrix Interleaving 32
5.6 Error control scheme for speech traffic channel with frame stealing activated 33
5.6.1 CRC code 33
5.6.2 RCPC codes 34
5.6.2.1 Puncturing scheme of the RCPC code of rate 8/17 35
5.6.3 Interleaving 35

6 Channel decoding for speech 35
6.1 General 35

6.2 Error control structure 35

7 Codec performance 36

8 Bit exact description of the TETRA codec 36

Annex A (informative): Implementation of speech channel decoding 38

A.1 Algorithmic description of speech channel decoding 38
A.1.1 Definition of error control codes 38
A.1.1.1 16-state RCPC codes 38
A.1.1.1.1 Obtaining the mother code from punctured code 38
A.1.1.1.2 Viterbi decoding of the 16-state mother code of the rate 1/3 38
A.1.1.2 CRC codes 39
A.1.1.3 Type-4 bits 39
A.1.2 Error control scheme for normal speech traffic channel 39
A.1.2.1 Matrix Interleaving 39
A.1.2.2 RCPC codes 39
A.1.2.2.1 Puncturing scheme of the RCPC code of rate 8/12 (equal to 2/3) 40
A.1.2.2.2 Puncturing scheme of the RCPC code of rate 8/18 40
A.1.2.3 CRC code 40
A.1.2.4 Speech parameters 40
A.1.3 Error control scheme for speech traffic channel with frame stealing activated 40
A.1.3.1 Interleaving 40
A.1.3.2 RCPC codes 40
A.1.3.2.1 Puncturing scheme of the RCPC code of rate 8/17 41
A.1.3.3 CRC code 41
A.1.3.4 Speech parameters 41

A.2 C Code for speech channel decoding 41

Annex B (informative): Indexes 42

B.1 Index of C code routines 42

B.2 Index of files 45

Annex C (informative): Bibliography 46

Annex D (informative): Codec performance 47

D.1 General 47

D.2 Quality 47
D.2.1 Subjective speech quality 47
D.2.1.1 Description of characterization tests 47
D.2.1.2 Absolute speech quality 47
D.2.1.3 Effect of input level 47
D.2.1.4 Effect of input frequency characteristic 47
D.2.1.5 Effect of transmission errors 47
D.2.1.6 Effect of tandeming 48
D.2.1.7 Effect of acoustic background noise 48
D.2.1.8 Effect of vocal effort 48
D.2.1.9 Effect of frame stealing 48
D.2.1.10 Speaker and language dependency 48
D.2.2 Comparison with analogue FM 48
D.2.2.1 Analogue and digital systems results 48
D.2.2.2 All conditions 49
D.2.2.3 Input level 49
D.2.2.4 Error patterns 50
D.2.2.5 Background noise 50

D.2.3 Additional tests 50
D.2.3.1 Types of signals 50
D.2.3.2 Codec behaviour 50

D.3 Performance of the channel coding/decoding for speech 51
D.3.1 Classes of simulation environment conditions 51
D.3.2 Classes of equipment 51
D.3.3 Classes of bits 52
D.3.4 Channel conditions 52
D.3.5 Results for normal case 52

D.4 Complexity 53
D.4.1 Complexity analysis 53
D.4.1.1 Measurement methodology 53
D.4.1.2 TETRA basic operators 53
D.4.1.3 Worst case path for speech encoder 55
D.4.1.4 Worst case path for speech decoder 56
D.4.1.5 Condensed complexity values for encoder and decoder 57
D.4.2 DSP independence 58
D.4.2.1 Program control structure 58
D.4.2.2 Basic operator implementation 58
D.4.2.3 Additional operator implementation 58

D.5 Delay 58

Annex E (informative): Results of the TETRA codec characterization listening and complexity tests 59

E.1 Characterization listening test 59
E.1.1 Experimental conditions 59
E.1.2 Tables of results 60

E.2 TETRA codec complexity study 69
E.2.1 Computational analysis results 69
E.2.1.1 TETRA speech encoder 69
E.2.1.2 TETRA speech decoder 77
E.2.1.3 TETRA channel encoder and decoder 80
E.2.2 Memory requirements analysis results 82
E.2.2.1 TETRA speech encoder 82
E.2.2.2 TETRA speech decoder 83
E.2.2.3 TETRA speech channel encoder 83
E.2.2.4 TETRA speech channel decoder 84

Annex F (informative): Description of attached computer files 85

F.1 Directory C-WORD 85

F.2 Directory C-CODE 85

History 86

Blank page

Foreword

This European Telecommunication Standard (ETS) has been produced by Terrestrial Trunked Radio (TETRA) Project of the European Telecommunications Standards Institute (ETSI), and is now submitted for the ETSI standards One-step Approval Procedure (OAP).

The sole purpose of the copyright statement below is to protect the documentation of the standard itself and not the technology which is described therein.

ETSI states that the technology described herein is in the sole ownership of THOMSON-CSF (subject to the right of it's associated partner) and the disclosure of the standard documentation cannot be construed as granting any right whatsoever by licence or otherwise on the said technology.

THOMSON-CSF has undertaken to grant licences for the technology described in the standard, on fair, reasonable and non-discriminatory terms and conditions to the users of the present standard in accordance with the ETSI IPR Policy. Such licences are subject to a licence agreement to be agreed upon and entered into with THOMSON-CSF.

This ETS consists of four parts as follows:

Part 1: "General description of speech functions";

Part 2: "TETRA codec";

Part 3: "Specific operating features";

Part 4: "Codec conformance testing".

Clause 4 provides a complete description of the full rate speech source encoder and decoder, whilst clause 5 describes the speech channel encoder, and clause 6 the speech channel decoder.

Clause 7 describes the codec performance.

Finally, clause 8 introduces the bit exact description of the codec. This description is given as an ANSI C code, fixed point, bit exact. The whole C code corresponding to the TETRA codec is given in computer files attached to this ETS, and are an integral part of this ETS.

In addition to these clauses, five informative annexes are provided.

Annex A describes a possible implementation of the speech channel decoding function.

Annex B provides comprehensive indexes of all the routines and files included in the C code associated with this ETS.

Annex C lists informative references relevant to the speech codec.

Annex D describes the actual quality, performance and complexity aspects of the codec.

Annex E reports detailed results from codec characterization listening and complexity tests.

Annex F contains instructions for the use of the attached electronic files.

Proposed transposition dates	
Date of latest announcement of this ETS (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Blank page

1 Scope

This European Telecommunication Standard (ETS) contains the full specification of the speech codec for use in the Terrestrial Trunked Radio (TETRA) system.

2 Normative references

This ETS incorporates by dated and undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] ETS 300 392-2: "Radio Equipment and Systems (RES); Trans-European Trunked Radio (TETRA) system; Voice plus Data; Part 2: Air Interface".

[2] CCITT Recommendation P.48 (1988): "Specifications for an Intermediate Reference System".

3 Abbreviations

For the purposes of this ETS, the following abbreviations apply:

ACELP	Algebraic CELP
ANSI	American National Standards Institute
BER	Bit Error Ratio
BFI	Bad Frame Indicator
BS	Base Station
CELP	Code-Excited Linear Predictive
CRC	Cyclic Redundancy Code
DSP	Digital Signal Processor
DTMF	Dual Tone Multiple Frequency
EQ	EQualizer test
EP	Error Pattern
FIR	Finite Impulse Response
HT	Hilly Terrain
IRS	Intermediate Reference System
LP	Linear Prediction
LPC	Linear Predictive Coding
LSF	Line Spectral Frequency
LSP	Line Spectral Pair
MER	Message Error Rate
MNRU	Multiplicative Noise Reference Unit
MOS	Mean Opinion Score
MS	Mobile Station
MSE	Mean Square Error
PDF	Probability Density Function
PUEM	Probability of Undetected Erroneous Message
RCPC	Rate-Compatible Punctured Convolutional
RF	Radio Frequency
TDM	Time Division Multiplex
TU	Typical Urban
VQ	Vector Quantization

4 Full rate codec

4.1 Structure of the codec

The TETRA speech codec is based on the Code-Excited Linear Predictive (CELP) coding model. In this model, a block of *N* speech samples is synthesized by filtering an appropriate innovation sequence from a codebook, scaled by a gain factor g_c, through two time varying filters. A simplified high level block diagram of this synthesis process, as implemented in the TETRA codec, is shown in figure 1.

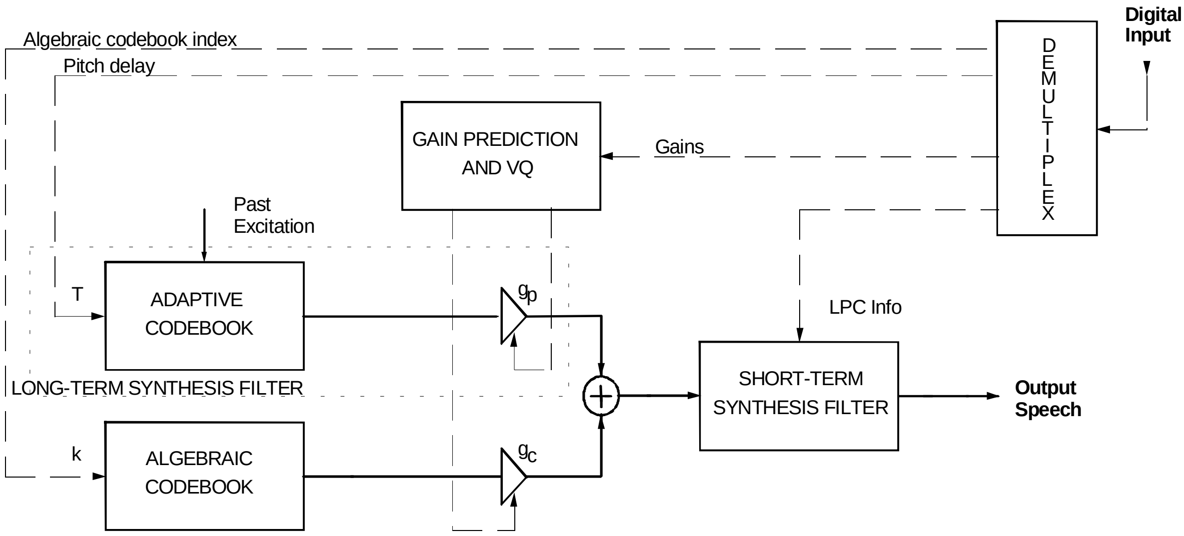

Figure 1: High level block diagram of the TETRA speech synthesizer

The first filter is a long-term prediction filter (pitch filter) aiming at modelling the pseudo-periodicity in the speech signal and the second is a short-term prediction filter modelling the speech spectral envelope.

The long-term or pitch, synthesis filter is given by:

$$\frac{1}{B(z)} = \frac{1}{1 - g_p z^{-T}} \tag{1}$$

where T is the pitch delay and g_p is the pitch gain. The pitch synthesis filter is implemented as an adaptive codebook, where for delays less than the sub-frame length the past excitation is repeated.

The short-term synthesis filter is given by:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{2}$$

where $a_i, i = 1,\ldots, p,$ are the Linear Prediction (LP) parameters and p is the predictor order. In the TETRA codec p shall be 10.

The TETRA encoder uses an analysis-by-synthesis technique to determine the pitch and excitation codebook parameters. The simplified block diagram of the TETRA encoder is shown in figure 2.

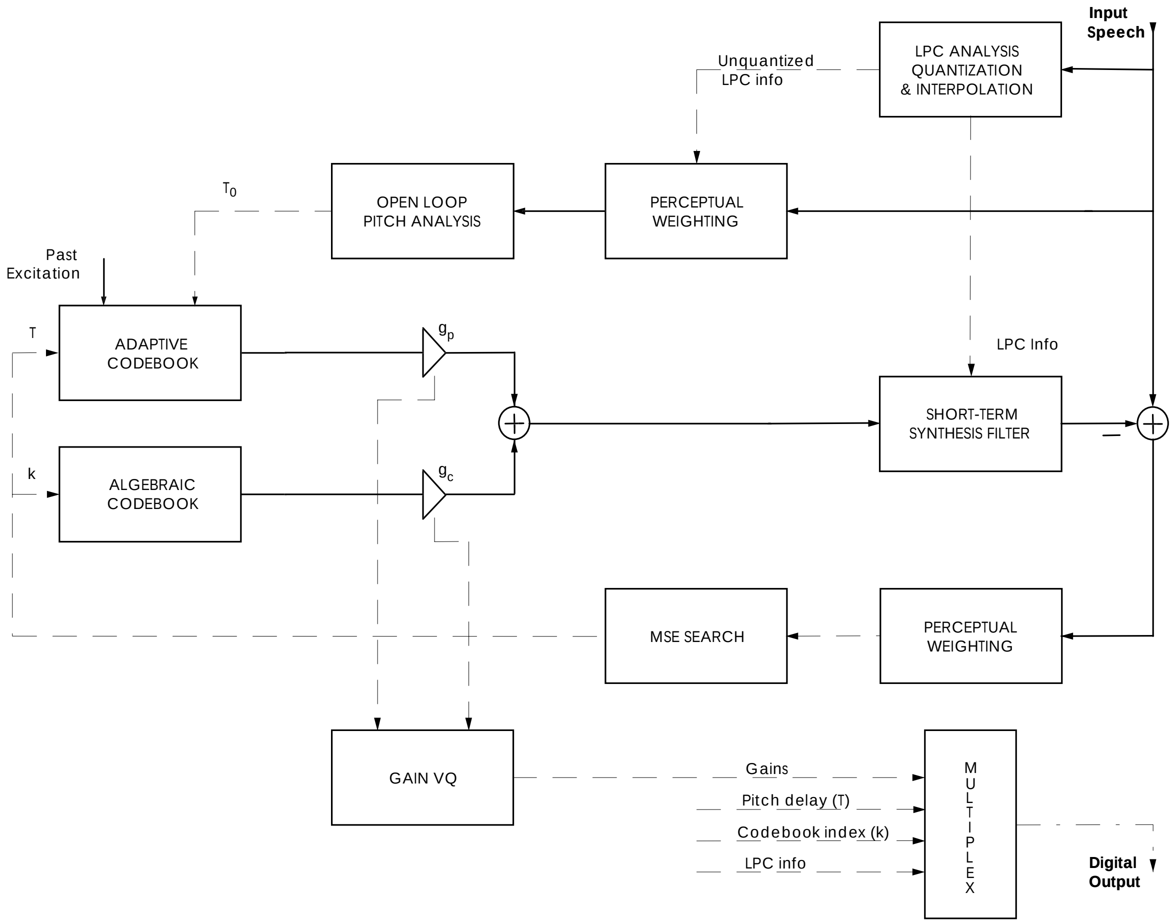

Figure 2: High level block diagram of the TETRA speech encoder

In this analysis-by-synthesis technique, the synthetic speech is computed for all candidate innovation sequences retaining the particular sequence that produces the output closer to the original signal according to a perceptually weighted distortion measure. The perceptual weighting filter de-emphasizes the error at the formant regions of the speech spectrum and is given by:

$$W(z) = \frac{A(z)}{A(z/\gamma)} \qquad (3)$$

where $A(z)$ is the LP inverse filter (as in Equation (2)) and $0 < \gamma \leq 1$. The value $\gamma_1 = 0{,}85$ shall be used. Both the weighting filter, $W(z)$, and formant synthesis filter, $H(z)$, shall use the quantized LP parameters.

In the Algebraic CELP (ACELP) technique, special innovation codebooks having an algebraic structure are used. This algebraic structure has several advantages in terms of storage, search complexity, and robustness. The TETRA codec shall use a specific dynamic algebraic excitation codebook whereby the fixed excitation vectors are shaped by a dynamic shaping matrix (see annex C {1}). The shaping matrix is a function of the LP model $A(z)$, and its main role is to shape the excitation vectors in the frequency domain so that their energies are concentrated in the important frequency bands. The shaping matrix used is a Toeplitz lower triangular matrix constructed from the impulse response of the filter:

$$F(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \qquad (4)$$

where $A(z)$ is the LP inverse filter. The values $\gamma_1 = 0,75$ and $\gamma_2 = 0,85$ shall be used.

In the TETRA codec, 30 ms speech frames shall be used. It is required that the short-term prediction parameters (or LP parameters) are computed and transmitted every speech frame. The speech frame shall be divided into 4 sub-frames of 7,5 ms (60 samples). The pitch and algebraic codebook parameters have also to be transmitted every sub-frame.

Table 1 gives the bit allocation for the TETRA codec. 137 bits shall be produced for each frame of 30 ms resulting in a bit rate of 4 567 bit/s.

Table 1: Bit allocation for the TETRA codec

Parameter	1st subframe	2nd subframe	3rd subframe	4th subframe	Total per frame
LP filter					26
Pitch delay	8	5	5	5	23
Algebraic code	16	16	16	16	64
VQ of 2 gains	6	6	6	6	24
Total					137

More details about the sequence of bits within the speech frame of 137 bits per 30 ms, with reference to the speech parameters, can be found in subclause 4.2.2.7, table 3.

4.2 Functional description of the codec

4.2.1 Pre- and post-processing

Before starting the encoding process, the speech signal shall be pre-processed using the offset compensation filter:

$$H_p(z) = \frac{1}{2}\left(\frac{1 - z^{-1}}{1 - \alpha z^{-1}}\right) \tag{5}$$

where α = 32 735/32 768. In the time domain, this filter corresponds to:

$$s'(n) = s(n)/2 - s(n-1)/2 + \alpha s'(n-1) \tag{6}$$

where $s(n)$ is the input signal and $s'(n)$ is the pre-processed signal. The purpose of this pre-processing is firstly to remove the dc from the signal (offset compensation), and secondly, to scale down the input signal in order to avoid saturation of the synthesis filtering.

At the decoder, the post-processing consists of scaling up the reconstructed signal (multiplication by 2 with saturation control).

4.2.2 Encoder

Figure 3 presents a detailed block diagram of the TETRA encoder illustrating the major parts of the codec as well as signal flow. On this figure, names appearing at the bottom of the various building blocks correspond to the C code routines associated with this ETS.

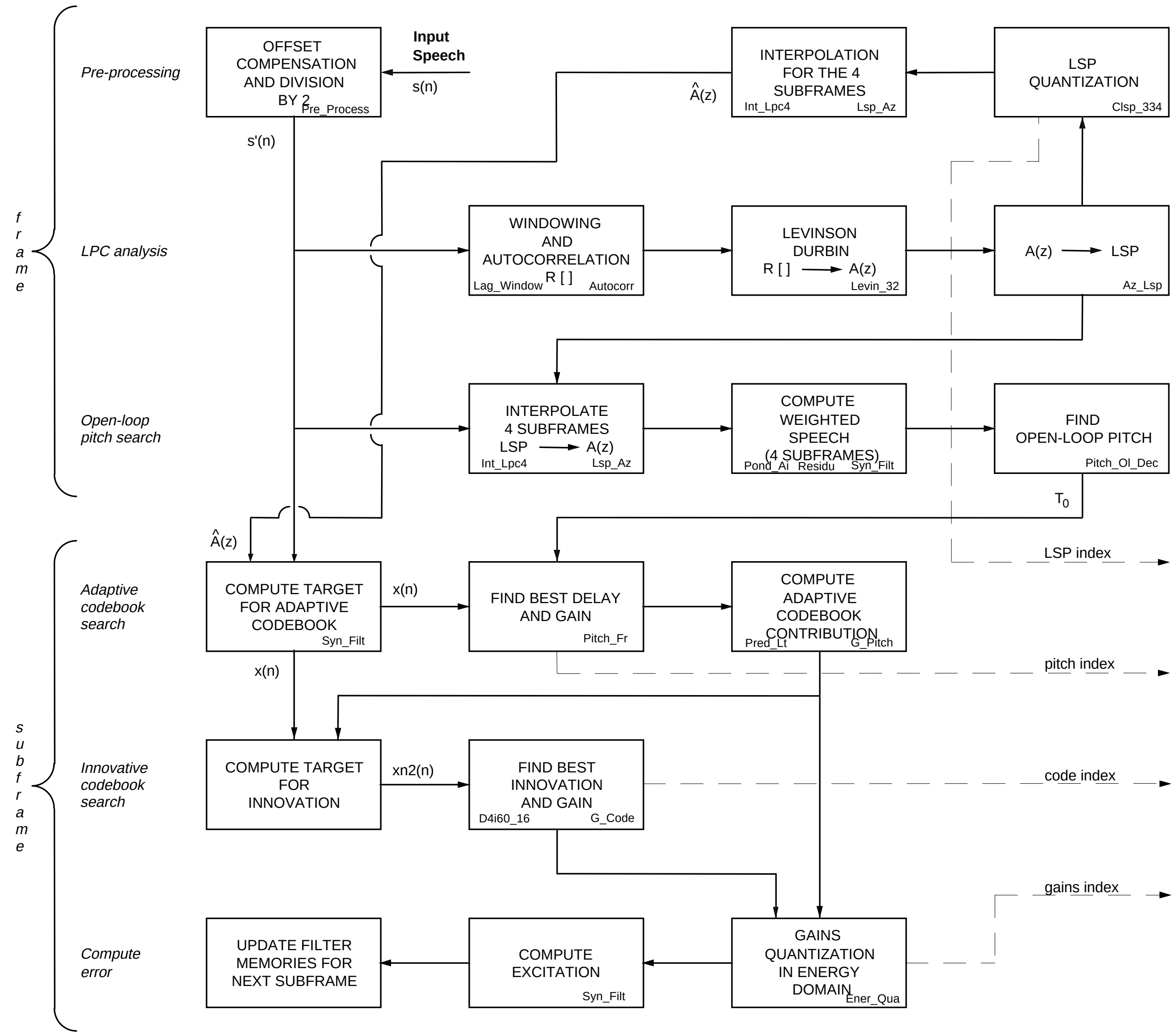

Figure 3: Signal flow at the encoder

4.2.2.1 Short-term prediction

Short-term prediction (LP or LPC analysis) shall be performed every 30 ms. The auto-correlation approach shall be used with an asymmetric analysis window. The LP analysis window consists of two halves of Hamming windows with different lengths. This window is given by:

$$
\begin{aligned}
w(n) &= 0{,}54 - 0{,}46\cos\left(\frac{\pi n}{L_1 - 1}\right), && n = 0,\ldots, L_1 - 1 \\
&= 0{,}54 + 0{,}46\cos\left(\frac{\pi(n - L_1)}{L_2 - 1}\right), && n = L_1,\ldots, L_1 + L_2 - 1
\end{aligned}
\qquad (7)
$$

A 32 ms analysis window (corresponding to 256 samples with the sampling frequency of 8 kHz) shall be used with values $L_1 = 216$ and $L_2 = 40$. The window shall be positioned such that 40 samples are taken from the future frame (look-ahead of 40 samples).

The auto-correlation of the windowed speech $s'(n), n = 0,\ldots,255$, are computed by:

$$r(k) \quad = \quad \sum_{n=k}^{255} s'(n)s'(n-k), \qquad\qquad k = 0,\ldots,10 \qquad (8)$$

and a 60 Hz bandwidth expansion has to be used by lag windowing the auto-correlation using the window (see annex C {2}):

$$w_{lag}(i) \quad = \quad \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 i}{f_s}\right)^2\right], \qquad\qquad i = 1,\ldots,10 \qquad (9)$$

where f_0 = 60 Hz is the bandwidth expansion and f_s = 8 000 Hz is the sampling frequency. Further, $r(0)$ is multiplied by 1,00005 which is equivalent to adding a noise floor at -43 dB. In the TETRA coder, this is alternatively performed by dividing the lag window as in equation (9) by 1,00005, resulting in $w'_{lag}(0) = 1$ and:

$$w'_{lag}(i) \quad = \quad w_{lag}(i)/1{,}00005 \qquad\qquad i = 1,\ldots,10 \qquad (10)$$

The modified auto-correlation:

$$r'(k) \quad = \quad r(k)w'_{lag}(k), \qquad\qquad k = 0,\ldots,10 \qquad (11)$$

are used to obtain the LP filter coefficients $a_k, k = 1,\ldots,10$, by solving the set of equations:

$$\sum_{k=1}^{10} a_k r'(|i-k|) \quad = \quad -r'(i), \qquad\qquad i = 1,\ldots,10 \qquad (12)$$

The set of equations in (12) shall be solved using the Levinson-Durbin algorithm (see annex C {3}).

4.2.2.2 LP to LSP and LSP to LP conversion

The LP filter coefficients of $A(z)$ $(a_k, k = 1,\ldots,10)$ shall be converted to the Line Spectral Pair (LSP) representation (see annex C {4}) for quantization and interpolation purposes. For a 10th order LP filter, the LSPs are defined as the roots of the sum and difference polynomials:

$$F'_1(z) \quad = \quad A(z) \quad + \quad z^{-11}A\left(z^{-1}\right) \qquad (13)$$

and

$$F'_2(z) \quad = \quad A(z) \quad - \quad z^{-11}A\left(z^{-1}\right) \qquad (14)$$

respectively. It can be proven that all roots of these polynomials are on the unit circle and they alternate each other (see annex C {5}). $F'_1(z)$ has a root $z = -1(\omega = \pi)$ and $F'_2(z)$ has a root $z = 1(\omega = 0)$.

To eliminate these two roots, new polynomials are defined:

$$F_1(z) = F_1'(z) \ / \ \left(1+z^{-1}\right) \tag{15}$$

and

$$F_2(z) = F_2'(z) \ / \ \left(1-z^{-1}\right) \tag{16}$$

Each polynomial has 5 conjugate roots on the unit circle $\left(e^{\pm j\omega_i}\right)$, therefore, the polynomials can be written as:

$$F_1(z) = \prod_{i=1,3,\ldots,9} \left(1-2q_i z^{-1}+z^{-2}\right) \tag{17}$$

and

$$F_2(z) = \prod_{i=2,4,\ldots,10} \left(1-2q_i z^{-1}+z^{-2}\right) \tag{18}$$

where $q_i = \cos(\omega_i)$, with ω_i being the Line Spectral Frequencies (LSFs). They satisfy the ordering property $0 < \omega_1 < \omega_2 < \ldots < \omega_{10} < \pi$. q_i are referred as the LSPs in the cosine domain.

The first five coefficients of each of the symmetric polynomials $F_1(z)$ and $F_2(z)$ are found by the recursive relations (for $i = 0$ to 4):

$$\begin{aligned} f_1(i+1) &= a_{i+1} + a_{p-i} - f_1(i) \\ f_2(i+1) &= a_{i+1} - a_{p-i} + f_2(i) \end{aligned} \tag{19}$$

The LSPs are found by evaluating the polynomials $F_1(z)$ and $F_2(z)$ at 60 points equally spaced between 0 and π and checking for sign changes. A sign change signifies the existence of a root and the sign change interval is then divided 4 times to better track the root. The Chebyshev polynomials have to be used to evaluate $F_1(z)$ and $F_2(z)$ (see annex C {6}). This method is very computationally efficient since it bypasses the cosine computations as the roots are found directly in the cosine domain $\{q_i\}$. In the TETRA codec, implementation, quantization and interpolation of the LSPs are performed in the cosine domain, thus no trigonometric computations are needed to convert to the frequency domain. The polynomials $F_1(z)$ or $F_2(z)$ are given by:

$$F(z) = 2e^{-j5\omega}\left(T_5(x) + f(1)T_4(x) + f(2)T_3(x) + f(3)T_2(x) + f(4)T_1(x) + f(5)/2\right) \tag{20}$$

where $T_m(x) = \cos(m\omega)$ is the mth order Chebyshev polynomial, and $f(i), i = 1,\ldots 5$, are the coefficients of either $F_1(z)$ or $F_2(z)$, computed using the equations in (19). The details of the Chebyshev polynomial evaluation method are found in (see annex C {6}). If this numerical process is not able to find enough roots, the previous computed set of LSPs is used.

Once the LSPs are quantized and interpolated, they are converted back to the LP coefficient domain $\{A(z)\}$. The conversion to the LP domain is done as follows. The coefficients of $F_1(z)$ and $F_2(z)$ are found by expanding equations (17) and (18) knowing the quantized and interpolated LSPs $q_i, i = 1,\ldots 10$.

The following recursive relation shall be used to compute $f_1(i)$:

for $i = 1$ to 5

$$f_1(i) = -2q_{2i-1} f_1(i-1) + 2 f_1(i-2)$$

for $j = i-1$ down to 1

$$f_1(j) = f_1(j) - 2q_{2i-1} f_1(j-1) + f_1(j-2)$$

with initial values $f_1(0) = 1$ and $f_1(-1) = 0$. The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-1} by q_{2i}. Once the coefficients $f_1(i)$ and $f_2(i)$ are found, $F_1(z)$ and $F_2(z)$ are multiplied by $1+z^{-1}$ and $1-z^{-1}$, respectively, to obtain $F_1'(z)$ and $F_2'(z)$; that is $f_1'(i) = f_1(i) + f_1(i-1)$ and $f_2'(i) = f_2(i) - f_2(i-1), i = 1,\dots,5$. Finally the LP coefficients are found by $a_i = 0{,}5 f_1(i) + 0{,}5 f_2(i), i = 1,\dots,5$ and $a_i = 0{,}5 f_1(i-5) - 0{,}5 f_2(i-5), i = 5,\dots,10$. This is directly derived from the relation $A(z) = \left(F_1'(z) + F_2'(z)\right)/2$, and considering the fact that $F_1'(z)$ and $F_2'(z)$ are symmetrical and anti-symmetrical polynomials, respectively.

4.2.2.3 Quantization and interpolation of LP parameters

The computed LP parameters have to be converted to LSPs and quantized with 26 bits using split-VQ.

NOTE: Both the quantization and interpolation are performed on the LSPs in the cosine domain; that is:

$$q_i = \cos(2\pi f_i / f_s), \qquad i = 1,\dots,10 \tag{21}$$

where f_i is the line spectral frequencies in Hz and f_s is the sampling frequency.

The LSP vector $\mathbf{q}$ shall be split into three sub-vectors of length 3, 3, and 4. The first sub-vector $\{q_1, q_2, q_3\}$ shall be quantized with 8 bits while the sub-vectors $\{q_4, q_5, q_6\}$ and $\{q_7, q_8, q_9, q_{10}\}$ shall be each quantized with 9 bits. The search is performed using Mean Square Error (MSE) minimization in the $\mathbf{q}$ domain with no LSP weighting.

The quantized LP parameters are used for the fourth sub-frame, whereas the first three sub-frames use a linear interpolation of the parameters of the present and previous frames. The interpolation is performed on the LSPs in the $\mathbf{q}$ domain. Let $\hat{\mathrm{q}}_n$ be the quantized LSP vector at the present frame and $\hat{\mathrm{q}}_{n-1}$ the quantized LSP vector at the past frame. The interpolated LSP vectors at each of the 4 sub-frames are given by:

$$\begin{aligned}
\mathrm{q}_1 &= 0{,}75\hat{\mathrm{q}}_{n-1} + 0{,}25\hat{\mathrm{q}}_n \\
\mathrm{q}_2 &= 0{,}50\hat{\mathrm{q}}_{n-1} + 0{,}50\hat{\mathrm{q}}_n \\
\mathrm{q}_3 &= 0{,}25\hat{\mathrm{q}}_{n-1} + 0{,}75\hat{\mathrm{q}}_n \\
\mathrm{q}_4 &= \hat{\mathrm{q}}_n
\end{aligned} \tag{22}$$

The initial values of the past quantized LSP vector are given in Q15 by $\hat{\mathrm{q}}_{-1}$ = {30 000, 26 000, 21 000, 15 000, 8 000, 0, -8 000, -15 000, -21 000, -26 000}. (Divide by 2^{15} to obtain the values in the range [-1,1]). The interpolated LSP vectors shall be used to compute a different LP filter at each sub-frame.

4.2.2.4 Long-term prediction analysis

The aim of the long term prediction analysis or adaptive codebook search is to find the best pitch parameters, which are the delay and gain values for the pitch filter. The pitch filter shall be implemented using the so-called adaptive codebook approach whereby the excitation is repeated for delays less than the sub-frame length (60). In this implementation the excitation is extended by the LP residual in the search stage to simplify the closed-loop search. In the first sub-frame, a fractional pitch delay is used with resolutions: 1/3 in the range $\left[19\frac{1}{3} \quad - \quad 84\frac{2}{3}\right]$ and integers only in the range [85 - 143]. For the other sub-frames, a pitch resolution of 1/3 is always used in the range $\left[T_1 - 5\frac{2}{3} \quad - \quad T_1 + 4\frac{2}{3}\right]$, where T_1 is the nearest integer to the fractional pitch lag of the first sub-frame.

To simplify the pitch analysis procedure, a two stage approach shall be used, comprising first an open loop pitch search followed by a closed loop search.

The open loop pitch has to be computed once every speech frame (30 ms) using a weighted speech signal $s_w(n)$. A pole-zero type weighting procedure shall be used to get $s_w(n)$. This procedure shall be performed with the help of a shaping filter $A(z/0{,}95)/A(z/0{,}60)$ for which the un-quantized LP parameters shall be used.

The open loop pitch search shall then be performed as follows. In a first step, 3 maxima of the correlation:

$$C_k = \sum_{j=0}^{120} s_w(2j)s_w(2j-k) \tag{23}$$

are found in the three ranges, [20 - 39], [40 - 79] and [80 - 142], respectively. The retained maxima $C_{k_i}, i = 1,\ldots,3,$ are normalized by dividing by $\sqrt{\sum_n s_w^2(n-k_i)}, i = 1,\ldots,3,$ respectively. The normalized maxima and corresponding delays are denoted by $(R_i, k_i), i = 1,\ldots 3$. The winner among the three normalized correlation is selected by favouring the delays in the lower ranges. That is, k_i is selected if $R_i > 0{,}85 R_{i+1}$. This procedure of dividing the delay range into 3 sections and favouring the lower sections is used to avoid choosing pitch multiples.

NOTE 1: The past weighted speech samples are initialized to zero.

Having found the open-loop pitch T_{op}, a closed-loop pitch analysis has to be performed around the open-loop pitch delay on a sub-frame basis. In the first sub-frame the range $T_{op} \pm 2$ bounded by [20 - 143] is searched. For the other sub-frames, closed-loop pitch analysis is performed around the pitch selected in the first sub-frame. As mentioned earlier, a pitch resolution of 1/3 is always used for the other sub-frames in the range $\left[T_1 - 5\frac{2}{3} \quad - \quad T_1 + 4\frac{2}{3}\right]$, where T_1 is the integer part of the first sub-frame pitch lag. The pitch delay shall be encoded with 8 bits in the first sub-frame while the relative delays of the other sub-frames shall be encoded with 5 bits per sub-frame.

The closed loop pitch search shall be performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term:

$$\tau_k = \frac{\sum_{n=0}^{59} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{59} y_k(n)y_k(n)}}, \tag{24}$$

where $x(n)$ is the target for the adaptive codebook search given by the weighted input speech after subtracting the zero-input response of the weighted synthesis filter $H(z)W(z)$ and $y_k(n)$ is the past filtered excitation at delay k (the past excitation is initialized to zero).

NOTE 2: The search range is limited around the open-loop pitch as explained earlier.

For delays $k < 60$ the excitation signal $u(n)$ is extended by the LP residual signal. Once the optimum integer pitch delay is determined, the fractions $-\frac{2}{3}$, $-\frac{1}{3}$, $\frac{1}{3}$, and $\frac{2}{3}$ around that integer are tested.

NOTE 3: For the first sub-frame, the fractions are tested only if the integer pitch lag is less than 85.

The fractional pitch search is performed by interpolating the normalized correlation in equation (24) and searching for its maximum. Once the non-integer pitch is determined, the adaptive codebook vector $v(n)$ is computed by interpolating the past excitation signal $u(n)$. The interpolation shall be performed using two FIR filters (Hamming windowed sinc functions); one for interpolating the term in equation (24) with the sinc truncated at ± 12 (8 multiplications per fraction) and the other for interpolating the past excitation with the sinc truncated at ± 48 (32 multiplications per sample). The pitch gain is then found by:

$$g_p = \frac{\sum_{n=0}^{59} x(n)y(n)}{\sum_{n=0}^{59} y(n)y(n)}, \qquad \text{bounded by} \qquad 0 \le g_p \le 1{,}2 \qquad (25)$$

where $y(n) = v(n) * h(n)$ is the filtered adaptive codebook vector (zero-state response of $H(z)W(z)$ to $v(n)$).

NOTE 4: Only positive pitch gains are allowed since by maximizing the term in equation (24) the negative correlations are eliminated.

4.2.2.5 Algebraic codebook: structure and search

A 16-bit algebraic codebook shall be used in the innovative codebook search, the aim of which is to find the best innovation and gain parameters. The innovation vector contains, at most, four non-zero pulses. The 4 pulses can assume the amplitudes and positions given in the following table:

Table 2

Codebook parameters	Positions of the pulses	Codebook bit allocation
Pulse amplitude: +1,4142	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58	5
Pulse amplitude: -1	2, 10, 18, 26, 34, 42, 50, 58	3
Pulse amplitude: +1	4, 12, 20, 28, 36, 44, 52, (60)	3
Pulse amplitude: -1	6, 14, 22, 30, 38, 46, 54, (62)	3
Global sign flag		1
Shift flag		1

The pulses shall have fixed amplitudes of +1,4142, -1, +1 and -1, respectively. The first pulse position shall be encoded with 5 bits while the positions of the other pulses shall be encoded with 3 bits. The positions of all pulses can be simultaneously shifted by one, to occupy odd positions. One bit shall be used to encode this shift and a global sign bit shall be used to invert all pulses simultaneously, giving a total of 16 bits.

NOTE 1: From table 2, it is possible to position the last two pulses outside the sub-frame which indicates that these pulses are not present.

The codebook is searched by minimizing the mean squared error between the weighted input speech and the weighted synthesis speech. The target signal used in the closed-loop pitch search is updated by subtracting the adaptive codebook contribution. That is, the target for the innovation is computed using:

$$x_2(n) \quad = \quad x(n) - g_p y(n), \qquad\qquad n = 0,\ldots 59 \tag{26}$$

where $y(n) = v(n) * h(n)$ is the filtered adaptive codebook vector, with $h(n)$ being the impulse response of the weighted synthesis filter $H(z)W(z) = 1 / A(z/\gamma)$.

As described in subclause 4.1 the algebraic codebook is dynamically shaped to enhance the important frequency regions. The used shaping matrix is a lower triangular convolution matrix consisting of the impulse response of the filter $F(z)$ in equation (4). Thus the shaping can be performed as a filtering process. To maintain the simplicity of the algebraic codebook search, the filter $F(z)$ is combined with the weighted synthesis filter $H(z)W(z)$ and the impulse response $h'(n)$ of the combined filter is computed (see annex C {1}). If c_k is the algebraic codeword at index k, then the algebraic codebook is searched by maximizing the term:

$$\tau_k = \frac{C_k^2}{\varepsilon_k} = \frac{\left(\mathrm{d}^t c_k\right)^2}{c_k^t \Phi c_k} \tag{27}$$

where $\mathbf{H}$ is a lower triangular Toeplitz convolution matrix with diagonal $h'(0)$ and lower diagonals $h'(1),\ldots,h'(59)$ and $\mathbf{d} = \mathbf{H}^t x_2$ is the backward filtered target vector and $\mathbf{\Phi} = \mathbf{H}^t\mathbf{H}$.

The algebraic structure of the codebook allows for very fast search procedures since the innovation vector c_k contains only 4 non-zero pulses. The search shall be performed in 4 nested loops, corresponding to each pulse positions, where in each loop the contribution of a new pulse is added. The correlation in equation (27) is given by:

$$C = ad(m_0) - d(m_1) + d(m_2) - d(m_3) \tag{28}$$

and the energy is given by:

$$\begin{aligned}\varepsilon \quad = \quad & a^2\phi(m_0,m_0) \\ & + \phi(m_1,m_1) - 2a\phi(m_0,m_1) \\ & + \phi(m_2,m_2) + 2a\phi(m_0,m_2) - 2\phi(m_1,m_2) \\ & + \phi(m_3,m_3) - 2a\phi(m_0,m_3) + 2\phi(m_1,m_3) - 2\phi(m_2,m_3)\end{aligned} \tag{29}$$

where m_i is the position of the ith pulse and a = 1,4142.

NOTE 2: The codebook gain is given by:

$$g_c = \frac{C}{\varepsilon} \tag{30}$$

A focused search approach shall be used to further simplify the search procedure.

In this approach pre-computed thresholds are tested before entering the last two loops and the loops are entered only if these thresholds are exceeded. The maximum number of times the loops can be entered is fixed so that a low percentage of the codebook is searched.

The two thresholds are computed based on the correlation C. The maximum absolute correlation due to the contribution of the first two pulses, max_2, and that due to the contribution of the first three pulses, max_3, are found prior to the codebook search.

The third loop is entered only if the absolute correlation (due to two pulses) exceeds $k_2\, max_2$, and similarly, the fourth loop is entered only if the absolute correlation (due to three pulses) exceeds $k_3\, max_3$, where $0 \le k_2, k_3 < 1$. The values of k_2 and k_3 control the percentage of codebook search, with higher values resulting in faster search time (setting $k_2 = k_3 = 0$ results in full search). The values $k_2 = k_3 = 0{,}586$ shall be used.

The focused search approach results in variable search time from one sub-frame to another. To control the worst case time, a down counter is set to 350 and it is decreased by 4 each time the third loop is completed and decreased by 3 each time the fourth loop is completed. In the worst cases where this counter could reach a value below 0 the codebook search is ended.

As the above described codebook search accounts for most of the complexity of the codec, a detailed flow diagram of this procedure is given in figure 4.

For the positions of 1st pulse:

Correlation due to 1st pulse;
Energy due to 1st pulse;

For the positions of 2nd pulse:

Add the contribution of 2nd pulse to correlation;
Add the contribution of 2nd pulse to energy;
If (correlation > first threshold) continue with 3rd pulse;

For the positions of 3rd pulse:

Add the contribution of 3rd pulse to correlation;
Add the contribution of 3rd pulse to energy;
If (correlation > second threshold) continue with 4th pulse;

For the positions of 4th pulse:

Add the contribution of 4th pulse to correlation;
Add the contribution of 4th pulse to energy;
Test for new maximum of square correlation divided by energy;
If new maximum, save optimum position of 4 pulses;

End of 4th pulse loop;

End of 3rd pulse loop;

End of 2nd pulse loop;

End of 1st pulse loop.

Figure 4: Flow diagram of the codebook search procedure

A special feature of the codebook is that, for pitch delays less than the sub-frame size 60, a fixed gain pitch contribution shall be added to the fixed excitation vector. That is, after the optimum algebraic code $c(n)$ is determined, it is modified by $c(n) \longleftarrow c(n) + 0{,}8c(n-T)$ with T being the integer pitch period. This was found to improve the performance of female speakers.

NOTE 3: Prior to the codebook search, the impulse response $h'(n)$ should be modified in a similar fashion if $T < 60$.

NOTE 4: Since the algebraic code-vector is to be passed through the shaping filter $F(z)$, this special feature is implemented by only modifying the impulse response of $F(z)$ $f(n) \longleftarrow f(n) + 0{,}8 f(n-T)$. In this case both the shaped code-vector and the impulse response $h'(n)$ will be implicitly modified.

4.2.2.6 Quantization of the gains

The adaptive and fixed codebook gains are quantized in terms of pitch excitation and innovative excitation energies using predictive Vector Quantization in the energy domain. The adaptive code-vector energy is given by:

$$E_a = \log_2\left(\left(\sum_{i=0}^{59} v^2(i) + \varepsilon\right) p_g\right) \tag{31}$$

where $\varepsilon = 1$ is used to avoid $\log_2(0)$ and p_g is the prediction gain of the quantized LP filter approximated by $\sum_{i=0}^{59} \hat{h}^2(i)$ where $\hat{h}(i)$ is the impulse response of the synthesis filter. The pitch excitation energy at frame n is defined as:

$$E_p^{(n)} = E_a + \log_2\left(g_p^2 + \varepsilon\right) \tag{32}$$

where g_p is the adaptive codebook gain.

Similarly, the fixed codebook energy is given by:

$$E_f = \log_2\left(\left(\sum_{i=0}^{59} c^2(i)\right) p_g\right) \tag{33}$$

and taking the codebook gain into account, the innovative excitation energy at frame n is given by:

$$E_c^{(n)} = E_f + \log_2\left(g_c^2 + \varepsilon\right) \tag{34}$$

where g_c is the fixed codebook gain.

The predicted pitch and innovative energies at sub-frame n are given by:

$$\begin{array}{lcllcl} \tilde{E}_p^{(n)} & = & 0{,}5\hat{E}_p^{(n-1)} & + & 0{,}25\hat{E}_c^{(n-1)} & - & 3{,}0 \\ \tilde{E}_c^{(n)} & = & 0{,}25\hat{E}_p^{(n-1)} & + & 0{,}5\hat{E}_c^{(n-1)} & - & 3{,}0 \end{array} \tag{35}$$

where $\hat{E}_p^{(k)}$ and $\hat{E}_c^{(k)}$ are the quantized energies at sub-frame k. Initially, $\hat{E}_p^{(-1)}$ and $\hat{E}_c^{(-1)}$ are both set to zero. The prediction errors on the pitch and code energies are given by:

$$\begin{array}{lcllcl} R_p^{(n)} & = & E_p^{(n)} & - & \tilde{E}_p^{(n)} \\ R_c^{(n)} & = & E_c^{(n)} & - & \tilde{E}_c^{(n)} \end{array} \tag{36}$$

The prediction errors $\left(R_p, R_c\right)$ shall be vector quantized with a 6-bit codebook to obtain $\left(\hat{R}_p, \hat{R}_c\right)$.

The quantized energies are given by:

$$\begin{aligned} \hat{E}_p^{(n)} &= \hat{R}_p^{(n)} + \tilde{E}_p^{(n)} \\ \hat{E}_c^{(n)} &= \hat{R}_c^{(n)} + \tilde{E}_c^{(n)} \end{aligned} \qquad (37)$$

These quantized energies $\hat{E}_p^{(n)}$ and $\hat{E}_c^{(n)}$ are limited respectively to 27 and 25 in order to avoid bursts of energy in case of non-recovered transmission errors.

Finally, the quantized adaptive and fixed excitation gains are found by:

$$\hat{g}_p = 2{,}0^a \text{; where } a = 0{,}5\left(\hat{E}_p^{(n)} - E_a\right) \qquad (38)$$

$$\hat{g}_c = 2{,}0^b \text{; where } b = 0{,}5\left(\hat{E}_c^{(n)} - E_f\right) \qquad (39)$$

4.2.2.7 Detailed bit allocation

The following table details the encoder output parameters in order of occurrence and bit allocation within the speech frame of 137 bits per 30 ms.

Table 3: Meaning of each bit within a frame

Parameter class	Parameter name	Number of bits	Bit number (MSB-LSB)
Filter	Codebook index: LSP1 to LSP3	8	B1 - B8
	Codebook index: LSP4 to LSP6	9	B9 - B17
	Codebook index: LSP7 to LSP10	9	B18 - B26
Sub-frame No 1	Pitch delay	8	B27 - B34
	Codebook index: pulse 4	3	B35 - B37
	Codebook index: pulse 3	3	B38 - B40
	Codebook index: pulse 2	3	B41 - B43
	Codebook index: pulse 1	5	B44 - B48
	Pulse global sign	1	B49
	Pulse shift	1	B50
	Codebook index: gains	6	B51 - B56
Sub-frame No 2	Pitch delay	5	B57 - B61
	Codebook index: pulse 4	3	B62 - B64
	Codebook index: pulse 3	3	B65 - B67
	Codebook index: pulse 2	3	B68 - B70
	Codebook index: pulse 1	5	B71 - B75
	Pulse global sign	1	B76
	Pulse shift	1	B77
	Codebook index: gains	6	B78 - B83
Sub-frame No 3	Pitch delay	5	B84 - B88
	Codebook index: pulse 4	3	B89 - B91
	Codebook index: pulse 3	3	B92 - B94
	Codebook index: pulse 2	3	B95 - B99
	Codebook index: pulse 1	5	B100 - B102
	Pulse global sign	1	B103
	Pulse shift	1	B104
	Codebook index: gains	6	B105 - B110
Sub-frame No 4	Pitch delay	5	B111 - B115
	Codebook index: pulse 4	3	B116 - B118
	Codebook index: pulse 3	3	B119 - B121
	Codebook index: pulse 2	3	B122 - B124
	Codebook index: pulse 1	5	B125 - B129
	Pulse global sign	1	B130
	Pulse shift	1	B131
	Codebook index: gains	6	B132 - B137

4.2.3 Decoder

Figure 5 presents a detailed block diagram illustrating the major components of the TETRA speech decoder as well as signal flow. On this figure, names appearing at the bottom of the various building blocks correspond to the C code routines attached to this ETS.

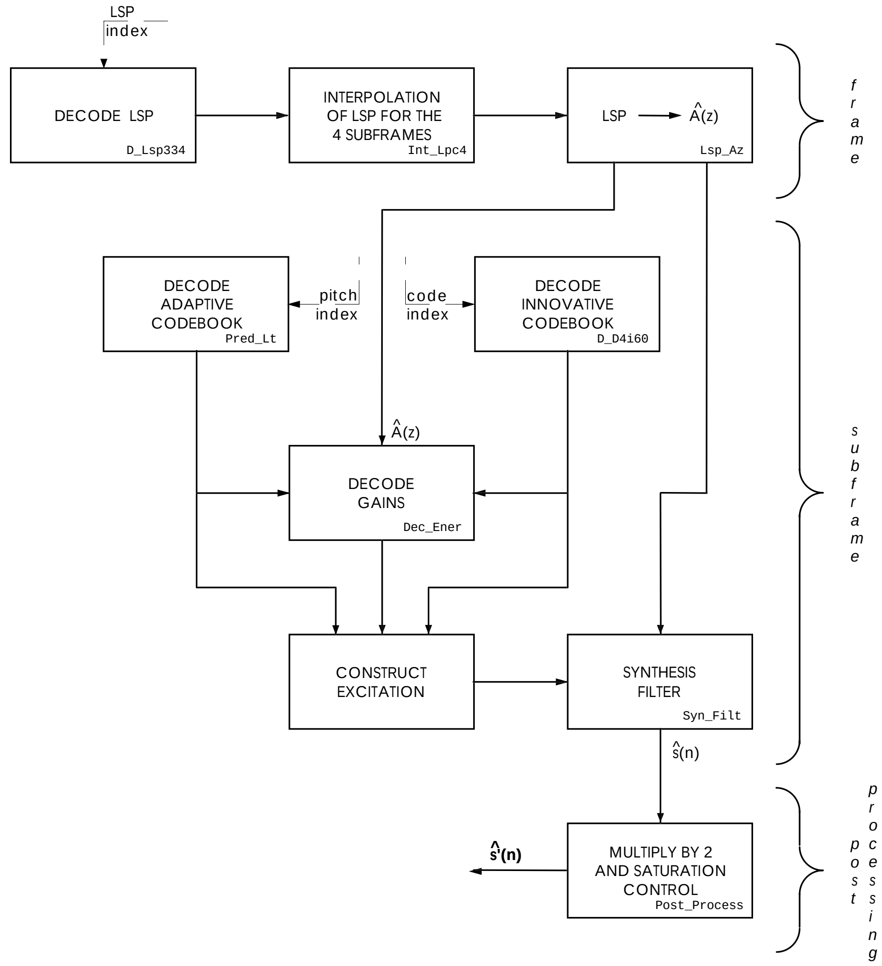

Figure 5: Signal flow at the decoder

4.2.3.1 Decoding process

The function of the decoder consists of decoding the transmitted parameters (LP parameters, adaptive codebook vector, algebraic code vector, and gains) and performing synthesis to obtain the reconstructed speech. The decoding process shall be as described in the following subclauses.

4.2.3.1.1 Decoding of LP filter parameters

The received indices of LSP quantization shall be used to reconstruct the quantized LSP vector $\hat{q}_n$. The interpolation between this vector and the one received in the previous frame $\hat{q}_{n-1}$ shall be performed as described in subclause 4.2.2.3 to obtain 4 interpolated LSP vectors (corresponding to 4 sub-frames). For each sub-frame, the interpolated LSP vector has to be converted to the LP filter $A(z)$ with coefficients a_k, which shall be used for synthesizing the reconstructed speech in the sub-frame.

4.2.3.1.2 Decoding of the adaptive codebook vector

The received pitch delay (adaptive codebook index) shall be used to find the integer and fractional parts of the pitch lag. The adaptive codebook vector $v(n)$ shall be found by interpolating the past excitation $u(n)$ (at the pitch delay) using the FIR filters described in subclause 4.2.2.4.

4.2.3.1.3 Decoding of the innovation vector

The received algebraic codebook index shall be used to extract the positions of the 4 non-zero pulses and to find the algebraic code-vector $c'(n)$. The impulse response $f(n)$ of the shaping filter $F(z)$ (given in equation (4)) shall be computed (using the interpolated LP coefficients). If the integer part of the pitch lag T is less than the sub-frame size (60 samples), the impulse response $f(n)$ shall be modified by $f(n) \longleftarrow f(n) + 0{,}8 f(n-T)$ to account for the fixed-gain pitch contribution to the code. Finally, the innovative vector $c(n)$ shall be found by convolving the algebraic code-vector $c'(n)$ with $f(n)$. This has to be performed by adding 4 delayed versions of $f(n)$ scaled by the pulse amplitudes since $c'(n)$ contains only 4 non-zero pulses.

4.2.3.1.4 Decoding of the adaptive and innovative codebook gains

The adaptive and innovative codebook gains shall be found according to the description given in subclause 4.2.2.6. The predicted pitch and innovative energies shall be computed as in equation (35). The adaptive code-vector energy E_a and the fixed code-vector energy E_f shall be computed as in equations (31) and (33), respectively. The received index of gain VQ shall be used to find the quantized prediction errors $\hat{R}_p$ and $\hat{R}_c$. The quantized energies $\hat{E}_p^{(n)}$ and $\hat{E}_c^{(n)}$ shall be found according to equation (37). Finally, the quantized adaptive codebook gain $\hat{g}_p$ and the quantized innovation gain $\hat{g}_c$ shall be computed according to equations (38) and (39), respectively.

4.2.3.1.5 Computation of the reconstructed speech

The excitation at the input of the synthesis filter is given by:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \qquad (40)$$

The reconstructed speech for a sub-frame of length 60 is given by:

$$\hat{s}(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i), \qquad n = 0,\ldots,59 \qquad (41)$$

where $\hat{a}_i$ are the interpolated LP coefficients of the synthesis filter as in equation (2).

4.2.3.2 Error concealment

When a Bad Frame Indicator (BFI) is received (indicating that the frame is badly corrupted or lost), the decoder shall perform an error concealment procedure utilizing the parameters of the last received "correct" frame. The error concealment procedure shall consist of the following steps:

- keep the previous "correct" LSP parameters;
- keep the previous "correct" pitch period (of the 4th sub-frame in the past frame) and repeat it for the 4 sub-frames in the present bad frame while setting the fraction to zero;
- in the decoding of the gains of the adaptive and innovative codebooks, the set of equations (37) is replaced by:

$$\begin{aligned} \hat{E}_p^{(n)} &= \hat{E}_p^{(n-1)} - 0{,}5 \\ \hat{E}_c^{(n)} &= \hat{E}_c^{(n-1)} - 0{,}5 \end{aligned} \qquad (42)$$

  This corresponds to decreasing the energies of the pitch and innovative vectors of the previous sub-frame by 1,5 dB;

- for the innovative codebook indices, keep the respective 4 indices of the previous frame.

5 Channel coding for speech

5.1 General

This clause shall apply to the speech traffic channel only.

A reference configuration of the TETRA transmission chain is given in ETS 300 392-2 [1], clause 19. Using this reference configuration, this clause defines the error control process which applies to the information bits (packed in MAC blocks, see definition in ETS 300 392-2 [1], clause 3), and which provides multiplexed bits (packed in multiplexed blocks).

This clause provides a definition of the error control process for the speech traffic channel (TCH/S). The definition of all the logical channels for the V+D system, including the speech traffic channel, is given in ETS 300 392-2 [1], clause 9.

This clause includes the specification of encoding, re-ordering and interleaving for the speech traffic channel, but does not specify any data processing in the receiver.

5.2 Interfaces in the error control structure

The definition of interfaces in the error control structure is given in figure 6.

The speech traffic channel has its own error control scheme. The information bits, corresponding to the input of the channel encoder, are referred to as type-1 bits. The type-1 bits are packed in MAC blocks, that are referred to as type-1 blocks; this defines interface 1 in figure 6.

The processing on the transmit part shall be as follows:

- the type-1 bits shall be ordered in three classes (sensitivity classes). Parity bits (CRC) shall be computed only on the third class (the most sensitive) and appended to it. In addition, 4 tail bits shall be globally appended. Bits in the three sensitivity classes, parity bits and tail bits are referred to as type-2 bits; this defines interface 2 in figure 6;

- the type-2 bits shall be encoded by convolutional codes, which provide the convolutionally-encoded bits. Convolutional coding shall be applied only to the bits corresponding to the two most sensitive classes, the less sensitive class being left unprotected. The convolutionally encoded bits along with the uncoded class 0 bits are referred to as type-3 bits; this defines interface 3 in figure 6;

- the type-3 bits shall be interleaved: this defines the interface 4 in figure 6;

- the type-4 bits shall be scrambled, into type-5 bits, which compose type-5 blocks: this defines interface 5 in figure 6. These bits are then mapped into multiplexed blocks.

  NOTE: Steps following interface 4 which are common to all traffic channels are described in ETS 300 392-2 [1], clause 8.

All these operations are made on a per type-1 block basis. The size of type-1, -2 and -3 blocks depends on the channel coding strategy. Two situations may occur in the case of the speech traffic channel.

In normal operative conditions two speech frames corresponding to one transmission time slot shall be encoded and interleaved together in order to improve the robustness of the speech channel coder.

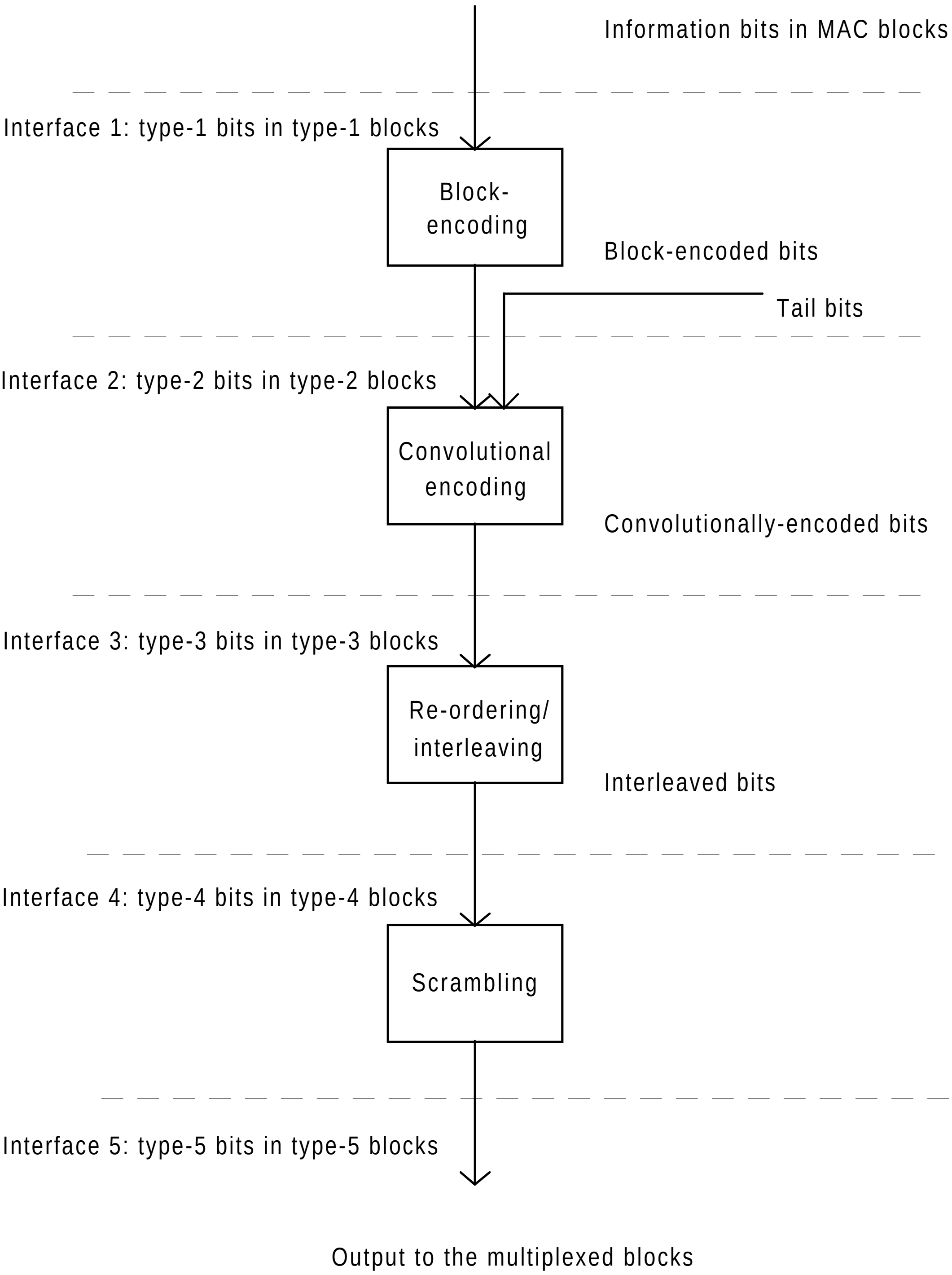

Figure 6: Interfaces in the error control structure

However, in some cases, capacity of the traffic channels (either speech or data circuits) may be stolen for signalling purposes. For the speech traffic channel such stealing shall only be performed on a speech frame basis. In principle, either just one or both the speech frames within a time slot may be stolen, even though for minimal degradation to speech quality, stealing only one speech frame is always preferable.

When the two speech frames are stolen, no speech parameters are present and therefore the speech traffic channel specifications no longer apply.

When only one speech frame is stolen, it is assumed that it is always the first one in the time slot. In that specific case the two half slots are encoded and interleaved separately.

The channel coding strategy for signalling data in a stolen half slot is described in ETS 300 392-2 [1], clause 8 (stealing channel - STCH).

The following subclauses specify the encoding, re-ordering and interleaving strategies to be applied in the two cases described above for the speech traffic channel.

5.3 Notations

For ease of understanding, the following notation for bits and blocks is used throughout this clause:

- x is the interface number, as defined in figure 6: x = 1, 2, 3, 4, 5;
- K_x is the number of bits that are carried by one type-x block;
- k is a bit number;
- $C_x(k)$ is the type-x bit number k in a type-x block, with $k = 1,2,\ldots,K_x$.

5.4 Definition of sensitivity classes and error control codes

5.4.1 Sensitivity classes

Based on a bit sensitivity study, the bits of the speech parameters corresponding to one speech frame (137 bits) have been assigned to a sensitivity class according to table 4. In table 4, class 0 corresponds to the least sensitive bits. Each sensitivity class shall be protected in a different way. In table 4, the column entitled "Bit number in speech frame" refers to the bit stream at the output of the source speech encoder. More details can be found in clause 4.

5.4.2 CRC codes

The CRC codes shall encode the type-1 bits of the third sensitivity class (the most sensitive one) leading from K_1 type-1 bits to K_2 type-2 bits ($K_2 = K_1 + p$, where p is defined according to the status of frame stealing).

The CRC codes are systematic codes computed by $F(X) = X^{n-K_1} I(X)$ mod $G(X)$ where:

- $I(X)$ is the codeword consisting of the third sensitivity class:
  $$I(X) = C_1(1) + C_1(2)X + \ldots + C_1(K_1)X^{(K_1-1)};$$
- $G(X)$ is the polynomial generator of the CRC code; and
- $n - K_1$ is the number of parity bits generated.

$F(X)$ is of degree $n - K_1 - 1$ with coefficients denoted by $f(0), f(1), \ldots, f(n-K_1-1)$: $F(X) = \sum_{i=0}^{n-K_1-1} f(i)X^i$.

The n type-2 bits consisting of the bits of the third sensitivity class followed by the CRC bits are given by:

$C_2(i) = C_1(i)$, for $i = 1,2,\ldots,K_1$; and

$C_2(i) = f(i - K_1 - 1)$, for $i = K_1 + 1, K_1 + 2, \ldots, n (= K_1 + n - K_1)$

Table 4: Assignment of the bits of the speech parameters to sensitivity classes

Speech parameter	Bit in parameter (LSB=b0)	Bit number in speech frame	Sensitivity class
Filter codebook index: LSP1 to LSP3	b7, b6, b5, b4	B1 - B4	2
"	b3, b2, b1, b0	B5 - B8	1
Filter codebook index: LSP4 to LSP6	b8, b7, b6, b5	B9 - B12	2
"	b4, b3, b2, b1, b0	B13 - B17	1
Filter codebook index: LSP7 to LSP10	b8, b7, b6, b5	B18 - B21	2
"	b4, b3, b2, b1, b0	B22 - B26	1
Pitch delay for sub-frame No 1 (sf1)	b7, b6, b5, b4, b3, b2	B27 - B32	2
"	b1, b0	B33 - B34	1
Pitch delay for sub-frame No 2 (sf2)	b4, b3, b2, b1	B57 - B60	1
"	b0	B61	0
Pitch delay for sub-frame No 3 (sf3)	b4, b3, b2, b1	B84 - B87	1
"	b0	B88	0
Pitch delay for sub-frame No 4 (sf4)	b4, b3, b2, b1	B111 - B114	1
"	b0	B115	0
Codebook index for sf1 (pulse 4)	b13, b12, b11	B35 - B37	0
Codebook index for sf1 (pulse 3)	b10, b9, b8	B38 - B40	0
Codebook index for sf1 (pulse 2)	b7, b6, b5	B41 - B43	0
Codebook index for sf1 (pulse 1)	b4, b3, b2	B44 - B46	1
"	b1, b0	B47 - B48	0
Codebook index for sf2 (pulse 4)	b13, b12, b11	B62 - B64	0
Codebook index for sf2 (pulse 3)	b10, b9, b8	B65 - B67	0
Codebook index for sf2 (pulse 2)	b7, b6, b5	B68 - B70	0
Codebook index for sf2 (pulse 1)	b4, b3, b2	B71 - B73	1
"	b1, b0	B74 - B75	0
Codebook index for sf3 (pulse 4)	b13, b12, b11	B89 - B91	0
Codebook index for sf3 (pulse 3)	b10, b9, b8	B92 - B94	0
Codebook index for sf3 (pulse 2)	b7, b6, b5	B95 - B97	0
Codebook index for sf3 (pulse 1)	b4, b3, b2	B98 - B100	1
"	b1, b0	B101 - B102	0
Codebook index for sf4 (pulse 4)	b13, b12, b11	B116 - B118	0
Codebook index for sf4 (pulse 3)	b10, b9, b8	B119 - B121	0
Codebook index for sf4 (pulse 2)	b7, b6, b5	B122 - B124	0
Codebook index for sf4 (pulse 1)	b4, b3, b2	B125 - B127	1
"	b1, b0	B128 - B129	0
Pulse global sign for sf1	b0	B49	1
Pulse global sign for sf2	b0	B76	1
Pulse global sign for sf3	b0	B103	1
Pulse global sign for sf4	b0	B130	1
Pulse shift for sf1	b0	B50	1
Pulse shift for sf2	b0	B77	1
Pulse shift for sf3	b0	B104	1
Pulse shift for sf4	b0	B131	1
Codebook index for sf1 gains	b5, b4, b3	B51 - B53	2
"	b2, b1	B54 - B55	1
"	b0	B56	0
Codebook index for sf2 gains	b5, b4, b3	B78 - B80	2
"	b2, b1	B81 - B82	1
"	b0	B83	0
Codebook index for sf3 gains	b5, b4, b3	B105 - B107	2
"	b2, b1	B108 - B109	1
"	b0	B110	0
Codebook index for sf4 gains	b5, b4, b3	B132 - B134	2
"	b2, b1	B135 - B136	1
"	b0	B137	0
NOTE: For one speech frame, class 2 = 30 bits, class 1 = 56 bits, class 0 = 51 bits.			

5.4.3 16-state RCPC codes

The RCPC codes encode K_2 type-2 bits $C_2(1), C_2(2), \ldots, C_2(K_2)$ into K_3 type-3 bits $C_3(1), C_3(2), \ldots, C_3(K_3)$. This encoding shall be performed in two steps:

- encoding by a 16-state mother code of rate 1/3;
- puncturing of the mother code to obtain a 16-state RCPC code of rate 8/(8+1).

A general description of these two steps is given in the following subclauses.

5.4.3.1 Encoding by the 16-state mother code of rate 1/3

The input to the mother code of any type-2 bit $C_2(k), k = 1,2,\ldots,K_2$ implies the output, by the mother code, of 3 bits, denoted by $V(3(k-1)+i), i = 1,2,3$, which shall be calculated as follows.

Any of the 3 generator polynomials of the mother code, $G_i(D), i = 1,2,3$, can be written as:

$$G_i(D) = \sum_{j=0}^{4} g_{i,j} D^j \qquad i = 1,2,3$$

where $g_{i,j} = 0$ or $1, j = 0,1,\ldots,4$.

This means that the encoded bits are defined by:

$$V(3(k-1)+i) = \sum_{j=0}^{4} C_2(k-j) g_{i,j} \qquad i = 1,2,3, k = 1,2,\ldots,K_2$$

where the sum is meant modulo 2, and where $C_2(k-j) = 0$ for $k \le j$.

The generator polynomials of the mother code shall be:

$$\begin{aligned} G_1(D) &= 1 + D + D^2 + D^3 + D^4 \\ G_2(D) &= 1 + D + D^3 + D^4 \\ G_3(D) &= 1 + D^2 + D^4 \end{aligned}$$

5.4.3.2 Puncturing of the mother code

The puncturing of the mother code into a 16-state RCPC code of rate (K_2/K_3) shall be achieved by selecting K_3 type-3 bits out of the $(3K_2)$ bits encoded by the mother code. This selection shall be as follows.

Denoting by $P(1), P(2), \ldots, P(t)$ the t puncturing coefficients (each one being equal to 1, 2, ..., 23, or 24), the type-3 bits are given by:

$$C_3(j) = V(k) \qquad j = 1,2,\ldots,K_3 \qquad \text{with}$$

$$k = Period * ((j-1)\,\mathrm{div}\,t) + P(j - t((j-1)\,\mathrm{div}\,t))$$

where $Period$ and t are defined according to the status of frame stealing.

5.5 Error control scheme for normal speech traffic channel

In this case two speech frames corresponding to one transmission time slot shall be encoded together. Therefore the size of type-1 blocks is 274 bits.

5.5.1 CRC code

The CRC code shall be applied to the 2*30 = 60 type-1 bits of the third sensitivity class.

The polynomial generator shall be the following:

$$G(X) = 1 + X^3 + X^7$$

It generates $(n - K_1) = 7$ parity bits, numbered b1 to b7 in table 5. An eighth parity bit, b8 in table 5, $(p = 8)$ shall be computed as an overall parity bit, i.e. as the sum (modulo 2) of the bits in the third sensitivity class with the 7 parity bits.

The size of type-2 blocks is therefore:

class 0	class 1	class 2	p	tail	
2*51	+ 2*56	+ 2*30	+ 8	+ 4	= 286 bits

The practical order of occurrence of these 286 type-2 bits is given in table 5, in relation to the bit number in the speech frame, as defined in table 4. In table 5, capital letters A and B refer to the two speech frames, A for the first speech frame of the transmission time slot, B for the second.

5.5.2 RCPC codes

Convolutional coding shall be applied to the bits corresponding to the two most sensitive classes, the least sensitive class being left unprotected.

A convolutional code of rate 2/3 shall be applied to the bits of class 1, while a convolutional code of rate 8/18 shall be applied to the bits of class 2. At that level, the p parity bits and the tail bits have to be included in class 2.

The size of type-3 blocks is therefore:

class 0	class 1	class 2	
2*51	+ (2*56)*3/2	+ (2*30 + 8 + 4)*18/8	= 432 bits

The change of the convolutional code rate is continuous, i.e. as soon as the first class 2 bit enters the encoder, the rate of the convolutional code is changed. Thus, encoding of the first bits of class 2 is affected by the values of the last bits of class 1.

5.5.2.1 Puncturing scheme of the RCPC code of rate 8/12 (equal to 2/3)

The $t = 3$ puncturing coefficients shall be: $P(1) = 1, P(2) = 2, P(3) = 4$ and $Period = 6$.

5.5.2.2 Puncturing scheme of the RCPC code of rate 8/18

The $t = 9$ puncturing coefficients shall be:

$$P(1) = 1, P(2) = 2, P(3) = 3, P(4) = 4, P(5) = 5,$$
$$P(6) = 7, P(7) = 8, P(8) = 10, P(9) = 11$$

and $Period = 12$.

5.5.3 Matrix Interleaving

The 432 type-3 bits shall be interleaved to produce 432 type-4 re-ordered bits, according to the following rule:

$$C_4(i*24+j)=C_3(j*18+i)$$

which corresponds to transposing the (24,18) matrix (24 lines, 18 columns) of the input type-3 bits.

Table 5: Meaning of each type-2 bits

Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame
1	B35,A	2	B35,B	3	B36,A	4	B36,B	5	B37,A
6	B37,B	7	B38,A	8	B38,B	9	B39,A	10	B39,B
11	B40,A	12	B40,B	13	B41,A	14	B41,B	15	B42,A
16	B42,B	17	B43,A	18	B43,B	19	B47,A	20	B47,B
21	B48,A	22	B48,B	23	B56,A	24	B56,B	25	B61,A
26	B61,B	27	B62,A	28	B62,B	29	B63,A	30	B63,B
31	B64,A	32	B64,B	33	B65,A	34	B65,B	35	B66,A
36	B66,B	37	B67,A	38	B67,B	39	B68,A	40	B68,B
41	B69,A	42	B69,B	43	B70,A	44	B70,B	45	B74,A
46	B74,B	47	B75,A	48	B75,B	49	B83,A	50	B83,B
51	B88,A	52	B88,B	53	B89,A	54	B89,B	55	B90,A
56	B90,B	57	B91,A	58	B91,B	59	B92,A	60	B92,B
61	B93,A	62	B93,B	63	B94,A	64	B94,B	65	B95,A
66	B95,B	67	B96,A	68	B96,B	69	B97,A	70	B97,B
71	B101,A	72	B101,B	73	B102,A	74	B102,B	75	B110,A
76	B110,B	77	B115,A	78	B115,B	79	B116,A	80	B116,B
81	B117,A	82	B117,B	83	B118,A	84	B118,B	85	B119,A
86	B119,B	87	B120,A	88	B120,B	89	B121,A	90	B121,B
91	B122,A	92	B122,B	93	B123,A	94	B123,B	95	B124,A
96	B124,B	97	B128,A	98	B128,B	99	B129,A	100	B129,B
101	B137,A	102	B137,B	103	B58,A	104	B58,B	105	B85,A
106	B85,B	107	B112,A	108	B112,B	109	B54,A	110	B54,B
111	B81,A	112	B81,B	113	B108,A	114	B108,B	115	B135,A
116	B135,B	117	B50,A	118	B50,B	119	B77,A	120	B77,B
121	B104,A	122	B104,B	123	B131,A	124	B131,B	125	B45,A
126	B45,B	127	B72,A	128	B72,B	129	B99,A	130	B99,B
131	B126,A	132	B126,B	133	B55,A	134	B55,B	135	B82,A
136	B82,B	137	B109,A	138	B109,B	139	B136,A	140	B136,B
141	B5,A	142	B5,B	143	B13,A	144	B13,B	145	B34,A
146	B34,B	147	B8,A	148	B8,B	149	B16,A	150	B16,B
151	B17,A	152	B17,B	153	B22,A	154	B22,B	155	B23,A

(continued)

Table 5 (concluded): Meaning of each type-2 bits

Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame
156	B23,B	157	B24,A	158	B24,B	159	B25,A	160	B25,B
161	B26,A	162	B26,B	163	B6,A	164	B6,B	165	B14,A
166	B14,B	167	B7,A	168	B7,B	169	B15,A	170	B15,B
171	B60,A	172	B60,B	173	B87,A	174	B87,B	175	B114,A
176	B114,B	177	B46,A	178	B46,B	179	B73,A	180	B73,B
181	B100,A	182	B100,B	183	B127,A	184	B127,B	185	B44,A
186	B44,B	187	B71,A	188	B71,B	189	B98,A	190	B98,B
191	B125,A	192	B125,B	193	B33,A	194	B33,B	195	B49,A
196	B49,B	197	B76,A	198	B76,B	199	B103,A	200	B103,B
201	B130,A	202	B130,B	203	B59,A	204	B59,B	205	B86,A
206	B86,B	207	B113,A	208	B113,B	209	B57,A	210	B57,B
211	B84,A	212	B84,B	213	B111,A	214	B111,B	215	B18,A
216	B18,B	217	B19,A	218	B19,B	219	B20,A	220	B20,B
221	B21,A	222	B21,B	223	B31,A	224	B31,B	225	B32,A
226	B32,B	227	B53,A	228	B53,B	229	B80,A	230	B80,B
231	B107,A	232	B107,B	233	B134,A	234	B134,B	235	B1,A
236	B1,B	237	B2,A	238	B2,B	239	B3,A	240	B3,B
241	B4,A	242	B4,B	243	B9,A	244	B9,B	245	B10,A
246	B10,B	247	B11,A	248	B11,B	249	B12,A	250	B12,B
251	B27,A	252	B27,B	253	B28,A	254	B28,B	255	B29,A
256	B29,B	257	B30,A	258	B30,B	259	B52,A	260	B52,B
261	B79,A	262	B79,B	263	B106,A	264	B106,B	265	B133,A
266	B133,B	267	B51,A	268	B51,B	269	B78,A	270	B78,B
271	B105,A	272	B105,B	273	B132,A	274	B132,B	275	parity b1
276	parity b2	277	parity b3	278	parity b4	279	parity b5	280	parity b6
281	parity b7	282	parity b8	283 to 286: 4 tail bits equal to 0					

5.6 Error control scheme for speech traffic channel with frame stealing activated

When frame stealing is activated, only the second half slot contains a speech frame. The first half slot contains signalling data. The speech frame and signalling data shall be encoded and interleaved separately. The size of type-1 blocks is then 137 bits.

5.6.1 CRC code

The CRC code shall be applied to the 30 type-1 bits of the third sensitivity class.

The polynomial generator shall be the following:

$$G(X) = 1 + X + X^4$$

It generates $(n - K_1) = p = 4$ parity bits.

The size of type-2 blocks is therefore:

class 0	class 1	class 2	p	tail	
51	+ 56	+ 30	+ 4	+ 4	= 145 bits

The practical order of occurrence of these 145 type-2 bits is given in table 6, in relation to the bit number in the speech frame. In table 6, conventions are the same as in table 5. However letters A and B are no longer used, since only one speech frame is now concerned.

Table 6: Meaning of each type-2 bits (in case of frame stealing)

Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame	Bit no.	Bit number in speech frame
1	B35	2	B36	3	B37	4	B38	5	B39
6	B40	7	B41	8	B42	9	B43	10	B47
11	B48	12	B56	13	B61	14	B62	15	B63
16	B64	17	B65	18	B66	19	B67	20	B68
21	B69	22	B70	23	B74	24	B75	25	B83
26	B88	27	B89	28	B90	29	B91	30	B92
31	B93	32	B94	33	B95	34	B96	35	B97
36	B101	37	B102	38	B110	39	B115	40	B116
41	B117	42	B118	43	B119	44	B120	45	B121
46	B122	47	B123	48	B124	49	B128	50	B129
51	B137	52	B58	53	B85	54	B112	55	B54
56	B81	57	B108	58	B135	59	B50	60	B77
61	B104	62	B131	63	B45	64	B72	65	B99
66	126	67	B55	68	B82	69	B109	70	B136
71	B5	72	B13	73	B34	74	B8	75	B16
76	B17	77	B22	78	B23	79	B24	80	B25
81	B26	82	B6	83	B14	84	B7	85	B15
86	B60	87	B87	88	B114	89	B46	90	B73
91	B100	92	B127	93	B44	94	B71	95	B98
96	B125	97	B33	98	B49	99	B76	100	B103
101	B130	102	B59	103	B86	104	B113	105	B57
106	B84	107	B111	108	B18	109	B19	110	B20
111	B21	112	B31	113	B32	114	B53	115	B80
116	B107	117	B134	118	B1	119	B2	120	B3
121	B4	122	B9	123	B10	124	B11	125	B12
126	B27	127	B28	128	B29	129	B30	130	B52
131	B79	132	B106	133	B133	134	B51	135	B78
136	B105	137	B132	138	parity b1	139	parity b2	140	parity b3
141	parity b4	142 to 145: 4 tail bits equal to 0							

5.6.2 RCPC codes

Convolutional coding shall be applied to the bits corresponding to the two most sensitive classes, the less sensitive class being left unprotected.

A convolutional code of rate 2/3 shall be applied to the bits of class 1, while a convolutional code of rate 8/17 shall be applied to the bits of class 2. At that level, the p parity bits and the tail bits have to be included in class 2.

The size of type-3 blocks is therefore:

class 0	class 1	class 2	
51	+ 56*3/2	+ (30 + 4 + 4)*17/8	= 216 bits

The puncturing scheme of the RCPC code of rate 2/3 has been already defined for the normal speech traffic channel.

The change of the convolutional code rate is continuous, i.e. as soon as the first class 2 bit enters the encoder, the rate of the convolutional code is changed. Thus, encoding of the first bits of class 2 is affected by the values of the last bits of class 1.

5.6.2.1 Puncturing scheme of the RCPC code of rate 8/17

The $t = 17$ puncturing coefficients shall be:

$$P(1)=1, P(2)=2, P(3)=3, P(4)=4, P(5)=5, P(6)=7,$$
$$P(7)=8, P(8)=10, P(9)=11, P(10)=13, P(11)=14, P(12)=16,$$
$$P(13)=17, P(14)=19, P(15)=20, P(16)=22, P(17)=23$$

and $Period = 24$.

5.6.3 Interleaving

For the sake of simplicity, the 216 type-3 bits shall be interleaved to produce 216 type-4 re-ordered bits following the scheme already adopted for the stealing channel for signalling data (STCH) entirely defined in ETS 300 392-2 [1], clause 8.

6 Channel decoding for speech

6.1 General

This clause shall apply to the speech traffic channel only.

This clause is the counterpart of clause 5 describing the channel coding for speech. Conventions and notations are the same.

An informative description of a possible implementation of channel decoding for speech is given in annex A. C code is provided as an example and can be found in computer files attached to this ETS.

6.2 Error control structure

The processing on the receive part shall be as follows:

- the type-1 bits correspond to the input of the channel decoder, i.e. the output of the demodulator. The type-1 bits shall be interleaved into type-2 bits;

- the type-2 bits shall be decoded by a convolutional decoder, which provides the convolutionally decoded bits type-3 bits;

- the type-3 bits shall be structured in three classes (sensitivity classes) plus additional bits. Parity bits (CRC) shall be computed only on the third class and compared to the additional bits. In case of discrepancy the BFI flag shall be set;

- the type-4 bits shall consist of the sensitivity classes together with the BFI. These bits are the ones delivered by the channel decoder.

All these operations are made on a per type-1 block basis. The size of type -1, -2, -3 and -4 blocks depend on the channel decoding strategy. Two situations described in clause 5 may occur in the case of the speech traffic channel. These two cases lead to two channel decoding strategies defined according to the status of frame stealing (activated/not activated).

In case of normal speech traffic channel, one transmission time slot corresponding to two speech frames shall be decoded. Therefore the size of type-1 blocks is 432 bits.

When frame stealing is activated, only the second half slot corresponding to one speech frame shall be decoded separately. The size of type-1 blocks is then 216 bits.

7 Codec performance

This clause shall apply to the TETRA speech traffic channel only.

The TETRA speech and channel encoding and decoding performance, as measured during the codec studies performed by ETSI sub-technical committee RES06 WG5, is described in annexes D and E.

Any practical implementation of the speech and channel coding and decoding processes will be required to satisfy normative conformance tests.

8 Bit exact description of the TETRA codec

This clause should be read in conjunction with annex F.

The various components of the TETRA codec are described in the form of an ANSI C code, fixed point, bit exact.

The C code was originally developed for a specific Digital Signal Processor (DSP). Later, the code was partly rewritten to ensure DSP independence.

The C code corresponding to the source coding component of the TETRA codec is given in computer files attached to this ETS. They are an integral part of this ETS.

For a better readability of the C code, these files are organized as follows:

- main program for source coder: scoder.c, scod_tet.c;

- main program for source decoder: sdecoder.c, sdec_tet.c;

- library of source coder/decoder subroutines: sub_sc_d.c;

- library of signal processing related subroutines: sub_dsp.c;

- libraries of basic and mathematics functions: fbas_tet.c, fexp_tet.c, fmat.c;

- tables and constants for source coding: source.h, clsp_334.tab,enr_qua.tab, grid_tab, inv_sqrt.tab, lag_wind.tab, log2.tab, pow2.tab, window.tab.

Two types of variables are used along the fixed point implementation. These two types are signed integers in 2's complement representation, defined by:

- var1, var2, ..., varn as 16 bit variables;

- L_var1, L_var2, ..., L_varn as 32 bit variables.

All the computations shall be done using a predefined set of basic operators, the descriptions of which are in the computer file: tetra_op.c.

The speech source encoder shall take its input as a 16 bit uniform PCM signal (16 bit 2's complement audio samples). In the associated C code simulation, binary files of 16 bit-samples are used to support these audio samples.

Results provided by the speech source encoder shall be made of frames of 137 bits for each speech frame of 30 ms. To allow the exact matching of the structures of the output file of the speech source encoder and the input file of the speech source decoder, a provision of one bit is added to support the BFI included in the input of the source decoder. Therefore, the complete frame is 138 bits long (with the first bit forced to 0 in the case of the encoder). In the binary files used for simulation, these 138 bits are encoded on 16 bit-samples. Each 16 bit-sample represents one encoded bit, with only the least significant bit used.

The encoded speech at the output of the speech source encoder is delivered to the channel coding function.

The C code corresponding to the channel coding component is given in computer files attached to this ETS, together with an example of implementation of the speech channel decoding component. For a better readability of the C code, files corresponding to the channel coding component are organized as follows:

- main program for speech channel coding: ccoder.c, ccod_tet.c;
- library of speech channel coding subroutines: sub_cc.c;
- tables and constants for speech channel: channel.h, const.tab, arrays.tab.

The interface between the channel coding component and the channel decoding component, as used in the C code description (representing binary information by +/-127) is defined primarily to facilitate bit exactness testing and to be compatible with the error insertion device simulation. In addition, the structure of the output file of the channel coding component is compatible with the error patterns formerly developed for the GSM codec. Actually, the 432 type-4 bits which form the output of the channel coding component are mapped in a file structure of 690 words (each type-4 bit being encoded through a 16 bit-word) as given in table 7.

Table 7: File structure of channel coded speech data

Word number	Contents
1	0x6B21
2 - 115	114 type-4 bits
116	0x6B22
117 - 230	114 type-4 bits
231	0x6B23
232 - 345	114 type-4 bits
346	0x6B24
347 - 436	90 type-4 bits
437 - 460	24 bits set to 0
461	0x6B25
462 - 575	114 bits set to 0
576	0x6B26
577 - 690	114 bits set to 0
NOTE: 0x6B21 to 0x6B26 are synchronization words expressed in hexadecimal notation.	

The physical implementation in a product may be different, and will depend upon the higher levels of the communication system.

In the receive direction, the inverse operations take place.

A "makefile" which can be used to compile the C code is included with the previously described computer files. This example has been written for an ANSI C compiler running within a UNIX environment. A similar "makefile" could be derived for other environments. The code actually included in this ETS been checked under the following conditions:

- UNIX (Sun OS 4.1.2), SUN SPARC station, ANSI C compiler (acc - version 03/91);
- AIX, CETIA station, xlc (version 1.3);
- MS DOS (6.20), PC HP-80486, Turbo C++ compiler (version 01/90);
- VMS (5.5-2), VAX station 4000.60, VAX C (version 11/90).

Clause B.1 contains an index in which are listed all the routines included in the C code. An index of computer files is also provided in clause B.2

Annex A (informative): Implementation of speech channel decoding

This annex describes a possible implementation of channel decoding for speech, and the presented solutions should be the preferred choice for implementation.

Clause A.1 provides an algorithmic description of this implementation, whilst an example, supplied as fixed point ANSI C code, is described in clause A.2.

A.1 Algorithmic description of speech channel decoding

A.1.1 Definition of error control codes

A.1.1.1 16-state RCPC codes

As discussed in clause 5 the RCPC codes encode K_3 type-3 bits $C_3(1), C_3(2), \ldots, C_3(K_3)$ into K_2 type-2 bits $C_2(1), C_2(2), \ldots, C_2(K_2)$.

Decoding of type-2 bits to type-3 bits can be performed in two steps:

- de-puncturing of 16-state RCPC code of rate 8/(8+1) to obtain the mother code of rate 1/3; and
- Viterbi decoding by a 16-state code of rate 1/3 (mother code).

A general description of these two steps is given in the following subclauses.

A.1.1.1.1 Obtaining the mother code from punctured code

The de-puncturing of the 16-state RCPC code of rate (K_3/K_2) can be achieved by selecting K_2 type-2 bits and inserting zeroes to get a $(3K_3)$ block size V. This insertion can be carried out as follows:

- denoting by $P(1), P(2), \ldots, P(t)$ the t puncturing coefficients (each one being equal to 1, 2, ..., 23, or 24), the de-punctured bits $V(i), 1 \le i \le 3K_3$ are given by:

$$
\begin{array}{lll l}
V(i) & = & 0 & i = 1,2,\ldots,3K_3 \\
V(k) & = & C_2(j) & j = 1,2,\ldots,K_2
\end{array}
$$

with

$$k = Period * ((j-1)\operatorname{div} t) + P(j - t((j-1)\operatorname{div} t))$$

where $Period$ and t are defined according to the status of frame stealing.

A.1.1.1.2 Viterbi decoding of the 16-state mother code of the rate 1/3

A classical Viterbi decoding of the 16-state mother code of rate 1/3 can be applied to the de-punctured $(3K_3)$ size block leading to K_3 type-3 bits.

The generator polynomials of the mother code are:

$$
\begin{aligned}
G_1(D) &= 1 + D + D^2 + D^3 + D^4 \\
G_2(D) &= 1 + D + D^3 + D^4 \\
G_3(D) &= 1 + D^2 + D^4
\end{aligned}
$$

A.1.1.2 CRC codes

The CRC codes encode the type-3 bits of the third sensitivity class (the most sensitive one) in order to deliver p parity bits (where p is defined according to the status of frame stealing).

By comparison of these p parity bits with the last p bits of the (Viterbi decoded) type-3 bits the BFI is set. BFI becomes 1 when at least one of the p parity bits is different from the corresponding Viterbi-decoded ones, and remains 0 else.

The CRC codes are systematic codes computed as the remainder in the Euclidean division of $X^{n-K_3}I(X)$ by $G(X)$ where:

- $I(X)$ is the codeword consisting of the third sensitivity class:

$$I(X) = C_1(1) + C_1(2)X + ... + C_1(K_3)X^{(K_3-1)};$$

- $G(X)$ is the polynomial generator of the CRC code;

- $n - K_3$ is the number of parity bits generated.

A.1.1.3 Type-4 bits

From K_3 type-3 bits, K_4 type-4 bits are built from the three sensitivity classes (that is excluding the type-3 additional bits) together with the Bad Frame Indicator as previously processed.

A.1.2 Error control scheme for normal speech traffic channel

In this case one transmission time slot corresponding to two speech frames has to be decoded. Therefore the size of type-1 blocks is 432 bits.

A.1.2.1 Matrix Interleaving

The 432 type-1 bits shall be interleaved to produce 432 type-2 re-ordered bits, according to the following rule:

$$C_2(j*18+i) = C_1(i*24+j)$$

which corresponds to transposing the (18,24) matrix (18 lines, 24 columns) of the input type-1 bits.

A.1.2.2 RCPC codes

Decoding of type-2 bits to type-3 bits are done with RCPC codes. Type-2 blocks of 432 bits have to be split in three sub-blocks corresponding to the three sensitivity classes used at the coding level.

No processing has to be performed on the first 102 bits. These bits correspond to the least sensitive class bits of two speech frames.

A convolutional code of rate 2/3 is applied to the following 168 bits (corresponding to the sensitivity class 1), while a convolutional code of rate 8/18 is applied to the remaining 162 bits (corresponding to class 2).

The size of type-3 blocks is therefore:

class 0	class 1	class 2		
102	+ 168*2/3	+ 162*8/18	=	286 bits

A.1.2.2.1 Puncturing scheme of the RCPC code of rate 8/12 (equal to 2/3)

The $t = 3$ puncturing coefficients are: $P(1) = 1, P(2) = 2, P(3) = 4$ and $Period = 6$.

A.1.2.2.2 Puncturing scheme of the RCPC code of rate 8/18

The $t = 9$ puncturing coefficients are:

$$P(1) = 1, P(2) = 2, P(3) = 3, P(4) = 4, P(5) = 5,$$
$$P(6) = 7, P(7) = 8, P(8) = 10, P(9) = 11$$

and $Period = 12$.

A.1.2.3 CRC code

The CRC code is applied to the 72 type-3 bits corresponding to the third sensitivity class.

The polynomial generator is the following:

$$G(X) = 1 + X^3 + X^7$$

It generates $(n - K_3) = 7$ parity bits. A eight parity bit $(p = 8)$ is computed as an overall parity bit, i.e. as the sum (modulo 2) of the bits in the third sensitivity class with the 7 parity bits.

A.1.2.4 Speech parameters

The type-4 bits, then the speech parameters for the two speech frames have to be reconstructed from the type-3 bits by using the tables of correspondence given in clause 5.

A.1.3 Error control scheme for speech traffic channel with frame stealing activated

When frame stealing is activated, only the second half slot corresponding to one speech frame has to be decoded separately. The size of type-1 blocks is then 216 bits.

A.1.3.1 Interleaving

The 216 type-1 bits shall be interleaved to produce 216 type-2 re-ordered bits following the scheme adopted for the stealing channel for signalling data (STCH).

A.1.3.2 RCPC codes

Decoding of type-2 bits to type-3 bits are done with RCPC codes. Type-2 blocks of 216 bits have to be split in three sub-blocks corresponding to the three sensitivity classes used at the coding level.

No processing has to be performed on the first 51 bits. These bits correspond to the least sensitive class bits of the speech frame.

A convolutional code of rate 2/3 is applied to the following 84 bits (corresponding to the sensitivity class 1), while a convolutional code of rate 8/17 is applied to the remaining 81 bits (corresponding to class 2).

The size of type-3 blocks is therefore:

class 0	class 1	class 2		
51	+ 84*2/3	+ 81*8/17	=	145 bits

The puncturing scheme of the RCPC code of rate 2/3 has been already defined for the normal speech traffic channel.

A.1.3.2.1 Puncturing scheme of the RCPC code of rate 8/17

The $t = 17$ puncturing coefficients are:

$$P(1)=1, P(2)=2, P(3)=3, P(4)=4, P(5)=5, P(6)=7,$$
$$P(7)=8, P(8)=10, P(9)=11, P(10)=13, P(11)=14, P(12)=16,$$
$$P(13)=17, P(14)=19, P(15)=20, P(16)=22, P(17)=23$$

and $Period = 24$.

A.1.3.3 CRC code

The CRC code is applied to the 38 type-3 bits corresponding to the third sensitivity class.

The polynomial generator is the following:

$$G(X) = 1 + X + X^4$$

It generates $(n - K_3) = p = 4$ parity bits.

A.1.3.4 Speech parameters

The type-4 bits, then the speech parameters for the speech frame have to be reconstructed from the type-3 bits by using the tables of correspondence given in clause 5.

A.2 C Code for speech channel decoding

C code corresponding to an example of implementation of the speech channel decoding component is given in computer files attached to this ETS.

For a better readability of the C code, these files are organized as follows:

- main program for speech channel decoding: cdecoder.c, cdec_tet.c;
- library of speech channel decoding subroutines: sub_cd.c.

The interface between the channel coding component and the channel decoding component, as used in the C code description (representing binary information by +/-127) is defined primarily to facilitate bit exactness testing and to be compatible with the error insertion device simulation. The physical implementation in a product may be different.

Annex B (informative): Indexes

B.1 Index of C code routines

In this index are listed all the routines included in the C code.

Routines are classified according to their names. The second column relates to the source file in which they are located.

All source files have the extension ".c".

A

abs_s	tetra_op
add	tetra_op
add_sh	fbas_tet
add_sh16	fbas_tet
Autocorr	sub_dsp
Az_Lsp	sub_dsp

B

Back_Fil	sub_dsp
Bfi	sub_cd
bin2int	fbas_tet
Bits2prm_Tetra	sub_sc_d
Build_Crc	sub_cc
Build_Sensitivity_Classes	sub_cc

C

Cal_Rr2	sub_sc_d
ccoder	ccoder
cdecoder	cdecoder
Channel_Decoding	cdec_tet
Channel_Encoding	ccod_tet
Chebps	sub_dsp
Clsp_334	sub_sc_d
Coder_Tetra	scod_tet
Combination	sub_cc
Combination	sub_cd
Convolve	sub_dsp

D

D_D4i60	sub_sc_d
D_Lsp334	sub_sc_d
D4i60_16	sub_sc_d
Dec_Ener	sub_sc_d
Decod_Tetra	sdec_tet
Desinterleaving_Signalling	sub_cd
Desinterleaving_Speech	sub_cd
div_32	fexp_tet
div_s	tetra_op

E

Ener_Qua	sub_sc_d
extract_h	tetra_op
extract_l	tetra_op

F

Fac_Pond	sub_dsp

G

G_Code	sub_sc_d
G_Pitch	sub_sc_d
Get_Lsp_Pol	sub_dsp

I

Init_Coder_Tetra	scod_tet
Init_Decod_Tetra	sdec_tet
Init_Rcpc_Coding	sub_cc
Init_Rcpc_Decoding	sub_cd
Int_Lpc4	sub_dsp
int2bin	fbas_tet
Inter32_1_3	sub_sc_d
Inter32_M1_3	sub_sc_d
Inter8_1_3	sub_sc_d
Inter8_M1_3	sub_sc_d
Interleaving_Signalling	sub_cc
Interleaving_Speech	sub_cc
inv_sqrt	fmat_tet

L

L_abs	tetra_op
L_add	tetra_op
L_comp	fexp_tet
L_deposit_h	tetra_op
L_deposit_l	tetra_op
L_extract	fexp_tet
L_mac	tetra_op
L_mac0	tetra_op
L_msu	tetra_op
L_msu0	tetra_op
L_mult	tetra_op
L_mult0	tetra_op
L_negate	tetra_op
L_shl	tetra_op
L_shr	tetra_op
L_shr_r	tetra_op
L_sub	tetra_op
Lag_Max	sub_sc_d
Lag_Window	sub_dsp
Levin_32	sub_dsp
Load_sh	fbas_tet
Load_sh16	fbas_tet
Log2	fmat_tet
Lpc_Gain	sub_dsp
Lsp_Az	sub_dsp

M

mpy_32	fexp_tet
mpy_mix	fexp_tet
mult	tetra_op
mult_r	tetra_op

N

negate	tetra_op
Norm_Corr	sub_sc_d
norm_l	tetra_op
norm_s	tetra_op
norm_v	fbas_tet

P

Pitch_Fr	sub_sc_d
Pitch_Ol_Dec	sub_sc_d
Pond_Ai	sub_dsp
Post_Process	sub_sc_d
pow2	fmat_tet
Pre_Process	sub_sc_d
Pred_Lt	sub_sc_d
Prm2bits_Tetra	sub_sc_d

R

Rcpc_Coding	sub_cc
Rcpc_Decoding	sub_cd
Read_Tetra_File	sub_cd
Residu	sub_dsp
round	tetra_op

S

sature	tetra_op
scoder	scoder
sdecoder	sdecoder
shl	tetra_op
shr	tetra_op
store_hi	fbas_tet
sub	tetra_op
sub_sh	fbas_tet
sub_sh16	fbas_tet
Syn_Filt	sub_dsp

T

Transform_Class_0	sub_cc

U

Unbuild_Sensitivity_Classes	sub_cd
Untransform_Class_0	sub_cd

W

Write_Tetra_File	sub_cc

B.2 Index of files

In this index are listed all the computer files containing C code for the TETRA speech codec.

Files are classified according to their names.

File extensions are given. By convention, extension "tab" is used for files containing only numerical data and extension "h" for files containing definitions and prototypes.

A

arrays.tab

C

ccoder.c ccod_tet.c cdecoder.c cdec_tet.c channel.h clsp_334.tabconst.tab

E

ener_qua.tab

F

fbas_tet.c fexp_tet.c fmat_tet.c

G

grid.tab

I

inv_sqrt.tab

L

lag_wind.tab log2.tab

M

makefile

P

pow2.tab

S

scoder.c scod_tet.c sdecoder.c sdec_tet.c source.h sub_cc.c sub_cd.c sub_dsp.c sub_sc_d.c

T

tetra_op.c

W

window.tab

Annex C (informative): Bibliography

{1} Laflamme C., Adoul J-P., Salami R., Morissette S. and Mabilleau P.: "16 kbps wide-band speech coding technique based on algebraic CELP", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toronto, Canada, May 1991, pp. 177-180.

{2} Tohkura Y. and Itakura F.: "Spectral smoothing technique in PARCOR speech analysis-synthesis", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-26, No 6, pp. 587-596, December 1978.

{3} Salami R. et al.: "Speech coding", Chapter 3 in *Mobile Radio Communications*, edited by R. Steele, Pentech Press, London, 1992.

{4} Itakura F.: "Line spectral representation of linear predictive coefficients of speech signals", J. Acoust. Soc. Amer., Vol. 57, Supplement No 1, S35, 1975.

{5} Soong F.K. and Juang B.H.: "Line spectrum pair (LSP) and speech data compression", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, San Diego, Cal., March 1984, pp. 1.10.1-1.10.4.

{6} Kabal P. and Ramachandran R. P.: "The computation of line spectral frequencies using Chebyshev polynomials", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-34, NO 6, pp. 1419-1426, December 1986.

Annex D (informative): Codec performance

D.1 General

In this annex the actual quality, performance and complexity aspects of the codec are described. As speech quality is still often defined using subjective testing methodology, any attempts to define it in normative form was considered to be inappropriate for the TETRA codec. Hence, in the following clauses and subclauses various aspects of the TETRA speech codec, as gathered from results of the codec studies performed by the ETSI sub-technical committee RES06 WG5, are presented in summarized form. The aim is to provide both manufacturers and users of the TETRA system with indicative performance estimates for the TETRA voice system.

D.2 Quality

D.2.1 Subjective speech quality

D.2.1.1 Description of characterization tests

The TETRA codec has been characterized under a wide range of operating conditions. The quality reference adopted is the MNRU (Multiplicative Noise Reference Unit). Various MNRU conditions were included to allow comparison of results between experiments. The results of the speech quality listening tests are given in terms of Q values in dB relative to the MNRU reference condition using speech processed with an Intermediate Reference System (IRS) filter (see CCITT Recommendation P.48 [2]).

The summarized conditions and results for the characterization listening test are given in annex E.

Although every effort has been made to ensure that the results of the characterization tests are reliable, any performance figures given here should be considered indicative only and are intended to aid the design and implementation of TETRA systems.

D.2.1.2 Absolute speech quality

For clean speech at a nominal input level of -22 dB the average Q value obtained for the TETRA codec is 13,0 dB for the linear input condition and 16,5 dB for the IRS input condition. For comparison purposes the corresponding values obtained for the Global System for Mobile communications (GSM) full-rate codec are 17,4 dB and 18,9 dB respectively.

Although the MOS scale is not an absolute reference, it is commonly used in comparing speech codecs. For clean speech at a nominal input level of -22 dB the average MOS value obtained for the TETRA codec is 3,02 for the linear input condition and 3,63 for the IRS input condition. The corresponding values obtained in these characterization tests for the GSM full-rate codec are 3,79 and 3,93 respectively.

D.2.1.3 Effect of input level

There is no change in performance with input level of practical significance, over the range -32 dB to -12 dB.

D.2.1.4 Effect of input frequency characteristic

The presence of the IRS input characteristic generally results in an improvement in performance, but there are exceptions to this effect and it is likely to depend on the nature of any background noise present.

D.2.1.5 Effect of transmission errors

The degradation in Q value due to Error Pattern 1 (EP1) is small, but larger for other more severe error patterns as expected.

D.2.1.6 Effect of tandeming

Tandeming degrades the speech quality of the TETRA codec and is preferably avoided whenever possible. For clean speech with the IRS input condition this amounts to a reduction of about 4,2 dB Q value on average.

D.2.1.7 Effect of acoustic background noise

Background noise causes degradation of speech quality which increases as the signal-to-noise ratio reduces. The robustness of the TETRA codec to background noise appears to be similar to that of the full rate GSM codec.

D.2.1.8 Effect of vocal effort

Increasing vocal effort when the talker is subjected to high levels of noise causes some degradation of speech quality and increases the required listener effort.

D.2.1.9 Effect of frame stealing

Regular frame stealing at a rate of one speech frame per TETRA multiframe degrades the speech quality slightly, by about 1 dB on average for clean speech. The presence of background noise or transmission errors reduces the audibility of the degradation.

D.2.1.10 Speaker and language dependency

There may be some variation in performance for different languages.

The performance for male speakers is better than for female speakers. As an example, for clean speech at the nominal input level of -22 dB the average difference in MOS score is 0,39 for the IRS input condition. This effect is not apparent from the Q values because of the perceived quality difference between the male and female MNRU reference conditions used to derive the Q values.

D.2.2 Comparison with analogue FM

This subclause summarizes results from the original TETRA codec selection competition test and uses them to compare the speech quality of the TETRA codec with an analogue FM reference system.

D.2.2.1 Analogue and digital systems results

The codec selection tests were designed to compare a number of digital codecs under conditions of transmission errors and acoustic background noise. In addition results were obtained for an analogue FM simulator. The channel conditions for the analogue simulator and for the digital error pattern generation are summarized in table D.1. It was unfortunately not possible to make the analogue channel conditions identical to the digital conditions. In order for the speech quality from the analogue system to be representative of realistic operating conditions, it was found necessary to raise the equivalent RF Signal-to-Noise Ratio (SNR) by 10 dB compared to the digital systems. Another difference was that the analogue system used a 12,5 kHz FM channel, while the digital systems were designed to transmit four channels in 25 kHz using Time Division Multiplex (TDM). Since the conditions applied for analogue and digital systems do not correspond to identical operating conditions, results given hereafter should be used with care. Figures given are valid only under assumptions of table D.1 and do not reflect perfectly the behaviour of respective systems in identical environments.

Table D.1: Channel conditions for analogue and digital systems

Channel model	Error pattern	Digital system			Analogue System	
		Es/No (dB)	**Vehicle speed (km/h)**	**Bit Error Ratio (BER)**	**SNR (dB)**	**Vehicle Speed (km/h)**
Perfect	EP0	Infinity	0	No errors	> 60	0
TU50	EP1	20,0	50	1 %	28,7	60
TU50	EP2	15,2	50	3 %	24,7	60
HT50	EP3	16,7	50	3 %	25,6	60
TU100	EP4	12,8	100	5 %	21,7	115
HT150	EP5	14,3	150	5 %	21,7	115

D.2.2.2 All conditions

The results for all the experimental conditions were taken from the TETRA codec selection results and are shown in table D.2, where for each condition the quality score from the selected digital system is shown against the score for the analogue system. Since the performance of the analogue system was found to depend strongly on the audio input signal level, data points for the three different input levels have been distinguished. The reference level is the overload point. An average speech level of -22 dB corresponds to the nominal input condition while at -12 dB some clipping is present.

The experiments involving the linear input condition used a reference with linear input characteristics and no IRS filtering. Therefore the results presented in table D.2 cannot be directly compared with those obtained during the later characterization tests as reported in subclause D.2.1.

From table D.2, it can be seen that the performance of the digital system is always at least comparable with the analogue system, and often much better, particularly with the low input level of -32 dB.

Table D.2: Q value scores for digital TETRA and analogue FM

Conditions		Digital TETRA	Analogue FM
A-Law IRS	-12 dB	14,6	15,5
	-22 dB	15,5	14,7
	-32 dB	14,8	9,5
	EP0	15,5	14,7
	EP1	13,6	7,2
	EP2	9,8	6,9
	EP3	12,2	7,1
	EP4	6,5	6,9
	EP5	7,5	5,2
Linear	-12 dB	22,6	26,6
	-22 dB	23,3	21,7
	-32 dB	22,8	13,2
	EP0	23,3	21,7
	EP1	22,7	12,4
	EP2	18,6	6,8
	EP3	19,7	10,8
Office noise		4,4	5,4
Vehicular noise		4,6	6,8
Traffic noise		3,1	2,6

D.2.2.3 Input level

The quality of the FM system was found to be more dependent on input level, whereas the quality achieved by the digital system was almost independent of audio input level.

D.2.2.4 Error patterns

The available configurations from the analogue system did not allow an exact match with the conditions used for the digital system, but they are close enough for a valid comparison, as indicated in table D.1.

The quality for the digital system is generally higher than for the analogue system.

D.2.2.5 Background noise

Tests with acoustic background noise were carried out. The conditions were office noise, which was a "babble" produced by adding multiple speech files, vehicular noise recorded inside a moving vehicle, and traffic noise obtained at a roadside. In all cases the background noise was added to the speech data to give an equivalent acoustic SNR of 10 dB. In addition the talkers in the case of the traffic noise were subjected to ambient noise to allow investigation of the effect of increased vocal effort on coded speech quality.

For a nominal input level of -22 dB with no transmission errors it was found that the analogue system performance is better for two of the three conditions, but that the differences are not very great between the digital and analogue.

D.2.3 Additional tests

In order to make sure that the TETRA codec is able to handle unusual types of signals, non-exhaustive sets of experiments were performed. The signals consisted of speech with added background noises and non-speech signals.

D.2.3.1 Types of signals

The following signals were generated as background noises:

1) sinusoids of constant frequency (400 Hz, 1 000 Hz, 3 000 Hz);

2) swept frequency sinusoid (3 000 Hz to 100 Hz);

3) rectangular pulses (On = 2 samples, Off = 20 samples or Off = 300 samples);

4) swept frequency pulse (On = 2 to 20 samples, Off = 20 to 200 samples);

5) pulsed swept sinusoid (On = 50 to 100 samples, Off = 100 to 200 samples, 2 000 Hz to 100 Hz);

6) dc level (16 000 or 2 000 where +/-32 767 is the maximum input magnitude);

7) noise - Gaussian, uniform Probability Density Function (PDF), two valued PDF (maximum +/-32 000);

8) DTMF signals.

In all cases various parameters were adjusted (e.g. amplitude, frequency, pulse durations).

The test signals were generated with large amplitude and added to typical speech files so that the speech to interference ratio was 3 dB. As well as the corrupted speech files, the background noise signals were passed alone through the speech codec.

D.2.3.2 Codec behaviour

Under the conditions as specified objective and informal subjective tests were carried out where appropriate. From the informal listening sessions, the quality of the reconstructed files were generally judged as good, and the SNR degradation of the signals through the codec was acceptable. Overall, no abnormal behaviour was exhibited by the TETRA speech codec.

D.3 Performance of the channel coding/decoding for speech

The channel coding for the speech traffic channel of the TETRA system is different to the other TETRA channel types. The definition of the TETRA speech channel coding and decoding is described in clauses 5 and 6.

This clause describes the speech channel decoder performance in terms of Bit Error Ratio (BER), Message Error Rate (MER) and Probability of Undetected Erroneous Message (PUEM).

For the purpose of this clause:

- BER is the percentage of bits in error after channel decoding;
- MER is the rate at which the CRC indicates an error in the class 2 bits;
- PUEM is the rate at which the CRC fails to detect erroneous class 2 bits.

D.3.1 Classes of simulation environment conditions

Three classes of simulation environment are specified, distinguished by their intended operating environments and testing speeds. The operating environments are:

- Typical Urban (TU);
- Hilly Terrain (HT);
- Equalizer Test (EQ);

and the testing speeds are:

- 50 km/h;
- 200 km/h.

The simulation environment HT200 is therefore interpreted as "hilly terrain at 200 km/h".

D.3.2 Classes of equipment

Three equipment classes are specified, distinguishing their intended operating environments and testing conditions. The classes have preferred operating conditions, as follows:

Class A:

equipment is optimized for use in urban areas and in areas with hilly or mountainous terrain. It is resilient to extreme propagation conditions (hilly terrain) and is specified in static, TU50 and HT200 conditions;

Class B:

equipment is optimized for use in built-up and urban areas. The specification guarantees good performance at the reference sensitivity and interference level in static and TU50 conditions, but not in extreme propagation conditions (hilly terrain);

Class E:

equipment comprises an equalizer and is specified in static, TU50 and EQ200 conditions. It is not applicable to BS equipment.

All classes are meant for use in rural areas.

D.3.3 Classes of bits

The bits from the speech encoder are broken into three classes, class 0, class 1 and class 2 depending on their relative importance in the encoded speech. Class 0 bits are not encoded, class 1 bits are subject to 8/12 FEC coding, and class 2 bits (the most important) are subject to 8/18 FEC encoding and an associated CRC check. Definitions of these classes are as specified in clause 5.

D.3.4 Channel conditions

The channel conditions for which the speech channel codec is specified shall be as specified in ETS 300 392-2 [1]. For the downlink case, the conditions are for the discontinuous downlink channel type.

The minimum required static reference sensitivity performance is specified according to the logical channel and the receiver class at the static reference sensitivity level. The static reference sensitivity level shall be:

- for Mobile Stations (MS): -112 dBm;
- for Base Stations (BS): -115 dBm.

The minimum required dynamic reference sensitivity performance is specified according to the logical channel, the propagation condition and the receiver class at the dynamic reference sensitivity level. The dynamic reference sensitivity level shall be:

- for MS: -103 dBm;
- for BS: -106 dBm.

The minimum required reference interference performance (for co-channel, C/Ic, or adjacent channel, C/Ia) is specified according to the logical channel, the propagation condition and the receiver class at the reference interference ratio. The reference interference ratio shall be, for BS and all types of MS:

- for co-channel interference: C/Ic = 19 dB;
- for adjacent channel interference: C/Ia = -45 dB.

In the case of co-channel interference these specifications apply for a wanted input signal level of -85 dBm, and in the case of adjacent channel interference for a wanted input signal level 3 dB above the dynamic reference sensitivity level. In any case the interference shall be a continuous TETRA random modulated signal subjected to an independent realization of the same propagation condition as the wanted signal.

D.3.5 Results for normal case

The following table gives the minimum required performance figures in terms of maximum allowable error probabilities expressed as a percentage for the speech bit classes and channel types as specified in the previous subclauses.

Table D.3: Maximum allowable error probabilities for TETRA codec bits for the various channel types

Equipment class	Simulation description		BER Class 0	BER Class 1	MER	PUEM
A, E, B	Static reference sensitivity	Uplink	3,3 %	0,15 %	0,02 %	<0,001 %
		Downlink	3,3 %	0,15 %	0,018 %	<0,001 %
A, E, B	TU50 dynamic reference sensitivity	Uplink	2,2 %	1,6 %	2,2 %	0,008 %
		Downlink	2,2 %	1,6 %	2,2 %	0,007 %
A	HT200 dynamic reference sensitivity	Uplink	3,9 %	1,8 %	2,7 %	0,011 %
		Downlink	3,8 %	1,7 %	2,6 %	0,01 %
E	EQ200 dynamic reference sensitivity					
		Downlink only	10,3 %	9,4 %	14,3 %	0,037 %
A, E, B	TU50 reference interference performance	Uplink	2,3 %	1,8 %	2,7 %	0,01 %
		Downlink	2,3 %	1,9 %	2,7 %	0,012 %
A	HT200 reference interference performance	Uplink	3,8 %	1,9 %	2,8 %	0,011 %
		Downlink	3,8 %	2,0 %	2,8 %	0,011 %
E	EQ200 reference interference performance					
		Downlink only	9,3 %	8,1 %	12,3 %	0,045 %

D.4 Complexity

The computational complexity of the TETRA speech and channel codec was assessed in order to gain an indicative estimated figure which can be used by designers of the TETRA system.

D.4.1 Complexity analysis

D.4.1.1 Measurement methodology

The computational complexity of the TETRA codec was measured with rules derived from those adopted for the half-rate GSM Speech Codec competition. The complexity evaluation was carried out using the fixed-point C code simulation of the TETRA codec which is included in the computer files attached to this ETS.

D.4.1.2 TETRA basic operators

The basic operators for TETRA are the same as those used for the half-rate GSM codec competition with additional basic operators specific to the TETRA codec. The operators and their computational weighting is given in tables D.4, D.5, D.6 and D.7. In addition, the terms array16, array32, var16, var32, pointer16, pointer32, return16 and return32 stand for:

1) array16, array32: array of Word16 or Word32;
2) var16, var32: variable of 16 or 32 bits;
3) pointer16, pointer32: pointer to variable of 16 or 32 bits;
4) return16, return32: return value of a function of 16 or 32 bits.

In tables D.4, D.5, and D.6 some operators have been grouped by type. The called functions are not included in the complexity of the calling functions; they appear independently. The complexity takes into account the number of calls specified, which includes calls made by calling functions as well as by the main routine. The grouping of the operators is given by table D.4.

Table D.4: Operator groupings and their weights

Group name	Instructions	Weight
S_DM (Short data move)	array16 var16 pointer16 return(Word16)	1
L_DM (Long data move)	array32 var32 pointer32 return(Word32)	2
add	add, sub negate	1
L_add	L_add, L_sub L_negate	2
sh	shl, shr	1
extract	extract_h extract_l	1
L_dep	L_deposit_h L_deposit_l	2
L_mac	L_mac, L_msu	1
L_mac0	L_mac0, L_msu0	1
A_test (Arithmetic test)	A_test	2
Slog (Logical operation)	Slog	1

Table D.5: TETRA function operators

Name	Weight
Load_sh	1
add_sh	1
sub_sh	1
Load_sh16	1
add_sh16	1
sub_sh16	1
Store_hi	3
norm_v	37
L_extract	5
L_comp	2
mpy_32	7
mpy_mix	4
div_32	52

Table D.6: TETRA codec basic operators and weightings as used in GSM

Name	Weight
add	1
sub	1
abs_s	1
shl	1
shr	1
mult	1
L_mult	1
negate	1
extract_h	1
extract_l	1
round	1
L_mac	1
L_msu	1
L_add	2
L_sub	2
L_negate	2
mult_r	2
L_shl	2
L_shr	2
L_deposit_h	2
L_deposit_l	2
L_shr_r	3
L_abs	3
norm_s	15
div_s	18
norm_l	30

Table D.7: TETRA codec additional basic operators

Name	Weight
L_mult0	1
L_mac0	1
L_msu0	1

D.4.1.3 Worst case path for speech encoder

The most computational intensive path through the TETRA speech encoder for one frame of speech is given in table D.8. As indicated by the table some of the routines are called more than once through the encoder, e.g. the Lsp_Az routine for instance is called eight times for each frame of processing.

Table D.8: Worst case path for the TETRA speech encoder

Name of function	Number of calls
Autocorr	1
Lag_Window	1
Levinson	1
Az_Lsp	1
Chebps	1*111
Clsp_334	1
Int_Lpc4	2
Lsp_Az	2*4
Get_Lsp_Pol	2*4*2
Pond_Ai	16
Residu	8
Syn_Filt	32
Pitch_Ol_Dec	1
Lag_Max	1*3
inv_sqrt	1*3*1
Pitch_Fr	4
Norm_Corr	1*1 + 3*1
Convolve	1*1*1 + 3*1*1
inv_sqrt	1*1*13 + 3*1*18
Inter8_M1_3	4*2
Inter8_1_3	4*2
Pred_Lt	4
Inter32_M1_3	4*60
G_Pitch	4
Cal_Rr2	4
Back_Fil	4
D4i60_16	4
G_Code	4
Ener_Qua	4
Lpc_Gain	4*1
Syn_Filt	4*1*1
Log2	4*4
pow2	4*2

D.4.1.4 Worst case path for speech decoder

The most computational intensive path through the TETRA speech decoder is given in table D.9.

Table D.9: Worst case path for the TETRA speech decoder

Name of function	Number of calls
D_Lsp334	1
Int_Lpc4	1
Lsp_Az	1*4
Get_Lsp_Pol	1*4*2
Pred_Lt	4
Inter32_M1_3	4*60
Pond_Ai	8
Syn_Filt	8
D_D4i60	4
Dec_Ener	4
Lpc_Gain	4*1
Syn_Filt	4*1*1
Log2	4*2
pow2	4*2

D.4.1.5 Condensed complexity values for encoder and decoder

The computational complexity and memory requirements for the TETRA speech and channel coding is given in table D.10. Read-Only Memory (ROM) stands for data tables; program code is not taken into account. ROM is the sum of all that is needed in speech and channel encoders and decoders. Since the codec works in half-duplex, the needs in static Random Access Memory (RAM) are summed up in the speech and channel encoders and respectively decoders, only. The need in scratch RAM is identified as the maximum of each stage.

NOTE: In this case, contrary to normal standard, one kbytes equals 1 000 bytes, i.e. 500 Word16.

Full set of tabulated results are given in annex E.

The formula used to evaluate the complexity of the TETRA codec is:

$$C = MOPS + 0,2*RAM + 0,05*ROM.$$

RAM is the data memory in kbytes, and is the sum of the (maximum of) Scratch RAM and (sum of) Static RAM. ROM is the memory for data tables in kbytes.

The results are:

1) for the encoder:

speech encoder 9,624 MOPS;
channel encoder 0,081 MOPS;

RAM 8,34 kbytes, 1,668 MOPS;
ROM 11,07 kbytes, 0,550 MOPS;

encoder complexity 11,923 MOPS;

2) for the decoder:

speech decoder 1,025 MOPS;
channel decoder 3,040 MOPS;

RAM 3,84 kbytes, 0,768 MOPS;
ROM 11,07 kbytes, 0,550 MOPS;

decoder complexity 5,383 MOPS.

Table D.10: Summarized complexity and storage requirements for TETRA codec

<table>
<tr><th colspan="2"></th><th>Speech encoder</th><th>Speech decoder</th><th>Channel encoder</th><th>Channel decoder</th></tr>
<tr><td rowspan="2">Complexity</td><td>Operations per frame</td><td>288 720</td><td>30 750</td><td>4 852</td><td>182 395</td></tr>
<tr><td>MOPS</td><td>9,624</td><td>1,025</td><td>0,081</td><td>3,040</td></tr>
<tr><td rowspan="3">Memory in Kbytes</td><td>Scratch RAM</td><td>2,39</td><td>1,40</td><td>1,45</td><td>1,47</td></tr>
<tr><td>Static RAM</td><td>4,54</td><td>0,93</td><td>1,41</td><td>1,44</td></tr>
<tr><td>ROM</td><td>10,044</td><td>9,24</td><td colspan="2">1,026</td></tr>
</table>

D.4.2 DSP independence

The implementation of the TETRA codec was evaluated with respect to typical DSP devices commonly available at the time of TETRA standardization. This included the Texas Instruments TMS320C5X, AT+T DSP16/DSP16A, Analog Devices ADSP-2100 and the Motorola DSP56016. This involved examining the C code for the TETRA codec in detail to assess whether any particular part of the TETRA codec possess any inherent computational difficulties when practical real-time solutions are constructed. A careful analysis of the results indicated that the feasibility of a full duplex implementation on any of the above mentioned DSP families will be guaranteed.

D.4.2.1 Program control structure

With respect to program control no peculiar aspects that would make implementation particularly favourable or unfavourable for any of the DSPs were found.

D.4.2.2 Basic operator implementation

All the DSP devices were capable of implementing the TETRA basic operators. However, as their architectures are different, the ease or difficulty in their execution of particular operators varied. Thus, the conclusion is that an assembler implementation of the TETRA codec may lead to quite different results in terms of complexity and memory usage.

D.4.2.3 Additional operator implementation

As most of the additional operators are derived starting from the basic operators, the results for the basic operator implementation also applies.

D.5 Delay

The TETRA codec's audio delay is broken down into four parts as shown below:

$$D = Da + Dt + Dl + Ds$$

where:

Da (algorithmic delay) = 30 ms (analysis of one speech frame) + 5 ms (40 speech samples for interpolation) = 35 ms;

Dt (transmission delay)) = 15 ms (duration of TDMA slot);

Dl (interleaving delay) = 30 ms (interleaving of 2 speech frames);

Ds (error smoothing delay) = 0 ms (no error smoothing delay).

The total delay due to the TETRA codec is therefore 80 ms.

NOTE: Although this partially takes into account the transmission delay, the real practical system audio delay calculation is more complicated and can be found in the TETRA Designers' Guide (Part 1: Overview, Technical Description and Radio Aspects).

Annex E (informative): Results of the TETRA codec characterization listening and complexity tests

E.1 Characterization listening test

E.1.1 Experimental conditions

In order to characterize the TETRA speech codec, listening tests were carried out in various languages and table E.1 shows the allocation of languages to experiments.

Table E.1: Allocation of languages to the listening test experiments

Experiment	Language
1	Swedish
2	Italian
3	Dutch
4	German
5	German
6	Dutch
7	English, Dutch, German, French, Italian, Swedish
8	English
9	French
10	French
11	English

In the IRS input condition the speech signal is processed by an IRS filter and an A law PCM characteristic. In the UPCM input condition a linear PCM characteristic with no IRS filter was used.

The input levels are referred to the overload point.

The vehicle noise was obtained inside a moving vehicle and added electronically to the speech to obtain the specified signal-to-noise ratio.

The office noise was generated by multiple simultaneous speakers and added electronically to the speech samples to obtain the specified signal-to-noise ratio.

The ambient noise condition was produced by adding noise generated by moving traffic to speech samples obtained from talkers subject to high acoustic noise levels. This was intended to assess codec performance with realistic vocal effort.

The error patterns correspond to the channel models in table E.2.

Table E.2: Error patterns used for the listening test experiments

Error pattern label	Channel model	Bit Error Ratio (BER)
EP0	Perfect channel	No transmission errors
EP1	TU50	1 %
EP2	TU50	3 %
EP3	HT50	3 %
EP4	TU100	5 %
EP5	HT150	5 %

A listening effort scale was used in experiment 8 which makes it difficult to directly compare results with those from other experiments.

The summarized results of the characterization listening experiments are given in the following tables. For comparison purposes results are also given for the full-rate GSM system speech codec. In each case the values are relative to a reference condition using the IRS input characteristic. The Q values have been averaged over male and female speakers. The "T" denotes a tandeming condition and "F" denotes a regular frame stealing condition where one speech frame is stolen for every 34 frames of speech (i.e. one sub-slot stolen in every TETRA multiframe).

E.1.2 Tables of results

There are entries in the tables where the Q value is zero. This is a result of the MOS to Q conversion curve where for many cases, a zero MOS score maps to a negative Q value. Hence, for any MOS scores which maps to less than or equal to a zero Q value, a zero Q value was assigned. Therefore, a zero Q value does not necessarily equate to a zero MOS value. Also, some Q values are very large. This again is a result of the MOS to Q conversion. Basically for MOS scores falling on the top part of the conversion curve, the Q values are magnified considerably. For both zero Q and large Q conditions, the results are not useful as they do not lie within the useful portion of the MNRU reference conditions.

Table E.3: Q values for TETRA codec under EP0 relative to MNRU IRS speech

Conditions			Experiment								
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**7**	**8**	**9**	**9T**	**11**	**11F**
A-Law IRS	No noise	-12				14,1		14,5	11,2	17,1	16,9
		-22				16,5		16,6	13,1	17,2	16,4
		-32				15,7		18,0	12,2	19,4	17,4
	Veh. 0 dB	-12		0							
		-22		0							
		-32		0							
	Veh. -10 dB	-12		4,3				2,4	1,5		
		-22		4,5				3,7	0,8		
		-32		4,9				0	0		
	Veh. -20 dB	-12		8,0		8,9				12,5	12,0
		-22		8,8		10,2				12,7	11,9
		-32		8,9		9,7				12,4	12,0
	Off. 0 dB	-12			0						
		-22			0,5						
		-32			0						
	Off. -10 dB	-12			7,2						
		-22			7,2						
		-32			7,9						
	Off. -20 dB	-12			10,2					14,5	12,8
		-22			11,4					14,5	13,9
		-32			11,6					14,7	14,2
	Amb. -10 dB	-12					9,6				
		-22					10,0				
		-32					10,0				

Table E.4: Q values for TETRA codec under EP0 relative to MNRU IRS speech

Conditions			Experiment								
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**7**	**8**	**9**	**9T**	**11**	**11F**
U-PCM No IRS	No noise	-12	12,47			12,61					
		-22	11,72			12,98					
		-32	12,72			12,44					
	Veh. 0 dB	-12									
		-22									
		-32									
	Veh. -10 dB	-12	3,92								
		-22	6,48								
		-32	4,10								
	Veh. -20 dB	-12				8,65					
		-22				9,30					
		-32				9,48					
	Off. 0 dB	-12									
		-22									
		-32									
	Off. -10 dB	-12									
		-22									
		-32									
	Off. -20 dB	-12	7,86								
		-22	9,39								
		-32	8,89								
	Amb. -10 dB	-12					14,20				
		-22					13,62				
		-32					15,65				

Table E.5: Q values for TETRA codec under EP1 relative to MNRU IRS speech

Conditions			Experiment							
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**8**	**10**	**10T**	**11**	**11F**
A-Law IRS	No noise	-12					14,6	9,7	15,4	15,8
		-22					16,3	11,9	18,0	15,9
		-32					15,8	11,5	16,8	16,2
	Veh. 0 dB	-12			0					
		-22			0					
		-32			0					
	Veh. -10 dB	-12			4,4		3,8	1,2		
		-22			4,2		4,6	2,2		
		-32			6,0		2,2	0,7		
	Veh. -20 dB	-12			7,5				11,2	11,7
		-22			9,7				12,0	12,0
		-32			8,6				12,6	11,7
	Off. 0 dB	-12		0						
		-22		0						
		-32		0						
	Off. -10 dB	-12		8,0						
		-22		6,8						
		-32		7,5						
	Off. -20 dB	-12		10,4					13,5	13,8
		-22		12,5					13,3	13,9
		-32		10,6					13,6	12,9
	Amb. -10 dB	-12				8,4				
		-22				8,0				
		-32				8,3				

Table E.6: Q values for TETRA codec under EP1 relative to MNRU IRS speech

Conditions			Experiment							
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**8**	**10**	**10T**	**11**	**11F**
U-PCM No IRS	No noise	-12	13,1							
		-22	14,1							
		-32	14,4							
	Veh. 0 dB	-12								
		-22								
		-32								
	Veh. -10 dB	-12	12,4							
		-22	11,7							
		-32	11,6							
	Veh. -20 dB	-12								
		-22								
		-32								
	Off. 0 dB	-12								
		-22								
		-32								
	Off. -10 dB	-12	6,7							
		-22	7,4							
		-32	7,3							
	Off. -20 dB	-12								
		-22								
		-32								
	Amb. -10 dB	-12				13,0				
		-22				12,3				
		-32				13,2				

Table E.7: Q values for TETRA codec under EP2 relative to MNRU IRS speech

Conditions			Experiment						
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**7**	**8**	**9**	**9T**
A-Law IRS	No noise	-12				10,4		11,0	7,8
		-22				11,7		12,7	9,2
		-32				10,9		12,6	9,7
	Veh. 0 dB	-12		0					
		-22		0					
		-32		0					
	Veh. -10 dB	-12		0				0,3	0
		-22		1,7				0,8	0
		-32		0				0	0
	Veh. -20 dB	-12		4,7		6,3			
		-22		6,8		7,2			
		-32		5,7		6,6			
	Off. 0 dB	-12			0				
		-22			0				
		-32			0				
	Off. -10 dB	-12			5,1				
		-22			5,9				
		-32			5,0				
	Off. -20 dB	-12			9,1				
		-22			8,7				
		-32			10,0				
	Amb. -10 dB	-12					4,7		
		-22					4,9		
		-32					4,5		

Table E.8: Q values for TETRA codec under EP2 relative to MNRU IRS speech

Conditions			Experiment						
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**7**	**8**	**9**	**9T**
U-PCM No IRS	No noise	-12	8,9			9,1			
		-22	7,5			9,3			
		-32	7,5			9,1			
	Veh. 0 dB	-12							
		-22							
		-32							
	Veh. -10 dB	-12	0,1						
		-22	0						
		-32	2,6						
	Veh. -20 dB	-12				5,9			
		-22				5,7			
		-32				6,3			
	Off. 0 dB	-12							
		-22							
		-32							
	Off. -10 dB	-12							
		-22							
		-32							
	Off. -20 dB	-12	3,7						
		-22	6,0						
		-32	4,6						
	Amb. -10 dB	-12					7,3		
		-22					5,9		
		-32					8,0		

Table E.9: Q values for TETRA codec under EP3 relative to MNRU IRS speech

Conditions			Experiment					
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**8**	**10**	**10T**
A-Law IRS	No noise	-12					9,6	7,5
		-22					11,0	7,0
		-32					11,5	7,4
	Veh. 0 dB	-12			0			
		-22			0			
		-32			0			
	Veh. -10 dB	-12			1,7		0	0
		-22			2,9		0	0
		-32			0,8		0	0
	Veh. -20 dB	-12			6,8			
		-22			6,4			
		-32			8,0			
	Off. 0 dB	-12		0				
		-22		0				
		-32		0				
	Off. -10 dB	-12		5,0				
		-22		5,2				
		-32		4,3				
	Off. -20 dB	-12		8,3				
		-22		7,8				
		-32		9,4				
	Amb. -10 dB	-12				6,4		
		-22				5,7		
		-32				5,1		

Table E.10: Q values for TETRA codec under EP3 relative to MNRU IRS speech

Conditions			Experiment					
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**8**	**10**	**10T**
U-PCM No IRS	No noise	-12	9,5					
		-22	10,5					
		-32	10,3					
	Veh. 0 dB	-12						
		-22						
		-32						
	Veh. -10 dB	-12						
		-22						
		-32						
	Veh. -20 dB	-12	9,2					
		-22	9,4					
		-32	9,0					
	Off. 0 dB	-12						
		-22						
		-32						
	Off. -10 dB	-12	4,8					
		-22	4,7					
		-32	4,5					
	Off. -20 dB	-12						
		-22						
		-32						
	Amb. -10 dB	-12				7,5		
		-22				9,0		
		-32				8,6		

Table E.11: Q values for TETRA codec under EP4 relative to MNRU IRS speech

Conditions			Experiment				
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**9**	**9T**
A-Law IRS	No noise	-12				10,4	7,1
		-22				11,2	7,6
		-32				12,2	7,3
	Veh. 0 dB	-12		0			
		-22		0			
		-32		0			
	Veh. -10 dB	-12		1,4		0	0
		-22		0,5		0	0
		-32		1,3		0	0
	Veh. -20 dB	-12		3,4			
		-22		4,5			
		-32		6,1			
	Off. 0 dB	-12			0		
		-22			0		
		-32			0		
	Off. -10 dB	-12			4,0		
		-22			4,0		
		-32			4,9		
	Off. -20 dB	-12			7,2		
		-22			8,7		
		-32			6,6		
	Amb. -10 dB	-12					
		-22					
		-32					

Table E.12: Q values for TETRA codec under EP4 relative to MNRU IRS speech

Conditions			Experiment				
I/P Char	**Noise SNR**	**Input level**	**1**	**3**	**5**	**9**	**9T**
U-PCM No IRS	No noise	-12	6,8				
		-22	7,6				
		-32	7,1				
	Veh. 0 dB	-12					
		-22					
		-32					
	Veh. -10 dB	-12	1,1				
		-22	0,4				
		-32	0,4				
	Veh. -20 dB	-12					
		-22					
		-32					
	Off. 0 dB	-12					
		-22					
		-32					
	Off. -10 dB	-12					
		-22					
		-32					
	Off. -20 dB	-12	2,3				
		-22	3,6				
		-32	5,0				
	Amb. -10 dB	-12					
		-22					
		-32					

Table E.13: Q values for TETRA codec under EP5 relative to MNRU IRS speech

Conditions			Experiment				
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**10**	**10T**
A-Law IRS	No noise	-12				8,9	5,1
		-22				10,2	5,4
		-32				9,5	6,1
	Veh. 0 dB	-12			0		
		-22			0		
		-32			0		
	Veh. -10 dB	-12			1,0	0	0
		-22			0,3	0	0
		-32			2,1	0	0
	Veh. -20 dB	-12			3,9		
		-22			5,0		
		-32			7,3		
	Off. 0 dB	-12		0			
		-22		0			
		-32		0			
	Off. -10 dB	-12		4,4			
		-22		3,7			
		-32		4,5			
	Off. -20 dB	-12		6,6			
		-22		9,1			
		-32		7,4			
	Amb. -10 dB	-12					
		-22					
		-32					

Table E.14: Q values for TETRA codec under EP5 relative to MNRU IRS speech

Conditions			Experiment				
I/P Char	**Noise SNR**	**Input level**	**2**	**4**	**6**	**10**	**10T**
U-PCM No IRS	No noise	-12	9,1				
		-22	9,9				
		-32	9,7				
	Veh. 0 dB	-12					
		-22					
		-32					
	Veh. -10 dB	-12					
		-22					
		-32					
	Veh. -20 dB	-12	8,3				
		-22	8,6				
		-32	7,5				
	Off. 0 dB	-12					
		-22					
		-32					
	Off. -10 dB	-12	3,3				
		-22	2,8				
		-32	2,4				
	Off. -20 dB	-12					
		-22					
		-32					
	Amb. -10 dB	-12					
		-22					
		-32					

Table E.15: Q values for full rate GSM codec under EP0 relative to MNRU IRS speech

Conditions			Experiment										
I/P Char	**Noise SNR**	**Input level**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
A-Law IRS	No noise	-12							16,8		21,1	19,8	20,9
		-22							18,9		19,7	18,8	20,4
		-32							16,8		20,6	21,4	21,4
	Veh. 0 dB	-12			0			0					
		-22			0			0					
		-32			0			0					
	Veh. -10 dB	-12			4,5			4,3					
		-22			5,2			6,4					
		-32			4,1			5,6					
	Veh. -20 dB	-12			10,4			11,3					
		-22			10,3			10,6					
		-32			9,7			10,9					
	Off. 0 dB	-12				0	1,8						
		-22				0	0,5						
		-32				0	0,8						
	Off. -10 dB	-12				8,3	7,6						
		-22				8,9	8,4						
		-32				8,6	8,7						
	Off. -20 dB	-12				14,1	12,9						
		-22				12,4	11,0						
		-32				13,3	12,6						
	Amb. -10 dB	-12								10,7			
		-22								7,2			
		-32								10,7			

Table E.16: Q values for full rate GSM codec under EP0 relative to MNRU IRS speech

Conditions			Experiment										
I/P Char	**Noise SNR**	**Input level**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
U-PCM No IRS	No noise	-12	19,0	18,2					18,3				
		-22	17,0	18,0					17,4				
		-32	18,1	17,6					17,9				
	Veh. 0 dB	-12											
		-22											
		-32											
	Veh. -10 dB	-12	8,8										
		-22	10,0										
		-32	9,5										
	Veh. -20 dB	-12		14,2									
		-22		14,7									
		-32		13,1									
	Off. 0 dB	-12											
		-22											
		-32											
	Off. -10 dB	-12		7,6									
		-22		8,9									
		-32		9,0									
	Off. -20 dB	-12	13,4										
		-22	14,6										
		-32	13,0										
	Amb. -10 dB	-12											
		-22											
		-32											

E.2 TETRA codec complexity study

E.2.1 Computational analysis results

E.2.1.1 TETRA speech encoder

Table E.17: TETRA speech encoder instructions breakdown analysis, part 1

Encoder function	no of calls	S_DM 1	L_DM 2	round 1	abs_s 1	sh 1	add 1	mult 1
Coder_Tetra	1	1 506					428	160
Autocorr	1	771	13			512		
Az_Lsp	1	428			10	10	158	
Back_Fil	4	4	724				8	
Cal_Rr2	4	728	124			244	4	
Chebps	111	1 554						
Clsp_334	1	1 305	1 283			2	4 360	
Convolve	4	240	240					
D4i60_16	4	14 892			808	12 980	4 220	3 240
Ener_Qua	4	268	300			4	548	
Get_Lsp_Pol	16		496					
G_Code	4	4	16			8	8	
G_Pitch	4	8	16			8	16	
Inter8_1_3	8		8					
Inter8_M1_3	8		8					
Inter32_M1_3	240		240	240				
Int_Lpc4	2	4					12	
inv_sqrt	70					70	280	
Lag_Max (1)	1	190	129					
Lag_Max (2)	1	121	83					
Lag_Max (3)	1	61	43					
Lag_Window	1							
Levin_32	1	119	9				19	
Log2()	16						48	
Lpc_Gain	4	240	4					
Lsp_Az	8	16	80					
Norm_Corr (1)	1	721	39				13	
Norm_Corr (2,3,4)	3	3 063	162				54	
Pitch_Fr (1)	1	17	8				9	
Pitch_Fr (2,3,4)	3	81	24				42	
Pitch_Ol_Dec	1	128	1			120	2	2
Pond_Ai	16	176		160				
pow2()	8						16	
Pred_Lt	4	240					8	
Residu	8							
Syn_Filt /upd=0	20	1 420						
Syn_Filt /upd=1	12	972						
Total 1		29 277	4 050	400	818	13 958	10 253	3 402
Pre_Process	4	480						240
Total 2		29 757	4 050	400	818	13 958	10 253	3 642

Table E.18: TETRA speech encoder instructions breakdown analysis, part 2

Encoder function	mult_r 1	L_sh 2	L_add 2	L_abs 3	L_shr_r 3	L_mult 1	L_mult0 1
Coder_Tetra		240	240		240	240	240
Autocorr	256	22					
Az_Lsp		10					110
Back_Fil		240	240	240			
Cal_Rr2							
Chebps		444					
Clsp_334			1 280				1 280
Convolve							
D4i60_16			480	1 376		3 232	728
Ener_Qua		24	260				264
Get_Lsp_Pol		160	320				
G_Code		496	480				480
G_Pitch		496	480				480
Inter8_1_3							
Inter8_M1_3							
Inter32_M1_3			240				
Int_Lpc4						60	
inv_sqrt		350					
Lag_Max (1)		1	63				
Lag_Max (2)		1	40				
Lag_Max (3)		1	20				
Lag_Window							
Levin_32		38	74	20			
Log2()		48					
Lpc_Gain							
Lsp_Az			160		80		
Norm_Corr (1)		13					708
Norm_Corr (2,3,4)		54					3 009
Pitch_Fr (1)			4				
Pitch_Fr (2,3,4)			12				
Pitch_Ol_Dec		1	1				
Pond_Ai						160	
pow2()		16			8		
Pred_Lt							
Residu		480					
Syn_Filt /upd=0		1 200					
Syn_Filt /upd=1		720					
Total 1	256	5 055	4 394	1 636	328	3 692	7 299
Pre_Process		240					
Total 2	256	5 295	4 394	1 636	328	3 692	7 299

Table E.19: TETRA speech encoder instructions breakdown analysis, part 3

Encoder function	L_mac 1	L_mac0 1	extract 1	L_dep 2	A_test 2	Slog 1	norm_s 15
Coder_Tetra		240	240		9		
Autocorr		3 273			6		
Az_Lsp		10	60		253		10
Back_Fil		7 320	240		244		
Cal_Rr2	3 720	240	3 720				
Chebps			111				
Clsp_334		3 072			1 282		
Convolve		7 320					
D4i60_16	3 232	4 608	7 840		5 844		
Ener_Qua		744	36		276		
Get_Lsp_Pol							
G_Code		480	16		8		
G_Pitch		480	16		12		
Inter8_1_3		64					
Inter8_M1_3		64					
Inter32_M1_3		7 680					
Int_Lpc4	60		60				
inv_sqrt		70	140	70	140	140	
Lag_Max (1)		7 680	1		63		
Lag_Max (2)		4 920	1		40		
Lag_Max (3)		2 520	1		20		
Lag_Window							
Levin_32					10		
Log2()		16	48	16	16	16	
Lpc_Gain		240					
Lsp_Az			80				
Norm_Corr (1)		1 560	13		13		
Norm_Corr (2,3,4)		6 480	54		54		
Pitch_Fr (1)					12		
Pitch_Fr (2,3,4)					51		
Pitch_Ol_Dec		191			4		
Pond_Ai							
pow2()	8		16	16		8	
Pred_Lt					12		
Residu	4 800		480				
Syn_Filt /upd=0		12 000	1 200		20		
Syn_Filt /upd=1		7 200	720		12		
Total 1	11 820	78 472	15 093	102	8 401	164	10
Pre_Process	240		720				
Total 2	12 060	78 472	15 813	102	8 401	164	10

Table E.20: TETRA speech encoder instructions breakdown analysis, part 4

Encoder function	norm_l 30	div_s 18	Load_sh 1	Load_sh16 1	add_sh 1	sub_sh 1
Coder_Tetra			240			
Autocorr	1					
Az_Lsp		10	60		45	5
Back_Fil	4					
Cal_Rr2	4					
Chebps			111		555	888
Clsp_334						
Convolve						
D4i60_16			2 860	416	9 696	7 840
Ener_Qua	8		16	16	32	20
Get_Lsp_Pol			16		80	
G_Code	16	4				
G_Pitch	16	4				
Inter8_1_3						
Inter8_M1_3						
Inter32_M1_3						
Int_Lpc4						
inv_sqrt	70					
Lag_Max (1)						
Lag_Max (2)						
Lag_Max (3)						
Lag_Window						
Levin_32					100	
Log2()	16					
Lpc_Gain						
Lsp_Az						
Norm_Corr (1)					708	
Norm_Corr (2,3,4)					3 009	
Pitch_Fr (1)			1			
Pitch_Fr (2,3,4)			3			
Pitch_Ol_Dec						
Pond_Ai						
pow2()						
Pred_Lt						
Residu			480		480	
Syn_Filt /upd=0			1 200		1 200	
Syn_Filt /upd=1			720		720	
Total 1	135	18	5 707	432	16 625	8 753
Pre_Process			240		240	480
Total 2	135	18	5 947	432	16 865	9 233

Table E.21: TETRA speech encoder instructions breakdown analysis, part 5

Encoder function	add_sh16 1	sub_sh16 1	L_comp 2	Store_hi 3	mpy_mix 4
Coder_Tetra					
Autocorr					
Az_Lsp	5	5		10	
Back_Fil					
Cal_Rr2					
Chebps					444
Clsp_334					
Convolve				240	
D4i60_16		1 376		728	
Ener_Qua		12		8	
Get_Lsp_Pol					160
G_Code					
G_Pitch					
Inter8_1_3					
Inter8_M1_3					
Inter32_M1_3					
Int_Lpc4					
inv_sqrt					
Lag_Max (1)					
Lag_Max (2)					
Lag_Max (3)					
Lag_Window					
Levin_32			20	10	
Log2()					
Lpc_Gain					
Lsp_Az					
Norm_Corr (1)				708	
Norm_Corr (2,3,4)				3 009	
Pitch_Fr (1)					
Pitch_Fr (2,3,4)					
Pitch_Ol_Dec					
Pond_Ai					
pow2()					
Pred_Lt					
Residu					
Syn_Filt /upd=0					
Syn_Filt /upd=1					
Total 1	5	1 393	20	4 713	604
Pre_Process					
Total 2	5	1 393	20	4 713	604

Table E.22: TETRA speech encoder instructions breakdown analysis, part 6

Encoder function	return16 1	return32 2	var16 1	var32 2
Coder_Tetra	13		2	
Autocorr			3	13
Az_Lsp	111		294	
Back_Fil			1	121
Cal_Rr2				31
Chebps				
Clsp_334			1 282	1 283
Convolve				60
D4i60_16			3 597	
Ener_Qua		3	66	72
Get_Lsp_Pol				
G_Code			1	4
G_Pitch			2	4
Inter8_1_3				1
Inter8_M1_3				1
Inter32_M1_3				1
Int_Lpc4			2	
inv_sqrt				
Lag_Max (1)		1	63	128
Lag_Max (2)		1	40	82
Lag_Max (3)		1	20	42
Lag_Window				
Levin_32				9
Log2()				
Lpc_Gain				1
Lsp_Az			1	
Norm_Corr (1)		13	1	26
Norm_Corr (2,3,4)		18	1	36
Pitch_Fr (1)		4	15	4
Pitch_Fr (2,3,4)		4	25	4
Pitch_Ol_Dec	3		5	1
Pond_Ai				
pow2()				
Pred_Lt				
Residu				
Syn_Filt /upd=0			1	
Syn_Filt /upd=1			1	
Total 1	127	45	5 423	1 924
Pre_Process			120	
Total 2	127	45	5 543	1 924

Table E.23: TETRA speech encoder instructions breakdown analysis, part 7

Encoder function	array16 1	array32 2	pointer 1	pointer32 2
Coder_Tetra	847		644	
Autocorr	768			
Az_Lsp	12		11	
Back_Fil		60		
Cal_Rr2	60		122	
Chebps				
Clsp_334			23	
Convolve	60			
D4i60_16	124		2	
Ener_Qua			1	
Get_Lsp_Pol				31
G_Code				
G_Pitch				
Inter8_1_3				
Inter8_M1_3				
Inter32_M1_3				
Int_Lpc4				
inv_sqrt				
Lag_Max (1)			127	
Lag_Max (2)			81	
Lag_Max (3)			41	
Lag_Window				
Levin_32	119			
Log2()				
Lpc_Gain	60			
Lsp_Az	1	10		
Norm_Corr (1)	720			
Norm_Corr (2,3,4)	1 020			
Pitch_Fr (1)			2	
Pitch_Fr (2,3,4)			2	
Pitch_Ol_Dec	120			
Pond_Ai	11			
pow2()				
Pred_Lt	60			
Residu				
Syn_Filt /upd=0	60		10	
Syn_Filt /upd=1	70		10	
Total 1	4 112	70	1 076	31
Pre_Process				
Total 2	4 112	70	1 076	31

Table E.24: TETRA codec encoder instructions breakdown analysis, part 8

Encoder function	L_extract 5	mpy_32 7	norm_v 37	div_32 52	total weight	complexity (MOPS)
Coder_Tetra					4 992	0,1664
Autocorr	11				5 235	0,1745
Az_Lsp					1 792	0,0597
Back_Fil					11 308	0,3769
Cal_Rr2					9 024	0,3008
Chebps	444				8 103	0,2701
Clsp_334					17 709	0,5903
Convolve					8 760	0,292
D4i60_16					96 928	3,2309
Ener_Qua	8				3 984	0,1328
Get_Lsp_Pol	160				3 488	0,1163
G_Code					3 548	0,1183
G_Pitch					3 568	0,1189
Inter8_1_3					80	0,0027
Inter8_M1_3					80	0,0027
Inter32_M1_3					8 880	0,2960
Int_Lpc4					196	0,0065
inv_sqrt					3 920	0,1307
Lag_Max (1)	2	1			8 400	0,28
Lag_Max (2)	2	1			5 387	0,1796
Lag_Max (3)	2	1			2 767	0,0922
Lag_Window	10	10			120	0,004
Levin_32	85	110	10	10	2 715	0,0905
Log2()					768	0,0256
Lpc_Gain					488	0,0163
Lsp_Az					816	0,0272
Norm_Corr (1)	26	13			6 198	0,2066
Norm_Corr (2,3,4)	108	54			26 154	0,8718
Pitch_Fr (1)					75	0,0025
Pitch_Fr (2,3,4)					300	0,01
Pitch_Ol_Dec					457	0,0152
Pond_Ai					496	0,0165
pow2()					136	0,0045
Pred_Lt					272	0,0091
Residu					7 200	0,24
Syn_Filt /upd=0					19 460	0,6487
Syn_Filt /upd=1					11 796	0,3932
Total 1	858	190	10	10	285 600	9,52
Pre_Process					3 120	0,104
Total 2	858	190	10	10	288 720	9,624

E.2.1.2 TETRA speech decoder

Table E.25: TETRA speech decoder instructions breakdown analysis, part 1

Decoder function	no of calls	S_DM 1	L_DM 2	round 1	sh 1	add 1	mult 1
Decod_Tetra	1	925				193	164
D_D4i60	4	256			16	12	
D_Lsp334	1	39			2	17	
Dec_Ener	4	8	32		4	20	
Get_Lsp_Pol	8		248				
Inter32_M1_3	240		240	240			
Int_Lpc4	1	2				6	
Log2()	8					24	
Lpc_Gain	4	240	4				
Lsp_Az	4	8	40				
Pond_Ai	8	88		80			
pow2()	8					16	
Pred_Lt	4	240				8	
Syn_Filt /upd=0	8	568					
Syn_Filt /upd=1	4	324					
Total 1		2 698	564	320	22	296	164
Post_Process	4					240	
Total 2		2 698	564	320	22	536	164

Table E.26: TETRA speech decoder instructions breakdown analysis, part 2

Decoder function	L_sh 2	L_add 2	L_shr_r 3	L_mult 1	L_mult0 1	L_mac 1
Decod_Tetra			240		240	
D_D4i60		240			240	
D_Lsp334						
Dec_Ener	16	4			8	
Get_Lsp_Pol	80	160				
Inter32_M1_3		240				
Int_Lpc4				30		30
Log2()	24					
Lpc_Gain						
Lsp_Az		80	40			
Pond_Ai				80		
pow2()	16		8			8
Pred_Lt						
Syn_Filt /upd=0	480					
Syn_Filt /upd=1	240					
Total 1	856	724	288	110	488	38
Post_Process						
Total 2	856	724	288	110	488	38

Table E.27: TETRA speech decoder instructions breakdown analysis, part 3

Decoder function	L_mac0 1	extract 1	L_dep 2	A_test 2	Slog 1	norm_l 30
Decod_Tetra	240			12		
D_D4i60				4	16	
D_Lsp334				12		
Dec_Ener	480	28		24		8
Get_Lsp_Pol						
Inter32_M1_3	7 680					
Int_Lpc4		30				
Log2()	8	24	8	8	8	8
Lpc_Gain	240					
Lsp_Az		40				
Pond_Ai						
pow2()		16	16		8	
Pred_Lt				12		
Syn_Filt /upd=0	4 800	480		8		
Syn_Filt /upd=1	2 400	240		4		
Total 1	15 848	858	24	84	32	16
Post_Process						
Total 2	15 848	858	24	84	32	16

Table E.28: TETRA speech decoder instructions breakdown analysis, part 4

Decoder function	Load_sh 1	Load_sh16 1	add_sh 1	sub_sh 1	add_sh16 1
Decod_Tetra					
D_D4i60			240	480	
D_Lsp334					
Dec_Ener	16	8	24	20	
Get_Lsp_Pol	8			40	
Inter32_M1_3					
Int_Lpc4					
Log2()					
Lpc_Gain					
Lsp_Az					
Pond_Ai					
pow2()					
Pred_Lt					
Syn_Filt /upd=0	480		480		
Syn_Filt /upd=1	240		240		
Total 1	744	8	984	540	0
Post_Process					
Total 2	744	8	984	540	0

Table E.29: TETRA speech decoder instructions breakdown analysis, part 5

Decoder function	return16 1	return32 2	var16 1	var32 2
Decod_Tetra			24	
D_D4i60				
D_Lsp334			12	
Dec_Ener	3	2	5	
Get_Lsp_Pol				
Inter32_M1_3				1
Int_Lpc4			2	
Log2()				
Lpc_Gain				1
Lsp_Az			1	
Pond_Ai				
pow2()				
Pred_Lt				
Syn_Filt /upd=0			1	
Syn_Filt /upd=1			1	
Total 1	3	2	48	
Post_Process				
Total 2	3	2	48	2

Table E.30: TETRA speech decoder instructions breakdown analysis, part 6

Decoder function	array16 1	array32 2	pointer16 1	pointer32 2
Decod_Tetra	255		646	
D_D4i60	60		4	
D_Lsp334	24		3	
Dec_Ener				
Get_Lsp_Pol				31
Inter32_M1_3				
Int_Lpc4				
Log2()				
Lpc_Gain	60			
Lsp_Az	1	10		
Pond_Ai	11			
pow2()				
Pred_Lt	60			
Syn_Filt /upd=0			10	
Syn_Filt /upd=1			10	
Total 1	471	10	673	31
Post_Process				
Total 2	471	10	673	31

Table E.31: TETRA speech decoder instructions breakdown analysis, part 7

Decoder function	sub_sh16 1	Store_hi 3	mpy_mix 4	L_extract 5	total weight	complexity (MOPS)
Decod_Tetra					2 506	0,0835
D_D4i60		240			2 468	0,0823
D_Lsp334					82	0,0027
Dec_Ener	8	8		8	1 080	0,036
Get_Lsp_Pol			80	80	1 744	0,0581
Inter32_M1_3					8 880	0,296
Int_Lpc4					98	0,0033
Log2()					384	0,0128
Lpc_Gain					488	0,0163
Lsp_Az					408	0,0136
Pond_Ai					248	0,0083
pow2()					136	0,0045
Pred_Lt					272	0,0091
Syn_Filt /upd=0					7 784	0,2595
Syn_Filt /upd=1					3 932	0,1311
Total 1	8	248	80	88	30 510	1,017
Post_Process					240	0,008
Total 2	8	248	80	88	30 750	1,025

E.2.1.3 TETRA channel encoder and decoder

Table E.32: TETRA speech channel encoder and decoder instructions breakdown analysis, part 1

Subroutine called	Encoder calls	Decoder calls	add 1	sub 1	negate 1	shl 1	shr 1	L_mult 1
Build_Sensitivity_Classes	1							
Build_Crc	1		240					
Init_Rcpc_Coding			64	2		66		
Init_Rcpc			64	2		66		
Rcpc_Coding	1		184	184			185	184
Transform_Class_0	1							
Eid				432				
Interleaving_Speech	1							
Init_Rcpc_Decoding			64	2		66		
Rcpc_Decoding		1	17 664	11 299	17 994	330		
Untransform_Class_0		1						
Bfi		1	240	8				
Desinter-leaving		1						
Unbuild_Sensitivity_Classes		1						
Combination					5	5		
Channel encoder								
Channel decoder								

Table E.33: TETRA speech channel encoder and decoder instructions breakdown analysis, part 2

Subroutine called	extract_l 1	extract_h 1	L_shl 2	L_dep_l 2	A_test 2	Slog 1	var16 1
Build_Sensitivity_Classes							
Build_Crc						8	8
Init_Rcpc_Coding						32	2
Init_Rcpc						32	2
Rcpc_Coding	184				698	184	27
Transform_Class_0					102		
Eid					432		
Interleaving_Speech							
Init_Rcpc_Decoding						32	1
Rcpc_Decoding		184	184	184	34 784		32 923
Untransform_Class_0					102		
Bfi					8	8	17
Desinter-leaving							
Unbuild_Sensitivity_Classes							
Combination					5	6	1
Channel encoder							
Channel decoder							

Table E.34: TETRA speech channel encoder and decoder instructions breakdown analysis, part 3

Subroutine called	S_DM 1	array16 1	Combination 27	Total weight
Build_Sensitivity_Classes	274	274		274
Build_Crc	16	8		264
Init_Rcpc_Coding		132	96	5 482
Init_Rcpc		132	96	5 482
Rcpc_Coding	973	946		3 474
Transform_Class_0	204	204		408
Eid		432		1 728
Interleaving_Speech	432	432		432
Init_Rcpc_Decoding		128	96	5 477
Rcpc_Decoding	63 217	30 294		180 992
Untransform_Class_0	204	204		408
Bfi	17			289
Desinter-leaving	432	432		432
Unbuild_Sensitivity_Classes	274	274		274
Combination				27
Channel encoder				4 852
Channel decoder				182 395

E.2.2 Memory requirements analysis results

E.2.2.1 TETRA speech encoder

Table E.35: TETRA speech encoder memory usage breakdown

Path	Encoder	Scratch RAM	Static RAM	ROM
worst	global variables	24	2 270	4 942
worst	Coder_Tetra	910		
worst	Autocorr	261		
	Lag_Window	3		
	Levin_32	59		
	Az_Lsp	30		
	Chebps	10		
	Clsp_334	9		
	Int_Lpc4	17		
|______	Lsp_Az	28		
|______	Get_Lsp_Pol	6		
	Pond_Ai	1		
	Residu	4		
	Syn_Filt	86		
|______	Pitch_Ol_Dec	131		
|______	Lag_Max	13		
|______	Inv_sqrt	8		
|______	Pitch_Fr	52		
|______	Norm_Corr	89		
| |______	Convolve	4		
| |______	Inv_sqrt	8		
|______	Inter8_M1_3	3		8
|______	Inter8_1_3	3		8
|______	Pred_Lt	1		
|______	Inter32_M1_3	3		32
|______	Inter32_1_3	3		32
	G_Pitch	10		
	Cal_Rr2	71		
	Back_Fil	126		
	D4i60_16	48		
	G_Code	10		
|______	Ener_Qua	24		
|______	Lpc_Gain	64		
| |______	Syn_Filt	86		
|______	Log2	8		
|______	pow2	8		
Worst case	total in Word16	1 195	2 270	5 022
Worst case	total in kbytes	2,39	4,54	10,044
NOTE: ROM for speech encoder is already stored in speech decoder.				

E.2.2.2 TETRA speech decoder

Table E.36: TETRA speech decoder memory usage breakdown

<table>
<tr><th>Path</th><th>Decoder</th><th>Scratch RAM</th><th>Static RAM</th><th>ROM</th></tr>
<tr><td>worst</td><td>global variables</td><td>264</td><td>463</td><td>4 556</td></tr>
<tr><td>worst</td><td>Decod_tetra</td><td>272</td><td></td><td></td></tr>
<tr><td></td><td>D_Lsp_334</td><td>3</td><td></td><td></td></tr>
<tr><td></td><td>Int_Lpc4</td><td>17</td><td></td><td></td></tr>
<tr><td>|______</td><td>Lsp_Az</td><td>28</td><td></td><td></td></tr>
<tr><td>|______</td><td>Get_Lsp_Pol</td><td>6</td><td></td><td></td></tr>
<tr><td>|______</td><td>Pred_Lt</td><td>1</td><td></td><td></td></tr>
<tr><td>|______</td><td>Inter32_M1_3</td><td>3</td><td></td><td>32</td></tr>
<tr><td>|______</td><td>Inter32_1_3</td><td>3</td><td></td><td>32</td></tr>
<tr><td></td><td>Pond_Ai</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td>Syn_Filt</td><td>86</td><td></td><td></td></tr>
<tr><td></td><td>D_D4i60</td><td>11</td><td></td><td></td></tr>
<tr><td></td><td>G_Code</td><td>10</td><td></td><td></td></tr>
<tr><td>|______ w</td><td>Dec_Ener</td><td>14</td><td>2</td><td></td></tr>
<tr><td>|______ w</td><td>Lpc_Gain</td><td>64</td><td></td><td></td></tr>
<tr><td>| |______w</td><td>Syn_Filt</td><td>86</td><td></td><td></td></tr>
<tr><td>|______</td><td>Log2</td><td>8</td><td></td><td></td></tr>
<tr><td>|______</td><td>pow2</td><td>8</td><td></td><td></td></tr>
<tr><td>Worst case</td><td>total in Word16</td><td>700</td><td>465</td><td>4 620</td></tr>
<tr><td>Worst case</td><td>total in kbytes</td><td>1,4</td><td>0,93</td><td>9,24</td></tr>
</table>

E.2.2.3 TETRA speech channel encoder

Table E.37: TETRA speech channel encoder memory usage breakdown

Path	Encoder	Scratch RAM	Static RAM	ROM
worst	global variables			513
worst	Channel_Encoding	718	706	
	Build_Sensitivity_Classes	1		
	Transform_Class_0	1		
	Build_Crc	2		
worst	Rcpc_Coding	7		
	Interleaving_Speech	2		
Worst case	total in Word16	725	706	513
Worst case	total in kbytes	1,45	1,412	1,026
NOTE: ROM for both channel encoder and decoder is stored only once.				

E.2.2.4 TETRA speech channel decoder

Table E.38: TETRA speech channel decoder memory usage breakdown

Path	Decoder	Scratch RAM	Static RAM	ROM
worst	global variables			513
worst	Channel_Decoding	718	718	
	Desinterleaving_Speech	2		
worst	Rcpc_Decoding	19		
	Bfi	3		
	Untransform_Class_0	1		
	Unbuild_Sensitivity_Classes	1		
Worst case	total in Word16	737	718	513
Worst case	total in kbytes	1,474	1,436	1,026

Annex F (informative): Description of attached computer files

F.1 Directory C-WORD

This directory contains a hard copy of the C code files associated with this ETS. It has been prepared by ETSI Project Team (PT) 29 and is written in Microsoft ™ Word for Windows 2 format. The master document file is named C_WORD.WIN and will automatically read in a number of sub-documents. These documents have not been edited by the ETSI Secretariat.

The master document may be printed, but to preserve the page referencing should only be printed on a HP Laserjet III D, or equivalent, on A4 paper.

F.2 Directory C-CODE

This directory contains the C code files as described in clause 8, and listed in the indexes provided in annex B.

History

Document history			
December 1996	First Edition		
September 1997	One-step Approval Procedure	OAP 9803:	1997-09-19 to 1998-01-16