

IEEE Standard for Advanced Mobile Speech and Audio

IEEE Computer Society

Sponsored by the
Standards Activities Board

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1857.5™-2015

IEEE Standard for Advanced Mobile Speech and Audio

Sponsor

Standards Activities Board
of the
IEEE Computer Society

Approved 5 December 2015

IEEE-SA Standards Board

Recognized as an American National Standard

Abstract: A set of tools to support specific audio coding functions including ACELP/TVC compression, bandwidth extension, stereo coding, frame erasure concealment, etc. is defined in this standard. The tool set defined in this standard provides high quality and efficient coding tool sets for compression, decompression, processing, and representation of speech, music or mixed audio signals to save transmission bandwidth and storage space.

The target applications and services include but not limited to mobile communication, wireless broadband multimedia communications, internet broadband streaming media service and applications.

Keywords: ACELP, bandwidth extension, IEEE 1857.5™, mobile speech and audio, TVC, vector quantization

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2017 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 30 January 2017. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-0821-9 STD20877
Print: ISBN 978-1-5044-0822-6 STDPD20877

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Xplore at <http://ieeexplore.ieee.org/> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this draft standard was submitted to the IEEE-SA Standards Board for approval, the Audio Video Coding Working Group had the following membership:

Wen Gao, *Chair*
Cliff Reader, *Vice Chair*

Haojiang Deng
Weibei Dou
Ge Gao
Ruimin Hu
Tiejun Huang
Lin Jiang

Xun Li
Fuhui Lin
Peilin Liu
Fuwei Ma
Weiping Tu

Xiaochen Wang
Jing Xiao
Yuhong Yang
Jie Zhan
Tao Zhang
Yong Zhang

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Pieter Botman
Juan Carreon
Wang Chen
Xilin Chen
Keith Chow
Weibei Dou
P. Eastman
Ge Gao
Wen Gao
Randall Groves
Zongming Guo
Werner Hoelzl
Ruimin Hu

Noriyuki Ikeuchi
Ming Jiang
Tingting Jiang
Piotr Karocki
Houqiang Li
Weiping Li
Fan Liang
Chong Luo
Siwei Ma
Luntian Mou
Michael Newman
Honggang Qi

Lei Qin
Guangming Shi
Walter Struppler
Jiande Sun
Yonghong Tian
Xushu Wang
Forrest Wright
Feng Wu
Jing Xiao
Ruiqin Xiong
Yuhong Yang
Li Yu
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 5 December 2015, it had the following membership:

John D. Kulick, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Jean-Philippe Faure
J. Travis Griffith
Gary Hoffman
Michael Janezic

Joseph L. Keopfinger
David J. Law
Hung Ling
Andrew Myles
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Annette D. Reilly

Stephen J. Shellhammer
Adrian P. Stephens
Yatin Trivedi
Philip Winston
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1857.5-2015, IEEE Standard for Advanced Mobile Speech and Audio.

This standard provides regular high quality and efficient coding tool sets for compression, decompression, and packaging of the multimedia data to save bandwidth and storage usage.

Contents

1. Overview	10
1.1 Scope	10
1.2 Purpose	10
1.3 Audio-coding tools	10
1.4 Document structure.....	10
2. Normative references.....	11
3. Definitions, symbols, and abbreviations.....	11
3.1 Definitions	11
3.2 Symbols and abbreviations.....	12
4. General overview.....	16
4.1 Input signal	16
4.2 Module composition	16
4.3 Internal sampling frequency and bit rate	16
4.4 Structure of encoder and decoder	17
4.5 Low complexity operation	19
4.6 Frame erasure concealment	19
4.7 Bit allocation.....	20
5. Encoder.....	22
5.1 Input signal pre-processing.....	22
5.2 Hybrid ACELP/TVC encoding mode.....	24
5.3 ACELP/TVC hybrid coding module	37
5.4 Mono signal high-band encoding (BWE).....	93
5.5 Parameters stereo encoding in frequency domain.....	95
5.6 Streaming format	101
6. Decoder	107
6.1 Mono low frequency synthesis	107
6.2 Mono signal high-band synthesis	118
6.3 Stereo low-frequency synthesis	120
6.4 Packet loss concealment	125
6.5 Post processing	129
6.6 Producing output signal	130
7. Detailed bit allocation.....	130
8. Storage and transport interface formats	138
8.1 Encode mode and bit rates	138
8.2 Transport format	139
8.3 Storage format	139
Annex A (normative) Tables of stereo parameters	142
Annex B (informative) Comfortable noise model	147
B.1 Summary	147
B.2 Encoder function.....	147
B.3 Background noise parameters calculation.....	148
B.4 Decoder function.....	149

B.5 Comfort noise generation and update.....	150
Annex C (normative) Voice activity detection.....	151
C.1 Summary.....	151
C.2 Function description.....	151

IEEE Standard for Advanced Mobile Speech and Audio

1. Overview

1.1 Scope

This standard describes a set of speech and audio compression, decompression and packaging tools and mechanism. It is applicable to the following areas:

- Mobile communication
- Wireless broadband multimedia communication
- Internet broadband streaming media business

1.2 Purpose

This standard provides regular low-bit rate speech and audio-coding tool sets for wireless and mobile speech and audio compression and decompression. It provides error masking ability, supports stereo effect, supports wideband audio quality and saves bandwidth for wireless transmission and mobile communication and memory space for storage.

1.3 Audio-coding tools

In this standard, a set of audio coding tools are defined for encoding, transmitting and decoding of recorded speech, music, and other audio signals.

The coding standard describes the detailed mapping from input blocks of monophonic or stereophonic audio samples with sampling rates of 8 kHz, 16 kHz, 24 kHz, 32 kHz, 48 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz in 16 bit uniform PCM format to encode blocks and from encoded blocks to output blocks of reconstructed monophonic or stereophonic audio samples. The available mono bitrates are from 10.4 kb/s to 24 kb/s and stereo bitrates are from 12.4 kb/s to 32 kb/s.

1.4 Document structure

The structure of this document is as follows:

- Clause 1: Overview
- Clause 2: Normative references (provisions of this standard through reference in the text)
- Clause 3: Terms, definitions, and notation
- Clause 4: Outline description
- Clause 5: Encoder
- Clause 6: Decoder
- Clause 7: Detailed bit allocation
- Clause 8: Storage and transport Interface formats
- Annex A: (Normative) Table of stereo parameters
- Annex B: (Informative) Comfortable noise model
- Annex C: (Informative) Voice activity detection

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ISO/IEC 2382-1:1993, Information technology—Vocabulary—Part 1: Fundamental terms.¹

ISO/IEC 2382-4:1987, Information technology —Vocabulary—Part 4: Organization of data.

ISO/IEC 2382-9:1995, Information technology—Vocabulary—Part 9: Data communication.

3. Definitions, symbols, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.² The terms specified in ISO/IEC 2382-1:1993,³ ISO/IEC 2382-4:1987, and ISO/IEC 2382-9:1995 also apply.

adaptive codebook: Adaptive codebook includes the excitation vector with each sub-frame updated once, lag value is the serial number of adaptive codebook.

audio buffer: Storage unit in decoder, used for storing compressed audio data.

bit rate: The rate of compressed bit stream transmitting to the input end of decoder.

¹ ISO/IEC publications are available from the ISO Central Secretariat (<http://www.iso.org/>). ISO publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

² *IEEE Standards Dictionary Online* subscription is available at: <http://ieeexplore.ieee.org/xpls/dictionary.jsp>.

³ Information on references can be found in Clause 2.

bit stream: Is a group of bits with a certain order, used as data coded representation.

closed-loop pitch search: The process of estimating the optimal pitch lag under closed-loop in the vicinity of the pitch lag obtained from open-loop pitch search.

coding: The process of reading in audio sample and generating coded bit stream.

decoder: The entity completing decoding processing.

decoding: The process of reading in coded bit stream and generating audio signal sample.

encoder: The entity completing coding processing.

filter bank: A group of band pass filters covering the entire audio frequency range.

fixed codebook: A kind of algebraic codebook with the fixed pulse amplitude of +1 or -1.

frequency-time transform: The transform of audio signal from frequency domain to time domain.

open-loop pitch search: The process of estimating the optimal pitch lag directly from the weighted input speech. Open-loop pitch search simplifies pitch lag analysis, and limits the closed-loop pitch search to the vicinity of the lag value of open-loop pitch search. In coding, search the open-loop pitch once for every two sub-frames.

perceptual weighting filter: Reduce the subjectively perceived noise by using the noise masking feature at the formant.

reserved: Define some specific syntax element values for future extension of this part.

sampling frequency: The number of times of sampling per unit time in the process of audio signal digitalization.

scale factor: The factor scaling a set of values before quantization.

sound activity detection: Sound signal classification refers to the classification of the input sound signal (i.e. unwanted signal, speech and music).

spectral coefficient: Analyze the discrete frequency domain data output from filter bank.

spectral envelope: Amplitude spectrum contour after transforming to frequency domain.

steady signal: In a frame of audio signal, the signal that shall be analyzed by using long window sequence due to the gradual changes in signal energy and other characteristics.

time-frequency transform: The transform of audio signal from time domain to frequency domain.

transient signal: In a frame of audio signal, the signal that shall be analyzed by using short window sequence due to the rapid changes in signal energy and other characteristics.

3.2 Symbols and abbreviations

Mathematical operators and precedence used in this document are similar to those used in C language. However, it conducts specific definitions for integer division and arithmetic shift operation. Unless otherwise stated, the agreed numbering and counting shall start from '0'.

3.2.1 Arithmetic operators

+	Plus
−	Minus (binary operator or negative (unary prefix operator
×	Multiply
− −	Decrement
a^b	Power, representing b^{th} power of a , also representing a superscript
$\wedge$	Power
/	Divisible operations, truncate along the value direction of 0 (e.g., $7/4$ and $-7/-4$ shall be truncated to 1, $-7/4$ and $7/-4$ shall be truncated to -1 .
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from zero unless otherwise specified (e.g., $3//2$ is rounded to 2, and $-3//2$ is rounded to -2
$\frac{a}{b}$	Division, without truncating or rounding. Division, without rounding or half adjust.
$ x $	Absolute value, when $x \geq 0$, $ x = x$, when $x < 0$, $ x = -x$.
sign()	Take sign $sign(x) = \begin{cases} 1; x \geq 0 \\ -1; x < 0 \end{cases}$
min (x, y)	Take the maximum value $\max(x,y) = \begin{cases} x; x \geq y \\ y; x < y \end{cases}$
max(x,y)	Take the maximum value $\max(x,y) = \begin{cases} x; x \geq y \\ y; x < y \end{cases}$
$\sqrt{\quad}$	Square root
$\sum_{i=a}^b f(i)$	Cumulative sum of the function $f(i)$ when the independent variable i takes all the integer values for a to b (including b)
$\log_{10}$	Denary logarithm
$\log_2$	Binary logarithm
sin()	Sine function
cos()	Cosine function
arccos()	Inverse function of cosine function

$\exp()$	Exponent
$\prod_{i=1}^n a_i$	Continued multiplication
$\otimes$	Convolution
C_a^b	The number of permutations and combinations
$\cup$	Union
$\subset$	A proper subset
$\in$	An element

3.2.2 Logic operators

$\parallel$	Logical OR
$\&\&$	Logical AND
$!$	Logical NOT

3.2.3 Relational operators

$>$	Greater than
$\geq$	Greater than or equal to
$<$	Less than
$\leq$	Less than or equal to
$=$	Equal to
$\neq$	Not equal to

3.2.4 Bitwise operators

$\&$	AND
$ $	OR
$-$	Neg
$a \gg b$	Move a expressed in the form of two's complement integer rightward for b places. This operation is defined only when b takes a positive number.
$a \ll b$	Move a expressed in the form of two's complement integer leftward for b places. This operation is defined only when b takes a positive number.

3.2.5 Assignment

=	Assignment operators
$x = a \dots b$	x takes the value from a to b (including a and b), of which x , a , and b are integers
++	Automatic plus, $x++$ is equivalent to $x = x + 1$. When it is used in array subscript, first calculate the variable value before automatic plus operation
--	Automatic minus, $x--$ is equivalent to $x = x - 1$. When it is used in array subscript, first calculate the variable value before automatic minus operation

Example 1: $x-- = 3$ is equivalent to $x = x - 3$.

Example 2: $x-- = (-3)$ is equivalent to $x = x - (-3)$

3.2.6 Constants

π	3.14159265358...
e	2.71828182845...

3.2.7 Abbreviations

ACELP	algebraic code excited linear prediction
ACELP/TVC	ACELP or TVC
AVS	audio and video coding standard
BWE	bandwidth extension
DTX	discontinuous transmission
FFT	Fast Fourier Transform
FIR	finite impulse response
IFFT	Inverse Fast Fourier Transform
ISF	immittance spectral frequencies
ISP	immittance spectral pairs
ISPP	interleaved single-pulse permutation
LP	linear predictive
LPC	linear predictive coding
LTP	long-term potentiation
MA	moving average
PCM	pulse code modulation
PSVQ	prediction split vector quantization
SAD	sound activity detection
SID	silence insertion descriptor
SNR	signal to noise ratio
TVC	transform vector coding
VAD	speech activity detection
VQ	vector quantization

4. General overview

4.1 Input signal

Mono or stereo input signal of encoder is 16-bit sampling PCM data. It decodes and outputs mono or stereo signal of the same format supports stereo decoding mono output.

4.2 Module composition

The codec defined in this part includes a series of required or optional modules. Required modules are necessary in any realization; optional modules could be omitted in some realizations. The modules included in codec and their properties are shown in Table 1.

Table 1—List of mobile speech and audio standard modules

Module name	Property
Sampling frequency conversion filter	Required
ISP coefficient quantization	Required
Adaptive codebook search	Required
Perceptual weighting filter	Required
Fixed codebook search and coding	Required
Adaptive and fixed codebook gain quantization	Required
Spectrum spread	Required
Stereo coding	Required
Multirate split table vector quantization	Required
Coding mode switching	Optional
Peak value pre-shaping/inverse shaping	Optional
Open-loop pitch analysis	Optional
Sound classifier	Optional
Decoding post processing filter	Optional
Frame loss treatment	Optional

4.3 Internal sampling frequency and bit rate

After the signal is inputted into encoder, its sampling frequency is converted to the internal sampling frequency F_s . Internal sampling rate is obtained through sampling rate conversion filter with frequency range 12.8 kHz to 38.4 kHz. Codec divides the 2048 sampling points processed each time into 2 superframes of 1024 points, respectively corresponding to low frequency and high frequency. Each superframe is then divided into 4 frames of 256 sampling points.

Low and high frequencies of input signal are coded and decoded with different methods. Low frequency is coded and decoded by a core codec module switching between ACELP/TVC. High frequency is coded and decoded by BWE module.

In the case of mono, coding bit rates including low frequency and high frequency coding information are shown in Table 2.

Table 2—Basic bit rates at mono mode

Mono rate(bits/frame)
208
240
272
304
338
384
416
480

In the case of stereo, coding bit rates including low frequency and high frequency coding information are shown in Table 3.

Table 3—Extended bit rates at stereo mode

Stereo extension rates(bits/frame)
40
48
56
64
72
80
88
96
104
112
120
128
136
144
152
160

4.4 Structure of encoder and decoder

Structure of the encoder defined in this part is shown in Figure 1. When coding, the input signal is first divided into low frequency and high frequency signal, both sampling frequency is $F_s/2$. Low-frequency signal is encoded by the core coding module switching between ACELP and TVC. High-frequency signal is encoded by BWE module.

Mode selection bit, low frequency parameter and high frequency parameters are transmitted to the decoder. Each superframe is divided into 4 frames of the same size to be processed. When the input signal is stereo signal, the encoder down mixes the left and right channel signals into one sum signal and one difference signal, sum signal is encoded by ACELP/TVC coding module; stereo coding module processes the sum signal and difference signal.

Structure of the decoder defined in this part is shown in Figure 2. The decoder will respectively decode the low and high frequencies, and then use a synthesis filter to combine the signals of the two frequency bands. If the signal output is limited to be mono, then do not use stereo decoding.

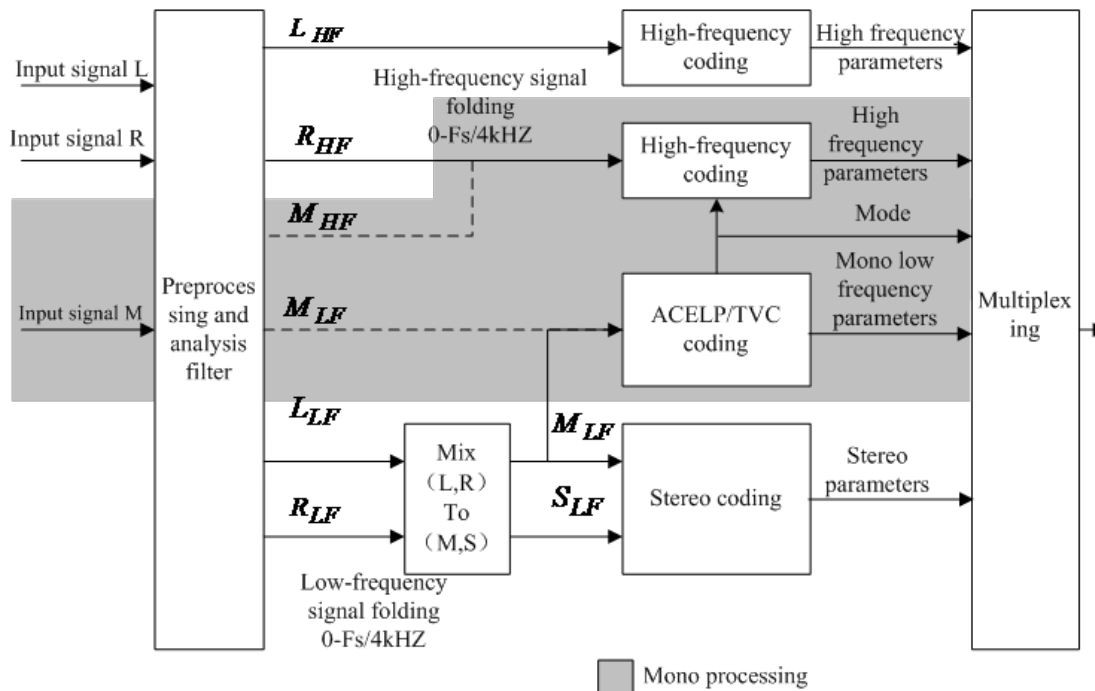


Figure 1—Structure of encoder

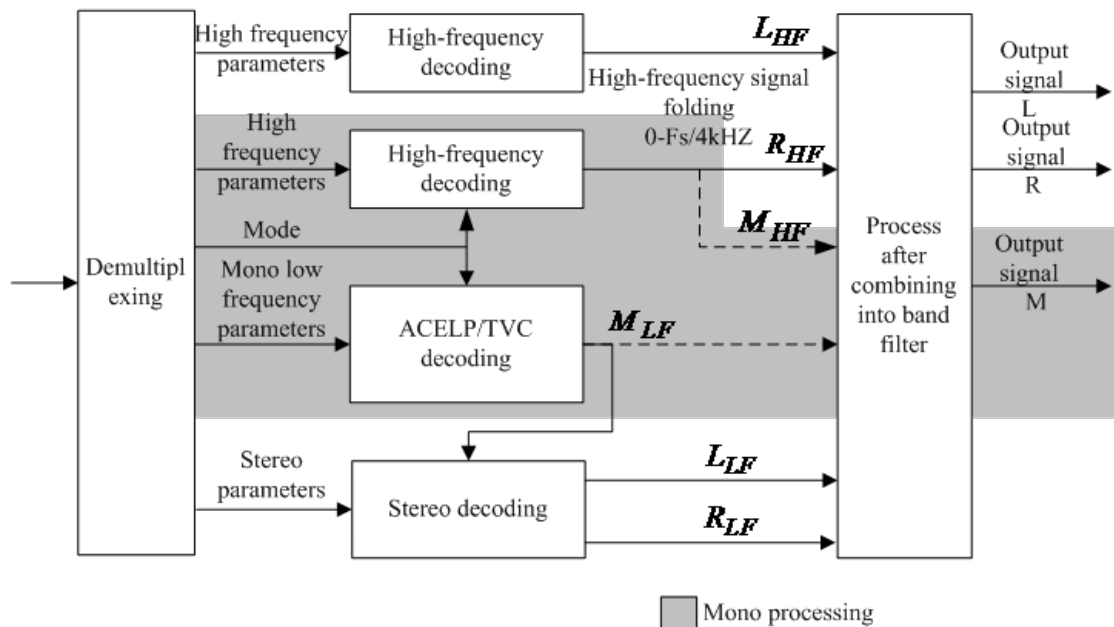


Figure 2—Structure decoder

4.4.1 LP analysis and synthesis of low frequency

Low-frequency coding adopts LP analysis, LP coefficient makes one linear interpolation with every 64 points as one sub-frame, LP analysis window is semi-cosine window of 384 points.

4.4.2 ACELP/TVC module

Mono low-frequency signal (0 to $F_s/4$ kHz) is encoded by ACELP or TVC. Selection of the coding mode is based on the closed-loop analysis–synthesis method. Only the frames with the frame size of 256 sampling points adopt ACELP mode, whilst frames with the frame size of 512 and 1024 sampling points can only adopt TVC coding mode.

ACELP coding includes LP analysis, adaptive codebook search, fixed codebook search etc. At TVC mode, perceptually weighted signal is encoded at the transform domain, and then the signal adopts split multi-bit-rate lattice vector quantization after FFT. The transform window length includes 256, 512, and 1024. The reconstructed excitation signal could be obtained by inverse filtering the quantized signal High-frequency coding.

High-frequency coding adopts BWE method. High-frequency signal is encoded by using the method of only transmitting spectral envelope and gain. The left and right channel high-frequency signals of the left and right channels of stereo are encoded by BWE.

4.4.3 Stereo coding

Stereo coding adopts frequency band decomposition method similar to mono. The left and right channels are divided into low frequency and high frequency. Low frequency signals of the two channels are downmixed into the low frequency signals (0 to $F_s/4$ kHz) of sum signals and difference signals. Low frequency signals of sum signals are encoded by ACELP/TVC module. Low frequency signals of sum signals and difference signals are further decomposed into ultra-low frequency signals (0 to $F_s \times (5/128)$ kHz and intermediate frequency signals ($F_s \times (5/128)$ kHz to $F_s/4$ kHz). Conduct downsampling for the ultra-low frequency signals of the difference signals, calculate the side information, conduct TVC of 40, 80, or 160 sampling points. Conduct parametric coding for the intermediate frequency signals of the difference signals referring to the intermediate frequency signals of sum signals. The decoder uses these parameters to recover the intermediate frequency signals of the two signals from mono excitation signal. High-frequency signals of the left and right channels are encoded by BWE.

4.5 Low complexity operation

In the low complexity operation, the decision on the usage of ACELP and TVC mode is done in an open-loop manner.

4.6 Frame erasure concealment

When missing packets occur at the receiver, the decoder applies concealment. The concealment algorithm depends on the mode of the correctly received packets preceding and following the missing packet. Concealment uses either time-domain coefficient extrapolation, or frequency-domain interpolation for some of the TVC modes.

4.7 Bit allocation

4.7.1 Bit allocation for ACELP/TVC encoding

The low-frequency part (0~fs/4 kHz) of ACELP/TVC encoding contains 4 coding types ACELP256, TVC256, TVC512, and TVC1024. Two bits are used to identify the frame coding type for each 256-sample frame, where ACELP256 denotes use ACELP to encode 256 sample frame, TVC256 denotes use TVC to encode 256 sample frame, TVC512 denotes use TVC to encode 512 sample frame, TVC1024 denotes use TVC to encode 2014 sample frame.

The bit allocation for the four coding types is shown in Table 4 to Table 7.

Table 4—Bit allocation for ACELP256 coding type (including BWE)

Parameter	Number of bits							
Mode bits	2							
ISF Parameters	46							
Mean Energy	2							
Pitch Lag	30							
Pitch Filter	4 × 1							
Fixed-codebook Indices	4 × 20	4 × 28	4 × 36	4 × 44	4 × 52	4 × 64	4 × 72	4 × 88
Codebook Gains	4 × 7							
HF ISF Parameters	9							
HF gain	7							
Total in bits	208	240	272	304	336	384	416	480

Table 5—Bit allocation for TVC256 coding type

Parameter	Number of bits							
Mode bits	2							
ISF Parameters	46							
Noise factor	3							
Global Gain	7							
Algebraic VQ	134	166	198	230	262	310	342	406
HF ISF Parameters	9							
HF gain	7							
Total in bits	208	240	272	304	336	384	416	480

Table 6—Bit allocation for TVC512 coding type

Parameter	Number of bits							
Mode bits	2+2							
ISF Parameters	46							
Noise factor	3							
Global Gain	7							
Gain redundancy	6							
Algebraic VQ	318	382	446	510	574	670	734	862
HF ISF Parameters	9×2							
HF Gain	7×2							
Total in bits	416	480	544	608	672	768	832	960

Table 7—Bit allocation for TVC1024 coding type

Parameter	Number of bits								
Mode bits	2+2								
ISF Parameters	46								
Noise factor	3								
Global Gain	7								
Gain redundancy	6								
Algebraic VQ	318	382	446	510	574	670	734	862	
HF ISF Parameters	9×2								
HF Gain	7×2								
Total in bits	416	480	544	608	672	768	832	960	

4.7.2 Bit allocation for stereo encoder

The bit allocation in stereo encoder is shown in Table 8 for stationary signals and Table 9 for transient signals.

Table 8—Bit allocation for stereo encoder for stationary signal

Parameter	Number of bits															
Signal Type	1															
Global Gain	7															
Blank bit	1															
IF Gain	40															
Winter filter coefficients	7															
Algebraic VQ	40	72	104	136	168	200	232	240	272	304	336	368	400	432	464	496
HF ISF Parameters	9×4															
HF Gain	7×4															
Total in bits	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576

Table 9—Bit allocation for stereo encoder for transient signal

Parameter	Number of bits															
Signal Type	1															
Global Gain	7×4															
Blank bit	1+1+1															
IF Gain	40															
Winter filter coefficients	4×4								7×4							
HF ISF Parameters	9×4															
HF Gain	7×4															
Algebraic VQ	8	40	72	104	136	168	200	220	252	284	316	348	380	412	444	476
Total in bits	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576

5. Encoder

5.1 Input signal pre-processing

Input signals are pre-processed in order to bring them to the internal sampling frequency of the encoder. The signal is up-sampled, filtered by a low pass filter and then down-sampled.

5.1.1 Sampling frequency conversion filter

The sampling frequency conversion filter is shown in Figure 3. This scheme supports audio input with sampling frequency of 8 kHz, 16 kHz, 24 kHz, 32 kHz, 48 kHz, 11 kHz, 22 kHz, and 44.1 kHz. For the consistency of the subsequent processing, it's necessary to resample the input signals into the internal sampling frequency. Similarly, sampling frequency converter is also used in the post-processing section for the decoder.

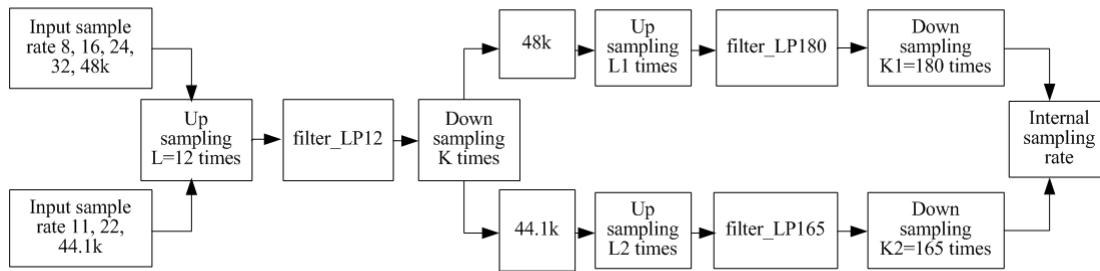


Figure 3—Sampling frequency converter

The weighted window function is shown Equation (1).

$$w(n) = h(n) \times w_{\text{hamming}}(n) \quad (1)$$

where $w_{\text{hamming}}(n)$ is the Common Hamming window, and it is shown in Equation (2).

$$w_{\text{hamming}}(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), n = 0, 1, \dots, N-1 \quad (2)$$

And $h(n)$ is the weighting function, which is described in Equation (3)

$$h(n) = \frac{1 + e^{-1} + e^{-(N-n)} + e^{-(n-1)}}{1 + e^{-1} - 2 \cdot e^{-\left(\frac{N+1}{2}\right)}} \quad (3)$$

5.1.2 High-pass filtering

The lower band signals are high pass filtered. The high-pass filter serves as a precaution against undesired low frequency components. A high pass filter is used, and it is given by Equation (4), where the filter parameters, such as a_1 , a_2 , b_0 , b_1 , b_2 , are dependent on the internal sampling rate.

$$H_{hl}(z) = \frac{b_0 - b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} + a_2 z^{-2}} \quad (4)$$

The corresponding relationship between internal sampling rate and high-pass filter parameters is shown in Table 10.

Table 10—Internal sampling rate and high-pass filter parameters

Internal sampling rate (Hz)	a_1	a_2	b_0	b_1	b_2
12800	1.979174	−0.979342	0.989629	−1.979258	0.989629
14400	1.979174	−0.979342	0.989629	−1.979258	0.989629
16000	1.979174	−0.979342	0.989629	−1.979258	0.989629
17067	1.979174	−0.979342	0.989629	−1.979258	0.989629
19200	1.979174	−0.979342	0.989629	−1.979258	0.989629
21333	1.979174	−0.979342	0.989629	−1.979258	0.989629
24000	1.979174	−0.979342	0.989629	−1.979258	0.989629
25600	1.979174	−0.979342	0.989629	−1.979258	0.989629
28800	1.986116	−0.986212	0.993082	−1.986164	0.993082
32000	1.986984	−0.987071	0.993514	−1.987027	0.993514
34133	1.987852	−0.987929	0.993945	−1.987891	0.993945
36000	1.988430	−0.988502	0.994233	−1.988466	0.994233
38400	1.988936	−0.989003	0.994485	−1.988970	0.994485
44100	1.989587	−0.989647	0.994809	−1.989617	0.994809

5.1.3 Stereo signal downmixing

When the input audio signal is stereo, the lower band mono signal is obtained by downmixing the left and right channels according to Equation (5).

$$x_{ML}(n) = 0.5(x_{LL}(n) + x_{RL}(n)) \quad (5)$$

The $x_{LL}(n)$, respectively $x_{RL}(n)$, is the lower band signal from the left, respectively right, channels. The lower band mono signal is supplied to the core low band encoder for TVC/ACELP encoding.

For stereo encoding, the obtained downmixed mono signal $x_{ML}(n)$, respectively the right channel signal $x_{RL}(n)$, will be used to get the residual signal $e_{ML}(n)$, respectively $e_{RL}(n)$ through linear filtering. And then by using time-frequency conversion they will further be split into two bands: a critically sampled low frequency band and an intermediate frequency band. After adding and subtracting the intermediate frequency (IF) bands between each other, the corresponding IF part of the left channel $e_{LMid}(n)$ and the right channel $e_{RMid}(n)$ are restored. The process is shown in the Figure 4.

The critically sampled low band output signals $e_{SLo}(n)$ is fed to the stereo low band encoder, while the signals $e_{LMid}(n)$ and $e_{RMid}(n)$ to the stereo mid band encoder.

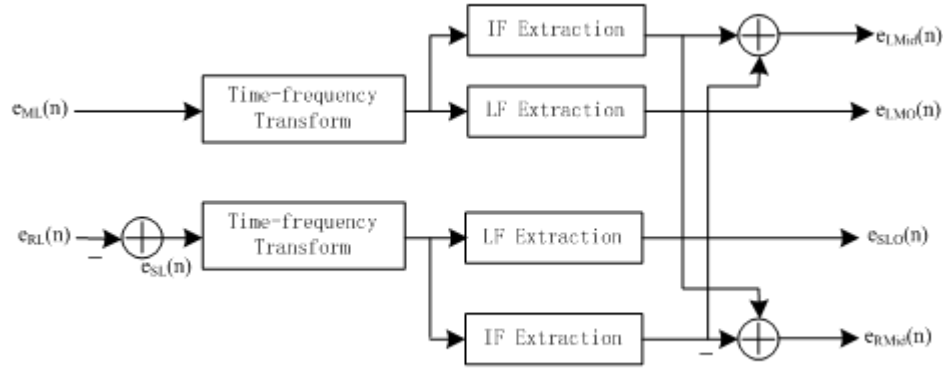


Figure 4—Downmixing process of stereo dual channel

5.2 Hybrid ACELP/TVC encoding mode

5.2.1 Outline description

The encoding algorithm at the core encoder is based on a hybrid ACELP/TVC model. For every block of input signal, the encoder decides (either in open-loop or closed-loop) which encoding model (ACELP or TVC) is best. The ACELP mode is a time-domain predictive encoder, and best suited for speech and transient signals. Alternatively, the TVC mode is a transform-based encoder, and is more appropriate for typical music samples.

5.2.2 Timing chart of the ACELP and TVC modes

The ACELP/TVC core encoder takes a mono signal as input, at a sampling frequency of $F_s/2$ kHz. This signal is processed in super-frames of 1024 samples in duration. Within each 1024-sample super-frame, several encoding modes are possible, depending on the signal structure. These types are: 256-sample ACELP, 256-sample TVC, 512-sample TVC and 1024-sample TVC. These encoding modes will be described further in 5.2.3.

The timing chart of all possible modes within a 1024-sample superframe is shown in Figure 5. As shown in the figure, each 256-sample frame within a superframe could be into one of four possible modes, which is denoted as ACELP256, TVC256, TVC512, and TVC1024. When in ACELP256 mode, the input signal is encoded with ACELP encoding of 256-sample frame. In TVC256 mode, the frame is encoded using TVC with a 256-sample support, and 32 samples of look-ahead used for overlap and add since TVC is a transform coding approach. The TVC512 mode means that two consecutive 256-sample frames are grouped to be encoded as a single 512-sample block, using TVC with a 512-sample support plus 64 samples look-ahead. Note that the TVC512 mode is only allowed by grouping either the first two 256-sample frames of the superframes, or the last two 256-sample frames. Finally, the TVC1024 mode indicates that all 256-sample frames with the superframe are grouped together to be encoded in a single block using TVC with a 1024-sample support plus 128 samples look-ahead.

5.2.3 ACELP/TVC mode combinations and mode encoding

Every superframe has 26 different ACELP/TVC coding type combinations within a 1024-sample superframe. These are shown in Table 11.

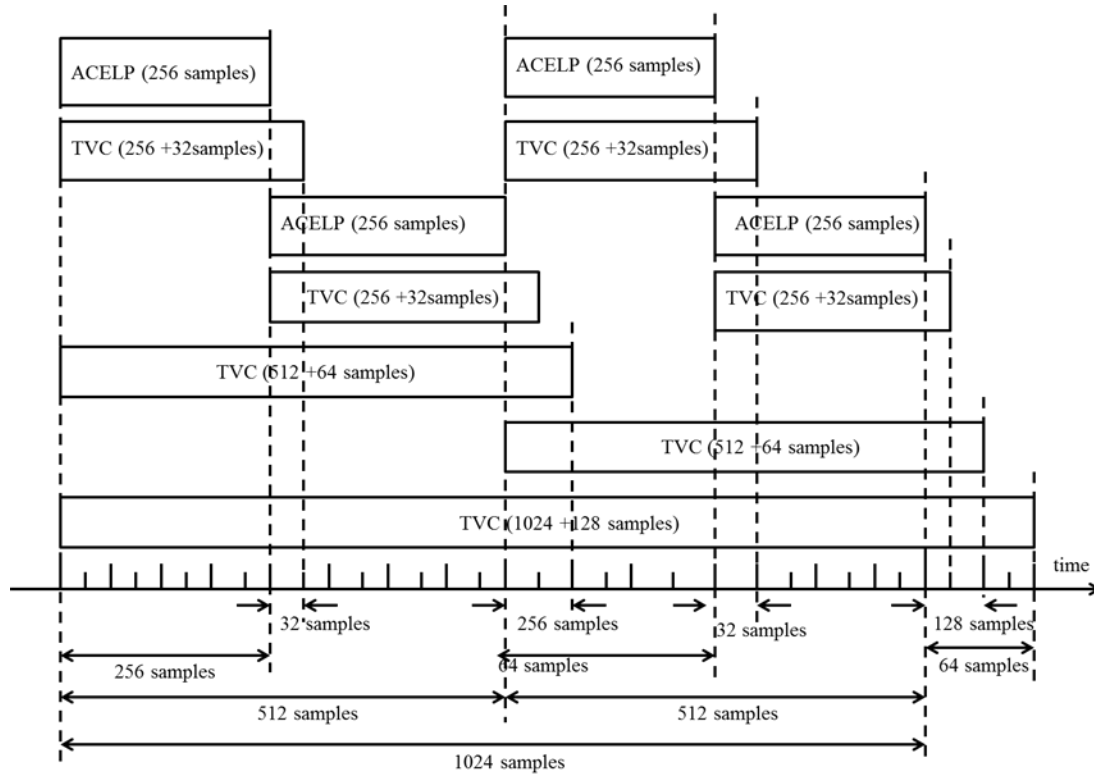


Figure 5—Timing chart of the frame types

Table 11—Possible mode combinations in a superframe

Serial number	(m_0, m_1, m_2, m_3)	Serial number	(m_0, m_1, m_2, m_3)	Serial number	(m_0, m_1, m_2, m_3)	Serial number	(m_0, m_1, m_2, m_3)
1	(0, 0, 0, 0)	9	(0, 0, 0, 1)	17	(2, 2, 0, 0)		
2	(1, 0, 0, 0)	10	(1, 0, 0, 1)	18	(2, 2, 1, 0)		
3	(0, 1, 0, 0)	11	(0, 1, 0, 1)	19	(2, 2, 0, 1)		
4	(1, 1, 0, 0)	12	(1, 1, 0, 1)	20	(2, 2, 1, 1)		
5	(0, 0, 1, 0)	13	(0, 0, 1, 1)	21	(0, 0, 2, 2)		
6	(1, 0, 1, 0)	14	(1, 0, 1, 1)	22	(1, 0, 2, 2)		
7	(0, 1, 1, 0)	15	(0, 1, 1, 1)	23	(0, 1, 2, 2)	25	(2, 2, 2, 2)
8	(1, 1, 1, 0)	16	(1, 1, 1, 1)	24	(1, 1, 2, 2)	26	(3, 3, 3, 3)

Each quadruplet of numbers (m_0, m_1, m_2, m_3) in Table 11 is interpreted as follows: m_k is the mode indication for the k^{th} 256-sample frame in the 1024-sample superframe, where m_k could take the following values:

- $m_k = 0$ means the mode for frame k is 256-sample ACELP
- $m_k = 1$ means the mode for frame k is 256-sample TVC
- $m_k = 2$ means the mode for frame k is 512-sample TVC
- $m_k = 3$ means the mode for frame k is 1024-sample TVC

Obviously, when the first 256-sample frame is in mode "2" (512-sample TVC), the second 256-sample frame should also be in mode 2. Similarly, when the third 256-sample frame is in mode "2" (512-sample TVC), the fourth 256-sample frame should also be in mode 2. And there is only one possible mode configuration including the value "3" (1024-sample TVC), namely all four 256-sample frames are in the same mode ($m_k = 3$ for $k = 0, 1, 2$, and 3).

As discussed above, the parameters for each 1024-sample super-frame are actually decomposed into four frames of identical size. To increase robustness, the mode bits are actually sent as two bits (the values of m_k) in each transmitted frame.

For example, if the super-frame is encoded in a full 1024-sample TVC frame, its bits will be allocated into four packets of equal size and each packet will contain the binary value "11" (mode $m_k = 3$) as mode indicator.

5.2.4 ACELP/TVC closed loop mode selection

The best mode combination out of the 26 possible combinations is determined in closed-loop. This means that the signal in each 256-sample frame within a 1024-sample super-frame has to be encoded in several modes before selecting the best combination.

The left portion of Figure 6 (Trials of encoding mode) shows what encoding mode is applied to each 256-sample frame in 11 successive trials. Note that each 256-sample frame is involved in only four of the 11 encoding trials. When more than 1 frame is involved in a trial (lines 5, 10, and 11 of Figure 6), the TVC of the corresponding length is applied (TVC512 or TVC1024). The right portion of Figure 6 gives an example of mode selection, where the final decision (after Trial 11) is 1024-sample TVC. This would result in sending a value of 3 for the mode in all four packets for this super-frame. Bold numbers in the example at the right of Figure 6 show at what point a mode decision is taken in the intermediate steps of the mode selection process. The final mode decision is only known after Trial 11.

Example: ACELP/TVC closed-loop mode selection process

step	Trials of encoding mode				Example of selection			
	Fr 0	Fr 1	Fr 2	Fr 3	Fr 0	Fr 1	Fr 2	Fr 3
1	ACELP256				ACELP256			
2	TVC256				ACELP256			
3		ACELP256			ACELP256	ACELP256		
4		TVC256			ACELP256	TVC256		
5	TVC512	TVC512			ACELP256	TVC256		
6			ACELP256		ACELP256	TVC256	ACELP256	
7			TVC256		ACELP256	TVC256	TVC256	
8				ACELP256	ACELP256	TVC256	TVC256	ACELP256
9				TVC256	ACELP256	TVC256	TVC256	TVC256
10			TVC512	TVC512	ACELP256	TVC256	TVC512	TVC512
11	TVC1024	TVC1024	TVC1024	TVC1024	TVC1024	TVC1024	TVC1024	TVC1024

Figure 6—Closed loop selection of ACELP/TVC mode combination

The mode selection process shown in Figure 6 proceeds as follows: First, in trials 1 and 2, ACELP256 then 256-sample TVC encoding are tried in the first 256-sample frame (Fr0). Then, a mode selection is made for Fr0 between these two modes. The selection criterion is the average segmental signal to noise ratio (SNR) between the weighted speech $x_w(n)$ and synthesized weighted speech $\hat{x}_w(n)$. The segmental SNR in sub-frame i is defined as shown in Equation (6).

$$segSNR_i = 20 \log_{10} \left(\frac{\sum_{n=0}^{N-1} x_w^2(n)}{\sum_{n=0}^{N-1} (x_w(n) - \hat{x}_w(n))^2} \right) \quad (6)$$

The N is the length of the sub-frame (equivalent to a 64-sample sub-frame in the encoder). Then, the average segmental SNR is defined as shown in Equation (7).

$$\overline{segSNR} = \frac{1}{N_{SF}} \sum_{i=0}^{N_{SF}-1} segSNR_i \quad (7)$$

The N_{SF} is the number of sub-frames in the frame. Since a frame may be 256, 512 or 1024 samples in length, N_{SF} may be 4, 8 or 16. In the example given in Figure 6, we assume that, according to the $segSNR$ decision criterion, mode ACELP256 was retained over TVC256. Then, in trials 3 and 4, the same mode comparison is made for Fr1 between ACELP and 256-sample TVC. Here, we assume that 256-sample TVC was better than ACELP256, based again on the segmental SNR measure described above. This choice is indicated in bold on line 4 of the example at the right of Figure 6. Then, in Trial 5, Fr0 and Fr1 are grouped together to form a 512-sample TVC. The algorithm now has to choose between 512-sample TVC for the first 2 frames, compared to ACELP in the first frame and TVC256 in the second frame. In this example, on line 5 in bold, the sequence ACELP-TVC256 was selected over TVC-512 according to the segmental SNR criterion.

The same procedure as Trial 1 to Trial 5 is then applied to the third and fourth frames (Fr2 and Fr3), in Trial 6 to Trial 10. After Trial 10, in the example of Figure 6, the four 256-sample frames are classified as: ACELP for F0, then TVC256 for F1, then TVC512 for F2 and F3 grouped together. A last trial (line 11) is then performed where all four 256-sample frames (the whole super-frame) are encoded with 1024-sample TVC. Using the segmental SNR criterion, again with 64-sample segments, this is compared with the signal encoded using the mode selection in Trial 10. In this example, the final mode decision is 1024 sample TVC for the whole frame. The mode bits for each 256-sample frame would then be (3, 3, 3, and 3).

5.2.5 ACELP/TVC open loop mode selection

The alternative method for ACELP/TVC mode selection is the low complexity open-loop method. The encoding mode selection is determined according to the extracting features of input signal. The procedure of choosing encoding mode with the signal features is described in 5.2.6.

5.2.6 Signal classifier

5.2.6.1 Algorithm

Sound activity detection (SAD) is a technique used to classify audio signals by noise, speech, and music. SAD consists of 4 modules: background noise estimation control (BNEC), initial classification, feature parameter extraction and signal type decision. The block diagram of SAD is shown in Figure 7.

SAD classify the signal with the features calculated by encoder, including the immittance spectral frequency $isf[m]$ ($m=1, 2, \dots, 16$), sub-band energy $level[n]$ ($n=1, 2, \dots, 12$), and the open-loop pitch parameter (open-loop pitch lag and open-loop pitch gain). The output of SAD is the type of the signal: noise, speech and music. Among them, speech and music are useful signals.

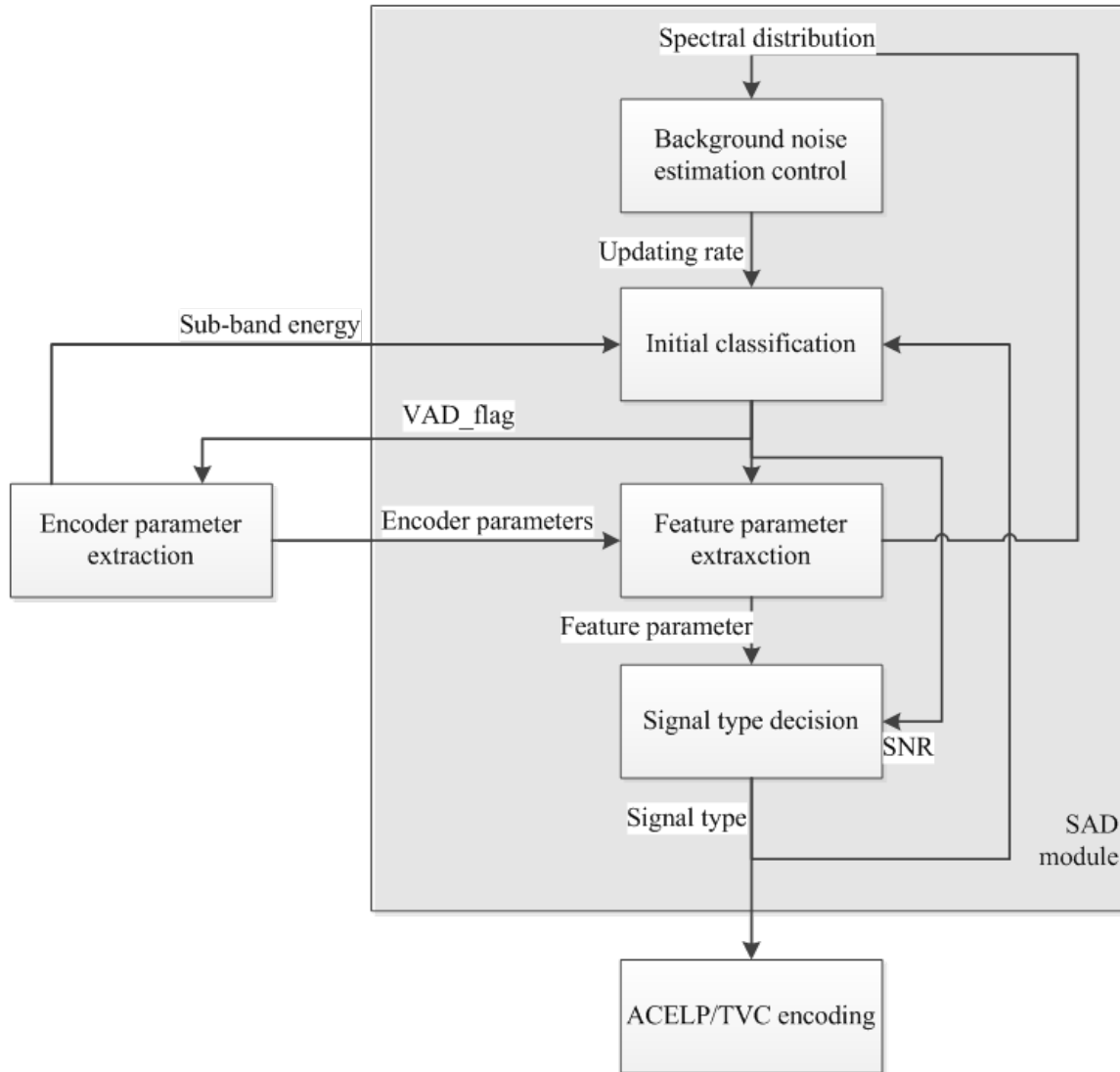


Figure 7—Block diagram of SAD

5.2.6.2 Encoder parameters extraction

To decrease the computational complexity, SAD makes full use of the parameters calculated by other modules of encoder. This module provides ISF, sub-band energy and open-loop pitch parameters to the feature parameter extraction module, and sub-band energy to initial classification module.

These parameters are defined as follows:

$level[i]$ –sub-band energy. The index i takes value from 1 to 12, which represents the band with frequency range of 0 Hz to 200 Hz, 200 Hz to 400 Hz, 400 Hz to 600 Hz, 600 Hz to 800 Hz, 800 Hz to 1200 Hz, 1200 Hz to 1600 Hz, 1600 Hz to 2000 Hz, 2000 Hz to 2400 Hz, 2400 Hz to 3200 Hz, 3200 Hz to 4000 Hz, 4000 Hz to 4800 Hz, 4800 Hz to 6400 Hz;

$isf_n[i]$ –the i th element of isf_n which means the isf coefficients vector of the frame n ;

ol_gain – the open-loop pitch gain;

ol_lag – the open-loop pitch lag;

tone_flag – tone flag.

If *ol_gain* is greater than threshold TONE_THR, *tone_flag* is assigned 1.

The flowchart of encoder parameters extraction is shown in Figure 8. After calculation of subband energy, SAD will decide whether to calculate the LSF coefficient according to the result of initial classification. If the current frame is noise, the process is according to the requirement of encoder. If the encoder needs LSF coefficient of noise frames, the calculation will be done. If not, the module ends. On the other hand, if the current frame is not noise, the LSF coefficient should be calculated.

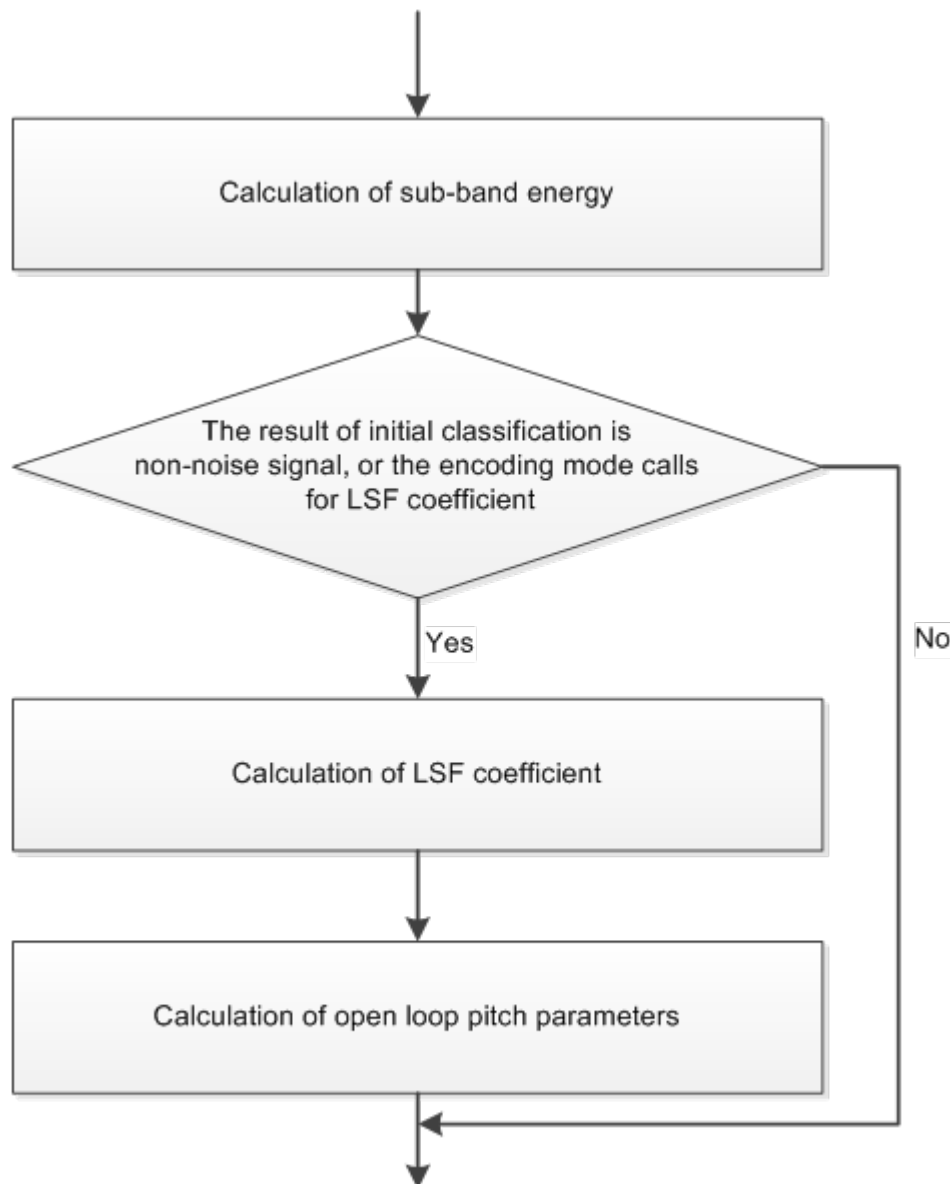


Figure 8—Encoder parameters extraction flowchart

5.2.6.3 Initial classification

Initial classification module classifies audio signal as non-noise and noise signal. The block diagram of initial classification is shown in Figure 9. The initial classification module is based on VAD algorithm. The detailed functional specification of VAD is in Annex C. Firstly, this module updates the noise parameters according to the updating rate received from BNEC. The noise parameters include noise estimation parameters and noise spectral distribution parameter etc. Then the module classifies the input audio signal according to sub-band energy and noise parameters updated. Finally the signal type is sent to the next module.

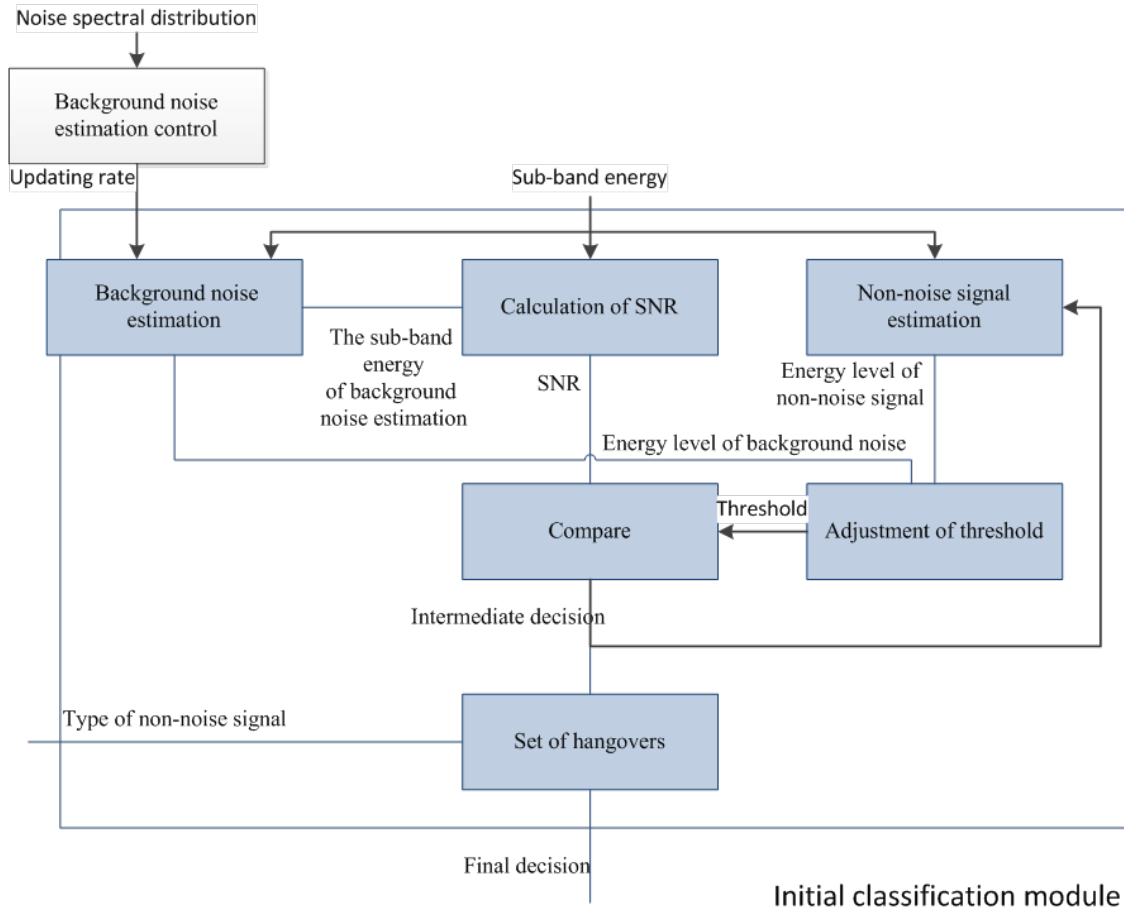


Figure 9—Block diagram of initial classification

If *update_{up}* and *update_{down}* respectively represent updating-up rate and updating-down rate of background noise. The updating algorithm could be described by the following pseudo codes:

```

If (bckr_est[m][n] < level[m-1][n] ) then
    update = update_up
else
    update = update_down
end

```

Noise estimation parameters could be updated by Equation (8).

$$bckr_est_{m+1}[n] = (1 - update) * bckr_est_m[n] + update * level_{m-1}[n] \quad (8)$$

And noise spectral distribution parameters could be updated by Equation (9).

$$\begin{aligned} \tilde{p}_{m+1}[i] &= (1 - \text{update_up} - \text{update_down}) \times \tilde{p}_m[i] + \text{update_up} \times p_m[i] \\ \tilde{q}_i^{(n,n+1)} &= (1 - \frac{i}{8}) \tilde{q}_i^{(n-1)} + \frac{i}{8} \tilde{q}_i^{(n+1)}, i = 0, 1, \dots, 7 \end{aligned} \quad (9)$$

where

- m is the frame index
- n is the sub-band index
- i is the index of spectral distribution parameter vector whose value is from 1 to 4, $bckr_est$ is the sub-band energy of background noise estimation
- $\tilde{p}$ is the estimation of background noise spectral distribution parameter vector, and P is the spectral distribution parameter vector of current signal

Generally, hangovers are used in order to avoid misjudging non-noise signal as noise. The length of hangovers is determined by the compromise between transmission efficiency and signal protection. In this part, the module decides the length of hangovers adaptively according to the result of SAD.

5.2.6.4 Feature parameter extraction

The feature parameter extraction module accepts signal type from initial classification, as well as ISF, sub-band energy and open-loop pitch parameters from encoder parameters extraction. After that, the spectral distribution of signal and background noise is calculated with these parameters and sent to BNEC.

The feature parameters are as follows:

- a) **Pitch:** The open-loop pitch lag is compared with the previous one. If the lag increment is smaller than a certain threshold, the lag count is added by 1. If the lag count of sequential two frames is big enough, the pitch flag is assigned 1. Otherwise the pitch flag is assigned 0.
- b) **Mean open-loop pitch gain:** If the mean gain is greater than a given threshold, the high order flag is assigned 1
- c) **Zero-crossing rate (zcr):** Is described in Equation (10) as follows:

$$zcr = \frac{1}{T} \sum_{i=1}^{T-1} \Pi \{x(i)x(i-1) < 0\} \quad (10)$$

where

$$\Pi \{A\} \quad \text{is 1 if } A \text{ is true, otherwise it is 0.}$$

- d) **Time domain fluctuation of sub-band (t_flux):** It is given in Equation (11) as follows:

$$t_flux = \frac{\sum_{i=1}^{12} |level_m(i) - level_{m-1}(i)|}{short_mean_level_energy} \quad (11)$$

where

$short_mean_level_energy$ is the mean of short-term energy.

- e) Sub-band energy ratio of high-band and low-band (ra): It is given in Equation (12) as follows:

$$ra = \frac{sublevel_high_energy}{sublevel_low_energy} \quad (12)$$

- f) Frequency domain fluctuation of sub-band (f_flux): It is given in Equation (13)

$$f_flux = \frac{\sum_{i=2}^{12} |level_m(i) - level_m(i-1)|}{short_mean_level_energy} \quad (13)$$

- g) **Short-term mean of ISF spectral deviation (Isf_meanSD)**: Calculate the mean value of ISF spectral deviation for each five continuous frames

5.2.6.5 Signal type decision

The signal type decision module accepts signal type from initial classification module as well as feature parameters (open loop pitch parameter, ISF and sub-band energy) from feature parameters extraction module, and then classifies non-noise signal into speech or music type.

The final flags of signal type decision include the following:

- NOISE—unwanted signal
- SPEECH—speech type
- MUSIC—music type

5.2.6.5.1 Feature parameter hangover

In order to make the signal type decision stable and avoid frequent changes of the results, the hangovers are set up as follows:

Hangovers for feature parameter flags are defined such as *pitch_flag*, *level_meanSD_high_flag*, *sf_meanSD_high_flag*, *lsf_meanSD_low_flag*, *level_meanSD_low_flag*, *meangain_flag*. The length of hangovers is controlled according to the error rate (ER) of each internal node of the trained parameter decision tree. The greater the ER is, the longer the corresponding hangover is.

5.2.6.5.2 First classification

The signal type (music or speech) of non-noise signal is determined based on the open-loop pitch parameter, sub-band energy and ISF in the first classification.

Speech detection is done firstly as shown in Figure 10. If the signal accords with the features of speech, it is classified as music and its speech flag is assigned 1.

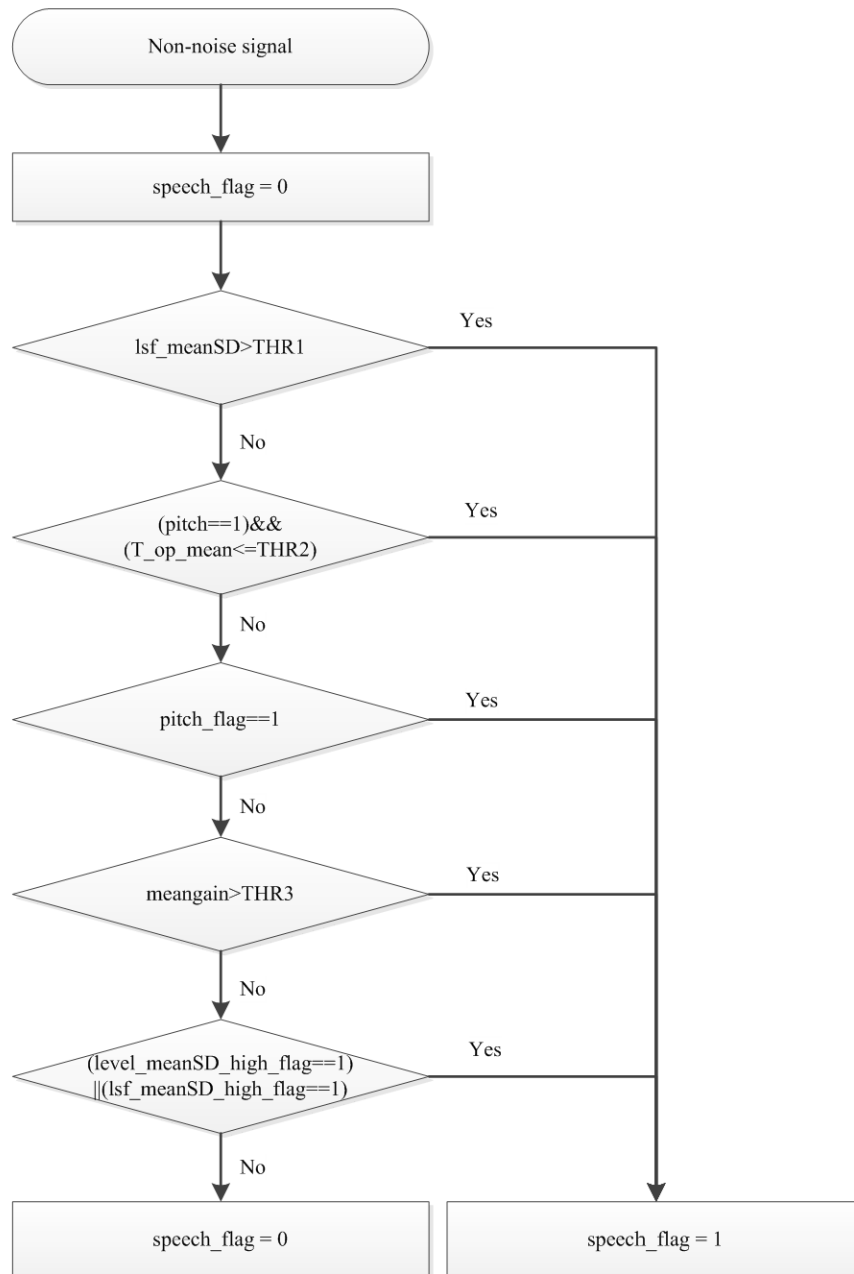


Figure 10—Speech detection

Music detection is done afterwards. If *lsf_meanSD_low_flag* and *level_meanSD_low_flag* are both 1, the signals classified as music and its music flag is assigned 1, as shown in Figure 11. The final type decision is made as shown in Figure 12.

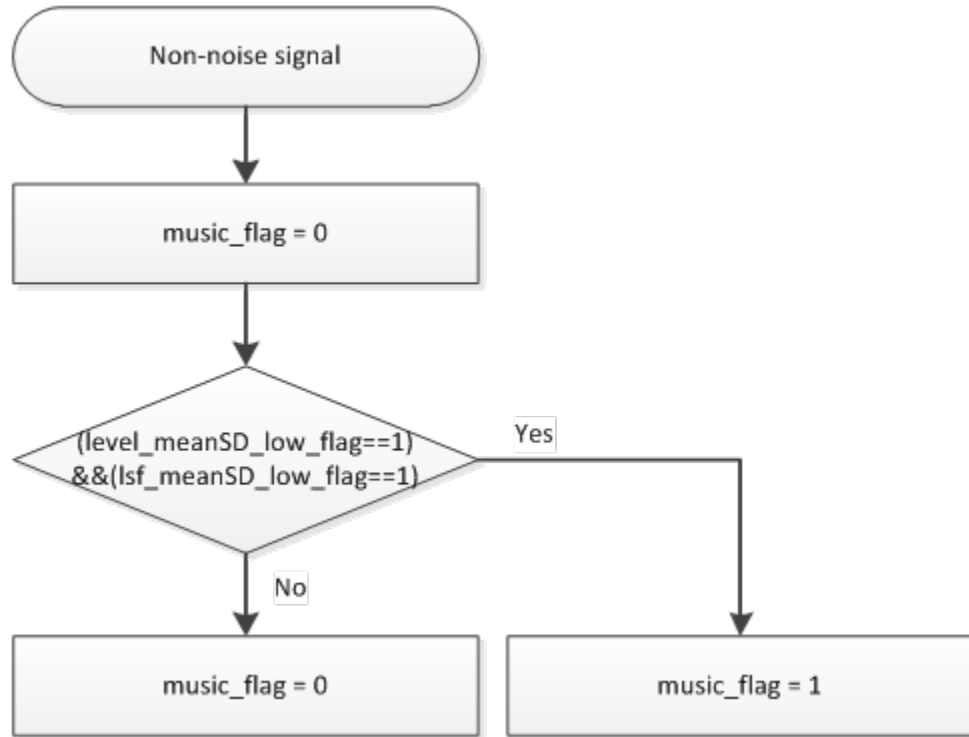


Figure 11 —Music detection

5.2.6.5.3 Classification correction

In classification correction, hangovers of speech and music are initially reset to 0. If the type of current frame is uncertain after first classification, further classification will be done, which is based on the context and some feature parameters. If the previous frames are strongly continuous speech and the feature parameters of current frame are more like speech, *speech_hangover_flag* is set to 1. Otherwise, if the previous frames are strongly continuous music and the feature parameters of current frame are more like music, *music_hangover_flag* is set to 1. So the current frame is classified to speech or music according to whose hangover flag is 1. If both of the hangover flags are 1, the current frame is classified as uncertain type. What's more, if there are more than two music frames before and the value of *lsf_meanSD* is small, the current frame is classified as music.

5.2.6.5.4 Final classification

Classification correction is followed by final classification. If the context is music that last for more than 3 seconds, the type of current frame will be forced to be revised according to the value of *lsf_meanSD*. Similarly, if the context is speech that last for more than 3 s, the type of current frame will be forced to be revised according to the value of *lsf_meanSD*. In addition, the current frame will be assigned the type of its previous frame if the short-term energy is too small.

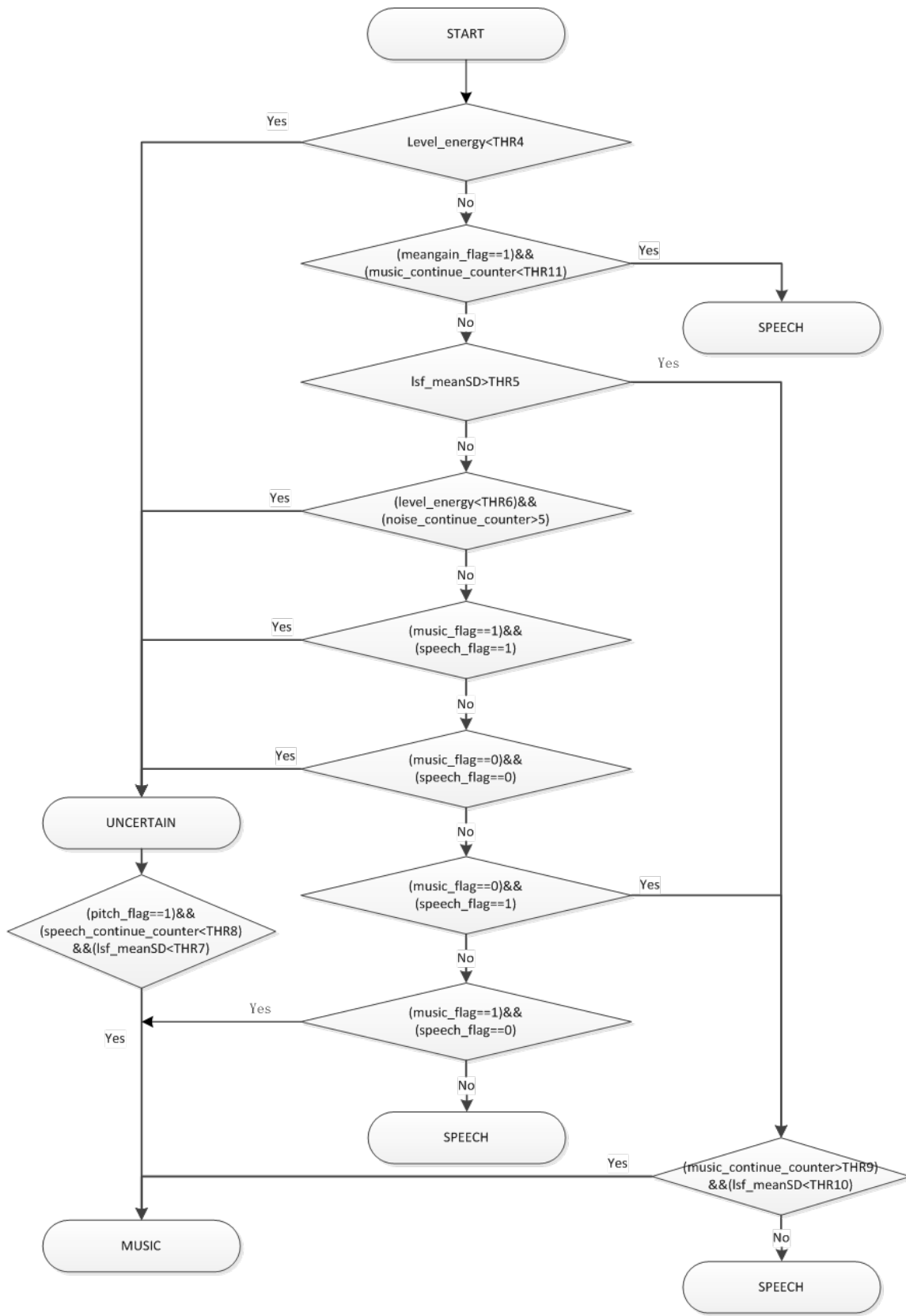


Figure 12—Flowchart of first classification

5.2.6.5.5 Parameter updating

The parameters to update include three type counters and thresholds in each classification module. If current frame is classified as music, *music_continue_counter* will be added by 1. Otherwise, it will be reset to 0. The other two type counters are updated in similar way. In addition, thresholds are updated according to the SNR of the initial classification module.

5.2.6.5.6 Background noise estimation control

In practical application, the energy level of background noise may rise suddenly. At this time, the background noise estimation probably shall not be updated because the signal is continuously classified as non-noise signal. So, background noise estimation control is necessary. In BNEC, the updating rate is calculated with the spectral distribution of current signal and the background noise, and then sent to initial classification module.

The spectral distribution parameter vector includes the following:

- *zcr_mean*: the short-term mean value of zero-cross rate.
- *ra_mean*: the short-term mean value of the ratio of high band and low band energy.
- *f_flux_mean*: the short-term mean value of frequency-domain fluctuation of subband.
- *t_flux_mean*: the short-term mean value of time-domain fluctuation of subband.

Among them, *zcr_mean* is calculated as follows. And the others are calculated in a similar way, as shown in Equation (14).

$$zcr_mean_m = ALPHA \times zcr_mean_{m-1} + (1 - ALPHA) \times zcr_m \quad (14)$$

where

m is the frame index

The updating rate of background noise is decided by the distance between spectral distribution parameter of current signal and background noise. In this part, the Manhattan distance is used, and it is described as shown in Equation (15).

$$d_{cb} = \sum_{i=1}^4 |p(i) - \tilde{p}(i)| \quad (15)$$

where

- p* is the spectral distribution parameter vector of current signal
- $\tilde{p}$* is the spectral distribution parameter vector estimation of background noise

When $d_{cb} < TH1$, the output rate of the module is acc1, which is the fastest updating rate; when $d_{cb} < TH2$, the output rate is acc2; when $d_{cb} < TH3$, the output rate is acc3; in other cases, the output rate is acc4. *TH1*, *TH2*, *TH3* are different thresholds.

5.3 ACELP/TVC hybrid coding module

5.3.1 Pre-emphasis

The input (mono) signal to the core ACELP/TVC encoder is first pre-processed through a high-pass filter. Then, a first order, fixed pre-emphasis filter is applied with transfer function which is described in Equation (16).

$$P(z) = 1 - 0.68z^{-1} \quad (16)$$

This pre-emphasis filter reduces the signal energy at low frequency and increases the signal energy at a high frequency. The result is a signal with a less spectral dynamics, which enhances the resolution of the linear predictive coding (LPC) analysis. This is particularly important in fixed-point implementations.

5.3.2 LP analysis, interpolation and quantify

5.3.2.1 LP analysis

LP analysis is performed using the auto correlation approach with asymmetric windows in 16 prediction order. The autocorrelations of windowed speech are converted to the LP coefficients using the Levinson-Durbin algorithm. Then the LP coefficients are transformed to the ISP domain for quantization and interpolation purposes. The interpolated quantized and unquantized filters are converted back to the LP filter coefficients. The corresponding 16-order LP synthesis filter could be represented by Equation (17).

$$H(z) = \frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^m \hat{a}_i z^{-i}} \quad (17)$$

where

$\hat{a}_i$ is the quantized LP coefficients, and $m=16$ is the prediction order

LP analysis is performed at 384 asymmetric window with 128 samples overlap, as shown in Figure 13. The window concentrate at 256-sample frame n with 64 samples look ahead from frame n-1 and another 64 samples from the next frame.

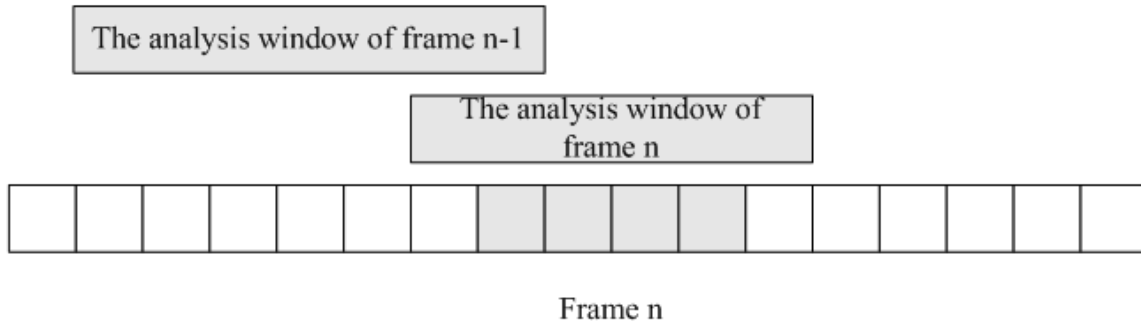


Figure 13—Frame structure figure of linear prediction

5.3.2.2 Windowing and auto correlation computation

LP analysis is performed once per frame using an asymmetric window. The block diagram of LP analysis and quantification is shown in Figure 14. The window has its weight concentrated at the fourth sub-frame, and it consists of two parts, the first part is half a half of a Hamming window and the second part is a quarter of a Hamming-cosine function cycle. The window is given as shown in Equation (18).

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{2L_1 - 1}\right) & n = 0, 1, \dots, L_1 - 1 \\ \cos\left(\frac{2\pi(n - L_1)}{4L_2 - 1}\right) & n = L_1, L_1 + 1, \dots, L_1 + L_2 - 1 \end{cases} \quad (18)$$

where

Values L_1 is 256 and $L_2=128$ are used

The window type of LP analysis window is shown in Figure 15.

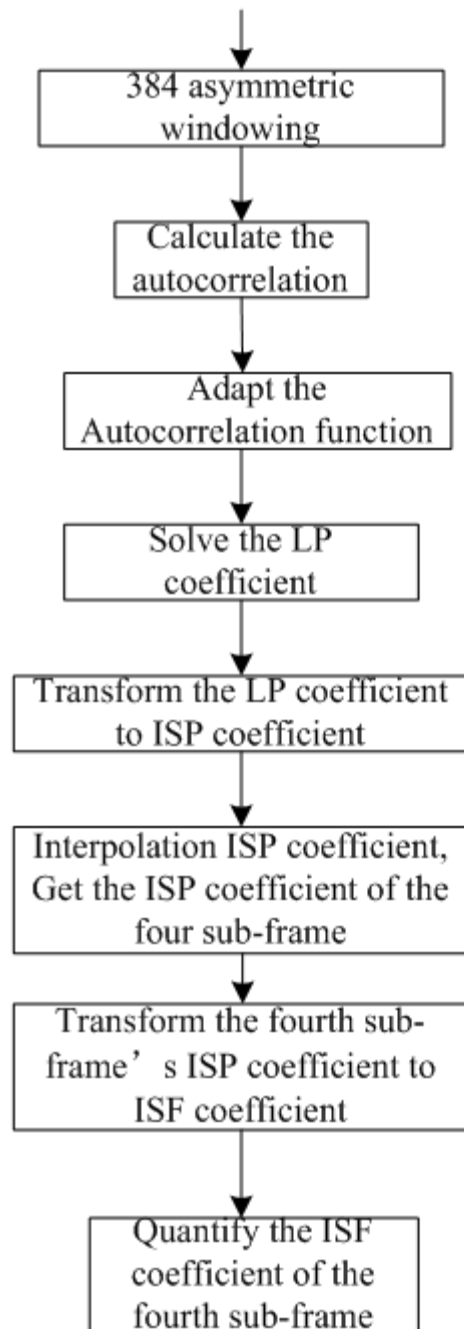


Figure 14—Block diagram of the encoder's LP analysis and quantification

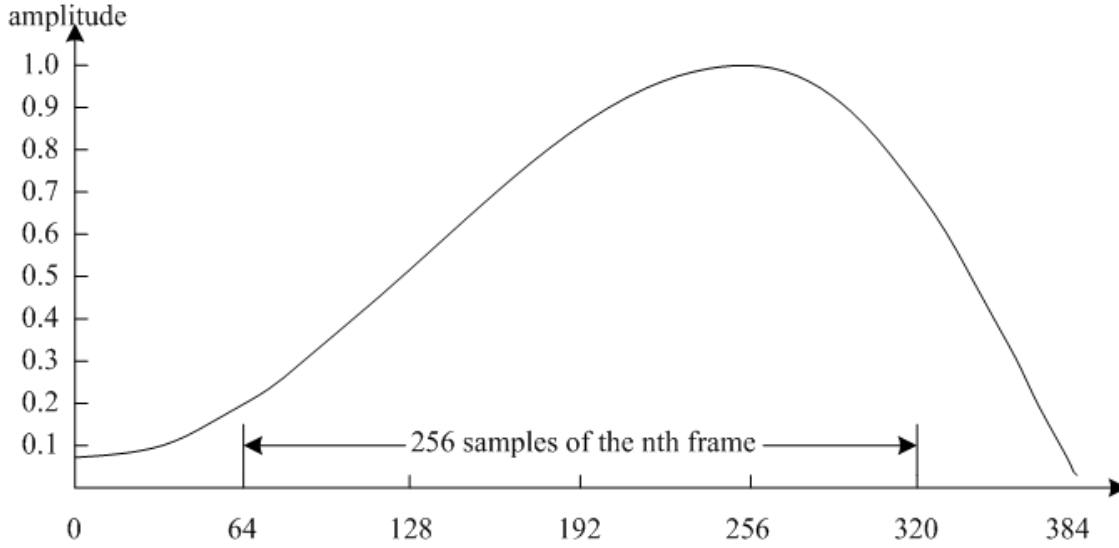


Figure 15—Window type of LP analysis window

Set the speech signal after windowing as $s'(n)$, $n = 0, 1, \dots, 383$, the calculating method is in Equation (19).

$$s'(n) = w(n)s(n), \quad n = 0, 1, \dots, 383 \quad (19)$$

where

$w(n)$ is the windowing function, $s(n)$ is the internal sampling audio signal after pretreatment

The corresponding autocorrelation function of $s'(n)$ is shown Equation (20).

$$r(k) = \sum_{n=k}^{383} s'(n)s'(n-k), \quad k = 0, 1, \dots, 16 \quad (20)$$

Then use the lag window $w_{lag}(i)$ to multiply the autocorrelation function in order to make it have 60 Hz bandwidth extension. The lag window $w_{lag}(i)$ is shown in Equation (21).

$$w_{lag}(i) = \exp \left[-\frac{1}{2} \left(\frac{2\pi f_0 i}{f_s} \right)^2 \right], \quad i = 1, 2, \dots, 16 \quad (21)$$

where

$f_0 = 60$ Hz is the bandwidth extension, and f_s is the internal sampling frequency. The revised autocorrelation function $r'(k)$ could be calculated as shown in Equation (22).

$$r'(0) = 1.0001 r(0), \quad r'(k) = r(k)w_{lag}(k), \quad k = 1, 2, \dots, 16 \quad (22)$$

5.3.2.3 Levinson-Durbin algorithm

The LP coefficient a_k , $k=1, 2, \dots, 16$ could be calculated by using the modified autocorrelation function, as shown in Equation (23).

$$\sum_{k=1}^{16} a_k r'(|i-k|) = -r'(i), \quad i=1, 2, \dots, 16 \quad (23)$$

The set of equations is solved using the Levinson-Durbin algorithm. This algorithm uses the following recursion:

```

E(0)=r'(0)
For i=1 to 16 do
    For j=1 to i-1 do
        Temp+=a[j][i-1]*r'[i-j];
    End
    k[i]=-(r'[i]+Temp)/E[i-1];
    a[i][i]=k[i];
    For i=1 to 16 do
        a[j][i]=a[j][i-1]+k[i]a[i-j][i-1];
    end
    E(i)=(1-k[i]*k[i])*E[i-1];

```

Finally get the LP coefficient: $a_j = a_j^{(16)}$ $j=1, 2, \dots, 16$ and transform LP coefficient to ISP coefficient, in order to quantify and interpolation.

5.3.2.4 LP to ISP conversion

For a 16-order LP synthesis filter, the ISPs are defined as the roots of the sum and difference polynomials, as shown in Equation (24) and Equation (25).

$$f_1'(z) = A(z) + z^{-16}A(z^{-1}) \quad (24)$$

$$f_2'(z) = A(z) - z^{-16}A(z^{-1}) \quad (25)$$

The polynomials $f_1'(z)$ and $f_2'(z)$ are symmetric and antisymmetric, respectively. It can be proven that all roots of these polynomials are on the unit circle and they alternate each other. $f_2'(z)$ has two roots, $z=1$ ($\omega=0$) and $z=-1$ ($\omega=\pi$). To eliminate these two roots, two new polynomials $f_1(z)$ and $f_2(z)$ are defined, as shown in Equation (26) and Equation (27).

$$f_1(z) = f_1'(z) \quad (26)$$

$$f_2(z) = f_2'(z) / (1 - z^{-2}). \quad (27)$$

Polynomials $f_1(z)$ and $f_2(z)$ have eight and seven conjugate complex roots ($e^{\pm j\omega_i}$) on the unit circle. Therefore, the polynomials could be written as in Equation (28) and Equation (29) where $q_i = \cos(\omega_i)$ with ω_i being the ISF. $a[16]$ is the last predictor coefficient. ISFs satisfy the ordering property $0 < \omega_1 < \omega_2 < \dots < \omega_{16} < \pi$. We refer to q_i as the ISPs in the cosine domain.

$$F_1(z) = (1 + a[16]) \prod_{i=0,2,\dots,14} (1 - 2q_i z^{-1} + z^{-2}) \quad (28)$$

$$F_2(z) = (1 - a[16]) \prod_{i=1,3,\dots,13} (1 - 2q_i z^{-1} + z^{-2}) \quad (29)$$

The pseudo-code of the solution procedure of ISP coefficient is described as follows.

```

For i=0 to 7 do
    f[1][i] = a[i] + a[m-i];
    f[2][i] = a[i] - a[m-i] + f[2][i-2];
end
f[1][8] = 2*a[8];

```

When $m = 16$ is the order of predictor, $f[2][-2] = f[2][-1] = 0$.

The solution procedure of ISP coefficient is described as follows. Firstly divide 0 to π into 24 equal parts. Find the interval of the sign changes, in each interval, applies Newton-chebyshev iterative method to find roots. Suppose the endpoints of inter zone is a, b, the solution of the initial value of iterative method ω_0 is shown in Equation (30).

$$\omega_0 = a - \frac{f(a)(b-a)}{f(b)-f(a)} \quad (30)$$

The method of iterative solution based on ω_0 is shown in Equation (31).

$$\omega_{i+1} = \omega_i - \frac{F_1(\omega_i)}{F_1'(\omega_i)}, i = 0, 1, 2 \quad (31)$$

where

$$F_1(\omega) = r_8 x^8 + r_7 x^7 + (r_6 - 8r_8) x^6 + (r_5 - 7r_7) x^5 + \\ (20r_8 - 6r_6 + r_4) x^4 + (14r_7 - 5r_5 + r_3) x^3 + (r_2 - 16r_8 + 9r_6 - 4r_4) x^2 + \\ (r_1 - 7r_7 + 5r_5 - 3r_3) x + (r_0 - 2r_2 + 2r_4 - 2r_6 + 2r_8) \big|_{x=\cos \omega} \quad (32)$$

$$F_1'(\omega) = (8r_8 x^7 + 7r_7 x^6 + 6(r_6 - 8r_8) x^5 + 5(r_5 - 7r_7) x^4 + 4(20r_8 - 6r_6 + r_4) x^3 + \\ 3(14r_7 - 5r_5 + r_3) x^2 + 2(r_2 - 16r_8 + 9r_6 - 4r_4) x + (r_1 - 7r_7 + 5r_5 - 3r_3) \big|_{x=\cos \omega}) \sin \omega \quad (33)$$

After eight iterations we shall the eight roots $F_1(\omega)$ corresponding could be obtained, and the seven roots to $F_2(\omega)$ could be obtained in the same way. These coefficients are all the ISP coefficients including $m = 16$.

5.3.2.5 ISP to LP conversion

Once ISP coefficients are quantified and interpolated, they are converted back to the LP coefficient domain $\{a_k\}$. The coefficients of $F_1(z)$ and $F_2(z)$ are found by expanding Equation (28) and Equation (29)

knowing the quantized and interpolated ISPs q_i , $i = 0, \dots, m-1$ ($m = 16$). The following recursive relation is used to compute $f_1(i)$

```

for i=2 to m/2 do
  f[1][i] = -2*q[2i-2]*f[1][i-1] + 2*f[1][i-2]
  for j=i-1 down to 2 do
    f[1][j] = f[1][j] - 2*q[2i-2]*f[1][j-1] + f[1][j-2]
  end
  f[1][1] = f[1][1] - 2*q[2i-2]
end

```

with initial value $f[1][0] = 1$, $f[1][1] = -2*q[0]$.

The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-2} by q_{2i-1} , and $m/2$ by $m/2-1$, and with initial conditions $f_2(0) = 1$ and $f_2(1) = -2q_1$.

Once the coefficients $f_1(i)$ and $f_2(i)$ are found, $F_2(z)$ is multiplied by $1 - z^{-2}$ to obtain $F'_2(z)$. It is shown in Equation (34).

$$\begin{aligned} f'_2(i) &= f_2(i) - f_2(i-2), & i = 2, 3, \dots, m/2-1, \\ f'_1(i) &= f_1(i) & i = 0, 1, \dots, m/2 \end{aligned} \quad (34)$$

Then $F'_1(z)$ and $F'_2(z)$ are multiplied by $1 + q_{m-1}$ and $1 - q_{m-1}$ respectively, which are shown in Equation (35).

$$\begin{aligned} f'_2(i) &= (1 - q_{m-1})f'_2(i), & i = 0, 1, \dots, m/2-1, \\ f'_1(i) &= (1 + q_{m-1})f'_1(i) & i = 0, 1, \dots, m/2 \end{aligned} \quad (35)$$

Considering the fact that $F'_1(z)$ and $F'_2(z)$ are respectively symmetric and anti-symmetric polynomials, according to $A(z) = (F'_1(z) + F'_2(z))/2$, finally the LP coefficients are found by Equation (36).

$$a_i = \begin{cases} 0.5f'_1(i) + 0.5f'_2(i) & i = 1, 2, \dots, m/2-1 \\ 0.5f'_1(i) - 0.5f'_2(i) & i = m/2+1, m/2+2, \dots, m-1 \\ 0.5f'_1(m/2) & i = m/2 \\ q_{m-1} & i = m \end{cases} \quad (36)$$

5.3.2.6 Quantization of the ISP coefficient

Before quantizing of the ISP coefficient, it should be transformed into ISF coefficient in frequency domain. The quantization process of ISF coefficients is as follows: the current frame LSF vector or current frame LSF residual vector need to be divided into two groups of sub-vector $\{a_i\}$, $\{b_j\}$, the current frame LSF residual vector described above is difference of the current frame LSF vector minus the mean LSF vector, then minus the contribution of the previous line spectral frequency vector to the current frame LSF vector; a group of sub-vector coefficients a_i^q could be gained after one of the group of sub-vector $\{a_i\}$ is quantitized, part of the sub-vector coefficients $a_{j,k}^q$ is chosen out to predict the set of sub-vector coefficients $\{b_j\}$, which is not quantified, and the predictive value of the sub-vector coefficients

$b_j^{predict}$ could be obtained afterwards; getting a set of not quantified sub-vector coefficients $\{b_j\}$ and the predictive value of the corresponding sub-vector coefficients $b_j^{predict}$ into the difference operator to get a set of predictive value of residual vector $\{b_j'\}$, finally this predictive value of residual vector $\{b_j'\}$ is quantized. The processing block diagram is shown in Figure 16.

The detailed ISF quantization processes area described as follows.

The expression of ISF is shown in Equation (37).

$$\begin{aligned} f_i &= \frac{f_s}{2\pi} \arccos(q_i), & i = 0, 1, \dots, 14 \\ &= \frac{f_s}{4\pi} \arccos(q_i), & i = 15 \end{aligned} \quad (37)$$

where

$f_i \in [0, f_s / 2]$ is the ISF coefficient

f_s is the internal sampling frequency, then the ISF vector could be expressed as $\mathbf{f}^T = [f_0, f_1, \dots, f_{15}]$, where T means transposition of vector

The first order moving average (MA) prediction method is used to calculate the ISF prediction residual vector of the current frame and to quantify the ISF prediction residual vector. The procedure of prediction and quantization is shown in Figure 17.

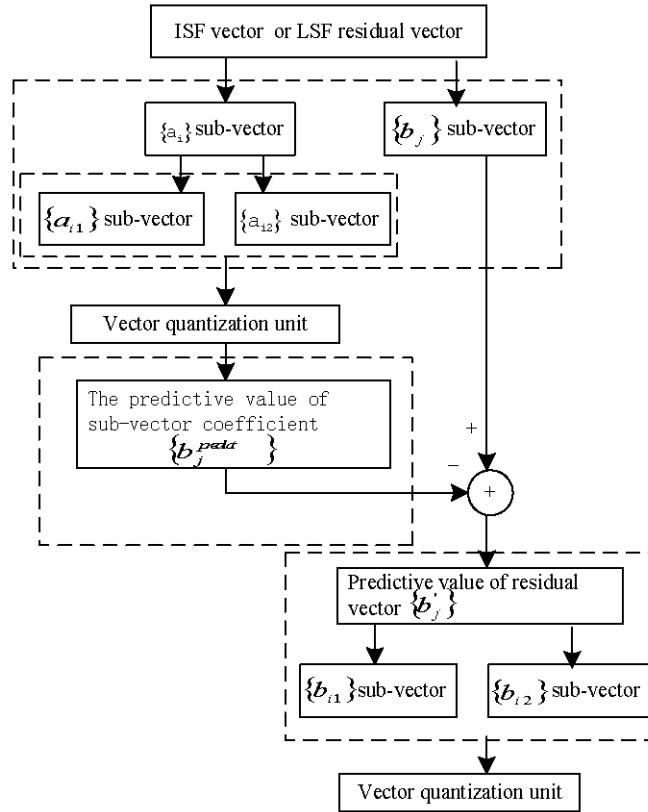


Figure 16—Basic block diagram of the ISF quantify

Here $\mathbf{z}(n)$ is defined as de-mean vector of the current frame, the predicted residual vector as $\mathbf{r}(n)$, their expression are described Equation (38) and Equation (39).

$$\mathbf{z}(n) = \mathbf{isf}_n - \mathbf{mean_isf} \quad (38)$$

$$\mathbf{r}(n) = \mathbf{z}(n) - \mathbf{p}(n) \quad (39)$$

where

$\mathbf{isf}_n$ is the ISF coefficient vector of frame n
 $\mathbf{mean_isf}$ is the mean value of ISF coefficient vector
 $\mathbf{p}(n)$ is the predicted ISF coefficient of the current frame and could be computed by first order MA prediction method which is shown in Equation (40)

$$\mathbf{p}(n) = \frac{1}{3} \hat{\mathbf{r}}(n-1) \quad (40)$$

The $\hat{\mathbf{r}}(n-1)$ is the quantified ISF residual vector of the previous frame.

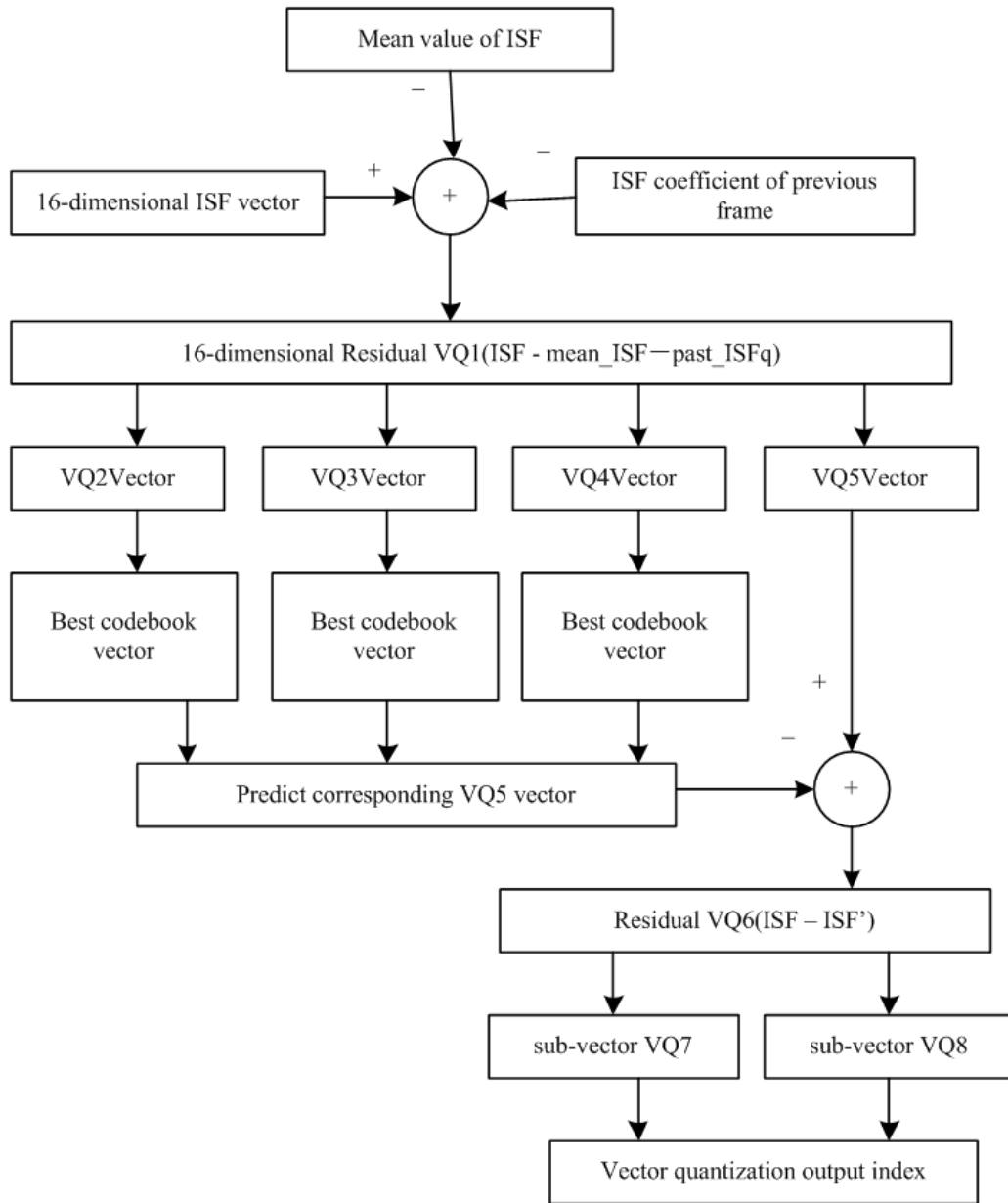


Figure 17—Block diagram of ISF coding

To take advantage of the correlation of ISF coefficients of one same frame, the 16 ISF coefficients (VQ_1 vector) are divided into two groups according to the parity of index number, which are shown as follows.

$$\{a_i\} = \{res_isf_0, res_isf_2, res_isf_4, res_isf_6, res_isf_8, res_isf_{10}, res_isf_{12}, res_isf_{14}\}$$

$$\{b_i\} = \{res_isf_1, res_isf_3, res_isf_5, res_isf_7, res_isf_9, res_isf_{11}, res_isf_{13}, res_isf_{15}\}$$