|  |  |
| --- | --- |
| 3GPP TS 26.403 V18.0.0 (2024-03) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  General audio codec audio processing functions;  Enhanced aacPlus general audio codec;  Encoder specification;  Advanced Audio Coding (AAC) part  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 5

2 Normative references 5

3 Definitions, symbols and abbreviations 6

3.1 Definitions 6

3.2 Symbols 6

3.3 Abbreviations 6

4 Outline description 6

5 AAC Encoder 7

5.1 Overview 7

5.2 Stereo Preprocessing 8

5.3 Filterbank 8

5.4 Psychoacoustic Model 8

5.4.1 Blockswitching 8

5.4.2 Threshold Calculation 10

5.4.2.1 Calculation of the energy spectrum 10

5.4.2.2 From energy to threshold 11

5.4.2.3 Spreading 11

5.4.2.4 Threshold in quiet 11

5.4.2.5 Pre-echo control 11

5.4.3 Spreaded Energy Calculation 12

5.4.4 Grouping 12

5.5 Tools 12

5.5.1 Temporal Noise Shaping (TNS) 12

5.5.1.1 TNS detection 12

5.5.1.2 TNS Stereo Synchronization 13

5.5.1.3 TNS Order 13

5.5.1.4 TNS Filtering 13

5.5.1.5 Threshold modification 13

5.5.2 Mid/Side Stereo 13

5.6 Quantization and coding 14

5.6.1 Reduction of psychoacoustic requirements 14

5.6.1.1 Principle of the threshold reduction strategy 14

5.6.1.1.1 Addition of noise with equal loudness 14

5.6.1.1.2 Avoidance of spectral holes 14

5.6.1.1.3 Relation between bit demand and perceptual entropy 15

5.6.1.2 Calculation of Bit Demand 15

5.6.1.3 Calculation of the reduction value 17

5.6.1.3.1 Preparatory steps of the perceptual entropy calculation 18

5.6.1.3.2 Calculation of the desired perceptual entropy 18

5.6.1.3.3 Selection of the bands for avoidance of holes 18

5.6.1.3.4 First Estimation of the reduction value 18

5.6.1.3.5 Second Estimation of the reduction value 19

5.6.1.3.6 Final threshold modification by linearization 19

5.6.1.3.7 Further perceptual entropy reduction 20

5.6.1.3.8 Possible failures 20

5.6.2 Scalefactor determination 20

5.6.2.1 Scalefactor Estimation 21

5.6.2.2 Scalefactor Improvement by Quantization 21

5.6.2.3 Scalefactor Difference Reduction 21

5.6.2.4 Final scalefactor determination 22

5.6.3 Noiseless coding 22

5.6.4 Out of Bits Prevention 22

Annex A (informative): Change history 23

# Foreword

The present document describes the detailed mapping of the general audio service employing the aacPlus general audio codec within the 3GPP system.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 Indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification;

# 1 Scope

This Telecommunication Standard (TS) describes the AAC encoder part of the Enhanced aacPlus general audio codec [1].

# 2 Normative references

This TS incorporates by dated and undated reference, provisions from other publications. These normative references are cited in 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 TS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1] 3GPP TS 26.401: "Enhanced aacPlus general audio codec; General Description".

[2] ISO/IEC 14496-3:2001: "Information technology - Coding of audio-visual objects - Part 3: Audio".

[3] ISO/IEC 14496-3:2001/Amd.1:2003: "Bandwidth Extension".

[4] ISO/IEC 14496-3:2001/Amd.1:2003/DCOR1.

[5] ISO/IEC 14496-3:2001/ Amd.2:2004: "Parametric Coding for High Quality Audio".

3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of this TS, the following definitions apply:

**frame:** time segment associated with one AAC single channel or channel pair element

**frequency coefficient:** output value of the MDCT transform

**scalefactor band:** a group of consecutive frequency coefficients, that will be coded with the same quantizer step size

## 3.2 Symbols

For the purposes of this TS, the following symbols apply:

 is the current index for the spectral coefficients

 is the index of the first spectral coefficient in scalefactorband 

 is the current scalefactor band

## 3.3 Abbreviations

For the purposes of this TS, the following abbreviations apply.

AAC Advanced Audio Coding

aacPlus Combination of MPEG-4 AAC and MPEG-4 Bandwidth extension (SBR)

Enhanced aacPlus Combination of MPEG-4 AAC, MPEG-4 Bandwidth extension (SBR) and MPEG-4 Parametric Stereo

KBD Kaiser-Bessel derived

PE perceptual entropy

SBR Spectral Band Replication

TNS Temporal Noise Shaping

# 4 Outline description

This TS is structured as follows:

Section 5.1 gives an encoder overview description. Section 5.2 gives a detailed description of the stereo preprocessing. Section 5.3 gives a detailed description of the filterbank used in the encoder. Section 5.4 gives a detailed description of the psychoacoustic model. Section 5.5 gives a detailed description of the temporal noise shaping and mid/side stereo tools. Section 5.6 gives a detailed description of the quantization and coding procedure used in the encoder.

# 5 AAC Encoder

## 5.1 Overview

The AAC encoder acts as the core encoding algorithm of the aacPlus system encoding at half the sampling rate of aacPlus. Since aacPlus implements the High Efficiency AAC Profile at Level 2 as defined in [3], the AAC LC object type is used. The AAC LC object type does not implement the Long Term Predictor (LTP) tool. The Level 2 implies a restriction to a maximum of two channels. Furthermore in case of SBR being used, the maximum AAC sampling rate is restricted to 24 kHz whereas if SBR is not used the maximum AAC sampling rate is restricted to 48 kHz.

The basic layout is depicted below.



Figure 1: AAC Encoder Block Diagram

## 5.2 Stereo Preprocessing

With stereo preprocessing, the stereo width of difficult to encode signals at low bitrates is reduced. Stereo preprocessing is active for bitrates less than 60kbit/s.

The side channel is attenuated with influence of the following parameters:

- The total perceptual entropy before the increase of the thresholds. This PE is smoothed over past frames and normalized. For a definition of the perceptual entropy see 5.6.1.1.3. .

- The energy ratio between side and mid channel smoothed over past frames. If the side channel is very strong, less attenuation of the side channel should happen.

- The energy ratio between the left and right channel. Less attenuation of the side channel occurs for signals that appear to be nearly on the left or the right.

- The smaller the bitrate, the more attenuation of the side channel takes place.

Depending on these parameters an attenuation factor is calculated for every frame. Always 1024 samples of one frame for the left and right channel are then modified as follows:



with ,  as left resp. right samples before and ,  as modified left and right samples after stereo preprocessing.

## 5.3 Filterbank

The filterbank is an MDCT as described in [2]. The window length of the MDCT is either 2048 for the ONLY\_LONG\_SEQUENCE, LONG\_START\_SEQUENCE and LONG\_STOP\_SEQUENCE window sequence or 256 for the EIGHT\_SHORT\_SEQUENCE window sequence. The spectral coefficients are defined as follows:

 , for 

with  as windowed input sequence, n as samples index and k as spectral coefficient index.

For long windows the window shape is always 1, that is a Kaiser-Bessel derived (KBD) window will be used. As window shape for the short windows always the sine window will be applied. For the definition of KBD and sine window see [2].

## 5.4 Psychoacoustic Model

The psychoacoustic model is simplified compared to the model presented in [2]. Note that the model works in combination with the quantization and coding strategy described in 5.6 below. The following sections describe the steps of the threshold calculation.

### 5.4.1 Blockswitching

The decision wether to use long windows with a window length of 2048 samples or a sequence of eight short blocks with a window length of 256 samples will be taken in the time domain. It is not possible to switch immediately between an ONLY\_LONG\_SEQUENCE and an EIGHT\_SHORT\_SEQUENCE. Thus when switching from the long window transform to frames with eight short windows a LONG\_START\_SEQUENCE has to be inserted, resp. when switching back from short to long a STOP\_WINDOW\_SEQUENCE is neeeded. Therefore there needs to be a lookahead of 1024+576 samples for the blockswitch decision (see figure below).



Figure 2: Blockswitch detection lookahead

A high pass IIR-Filter with the transfer function



is applied to the samples. After filtering, eight subblock energies are calculated by summing up 128 consecutive squared samples. These eight subblock energies represent the eight short windows of the next frame.

An attack is detected if one of these subblock energies exceeds a sliding average of the previous energies by a constant factor  and is greater than a constant energy level . The value for  depends on the bitrate and the number of channels:

for mono: 

for stereo: 

The window sequence of the next frame is now either set to ONLY\_LONG\_SEQUENCE or EIGHT\_SHORT\_SEQUENCE. Now the final window sequence of the actual frame can be determined by obeying the following the rules:

1) after a long window there can be a long or a start window

2) after a start window there will always be a short window sequence

3) after a short window sequence follows another short window sequence or a stop window

4) after a stop window there can be a long window or a start window

If the current window sequence is and EIGHT\_SHORT\_SEQUENCE, the eight windows will be grouped to reduce the sideinfo for transmitting scalefactors etc. If no attack has been detected in the actual short window sequence, there will be 3 groups with the first 3 subwindows in the first group, the next 3 in the second group and the last 2 subwindows will be in the third group. For short window sequences with attack there will always be 4 groups. The number of subwindows in each group depends on the position of the attack subwindow:

Table 1: Grouping of subwindows in an EIGHT\_SHORT\_SEQUENCE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| position of attack | nr of subwindows group 1 | nr of subwindows group 2 | nr of subwindows group 3 | nr of subwindows group 4 |
| 0 | 1 | 3 | 3 | 1 |
| 1 | 1 | 1 | 3 | 3 |
| 2 | 2 | 1 | 3 | 2 |
| 3 | 3 | 1 | 3 | 1 |
| 4 | 3 | 1 | 1 | 3 |
| 5 | 3 | 2 | 1 | 2 |
| 6 | 3 | 3 | 1 | 1 |
| 7 | 3 | 3 | 1 | 1 |

In case of stereo encoding, the blocktype for both channels must be the same to be able to apply the M/S stereo tool. The final common blocktype is chosen as shown in the following table:

Table 2: window sequence synchronization for stereo

|  |  |  |
| --- | --- | --- |
| blocktype channel 0 | blocktype channel 1 | final blocktype for both channels |
| Long | long | long |
| Long | start | start |
| Long | short | short |
| Long | stop | stop |
| Start | long | start |
| Start | start | start |
| Start | short | short |
| Start | stop | short |
| Short | long | short |
| Short | start | short |
| Short | short | short |
| Short | stop | short |
| Stop | long | stop |
| Stop | start | short |
| Stop | short | short |
| Stop | stop | stop |

If the final window sequence is EIGHT\_SHORT\_SEQUENCE the grouping is chosen from the channel containing the higher maximum subblock energy.

### 5.4.2 Threshold Calculation

The following are the necessary steps for the calculation of the psychoacoustic threshold , which is an upper limit for the quantization noise of the coder.

#### 5.4.2.1 Calculation of the energy spectrum

The energy spectrum in the coder scalefactor band domain  is calculated by using the output values  of the MDCT-transform that are later quantized and coded. This is done by the following equation:



Here  is the first spectral line of scalefactor band .

In this psychoacoustic model no threshold partition is used. The threshold calculation is performed directly in the scalefactor band domain.

#### 5.4.2.2 From energy to threshold

No difference is made between tonal and noisy components in the signal. Therefore the "worst case" is assumed, i.e. the signal is tonal for the complete frequency range. Thresholds must be achieved that result in a "transparent" audio quality.

The decrease of the energy is done by a constant required signal to noise ratio  which is 29dB for AAC . The scaled thresholds  are:



#### 5.4.2.3 Spreading

Instead of a convolution of the spectral energy with a spreading function, a simpler spreading is calculated. Here the slope to higher frequencies is created by weighting the previous threshold value with a frequency dependent factor  and by building the maximum of the threshold value of the actual band with this weighted threshold of the previous band.



Accordingly the steeper slope towards the low frequencies is computed by another pass beginning at the highest band and weighting the energy values by a factor .



The values for  resp.  are calculated by the distance of the adjacent bands in Bark and a constant slope that is 15dB/Bark for the first and 30dB/Bark for the second equation.

#### 5.4.2.4 Threshold in quiet

The comparison with the threshold in quiet  is a simple maximum operation.



The threshold in quiet is given as array for the Bark scale. Because of the difference of the scalefactor band scale compared to the Bark scale, the minimum of the threshold in quiet for the Bark values at the lower and the upper end of the scalefactor band is used.

#### 5.4.2.5 Pre-echo control

The pre-echo control operates as in the psychoacoustic model of [2]. To avoid pre-echos the actual threshold  is compared to the previous threshold :



with the parameters  and .

No pre-echo control can be calculated in case of blockswitching, because the psychoacoustic model doesn't calculate the thresholds for both long and short blocks simultaneous and the pre-echo control needs the thresholds of the previous block with the same scalefactor band partition as the actual block. Thus the pre-echo control is inactive for the first short window (but not all short windows in a short frame) after a start block and for all frames with a stop window sequence.

### 5.4.3 Spreaded Energy Calculation

After an eventual filtering of the mdct spectrum with the TNS analysis filter (see 5.5.1 ), the energy calculation of section 5.4.2.1 has to be performed again. Spreading this energy  the same way as the thresholds (see 5.4.2.3) yields the spreaded energy  in the scalefactor band domain. The values for  resp.  are dependent on the blocktype and are derived from constant slopes in the bark domain.

For long block  and 

and for short blocks  and , with

 as the width in bark of the scalefactor band 

Both the scalefactor band energy after TNS and the spreaded energy  are important input values for the determination of which bands must not be quantized to zero (see section 5.6.1.1.2. for more details on the avoidance of spectral holes).

### 5.4.4 Grouping

If the window sequence of the current frame is an EIGHT\_SHORT\_SEQUENCE, a grouping configuration has been determined by the blockswitching algorithm described above. The psychoacoustic model calculates thresholds, energies and other variables in the subwindow domain. The scalefactor band based thresholds and energies are grouped by adding up the values of all subwindows belonging to one group. The spectrum has to be reordered to match the new combined scalefactor bands.

## 5.5 Tools

### 5.5.1 Temporal Noise Shaping (TNS)

For a general description of TNS see [2]. If TNS is active in this encoder, only one filter per MDCT-spectrum will be applied. The steps in TNS encoding are described below. TNS is always calculated on a per subwindow basis, so in case of an EIGHT\_SHORT\_SEQUENCE window sequence these steps have to be applied once for each of the eight subwindows.

#### 5.5.1.1 TNS detection

Out of the spectral coefficients  a weighted spectrum  is calculated. The weighting factors are determined from the energy of the appropriate scalefactor band . For a definition of the scale factor band energy  see section 5.4.2.1. The factors are smoothed by filtering down:

for (k=lpcStopLine-2; k>=lpcStartLine; k--) {

wfac[k] = (wfac[k] + wfac[k+1]) / 2;

}

and up:

for (k=lpcStartLine+1; k<lpcStopLine; k++) {

wfac[k] = (wfac[k] + wfac[k-1]) / 2;

}

The lower and upper limits lpcStartLine and lpcStopLine depend on the bitrate and the blocktype. Next steps are an autocorrelation calculation and a LPC calculation using the Levinson-Durbin algorithm. As result so called parcor or reflection coefficients  and the prediction gain are available. TNS will be used only if the prediction gain is greater than a given threshold, which is bitrate dependent and varies between 1.2 and 1.41.

#### 5.5.1.2 TNS Stereo Synchronization

If prediction gains for the left and right channel differ only less than 3%, the same TNS filter coefficients are chosen for both channels by copying the TNS data of the left channel to the right channel.

#### 5.5.1.3 TNS Order

The TNS parcor coefficients will be quantized with a resolution of 4 bits for long blocks and 3 bits for short blocks. The order of the coefficients is now determined by going down from the maximum order until the first coefficient that exceeds an absolute value of 0.1 has been reached.

#### 5.5.1.4 TNS Filtering

The spectral coefficients  will now be replaced by filtering with the parcor coefficients. The first scalefactor band affected corresponds to a frequency of 1275Hz for long blocks resp. 2750Hz for short blocks. The filtering is done with the help of a so called lattice filter, no conversion from parcor coefficients  to linear prediction coefficients is required.

#### 5.5.1.5 Threshold modification

In the frequency range from 380Hz to the start frequency of the TNS filter the coding demands will be increased by multiplying a factor of 0.25 to the thresholds  calculated by the psychoacoustic model.

### 5.5.2 Mid/Side Stereo

Normal stereo operation, and thus Mid/Side Stereo, is only required when operating the encoder at bitrates at or above 44 kbit/s. Below 44 kbit/s the Parametric Stereo coding tool [5] is used instead where the AAC core is operated in mono.

Within Mid/Side Stereo, for each scalefactor band the left and right channel coefficients are either coded as L and R or as mid and side channel

 and .

For stereo in the psychoacoustic model in addition to the left and right energies also the mid and side energies  are calculated. The threshold for coding mid and side channel is simply the minimum of left and right thresholds . M/S coding is actually used if



is fulfilled. In such a case left channel values for spectral coefficients, energies and thresholds will be replaced by the mid channel values, resp. right channel values will be replaced by the side channel values. The spreaded energy  for mid and side channel will be the minium of the spreaded energy of left and right channel.

## 5.6 Quantization and coding

### 5.6.1 Reduction of psychoacoustic requirements

Usually the requirements of the psychoacoustic model are too strong for the desired bitrate. Thus a threshold reduction strategy is necessary, i.e. the strategy reduces the requirements by increasing the thresholds given by the psychoacoustic model. An overcoding, i.e. decreasing the thresholds for a finer quantization, doesn't take place in this encoder.

In this section the strategy to reduce the requirements for the quantization accuracy is presented. The first section explains the technique to modify the thresholds calculated by the psychoacoustic model. The reduction strategy has to operate with estimations of the bit demand to avoid multiple quantization and bit counting. This is done by an estimation of the used bits by the perceptual entropy described in the second section. Finally the method how to find the amount of threshold reduction for a given bit demand is presented.

#### 5.6.1.1 Principle of the threshold reduction strategy

##### 5.6.1.1.1 Addition of noise with equal loudness

An increase of the thresholds  from the psychoacoustic model is done in the form that the loudness of the disturbance is equal for all bands. Here the loudness  for additional noise is approximated by the equation , where  is the energy of the noise. To increase the masking threshold equally loud over the whole frequency range, to each scalefactor band the constant loudness  will be added:



The thresholds are converted to the loudness domain and after the addition of the constant loudness there is a conversion back to the energy domain.

##### 5.6.1.1.2 Avoidance of spectral holes

The basic form of threshold reduction described above is insufficient to guarantee an adequate audio quality. There will be too many bands where the quantization sets all spectral values to zero, i.e. audible holes in the frequency domain will occur. This problem can be solved with the help of an additional strategy to avoid holes. The bands that must not be quantized to zero are selected. The value of the increased threshold , which is determined by the previous equation, in such bands must not exceed the energy  in this band diminished by a minimum signal to noise ratio . This is done by building the minimum:



The minimum requirements  are frequency dependent and will be calculated for the given bitrate on initialization of the encoder. First the number of average bits per channel and transform block  have to be converted in a perceptual entropy  by calculating:



Of this total  60% are equally divided among the number of active barks , that is determined by the maximum bandwidth of the AAC encoder.



For each scalefactor band the corresponding part of the perceptual entropy called  is calculated by converting  to the width in Bark of the appropriate band. With the following equation  is calculated from the :



The value  holds the number of spectral lines in the scalefactor band .

Finally the value of  is limited to a maximum of 25dB and a minimum of 1dB.

A signal dependent modification of the minimum requirements is performed by increasing the minimum distance between energy and threshold on local peaks and decrease it for local valleys of the scalefactor band energy .

In case of M/S stereo another modification of  takes place. Depending on the energy difference between mid and side channel the requirements for the weaker channel will be released.

##### 5.6.1.1.3 Relation between bit demand and perceptual entropy

As there is only one quantization and the used bits will only be counted thereafter, the reduction strategy works with an estimation of these used bits. This is done by using the so called perceptual entropy PE. The PE used in this encoder is computed per scalefactor band:



with , , . The estimated number of lines that won't be zero after the quantization is called . This number is derived from the form factor  that is also needed by the scalefactor estimator:



The total PE  of one frame is the sum of the scalefactor band perceptual entropies:



To get a more linear relation between PE and the number of bits needed a constant value  is added to the scalefactorband perceptual entropies. The value for  is determined at initialization time:



with  as the bitrate per channel in bits per second

An approximation for the conversion from actual needed bits to perceptual entropy is:



#### 5.6.1.2 Calculation of Bit Demand

Since the AAC encoder uses a bitreservoir technique, the number of bits used for the actual frame will be variable. The bit demand of the current frame depends on:

- the current fullness of the bitreservoir, a number between 0 (empty) and 1 (full)

- the relative difficulty of the frame, a measure for this is the perceptual entropy  based on the unmodified psychoacoustic thresholds

The steps to calculate the bit demand for the current frame are:

With help of the bitreservoir fullness, the variables  and  are calculated according to the two figures below:



Figure 3: Calclation of bitSave



Figure 4: Calclation of bitSpend

The parameters are different for long and short blocks:

Table 3: Parameters for bitreservoir control

|  |  |  |
| --- | --- | --- |
| Parameters for bitSave calculation | | |
|  | Long block parameters | Short block parameters |
| clipLow | 0.2 | 0.2 |
| clipHigh | 0.95 | 0.75 |
| minBitSave | -0.05 | 0 |
| maxBitSave | 0.3 | 0.2 |
| Parameters for bitSpend calculation | | |
|  | Long block parameters | Short block parameters |
| clipLow | 0.2 | 0.2 |
| clipHigh | 0.95 | 0.75 |
| minBitSpend | -0.1 | -0.05 |
| maxBitSpend | 0.4 | 0.5 |

A factor  is calculated out of ,  and the current perceptual entropy . For a relatively low perceptual entropy (easy to encode frame) this means that bitfac is less than 1 and bits will be put to the bitreservoir. If the perceptual entropy is above the average which is an indication of a diffcult frame, bits will be taken out of the bitreservoir (bitfac>1). See the next figure on how to calculate this factor.



Figure 5: Calclation of bitFac

The variables  and  are adjusted after each calculation of .

The desired number of bits for the actual frame is:



where  is the average number of bits per frame matching the given constant bitrate

and  is the actual number of bits in the bitreservoir.

#### 5.6.1.3 Calculation of the reduction value

The actual problem of the reduction strategy is to find the loudness value  of the equation defined in section 5.6.1.1.2. so that the requirements of the granted bits resp. the appropriate PE  are fulfilled. In the following an iterative process to find the reduction value  and the thresholds used for the quantization is described.

##### 5.6.1.3.1 Preparatory steps of the perceptual entropy calculation

The calculation of the scalefactor band PEs  can be split up in a constant part  and a variable, threshold dependent part .



with 

and 

It is possible to calculate the constant part once at the start of the iteration.

##### 5.6.1.3.2 Calculation of the desired perceptual entropy

The goal to spend the number of bits calculated with the method described above in 5.6.1.2 is approximately achieved by increasing the tresholds in such a way that the resulting perceptual entropy equals the desired PE . The desired PE is determined by the relation between bit demand and perceptual enropy (5.6.1.1.3. ).



The actual number of used bits will only approximately match the desired bits. Small differences will be balanced out with the help of the bitreservoir. Systematic differences are compensated by applying a correction factor to the desired perceptual entropy. This factor is obtained by taking into account the real relation between number of used bits and perceptual entropy of the previous frames. The allowed range of the correction factor is between 0.85 and 1.15.

##### 5.6.1.3.3 Selection of the bands for avoidance of holes

First all bands are marked to participate in an eventually occurring avoidance of spectral holes. Then by means of different criteria individual bands are excluded.

- For long blocks no avoidance of holes in band , if the spreaded energy  exceeds the energy 

- For short blocks no avoidance of holes in band , if the spreaded energy  exceeds the energy 

- The minimum requirement  has to be greater than 0dB

For a definition of the scalefactor band energy  and the spreaded energy , see section 5.4.2.1 resp. section 5.4.3 .

##### 5.6.1.3.4 First Estimation of the reduction value

In the following an approximation for the total PE is used. It is derived from

the equation of the scalefactor band PE:



with  as the sum of the constant parts of  and  as the total number of estimated lines that will be unequal zero after quantization. The estimated average total threshold is .

In the first iteration the loudness  of this average threshold is calculated with help of the PE  without reduction ():



The estimation of the reduction value  follows from the desired PE :



With this  the new thresholds  can be calculated using the equation from section 5.6.1.1.2. and also the resulting PE .

##### 5.6.1.3.5 Second Estimation of the reduction value

Usually the value of  will be greater than the desired PE . In scalefactor bands that avoid holes after the first iteration the thresholds can not be reduced further. By repeating the two equations above a second guess  for the reduction value can be found, if only the bands that actually do not avoid holes are considered. Therefore the contributions of bands with active avoidance of holes have to be subtracted from , , and . The modified values are then called , , and . The loudness  of a new average threshold is calculated by:



The second estimation of the reduction value  is:



Using , new thresholds  and PE  can be calculated.

The calculation of  may be repeated once more if the absolute difference between desired and actual PE is greater than 5%.

##### 5.6.1.3.6 Final threshold modification by linearization

If the PE resulting from the second iteration  is already close to the desired value  (that means the difference to the desired PE is less than 15%), the desired PE can be reached by a linearization of the logarithms.

The formula for the PE after the second guess is:



Accordingly the desired PE  can be written as:



with  as the difference between the reduction value  and the latest guess .

The difference of the two perceptual entropies is:



A linearization of the logarithm around the zero results to the following:



Now the difference of the total PE can be divided among the individual bands,

that actuallly do not avoid holes:



with



For each band a final modification of the threshold is performed:



##### 5.6.1.3.7 Further perceptual entropy reduction

If the conditions for section 5.6.1.3.6. can not be reached, i.e. the actual perceptual entropy  exceeds the desired  by more than 15%, then it seems that the constraints given by the minum requirements  or the number of bands with active avoidance of holes are too strong for the desired PE.

In a first step the values of  are limited to a maximum value of 1dB starting from the scalefactor band with the highest frequency. By doing so, thresholds can be increased and the perceptual entropy will decrease.

If the actual perceptual entropy is still too large after having changed  for the whole spectrum, more spectral holes have to be allowed. In case of M/S stereo always the scalefactor band of the channel with less energy is now quantized to zero. Afterwards for mono and stereo subsequently bands with low engergies get erased. Therefore the bands are partitioned into four categories with different energy levels. Starting from the highest band all bands falling in the category with the lowest category get erased. This process is eventually repeated with the next energy categories, until the resulting perceptual entropy is as small as the desired value of .

##### 5.6.1.3.8 Possible failures

In general the difference of the resulting perceptual entropy to the desired is negligible. The described algorithm works fine for reasonable combinations of bitrate, samplerate and bandwidth. Normally inaccuracies especially in the relation between the perceptual entropy and the really used bits, can be balanced out by the bitreservoir. But there is no guarantee that there are always enough bits available to fulfill the requirements of the increased threshold. For measures that are available to avoid an abort of the encoder in such cases, see section 5.6.4 .

### 5.6.2 Scalefactor determination

The scalefactors determine the quanization step size for each scalefactor band. By changing the scalefactor, the quantization noise will be controlled. The equation for the quantization of the spectral coefficients is:



 is defined to 0.4054 and  is one of the spectral coefficients that is calculated from the MDCT filterbank. In the following three steps of combined scalefactor determination and quantization, always  is calculated. Only at the end, scalefactors  and the  values are derived from the values for .

The formula for the inverse quantization is:



It is needed for calculating the quantization noise.

#### 5.6.2.1 Scalefactor Estimation

A first guess of  that results in quantization noise approximately equal to the threshold  is given by the following equation:



with the form factor  that is also needed by the calculation of the perceptual entropy (see 5.6.1.1.3. ).

#### 5.6.2.2 Scalefactor Improvement by Quantization

The following steps of scalefactor changes include always a quantization and inverse quantization procedure to be able to calculate and compare the quantization error. The equation for the distortion is:



After quantizing with the scalefactor value calculated in 5.6.2.1, the resulting distortion  may be greater than the threshold . By trying increased and decreased values for  a lower distortion is searched for. If the distortion was already below the threshold, a search will only be done for smaller values of  to try to further improve the distortion.

#### 5.6.2.3 Scalefactor Difference Reduction

Above each scale factor band is treated individually. The next algorithms take into acount that finally the difference of the scalefactors will be encoded. A smaller difference between two adjacent scale factors costs less bits.

In a first step it is searched for single scalefactor bands, where the number of bits gained by using a smaller  is greater than the estimated increased bit demand for the noiseless coding of the quantized spectral coefficients. The estimation of needed bits for the noiseless coding is based on the equation for the perceptual entropy:



with , , . With  the estimated number of lines that won't be zero after the quantization is meant. This number has already been calculated during the adaptation of the thresholds to the bitrate, for a definition of  see 5.6.1.1.3.

If such a band is found and in addition the quantization error is smaller, the new value for  is accepted.

In a second assimilation step the same procedure as above is repeated but now trying to increase the  values for a complete region of scale factor bands instead of improving only single bands.

#### 5.6.2.4 Final scalefactor determination

The conversion from  to scalefactor values  and a value for  is done after limitation of the maximum scalefactor difference to a value of 60. This is achieved by limiting all values of  to a maximum allowed value of  with  as the minium over all bands.

The value for  is now chosen as the maximum of all  values and the scalefactors result in:



### 5.6.3 Noiseless coding

Coding of the quantized spectral coefficients is done by the noiseless coding. The encoder uses a so called greedy merge algorithm to segment the 1024 coefficients of a frame into section and to find the best huffman codebook for each section. For the sectioning and the huffman codebooks see [2].

### 5.6.4 Out of Bits Prevention

Only after the MDCT values are quantized according to the increased thresholds  and after the following noiseless coding, the number of really needed bits is counted. If this number is too high, the number of bits have to be reduced. This is achieved by increasing the global gain value and a new quantization of the whole spectrum plus additional noiseless coding in a loop until the bit demand is small enough to match the constraints of the bitreservoir.

Annex A (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **TSG SA#** | **TSG Doc.** | **CR** | **Rev** | **Subject/Comment** | **Old** | **New** |
| 2004-09 | 25 | SP-040635 |  |  | Approved at SA#25 Plenary | 2.0.0 | 6.0.0 |
| 2006-06 | 32 | SP-060360 | 0002 |  | Modification of written specification: Change of encoder bitrate border for Parametric Stereo usage | 6.0.0 | 7.0.0 |
| 2008-12 | 42 |  |  |  | Version for Release 8 | 7.0.0 | 8.0.0 |
| 2009-12 | 46 |  |  |  | Version for Release 9 | 8.0.0 | 9.0.0 |
| 2011-03 | 51 |  |  |  | Version for Release 10 | 9.0.0 | 10.0.0 |
| 2012-09 | 57 |  |  |  | Version for Release 11 | 10.0.0 | 11.0.0 |
| 2014-09 | 65 |  |  |  | Version for Release 12 | 11.0.0 | 12.0.0 |
| 2015-12 | 70 |  |  |  | Version for Release 13 | 12.0.0 | 13.0.0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2017-03 | 75 |  |  |  |  | Version for Release 14 | 14.0.0 |
| 2018-06 | 80 |  |  |  |  | Version for Release 15 | 15.0.0 |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | **16.0.0** |
| 2022-04 | - | - | - | - | - | Update to Rel-17 version (MCC) | **17.0.0** |
| 2024-03 | - | - | - | - | - | Update to Rel-18 version (MCC) | **18.0.0** |