**3GPP TSG-SA4 Meeting #134 *S4-252005***

**Dallas, United States, 17th Nov 2025 - 21st Nov 2025**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.4* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.253** | **CR** | **0026** | **rev** | **1** | **Current version:** | **19.0.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*https://www.3gpp.org/Change-Requests*](https://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Updates to the IVAS algorithmic description | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Dolby Laboratories Inc., Ericsson LM, Fraunhofer IIS, Huawei Technologies Co Ltd., Nokia, NTT, Orange, Panasonic Holdings Corporation, Philips International B.V., Qualcomm Incorporated, VoiceAge Corporation | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | IVAS\_Codec | | | | |  | ***Date:*** | | | 2025-11-19 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **A** |  | | | | | ***Release:*** | | | Rel-19 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The completion of the fixed-point C code and the corresponding update to the floating-point C code include corrections that are missing from the algorithmic description. In addition, several identified errors and some previous omissions are corrected. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | * Updates relating to object-based audio operation (also in conjunction with MASA/SBA) including object editing description * Output format processing for mono/stereo streams, specifying related algorithmic delays in the overview table * Default room reverb description for binaural rendering, room effect parameter adjustment, clarifying referencing of binary file format for parametrization of binaural renderers * Reference to EVS mono decoding of interoperable mono streams * Corrections and clarifications on some DTX operations * Clarification on split rendering configurations * Editorial corrections | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Discrepancies between C code specifications and algorithmic description will be introduced. Existing errors and omissions will remain. These aspects can confuse implementors. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 4.2.1, 4.3.2, 4.4, 5.3.5.2, 5.8.1, 5.9.6.3.4, 6.3.3.1, 6.3.5.2, 6.6.1, 6.6.6.1, 6.9.12 (new), 6.10 (new), 7.2.1.5 (new), 7.2.2.2.6, 7.2.2.3.1, 7.2.2.3.5, 7.2.2.3.11 (new), 7.2.2.4.3.2, 7.2.2.6 (new), 7.2.3 (new), 7.4.7.2, 7.4.8.1, 7.4.8.4 (new), 7.4.10 (new), 7.6.2.2 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **x** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **x** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **x** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | Rev 1: Clarification of change in clause 7.6.2.2, clause 7.4.7.2 affected, corrected cover page | | | | | | | | |

CHANGE 1

# 2 References

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.441: "Codec for Enhanced Voice Services (EVS); General Overview".

[3] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[4] 3GPP TS 26.447: "Codec for Enhanced Voice Services (EVS); Error concealment of lost packets".

[5] 3GPP TS 26.448: "Codec for Enhanced Voice Services (EVS); Jitter Buffer Management"

[6] 3GPP TS 26.250: "Codec for Immersive Voice and Audio Services (IVAS); General overview".

[7] 3GPP TS 26.251: "Codec for Immersive Voice and Audio Services (IVAS); C code (fixed-point)".

[8] 3GPP TS 26.252: "Codec for Immersive Voice and Audio Services (IVAS); Test Sequences".

[9] 3GPP TS 26.254: "Codec for Immersive Voice and Audio Services (IVAS); Rendering".

[10] 3GPP TS 26.255: "Codec for Immersive Voice and Audio Services (IVAS); Error concealment of lost packets".

[11] 3GPP TS 26.256: "Codec for Immersive Voice and Audio Services (IVAS); Jitter Buffer Management".

[12] 3GPP TS 26.258: "Codec for Immersive Voice and Audio Services (IVAS); C code (floating-point)".

CHANGE 2

### 4.2.1 General

The IVAS codec is an extension of the 3GPP Enhanced Voice Services (EVS) codec [2]. It provides full and bit exact EVS codec functionality for mono speech/audio signal input. It further provides:

- Encoding and decoding of stereo and immersive audio formats such as multi-channel audio, scene-based audio (Ambisonics), metadata-assisted spatial audio (MASA), object-based audio (ISM), and their combination.

- VAD/DTX/CNG for rate efficient stereo and immersive conversational voice transmissions

- Error concealment mechanisms to combat the effects of transmission errors and lost packets. Jitter buffer management is also provided.

- The IVAS codec operates on 20-ms audio frames. In addition, rendering is possible with 5-ms granularity.

- Support for bit rate switching upon command.

- Stereo and immersive audio coding at the following discrete bit rates [kbps]: 13.2, 16.4, 24.4, 32, 48, 64, 80, 96, 128, 160, 192, 256, 384, and 512, with supported bit rate ranges listed in Table 4.2‑1.

- Support for WB, SWB and FB audio, with the supported bitrate range listed in Table 4.2‑2.

Table 4.2‑1: Ranges of supported bitrates for stereo and immersive coding of the IVAS codec

|  |  |
| --- | --- |
| Input audio format | Range of supported bitrates [kbps] |
| Stereo | 13.2 – 256 |
| Scene-based audio (SBA) | 13.2 – 512 |
| Metadata assisted spatial audio (MASA) | 13.2 – 512 |
| Object-based audio (ISM) | 13.2 – 512 |
| Multi-channel audio (MC) | 13.2 – 512 |
| Combined ISM and MASA (OMASA) | 13.2 – 512 |
| Combined ISM and SBA (OSBA) | 13.2 – 512 |

Note, that for the object-based audio format (ISM) the range of supported bitrates varies based on the number of objects as follows: 13.2 kbps – 128 kbps for 1 ISM, 16.4 kbps – 256 kbps for 2 ISMs, 24.4 kbps – 384 kbps for 3 ISMs, 24.4 kbps – 512 kbps for 4 ISMs.

Table 4.2‑2 Supported audio bandwidth per input audio format and bitrate

|  |  |  |  |
| --- | --- | --- | --- |
| Input audio format | Bitrates supporting WB  [kbps] | Bitrates supporting SWB  [kbps] | Bitrates supporting FB  [kbps] |
| Stereo | 13.2 – 256 | 13.2 – 256 | 32 – 256 |
| Scene-based audio (SBA) | 13.2 – 512 | 13.2 – 512 | 32 – 512 |
| Metadata assisted spatial audio (MASA) | 13.2 – 512 | 13.2 – 512 | 32 – 512 |
| Object-based audio (ISM), 1 object | 13.2 – 128 | 13.2 – 128 | 16.4 – 128 |
| Object-based audio (ISM), 2 objects | 16.4 – 256 | 24.4 – 256 | 32 – 256 |
| Object-based audio (ISM), 3 objects | 24.4 – 384 | 24.4 – 384 | 48 – 384 |
| Object-based audio (ISM), 4 objects | 24.4 – 512 | 24.4 – 512 | 64 – 512 |
| Multi-channel audio (MC) | 13.2 – 512 | 13.2 – 512 | 32 – 512 |
| Combined ISM and MASA (OMASA) | 13.2 – 512 | 13.2 – 512 | 32 – 512 |
| Combined ISM and SBA (OSBA) | 13.2 – 512 | 13.2 – 512 | 32 – 512 |

CHANGE 3

### 4.3.2 Input/output audio formats

The IVAS coder accepts the following input audio formats:

- single-channel mono audio format denoted as

- two-channel stereo and binaural input audio format, where the left channel is denoted as and the right channel is denoted as

- scene-based (ambisonic) input audio format, where the ambisonic order is denoted as and the number of individual input channels is  . The individual input channels are stored in the ACN component ordering, denoted as where is the ambisonic degree. In case of the first-order ambisonic format (FOA), the individual input channels may also be denoted as:

For both, first-order ambisonic signals and higher-order ambisonic signals (HOA), the notation may be simplified to:

- object-based input audio format (ISM), where the number of objects is denoted as and the individual “streams” related to the objects are denoted as where . Each individual stream is associated with its input metadata signal, denoted as where is the frame index.

- multi-channel input audio format (MC), where the individual channels are denoted as where . The input channels correspond to one of the following loudspeaker layouts, where the loudspeaker positions are assumed to have azimuth and elevation as per ISO/IEC 23091-3:2018 Table 3 [24], unless otherwise noted below. The channel order is as per ISO/IEC 23008-3:2015 Table 95 [25]:

Table 4.2‑4: Ordering of input channels in MC format

|  |  |
| --- | --- |
| **loudspeaker layout** | **CICP layout** |
| 5.1 | CICP6 |
| 5.1+2 | CICP14, 35°elevation |
| 5.1+4 | CICP16, 35°elevation |
| 7.1 | CICP12 |
| 7.1+4 | CICP19, 35°elevation |

- metadata-assisted spatial audio (MASA) format, where for mono-MASA (MASA1), the mono channel is denoted as and, for stereo-MASA (MASA2), the left channel is denoted as and the right channel is denoted as . The MASA audio channel(s) is/are associated with the MASA metadata that is denoted as where is the frame index

- objects with metadata-assisted spatial audio (OMASA), which is a combination of MASA and object-based input audio format (ISM)

- objects with scene-based (ambisonic) input audio format (OSBA), which is a combination of SBA and object-based input audio format (ISM)

The IVAS codec is capable to output the audio into the following output formats:

- single-channel mono audio format

- two-channel stereo audio format

- two-channel binaural audio format (Binaural output without room acoustic synthesis, Binaural output with room acoustics synthesized using impulse responses, Binaural output with room acoustics synthesized using parametric reverberator)

- scene-based (ambisonic) audio format (ambisonics order *l* = 1, 2, 3)

- multi-channel audio format (loudspeaker layouts listed in Table 4.2-4 + custom layouts)

- split-rendering intermediate bitstream (supported for all IVAS formats except of stereo and EVS mono)

The overview of supported combination of input/output formats is provided in Table 4.4-1.

In addition, a pass-through operation is supported at which the codec outputs the audio in an output format corresponding to the input format including a separate metadata (when available). This operation primarily intends to skip the internal renderering and use of an external or a custom renderer. A pass-through operation is supported for all input audio formats, and it is also referred as an external (EXT) processing output.

CHANGE 4

## 4.4 Algorithmic delay

The input signals (audio, or audio and metadata) are processed using 20-ms frames. The codec algorithmic delay depends on the input/output audio formats as described in Table 4.4‑1.

**Table 4.4‑1: IVAS algorithmic delay for different input/output format combinations (rounded to integer milliseconds; in case multiple values are provided they depend on the bitrate)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Decoder output format | | | | | |
|  |  | Mono | Stereo | Multi-Channel | Binaural audio | Scene-based audio | External (EXT) output |
| Encoder input format | Mono | 32 | 32\* | 32\* | 32\* | 32\* | 32\*  (mono) |
| Stereo (NOTE1 | 32 | 32 | 32 | 32\* | 32\* | 32  (stereo) |
| Multi-channel audio | 32 | 32 | 32 / 37 | 32 / 37 | 32 / 37 | 32 / 37  (MC audio) |
| Scene-based audio | 33 | 33 | 38 | 38 | 38 | 38  (SBA audio) |
| Object-based audio | 32 | 32 | 32 / 37 | 32 / 37 | 32 / 37 | 32 / 37  (object-based audio) |
| Metadata-assisted spatial audio (NOTE2 | 32 / 37 | 37 | 37 | 37 | 37 | 32  (MASA audio) |
| OSBA | 33 | 33 | 38 | 38 | 38 | 38  (object-based + SBA audio) |
| OMASA | 32 / 37 | 37 | 37 | 37 | 37 | 32  (object-based + MASA audio) |

NOTE1: Stereo input format also includes binaural input format.

NOTE2: Metadata-assisted spatial audio (MASA) decoder output allows also for mono or stereo decoder output at 32 ms algorithmic delay by stripping the metadata file.

NOTE3: Mono and Stereo outputs to other formats marked with [\*] are further described in clauses 7.2.1.5, 7.2.2.6 and 7.2.3.

The algorithmic delay related to the core-coder coding in IVAS is 32 ms similarly as in EVS though its splitting between the encoder and the decoder is slightly different. It consists of 8.75 ms for the encoder look-ahead and 3.25 ms for the decoder delay related to the time-domain BWE and resampling in the DFT domain.

Further, the IVAS delay consists of 5 ms delay related to the rendering to the related output configuration, thus making the overall delay of 32 ms in some set-ups and 37 ms in other set-ups.

Finally, in SBA format, an additional encoder delay of 1 ms is present and it is related to the filter-bank analyses prior to the encoding.

It is also noted that the delay figures exclude any HRIR/BRIR induced delay.

The codec delay for mono (EVS) operation and stereo downmix operation in EVS compatible operation is 32 ms described in clause 4.3 in [3].

CHANGE 5

5.3.5.2 DTX in MDCT-based stereo

5.3.5.2.1 Overview

DTX operation in MDCT-based stereo is based on the FD-CNG functionality from EVS. It is extended by encoding frequency-dependent information about the coherence between the two channels in the SID frame to enable generation of stereo comfort noise with the same frequency-dependent coherence in the decoder. Additionally, an efficient way of transmitting the parametric background noise description for both channels is employed to encode the spectral shape parameters for both channels of the background noise with low additional bitrate demand.

Similar to the EVS, the MDCT-based stereo DTX coding employs a more efficient DTX coding (“regular”) strategy at bitrates lower or equal to 80 kbps while a conservative coding strategy is used at bitrates higher than 80 kbps. The conservative coding means that the DTX segments correspond to signals with a very low energy.

5.3.5.2.2 VAD in MDCT-based Stereo

The stereo signal is analyzed by a voice activity detector that determines a frame to be an inactive frame or an active frame. The activity detector analyzes the two channels separately to classify them as either active or inactive. Then, it determines the whole frame to be inactive if both the first channel and the second channel are classified as inactive. Otherwise, the frame is classified as active. The decision to classify a single channel as active or inactive is done the same as described in clause 5.2.2.2.5.

The VAD functionalities in each of the channels operate completely independent of each other. This also includes the mechanism for determining when to send the next SID frame in variable-DTX-update-interval operation (see clauses 5.6.1.1 and 5.6.1.2 of [3]). To prevent the two channel VADs to go out of synch, a synchronization mechanism is employed as depicted in Figure 5.3-46. If any of the two channels is classified as active, the current frame is classified as active and will be encoded using the discrete channel encoding of MDCT-based stereo as described in clause 5.3.3. If any of the two channel VADs decides on sending an SID in the current frame, the current frame will be an SID frame, even if the other channel’s VAD decides that based on the other channel, the current frame would be a NO\_DATA frame.

****

**Figure 5.3‑46: Frame type decision in MDCT-based Stereo DTX**

5.3.5.2.3 Coherence estimation

Frequency-dependent coherence information for the background noise signal is measured in inactive frames by calculating the coherence between the left and the right channel signal in 5 frequency bands. The bands are adapted to the psychoacoustic ERB scale. They are non-overlapping with monotonically increasing size and were derived by combining multiple of the stereo bands defined in table 5.3-17 which follow the ERB scale approximately. The first band starts at 50 Hz, so the calculated coherence parameters only define the coherence between the channels above this frequency. The coherence is calculated using the FFT spectrum from the pre-processing spectral analysis calculated on the 12.8 kHz downsampled signal, so the bands only cover that frequency range. See table 5.3-25 for the band definitions.

**Table 5.3‑25: Frequency bands for the background noise coherence estimation**

| **Band** |  |  |
| --- | --- | --- |
| **1** | 2 | 6 |
| **2** | 8 | 8 |
| **3** | 16 | 16 |
| **4** | 32 | 16 |
| **5** | 48 | 80 |

A coherence calculator is run on the two input channel spectra in each frequency band in every inactive frame to calculate a coherence value for the respective band. The calculation is done the same in each band and only differs in which part of the spectrum is processed (according to the band definitions in table 5.3-25). It is explained for a single band here and applied analoguously for the other bands. In each of two subframes, four intermediate values (a real intermediate value, *creal*, and an imaginary intermediate value, *cimag*, as well as a first energy value for the first channel, *eL*, and a second energy value for the second channel, *eR*) are calculated as

(5.3-325)

(5.3-326)

(5.3-327)

(5.3-328)

where *L* and *R* denote the subframe's FFT spectrum as calculated in clause 5.1.5 of [3]. The final coherence value is calculated using the real intermediate value, the imaginary intermediate value, the first energy value and the second energy value after smoothing all the intermediate values. The smoothed values are calculated as

(5.3-329)

(5.3-330)

(5.3-331)

(5.3-332)

where, , and denote the respective intermediate values as calculated for the previous frame. If the current frame is the first frame to be encoded or a bit rate switch has happened from a unified stereo bit rate, all the previous-frame values are initialized to 10-15. The coherence value is calculated using the smoothed values as

(5.3-333)

In the first 50 frames after starting the encoding process, the coherence calculation is always run regardless of the frame classification as active or inactive to initialize the buffers and converge to more reasonable default values in case no inactive frame is encountered for a longer period at the start of encoding.

The coherence value for each band is quantized as

(5.3-334)

and encoded in the SID bitstream using four bits each.

5.3.5.2.4 Noise parameter estimation and encoding

Information about the spectral shape of the background noise is derived by a noise parameter calculator that calculates first parametric noise data for the first channel of the stereo signal and second parametric noise data for the second channel of the stereo signal. The parameter calculator consists of the noise estimation algorithm as described in clause 5.6.3.2 of [3] which is run separately on the two channels of the stereo signal to generate two sets of parametric noise data - *NL,FD-CNG*for the left channel and *NR,FD-CNG*for the right channel. Additionally, the noise parameter calculator is configured to convert the first parametric noise data and second parametric noise data from a left/right representation to a mid/side representation. This conversion is applied after converting the parametric noise data values to dB as

(5.3-335)

(5.3-336)

From these, parametric noise data in a mid/side representation are calculated:

(5.3-337)

(5.3-338)

This parametric noise data is encoded using the MSVQ described in clause 5.6.3.5 of [3] with a modified first stage. For encoding the mid noise data, all 6 stages of the modified MSVQ are used while for the side noise data only the first four are used. The modified first stage MSVQ operation for FD-CNG is described in clause 5.2.2.3.5.2.

The output of the MSVQ decoder is denoted as *NSIDM,FD-CNG(i)* and *NSIDS,FD-CNG(i)*, respectively. The MSVQ output is reconverted from the mid/side representation back into a left/right representation. If the energy of the unquantized side noise data *NdBS,FD-CNG* is smaller than 0.1, all values of *NSIDS,FD-CNG* are set to zero prior to reconverting to the left/right representation. A flag (the "no-side" flag) is encoded in the SID bitstream to signal this to the decoder. If the energy of the unquantized side noise data *NS,FD-CNG* is smaller than 0.1, the no-side flag is one, otherwise it is zero. The value is encoded using a single bit. Thus, the reconverted parametric noise data for each channel in left/right representation is calculated as

(5.3-339)

(5.3-340)

From the reconverted left/right representation of the parametric noise data, a first global gain value for the first channel and a second global gain value for the second channel are calculated using equation (1399) of [3] with and for the left channel and and for the right channel. The global gain values for both channels are then quantized according to equation (5.2-240).

CHANGE 6

### 5.8.1 OSBA format overview

The encoder supports combined input with 1 – 4 ISMs and an SBA signal of order 1 – 3. Depending on the IVAS total bitrate and number of ISMs, different OSBA coding modes summarized in Table 5.8-1 are employed to combine these input signals. In OSBA format, DTX is not supported and only the first group of ISM metadata (i.e., azimuth and elevation) is considered.

**Table 5.8-1: Overview of coding modes in OSBA format**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IVAS bitrate**  **[kbps]** | **number of ISMs** | | | |
| **1** | **2** | **3** | **4** |
| **13.2 – 80** | Pre-rendering | Pre-rendering | Pre-rendering | Pre-rendering |
| **96** | Discrete | Pre-rendering | Pre-rendering | Pre-rendering |
| **128 – 512** | Discrete | Discrete | Discrete | Discrete |

Input to IVAS, consisting of audio signals in SBA and ISM formats and the associated metadata, is processed through a simplification stage and an encoding stage. At the simplification stage, the SBA and ISM signals are converted into a mezzanine format, as described in clauses 5.8.2 and 5.8.3. In the pre-rendering OSBA mode, the mezzanine format is the First order Ambisonics (FOA) format. In the discrete OSBA mode, the mezzanine format includes an Ambisonics signal and all discrete ISM objects. At the encoding stage, the audio output of the simplification stage is encoded into the IVAS bitstream which is then transmitted to the decoder.

CHANGE 7

##### 5.9.6.3.4 Encoding of ISM metadata

In the one separated audio object with parametric representation coding mode, the directional metadata associated with the audio objects are quantized and encoded. As mentioned in clause 5.9.6.3.1 when the IVAS codec is operating in the one separated audio object mode the audio object stream is configured into a stream consisting of the separated audio object and a stream consisting of the remaining audio objects and the MASA format audio.

The MASA-to-total energy ratio represents the distribution of the MASA format contribution to the total audio environment for a TF tile, while the ISM energy ratio for each remaining audio object represents the distribution of the remaining audio objects within the audio object contribution to the total audio environment. As a result, after the quantization and encoding of the MASA-to-total energy ratio and of the ISM energy ratios, the distribution of each object within the total audio content for each TF tile is known. Based on this information, an object priority order is determined that is further used to determine the bit allocation for the directional metadata associated with each audio object.

The priority order of each audio object is obtained by:

(5.9-1)

where is the ISM ratio for audio object for the TF tile , and is the MASA-to-total energy ratio for the TF tile . This is performed for all audio objects which encompasses all remaining audio objects and the separated audio object.

In addition, the priority value for the separated audio object with index , (where is an index from the set is determined separately based on the above determined priority values of all the audio objects:

(5.9-2)

where is the ISM importance class for the separated audio object as assigned in clause 5.6.2.3.2 based on the audio signal type given by the *coder\_type*.

The number of bits allocated for the quantization of the directional metadata for each audio object, i.e., for all audio objects including the separated audio object, is given by:

(5.9-3)

where the index incorporates the index of the separated audio object and stands for the integer part operation.

For each audio object, quantization of the audio directional parameter for the current frame, where the directional parameter consists of an azimuth value and an elevation value, is performed as follows:

If the bit allocation is less than 8 bits, the directional parameter of the audio object is compared to the directional parameter of the same audio object from the previous frame. If they are the same, that is within a threshold of 0.01 degrees then one bit is sent to signal that they are the same. If at least one of elevation value or azimuth value of the directional parameter is different from the corresponding elevation value or azimuth value of the previous frame’s directional parameter, then one bit is sent to signal they are different, and the current frame’s directional parameter is quantized with the spherical grid quantizer (described in clause 5.2.4.3.2) using the above allocated number of bits.

If the bit allocation for the audio directional parameter of the audio object is higher or equal to 8, then the audio directional parameter is quantized and encoded using a spherical grid quantizer based the allocated number of bits as described in clause 5.2.4.3.2.

CHANGE 8

6.3.3.1 General Overview

The decoding process for the MDCT-based stereo comprises of the decoding of all the side parameters, namely, the parameters of the tools like TNS, SNS, TCX LTP, the IGF parameters and the stereo parameters. Then the bitrate ratio decoded from the bitstream determines the number of bits to be read for each channel and the decoding of the spectral data follows using the range decoder. After noise filling, IGF is applied, the inverse discrete stereo processing takes place and resulting intermediate signal is de-normalized using the global ILD value parsed by the bitstream. Finally, spectral noise shaping is applied to de-whiten the signals, TNS filtering is also applied, if it is active, either before or after the dewhitening process and the inverse MDCT transform is carried out to obtain the original signals of the two channels in time-domain.

CHANGE 9

6.3.5.2 DTX in MDCT-based stereo

6.3.5.2.1 General overview

In MDCT-based stereo, the SID payload consists of parametric noise data for the two input channels in a mid/side representation, frequency-dependent coherence information indicating how correlated the background noise at the encoder is and a one-bit value indicating whether the energy of the side representation of the noise data is lower than a threshold (the "no-side flag").

For inactive frames, comfort noise is generated by first decoding the coherence information consisting of one coherence control parameter per frequency band and the parametric noise data from the bitstream. After generating the first and the second channel noise signals by mixing three uncorrelated noise signals according to the coherence information, the two noise signals are spectrally shaped according to spectral shape parameters derived from the parametric noise data,

6.3.5.2.2 Decoding of parametric noise data

Decoding of the noise data follows the MSVQ decoding approach described in clause 5.3.5.2.4. For decoding the mid noise data, all 6 stages of the MSVQ are used while for the side noise data only the first four are used. The no-side flag is represented by a single bit and thus read directly from the bitstream. If the value of the no-side flag is 1, the mid noise data vector is set to zero. Then, the MSVQ-decoded parametric noise data is reconverted from a mid/side representation to a left/right representation to derive the spectral shape parameters for both channels. The reconstructed parametric noise data for each channel is calculated as

(6.3-181)

(6.3-182)

Before generating the actual stereo CNG signal, processing steps described in clauses 6.7.3.1.2 and 6.7.3.1.3 of [3] are applied separately on the noise data for each channel.

6.3.5.2.3 Stereo CNG

Stereo Comfort noise is generated band-wise using the frequency bands defined in table 5.3-25. The two channels in each frequency band of the CNG output signal are generated using a stereo signal generator which consists of three uncorrelated gaussian noise sources and a mixer to mix the generated noise signals controlled by the coherence parameter for the respective frequency band decoded from the last SID.

****

**Figure 6.3‑18: Stereo noise signal generator for CNG in MDCT-based Stereo DTX**

The comfort noise generation process in a frequency band is illustrated in Figure 6.3‑18. For each of the two channels, a separate gaussian noise source generates a separate noise signal (*N1(i)* or *N3(i)*, respectively, the "channel noise signals") with both noise signals being decorrelated between each other. A third gaussian noise source (the "mixing noise source", *N2(i)*) generates a third noise signal that is also decorrelated from each of the two other noise signals. The two channel noise signals are then mixed with the mixing noise signal, i.e., the mixing noise signal is added to both channel noise signals after weighting all of the signals. The coherence value for the respective frequency band decoded from the last received SID frame serves as a control parameter for the weighting. With *Nl(i)* being the noise signal for the left channel, *Nr(i)* the noise signal for the right channel and *coh* being the coherence value for the respective frequency band, mixing of the noise signals in a frequency band is done according to:

(6.3-183)

(6.3-184)

, where *i* goes from *ilow* to *ihigh* for the frequency band as defined in table 5.3-25.A higher coherence between the channels thus leads to more correlated noise being generated in the channels for the respective frequency band, while a lower coherence value leads to a higher amount of uncorrelated noise in the stereo output for the respective frequency band. This way, the comfort noise signal has a frequency-dependent coherence as given by the coherence information decoded by from the SID which was derived from the background noise seen in the encoder and a similar spatial expression is achieved.

Since the FD-CNG configurations of EVS are re-used in IVAS, no comfort noise is generated below 50Hz as defined in clause 5.6.3.1.3 of [3]. The spectral values below are set to zero for generated noise signals.

Next, both *NL* and *NR* are spectrally shaped according to the parametric noise data decoded from the SID by employing FD-CNG as described in clause 6.7.3.3 of [3]. This is done separately on both channels.

CHANGE 10

### 6.6.1 Discrete ISM decoding mode

Figure 6.6‑1 is a schematic block diagram illustrating the ISM decoder in the DiscISM mode.



Figure 6.6‑1: Block diagram of the DiscISM decoder

The bitstream demultiplexer receives a bitstream which is in the structure from Figure 5.6‑4. When the IVAS format corresponds to the ISM format, the following is read from the bitstream in a sequential order: a) ISM common signaling incl. the number of audio streams, *NISM*, ISM importance classes, *classISM*[*n*], and metadata presence flags, *flagmeta*[*n*], *n*= 0, …, *NISM* – 1, b) the coded metadata for *NISM* streams, c) core-coder payloads for *NISM* streams. It is noted that the ISM mode is not part of the bitstream in active frames, but it is derived from the number of coded streams *NISM* and the *ism\_total\_brate* parameter.

Once the metadata are decoded, the information about respective bit-budgets and ISM classes per stream are supplied from the metadata processing module to the configuration module which comprises the bit-budget allocator. The bit-budget allocator at the decoder uses the same procedure as in the bit-budget allocator at the encoder to determine the core-decoder bitrates (see clause 5.6.2.3). The *NISM* decoded metadata – before they are supplied to the renderer – are then optionally subject to the object metadata editing, in which the metadata parameter values are modified based on metadata controlling as described in clause 7.4.10. Next, the *NISM* transport channels from the bitstream demultiplexer are sequentially decoded using *NISM* fluctuating bitrate SCE core-decoders (clause 6.2.3.2). These core-coder channels (corresponding to the transport channels) are finally supplied to the renderer.

It is noted that Figure 6.6‑1 contains arrows indicating “output set-up” parameters. These parameters are e.g. output audio configuration, output sapling rate, etc. and they are used for simplifying some steps during the decoding process.

CHANGE 11

#### 6.6.6.1 Metadata handling in bitrate switching

The direction metadata is encoded independent of the bitrate conditions. However, gain metadata coding operation is only active for bitrates higher than 64 kbps. Therefore, a mechanism is introduced to ensure smooth transition between the detail level of metadata in case of bitrate switching. Two detail levels are described for this operation. To support varying detail levels of metadata, the metadata parameters are initialized to azimuth , elevation , radius , yaw and pitch , matching the initialization done in the encoder as described in 5.6.4.3.

In the case of level 1 detail of metadata, only the common metadata (azimuth and elevation) decoder is active. If the second or extended level of metadata detail is used, the radius and the orientation angles and are also decoded with extended metadata decoder. Note that the first level of detail, azimuth and elevation is a subset of the extended level of detail. The level of detail for metadata to be decoded is indicated by the parameter received via bitstream, . If is set, the encoded metadata includes the extended metadata, level 2 detail.

A variable to track the state of the level of metadata is introduced as . Active state variable, , determines the detail level of the metadata to be used for rendering. At the beginning of the decoding process, is set to -1 to indicate the decoding process of the first frame.

A counter is introduced to keep track of the number of received frames from different levels of detail. The level 1 detail of metadata is always encoded and decoded independent of the transmission conditions. However, some parts of the level 2 detail are only encoded for high bitrates (≥ 64 kbps). In the event of the changes in the bitrate conditions, the metadata change counter, keeps track of the duration for such changes before updating the active state, . The counter, controls the status of the metadata detail state and allows the updates in the metadata memory accordingly via the variable .

Before the decoding process, all metadata parameters (level 1 and level 2) are set to their default values and stored in the metadata memory.

For the first received frame, indicated by is set to 'zero' and the state is updated with the bitstream level, :

(6.6-49)

(6.6-50)

If indicates level 2 detail, the metadata (both level 1 and 2) is decoded according to the level encoded, their values are used for rendering and stored in the memory.

In the case of indicating level 1 detail, the level 2 metadata parameters (yaw, pitch, and radius) are reset to their default values to be used in the rendering process. The metadata memory is updated with the default values of level 2 metadata in this case. The level 1 metadata parameters are updated according to the bitstream in the memory and those values are used for rendering.

For the static bitrate condition where the metadata detail state currently being used is the same with the bitstream metadata detail level, i.e., , the operations follow the logic in the previous clause with Equations (6.6-49) and (6.6-50).

If the received metadata detail level is different from the current state, when , it corresponds to the change in the bitstream conditions. In this case, is incremented by one:

.

(6.6-51)

Following the incrementation, is subjected to a threshold of 5 to see if the change is persistent over a period (5 frames in this case). If counter , meaning the change has been effective for less than 5 frames, the metadata values from the memory are used for rendering. The update on the level 2 metadata parameters is withheld for a short time window. When the counter reaches the threshold, meaning , Equations (6.6-49) and (6.6-50) are applied followed by the same operations in the first frame and the renderer uses the received and decoded metadata. In case level 1 metadata is received, the default values are used for the extended metadata in level 2. The renderer uses the received and decoded values of level 2 detail metadata when indicates level 2 detail. The metadata memory is updated with the metadata values used for rendering.

CHANGE 12 (new clause)

### 6.9.12 OMASA object editing

#### 6.9.12.1 Overview

The OMASA format supports editing of the objects at the decoder. Object editing is supported both for Param OMASA (see clause 6.9.4) and Disc OMASA (see clause 6.9.5) decoding modes. The direction and the gain of the objects can be manipulated. Furthermore, the gain of the MASA part can be manipulated to fully edit the decoded OMASA scene.

In object editing, object editing input information, described in clause 7.4.10.2, and associated MASA gain editing information received via a decoder interface are used to edit the decoded object metadata parameter values and the MASA gain. However, these input values are not applied directly, instead they go through a conditional application decision described in clause 6.9.12.2.

Object editing in the Param OMASA decoding mode is described in clause 6.9.12.3, and object editing in the Disc OMASA decoding is described in clause 6.9.12.4.

#### 6.9.12.2 Conditional application of object editing

To avoid perceptual artefacts due to excessive object manipulation, conditional application of object editing is used for OMASA. The decision if an edit of an object metadata parameter (azimuth, elevation, or gain) or MASA component gain should be applied or prevented is determined by obtaining the edit control information defining the edited value, determining the magnitude of the change caused by the edit, comparing this with a threshold value, and selecting to edit or to prevent editing the metadata parameter based on the result of this comparison.

The threshold values used in the comparison are defined as follows. The edit threshold for the MASA component gain is , and the edit threshold for the gain of object is . In the Disc OMASA decoding mode, the azimuth angle edit threshold is , and the elevation angle edit threshold is . In the Param OMASA decoding mode, the azimuth and elevation angle threshold values and are determined based on the quantization resolutions of the azimuth and elevation parameter values which depend on the number of bits used to code the spherical index describing the azimuth and elevation angle values of object . The number of bits is compared with the values in the column “No. bits” in table 5.2-31 and the matching row is selected. The value from column is assigned to , indicating the elevation angle quantization resolution. The index is determined with denoting the rounding to the nearest integer. The element at position from the list in the column "No. azimuth for each elevation” is selected and assigned into . This value indicates the number of distinct values in the quantized representation of azimuth angle. The edit threshold for the azimuth angle value is determined based on the quantization resolution computed from the number of distinct azimuth values with

The edit threshold for the elevation angle value of the object can be determined based on the quantization resolution with

The edit control values for the parameters object azimuth angle , elevation angle , object gain , and MASA component gain are obtained. These are used to indicate an expected change in the spatial metadata parameter. The real-valued object azimuth and elevation values are rounded to the closest integer values and .

An indication of the magnitude of the expected spatial metadata parameter change is obtained by subtracting the editing control value from the original spatial metadata parameter value and taking the absolute value:

The edit control information containing the magnitude of the expected change is compared with the corresponding edit threshold value. The object azimuth edit magnitude is compared with the azimuth angle edit threshold and the object elevation edit magnitude for that object is compared with the elevation edit threshold for that object . If one or both magnitudes are larger than the corresponding edit threshold, the edit control values are the edited values for the direction of the object

Otherwise, there is no editing of the direction of object .

The object gain edit magnitude is compared with the gain edit threshold . If the magnitude is larger than the threshold, the edit control value is the edited value for the gain of the object

Otherwise, there is no editing of the gain of the object .

The MASA gain edit magnitude is compared with the gain edit threshold . If the magnitude is larger than the threshold, the edit control value is the edited value for the gain of the MASA component

Otherwise, there is no editing of the gain of the MASA component.

Conditional application of object spatial metadata edit is not performed in Disc OMASA decoding mode when using binaural output or for the object metadata values for EXT output.

#### 6.9.12.3 Editing in Param OMASA decoding mode

In the Param OMASA decoding mode, a Param OMASA spatial audio stream is obtained. The spatial audio stream contains two transport audio signals (the audio signals containing an object portion and a non-object (i.e., other spatial audio) portion), metadata associated with the audio signals (containing object metadata (which defines audio object portion positions and audio object portion energy proportions), MASA metadata, and MASA-to-total energy ratio metadata), and one separated object audio signal.

In addition, object position control information, object gain control information, and MASA gain control information are obtained. The object position control information contains modified (or edited) positions for the audio objects in the form of edited object azimuth for the current frame and edited object elevation for the current frame, where is the object index. The object gain control information contains edited gain for the audio object portion for the current frame. The MASA gain control information contains edited gain for the other spatial audio portion for the current frame.

The obtained audio signals are transformed to the time-frequency domain (see clause 6.2.5 for details), as is done in “normal” rendering as well (see, e.g., clause 7.2.2.3.1), resulting in for the transport audio signals and for the separated object audio signal, where is the frequency bin index, is the temporal slot index, and is the transport audio signal channel index.

The two time-frequency domain transport audio signals are processed based on the obtained object position control information , , object gain control information , MASA gain control information , and the obtained metadata associated with the two transport audio signals. The processing includes determining position and gain processing information based on the obtained position and gain control information and the obtained metadata (including audio object portion positions and energy proportions). This processing is described in clause 6.9.12.5. This resulting processed time-frequency domain transport audio signals are denoted .

The gain of the separated object audio signal is modified using the edited object gain

where is the object index corresponding to the separated object audio signal.

Spatial audio signals are rendered using the processed time-frequency domain transport audio signals , the gained separated object audio signal , object position control information , , and the obtained metadata associated with the two transport audio signals. The rendering is performed as described in clause 6.9.7, but the edited object positions , are used instead of the non-edited object positions , , the processed time-frequency domain transport audio signals are used instead of the non-processed time-frequency domain transport audio signals , and the gained separated object audio signal is used instead of the non-gained separated object audio signal.

#### 6.9.12.4 Editing in Disc OMASA decoding mode

In the Disc OMASA decoding mode, a Disc OMASA spatial audio stream is obtained. The spatial audio stream contains two transport audio signals (the audio signals containing the non-object portion), metadata associated with the two transport audio signals (containing MASA metadata), audio signals (the audio signals containing the object portion), and object metadata (containing object position).

In addition, object position control information, object gain control information, and MASA gain control information are obtained. The object position control information contains edited object azimuth for the current frame and edited object elevation for the current frame, where is the object index. The object gain control information contains edited object gain for the current frame. The MASA gain control information contains edited MASA gain for the current frame.

The gains of the audio signals related to MASA are modified using the edited MASA gain

where is the transport audio signal channel. The gains of the audio signals related to objects are modified using the edited object gain

Spatial audio signals are rendered using the gained audio signals and , object position control information , , and the obtained metadata (MASA and object metadata). The rendering is performed as described in clause 6.9.7, but the edited object positions , are used instead of the non-edited object positions , , and the gained audio signals are used instead of the non-gained audio signals.

#### 6.9.12.5 Stereo signal pre-processing for object editing

Stereo transport audio signal pre-processing is performed by determining mixing information (containing mixing gain values) based on the obtained object position control information (containing modified object positions), object gain control information, audio object positions, audio object energy proportions and MASA gain control information, and processing the stereo transport audio signals based on the determined mixing information. The object position control information contains edited object azimuth for the processed frame and edited object elevation for the processed frame, where is the object index. The object gain control information contains edited object gain for the processed frame. The MASA gain control information contains edited MASA gain for the processed frame.

First, the non-edited stereo amplitude panning gains and the edited stereo amplitude panning gains the for each object and stereo transport audio channel are obtained, based on the original audio object positions (azimuth and elevation ) and edited audio object positions (azimuth and elevation ) as described in clause 7.2.2.3.6 (for “stereo” mode operations).

Then, panning energies , for non-edited and edited directins for each object and stereo transport audio channel are obtained, based on the determined stereo amplitude panning gains respectively:

Energy in MASA frequency bands and subframes for stereo transport audio channel are determined based on the input stereo transport audio signal by:

The total energy of stereo transport audio signal in MASA frequency bands and subframes is determined:

Then, gain processing information (in the form of ISM target energies ) is determined based on the gain control information and the audio object portion energy proportion , as follows. For each object, the total original object energy and total original object energy per transport audio channel is determined:

where is the rendering direct-to-total energy ratio for the object (determined in clause 6.9.6 based on the audio object portion energy proportions and the MASA-to-total energy ratios ).

Target object energy is then determined based on the obtained object gain control information, i.e., the audio object portion gain and the total original object energy :

The gain processing information in the form of ISM target energies for each transport audio channels is determined by:

This gain processing information is then used to render spatial audio, using the audio signals and the metadata defined in clause 6.9.12.3, as follows. Based on the obtained ISM target energy values, target energy of each transport channel is then calculated:

Furthermore, original MASA energy per each channel is determined:

where is the total number of ISM objects. The new target MASA energy per each transport audio channel is then determined:

Based on the obtained MASA target energy, target energy of each transport channel is further calculated:

The metadata is also processed based on the gain control information. New direct-to-total ISM ratios for the audio object portion and new direct-to-total MASA ratios for the other spatial audio portion are determined based on the original ratios as follows.

New total ratio value is given as:

where

New ratios for ISMs are then determined as:

And for MASA:

Based on the determined target energies of transport audio channels, total target energy = is determined. In addition, normalized target energies for both transport audio channels are determined.

If centering is enabled, centering factor is determined: , and panning energies are modified: .

Based on the object editing control information amount of moved energy, , between stereo transport audio channels is determined:

And further preserved energy is estimated:

From the normalized target channel energies, object parts are subtracted:

Any remaining non-object energy is estimated and added to the preserved energy estimate:

This enables to substantially not move energy for the non-object portion, at least for one of the channels.

Normalization value for scaling moved and preserved energies is determined, based on the estimated total target energy, original energy, moved energy and preserved energy:

Where . Moved and preserved energies per channel are then normalized , .

The target total energy, preserved energies and moved energies are then temporally averaged:

Stereo transport audio mixing information, in form of a preprocessing matrix containing four mixing gain values (direct and cross channel mixing values), is then determined based on the temporally averaged energies:

In addition, the mixing information having the mixing gain values is further temporally interpolated, before applying it to process the input signal containing the audio object and other spatial audio portions as a mixture, by , where . This processing based on the mixing information enables the object portion in one these two audio signals to be at least partially moved to another of these two audio signals.

The equalization gain is determined based on the target total energy and temporally averaged realized energy after applying the preprocessing matrix:

where

Determined equalization gain is then applied to the output signal:

The gain applied output signal is then subsequently used to render a spatial audio signal using the associated MASA metadata.

CHANGE 13 (new clause)

## 6.10 EVS-compatible mono audio operation

The IVAS codec supports mono operation with EVS compatibility by incorporating EVS functionality in a bit-exact manner. The EVS-compatible mono audio decoder operation is described in clause 6 of [3].

In addition to direct mono decoding, IVAS supports upmix rendering of the decoded mono to multichannel (described in clause 7.2.1.5), binaural (described in clause 7.2.2.6), and ambisonics (described in clause 7.2.3.1).

CHANGE 14 (new clause)

#### 7.2.1.5 Multichannel Upmix for Mono and Stereo inputs

##### 7.2.1.5.1 Rendering mono to stereo

For the special case of rendering mono to stereo, the non-diegetic panning gain according to clause 5.6.4.4 is used with an azimuth of zero. This means the mono input audio is copied to both stereo output channels with a gain factor of 0.5 according to eqs. 5.6-52 and 5.6-53.

##### 7.2.1.5.2 Rendering to supported IVAS loudspeaker layouts

The rendering for mono and stereo follows the procedures described in clause 6.7.7 for multichannel output format conversion. This rendering takes place in the time domain, similar to what is performed for the discrete MC decoding mode. Specifically, all supported IVAS output multichannel formats are supersets of mono (CICP1) and stereo (CICP2), therefore channel rerouting according to the superset upmix case of clause 6.7.7.4.3 is performed. Mono is routed to the center loudspeaker (CICP1) and stereo to the frontal left/right speakers (CICP2).

##### 7.2.1.5.3 Rendering to a custom loudspeaker layout

Rendering to a custom loudspeaker layout is supported and it is performed as described in clause 6.7.7.5.

CHANGE 15

##### 7.2.2.2.6 ITD synthesis

The ITD synthesis adjusts the timing of the signals such that the desired ITD is achieved in the rendered signal. For audio segments where the ITD remains the same, this can is realized by buffering and delaying the signal with the later time of arrival. To keep the delay at a minimum, the signal with the later time of arrival is delayed while the other channel is rendered with zero delay. When the ITD value changes, a time scaling operation needs to be performed in order to change the alignment of the channels. In case the ITD value changes sign, this means the delayed channel needs to be adjusted to zero delay and the other channel is adjusted to be delayed with the new target ITD.

For each object signal , either input from the decoder or fed to the external renderer, the new object signal frame is fed into a processing buffer. The samples of memory from the preceding frame is appended in front of the new signal frame. The length of the memory is

(7.2-49)

where is the maximum ITD that can be synthesized and is the number of samples used in the polyphase resampling stage. First, a buffer length is calculated to leave look-ahead room for the resampling filter according to

(7.2-50)

where is the ITD value of the current subframe , is the total transition time in samples and is the length of the 5 ms subframe. The ITD transition is set to complete within the subframe, which means is the maximum allowed transition length. At 48 kHz sampling rate, . If the sign of the ITD did not change from the previous subframe, or one of them is zero , the time scaling operation only needs to be done on one channel. In that case the number of transition times in samples and are calculated according to

(7.2-51)

where denote the starting indices of each time shift segment assuming that the current input subframe starts at , is the length of the resampling segments and . If the previous and current subframe ITD is non-zero and changes sign the transition times and are calculated according to

(7.2-52)

where denotes rounding to the nearest integer. If , it means the target ITD cannot be reached and the target ITD is updated to . Next, the output buffers A and B are assembled by time-shifting the signal using the transition lengths and as illustrated in figure 7.2‑7. First, a resampling is done of the buffer starting from of length to the first samples of output buffer A. The last samples are populated by copying the remainder of the input buffer to output buffer A. Then, the first samples of the input buffer starting from index 0 are copied to the first samples of output buffer B. The next samples of output buffer B are created by resampling the samples starting from in the input buffer of length . Finally, the last samples of the input buffer are copied to output buffer B. The last samples of the input frame are stored in memory for processing the next subframe. Output buffers A and B are assigned to the output channels 0 and 1 for left and right HRIR filtering respectively. The assignment of the output channels is done based on the signs of and as follows,

(7.2-53)

where denotes logical AND and denotes inclusive OR. The resampling operations are implemented using a polyphase filter with a sinc function from a lookup table, thereby performing the ITD transition and generating the two output buffers . The output buffers populate the corresponding output object channels and with the target ITD applied.

Figure 7.2‑7: Resampling and copying operations when changing ITD.

CHANGE 16

##### 7.2.2.3.1 Overview

The parametric binauralizer and stereo renderer operates on the following IVAS formats and operations: MASA, OMASA, multi-channel (in McMASA mode), SBA, OSBA, and ISM, i.e., the input to the encoder has been spatial audio (containing audio signal(s), and spatial metadata in case of MASA, OMASA, OSBA, and ISM formats) in one of these formats, and it is now being rendered to binaural or stereo output. The IVAS format being processed (i.e., whether operating on MASA, OMASA, multi-channel (in McMASA mode), SBA, OSBA, or ISM format) is determined by decoding it from the received IVAS bitstream (see clause 6.1), and the renderer obtains (or receives) this information (i.e., the IVAS format).

The parametric binaural (and stereo) rendering is performed in subframes, where denotes the subframe index. A subframe containsCLDFB slots (in non-JBM operation, , in JBM operation 7). The data determined at the rendering of the previous subframes (i.e., subframes -1 and earlier) affects the rendering of the present subframe due to temporal averaging and interpolation. The binaural renderer system described in the following is also capable to render a stereo signal instead of a binaural signal. Also, binaural sound with and without room effect can be reproduced.

As an input, the renderer obtains (or receives) a spatial audio signal containing one or two audio signals (i.e., transport audio signals) and associated spatial metadata. The number of transport audio signals (i.e., one or two) is determined by decoding it from the received IVAS bitstream (see clause 6.1), and the renderer receives this information (i.e., the number of transport audio signals). The spatial metadata obtained (or received) by the renderer contains the following parameters in frequency bands : direction (as azimuth and elevation ), direct-to-total energy ratio , spread coherence , and surround coherence In case of the SBA and OSBA formats, the spatial metadata contains SPAR metadata. The audio signals and the spatial metadata are used for providing spatial audio reproduction (i.e., to enable the rendering of spatial audio).

In addition, the input to the renderer includes a separated centre channel audio signal when the IVAS format is multi-channel and operating in McMASA “separate channel” mode. In addition, the input to the renderer contains a separated object audio signal and associated object metadata when the IVAS format is OMASA and operating in “one object with MASA representation coding mode” or “parametric one object coding mode”. In addition, the input to the renderer contains all object audio signals and associated object metadata when the IVAS format is OMASA and operating in “discrete coding mode”.

Furthermore, head orientation data and external orientation data may be received. Head tracking is described in clause 7.4.3, external orientation input is described in clause 7.4.5, and rotation combining functionality for determining the final rotation data to be used in the binaural rendering is described in clause 7.4.6. Head tracking (and external orientation information) should be used if they are available for improved spatial audio experience. In the following it is referred to head orientation and head tracking for simplicity regardless of which components the orientation data is originally composed of, and which processing has been applied to it prior to the rendering step.

In addition, the renderer (when rendering binaural output) obtains a room effect control indication, and based on this indication, it is determined whether to apply a room effect to the audio signal(s) (of the spatial audio signal) or to not apply the room effect. For rendering binaural audio with a room effect, two data sets related to binaural rendering are obtained. The first is a pre-defined data set containing spherical harmonics to binaural conversion matrices. It has been created based on head related impulse responses or transfer functions (HRTF). The second data set contains reverberation early part energy correction gains (which are used for modifying the resulting spectrum that is obtained from the rendering according to the first data set), and late reverberation energy correction gains and reverberation times. It has been created based on binaural room impulse responses or transfer functions (BRIR). Thus, the binaural output signal is generated based on a combination of the spatial audio signal and the mentioned two data sets, i.e., the binaural output signal is generated using the transport audio signal(s) and associated metadata according to a combination of these two data sets.

The binaural audio signal with the room effect is rendered by generating a first part binaural signal based on the transport audio signal(s) and the spatial metadata, generating a second part binaural signal by applying room effect to the transport audio signal(s) (using a reverberator), and combining the first part binaural audio signal and second part binaural signal to generate a combined binaural audio signal (see clause 7.2.2.3.5 for details).

CHANGE 17

##### 7.2.2.3.5 Processing audio signals with the processing matrices

The binaural (or stereo) audio output is produced by generating different binaural audio signal parts (direct input signal part, decorrelated input signal part, separated centre/object channel part, and room-effect part) and combining the different binaural audio signal parts to generate a combined binaural audio signal (which is the output of the processing)

where

where is the transport audio signals, is the one or more separate channel/object signals (if there is none of them, it is set to zero), is a decorrelated version of , denotes applying a room effect using a reverberator, the processing coefficients in ,andare based on the spatial metadata as described in the foregoing clauses, and is an interpolation value determined by

As an exception, when the IVAS format is OMASA, and the ISM mode is ISM\_MASA\_MODE\_MASA\_ONE\_OBJ or ISM\_MASA\_MODE\_PARAM\_ONE\_OBJ. Otherwise is determined using the equation above.

The decorrelated audio signals are generated by processing the transport audio signal(s) as described in clause 7.2.2.3.8.

is generated by processing the input signals with a room effect when the room effect indication says to apply it. Applying the room effect is described in clause 7.3.3. The reverberator is initialized based on the late reverberation energy correction gains and reverberation times obtained from the data set in variables *parametricReverberationEneCorrections* and *parametricReverberationTimes* in the reference C code. In case the reverberation times are smaller than a determined threshold value, the room effect parameters (early part energy correction gains, late reverberation energy correction gains, and reverberation times) are adjusted as described in clause 7.2.2.3.11. When the room effect indication says to not apply a room effect, is set to zero.

The time-frequency domain binaural (or stereo) audio signal is converted to the time domain via the inverse CLDFB (see clause 6.2.5 for details), yielding .

CHANGE 18 (new clause)

##### 7.2.2.3.11 Room effect parameter adjustment

Room effect parameters, which include reverberation times (where is the frequency bin index), early part energies (or, in other words, early part energy correction gains), and late part energies (or, in other words, late reverberation energy correction gains), are obtained, and they are adjusted when the reverberation time is very small, to avoid spectral artefacts with the synthetic reverberator. For each reverberation time RT60 within the room effect parameters, it is first checked whether the reverberation time RT60 is less than a determined threshold value of 0.2, and when this is the case, the room effect parameters are adjusted as follows.

A reverberation time modifier parameter is formulated by

Then, an adjusted reverberation time is formulated by

An energy modification value is formulated by

Then, the adjusted early part energy correction coefficient and the adjusted late part energy correction coefficient are formulated by

The binaural spatial audio signal is then (see clauses 7.2.2.3.1, 7.2.2.3.3, and 7.2.2.3.5) generated using the adjusted room effect parameters (reverberation times), (early part energy parameters), and (late part energy parameters), based on the received spatial audio stream having audio signal(s) and associated spatial metadata. The process is performed in two parts, i.e., two spatial audio portions are generated, which are then combined. The first spatial audio portion (direct/early reverberation portion) is generated using HRTF processing based on the spatial metadata and the audio signals so that the early part energy parameters of the adjusted room effect parameters affect the resulting spectrum. The second spatial audio portion (late reverberation portion) is separately generated based on the audio signal using a reverberator that is configured based on the adjusted room effect parameters (reverberation times and late part energies).

CHANGE 19

###### 7.2.2.4.3.2 Model parameter derivation

The above processing strongly depends on the choice of the model parameters:

 the length of the delay lines of each frequency band,

 the number of taps in the delay lines

 the positions of the taps along the delay line

 the phase factors

 the gains and

 the attenuation factors

These parameters are determined from the BRIRs

The delay-line length is determined according to the formula with =.

The number and positions of the taps, and , the phase factors , and the reverb equalization gains are determined randomly according to the following pseudo-code, which is run for each band index .

for each of L and R channels

for each bin with index k up to

{

energyBuildup = 0

currentEnergy = 1

= 0;

for each time slot index in the delay line sample up to

{

intendedEnergy += currentEnergy;

/\* The randomization at the energy build up affects where the sparse taps are located \*/

energyBuildup += currentEnergy + 0.1 \* random\_number

if ( energyBuildup >= 1.0f ) /\* A new filter tap is added at this condition \*/

{

/\* Four efficient phase operations: n\*pi/2, n=0,1,2,3 \*/

= n\*pi/2 with random number 1, 2, 3 ,or 4

/\* Set the tap pointer to point to the determined sample at the loop buffer \*/

=

energyBuildup -= 1.0f; /\* A tap is added, thus remove its energy from the buildup \*/

++;

actualizedEnergy += 1.0f;

}

currentEnergy \*= ;

}

If (== 0)

{

/\* Ensure at least 1 filter tap. \*/

= n\*pi/2 with random number 1, 2, 3 ,or 4

=

actualizedEnergy = 1.0f;

}

}

= /\* Determined reverb spectrum \*/

/\* Correction of random effects at the decorrelator design \*/

/\* Correction of IIR decay rate \*/

The sequence of pseudo-random numbers is generated in a reproducible way by the following simple C-language function.

static uint16\_t binRend\_rand(

REVERB\_STRUCT\_HANDLE hReverb /\* i/o: binaural reverb handle \*/

)

{

hReverb->binRend\_RandNext = hReverb->binRend\_RandNext \* 1103515245 + 12345;

return (uint16\_t) ( hReverb->binRend\_RandNext / 65536 ) % 32768;

}

The initial value of binRend\_RandNext is 1. The attenuation factor for each CLDFB band k is calculated according to the formula

=

with the number of CLDFB time slots in the reverberation time The loop attenuation factor is given by

The RT60 per CLDFB band is listed in Table 7.2‑11. They have been extracted from the measured BRIRs

Table

7.2‑11

: RT60 per CLDFB band

0.429201, 0.205110, 0.202338, 0.208383, 0.215664, 0.236545, 0.230598, 0.228400, 0.227467, 0.218956, 0.226083, 0.220702, 0.221501, 0.223471, 0.223705, 0.227063, 0.227899, 0.223071, 0.220000, 0.218583, 0.220417, 0.218250, 0.213250, 0.210333, 0.207417, 0.198750, 0.196250, 0.194917, 0.190333, 0.184500, 0.180333, 0.176167, 0.176500, 0.177583, 0.183583, 0.195917, 0.203250, 0.208417, 0.214667, 0.220000, 0.222917, 0.230417, 0.233928, 0.233647, 0.236333, 0.237428, 0.241629, 0.241118, 0.238847, 0.242384, 0.246292, 0.245948, 0.246100, 0.245396, 0.243951, 0.244123, 0.239270, 0.241474, 0.234824, 0.253040

The long-tail reverberation energy is given by = . The values found from the filters used in IVAS are listed in Table 7.2‑12.

Table

7.2‑12

: per CLDFB band

0.000584, 0.000210, 0.000233, 0.000212, 0.000257, 0.001518, 0.001154, 0.001097, 0.001265, 0.001298, 0.002320, 0.002432, 0.002686, 0.002702, 0.002632, 0.002564, 0.002732, 0.002727, 0.002609, 0.002524, 0.003417, 0.001783, 0.000987, 0.000699, 0.000606, 0.000536, 0.000511, 0.000569, 0.000600, 0.000543, 0.001257, 0.001209, 0.000957, 0.000601, 0.000274, 0.000106, 0.000072, 0.000051, 0.000040, 0.000030, 0.000024, 0.000018, 0.000014, 0.000013, 0.000012, 0.000011, 0.000009, 0.000009, 0.000008, 0.000008, 0.000007, 0.000006, 0.000005, 0.000003, 0.000002, 0.000002, 0.000001, 0.000001, 0.000000, 0.000000

CHANGE 20 (new clause)

#### 7.2.2.6 Binaural Upmix for Mono and Stereo inputs

Mono and Stereo inputs support all IVAS binaural output configurations (except for split rendering). For mono, the same upmix to stereo is performed as described in clause 7.2.1.5, which is then passed through to binaural output. For stereo, no rendering is performed, and the stereo output is passed through to binaural output.

CHANGE 21 (new clause)

### 7.2.3 Rendering to other supported output formats

#### 7.2.3.1 Mono and Stereo rendering to ambisonics

The IVAS decoder supports rendering of mono and stereo inputs to ambisonics. For mono, a routing to the omni (W, channel index 0) channel of ambisonics is performed, with the remaining channels zeroed out. For stereo, a rendering is performed to the W and Y (channel index 1) channels according to the following (equivalent to clause 7.5.2.2 for loudspeakers at ±90º):

CHANGE 22

#### 7.4.7.2 Parametrization of Binaural renderers using binary file

Head related filters for the binaural rendering may be provided to the decoder or the renderer by using dynamic loading of external binary file. The way to generate such a binary file from a set of SOFA file and the binary file format are described in TS 26.258 [12], clause 5.10.

CHANGE 23

#### 7.4.8.1 Overview

The late reverb is driven by the set of parameters comprising of:

- RT60 – indicating the time that it takes for the reflections to drop 60 dB in energy level,

- DSR – diffuse to source signal energy ratio,

- Pre-delay – delay at which the computation of DSR values was made. Can be interpreted as the threshold between early reflections and late reverberation phase.

Spatialized, rotation-responsive, first-order early reflections can be added when using multichannel input (any configuration accepted). The early reflections rendering is determined by several parameters that drive a shoebox model using the image-source method. The set of parameters consists of:

- 3D rectangular virtual room dimensions,

- Broadband energy absorption coefficient per wall,

- Listener origin coordinates within room (optional),

- Low-complexity mode (optional) – favours efficient early reflection rendering over spatial accuracy.

Figure 7.4‑9 illustrates the main reverberation properties with relevant reverberation synthesis control parameters indicated.



Figure

7.4‑9

: Simple representation of the main reverberation properties

Room acoustics parameters are provided to the renderer as metadata. Two metadata formats are supported in the IVAS decoder/renderer implementation: binary renderer config metadata format, and text renderer config metadata format (see [12], clause 5.14.1 and clause 5.14.2 respectively). Regardless of the metadata format, the general metadata processing is shared, as discussed in clause 7.4.8.2 for late reverb, and in clause 7.4.8.3 for early reflections. Both metadata formats support multiple acoustic environment datasets, allowing for selecting between such acoustic environments.

CHANGE 24 (new clause)

#### 7.4.8.4 Default binaural room parameters

Room acoustics parameters are provided to the renderer as metadata. In absence of a specific metadata input or other selection to overwrite the default, the IVAS renderer supports three sets of default parameters corresponding to three room sizes: small, medium, and large. A default room size is allocated for each of the IVAS encoder input formats as specified in table 7.4-6.

Table 7.4‑6: Default binaural room size per input audio format

|  |  |
| --- | --- |
| Input audio format | Default room size |
| Scene-based audio (SBA) | SMALL |
| Metadata assisted spatial audio (MASA) | SMALL |
| Object-based audio (ISM) | LARGE |
| Multi-channel audio (MC) | MEDIUM |
| Combined ISM and MASA (OMASA) | MEDIUM |
| Combined ISM and SBA (OSBA) | MEDIUM |

CHANGE 25 (new clause)

### 7.4.10 Object editing

#### 7.4.10.1 Overview

Object editing, or object manipulation, is a functionality that relates to the object-based audio (ISM) format and the combined formats OMASA and OSBA that utilize ISMs in addition to their underlying spatial audio formats. Object editing refers to the modification of at least one ISM rendering parameter value for at least one ISM: azimuth, elevation, and gain. In addition, the combined formats allow for editing of the rendering gain of the spatial audio (MASA or SBA) separate of the ISM(s) for full rendering control of the spatial audio scene components.

Object editing is supported by the IVAS decoder through a dedicated interface for controlling the relevant metadata parameter values. Edited metadata is furthermore supported for the EXT processing output. As the IVAS external renderer directly operates on IVAS input formats (clause 4), manipulation of the metadata prior to rendering, e.g., using IVAS decoding to EXT output, is possible and no separate object editing functionality within the renderer is required.

Object editing requires separation of the manipulated audio object(s) at the decoder. Due to use of downmixing in certain operations, availability of separated object(s) for editing depends on the coded format, number of ISMs, and bit rate as detailed in table 7.4-6.

The gain adjustment range associated with each input audio format is furthermore limited based on the ISM coding (parametric or discrete) as described in table 7.4-7.

Table 7.4-6: Object editing bit rate ranges in IVAS decoder per input format and number of ISMs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input audio format | Object editing bit rates [kbps] | | | |
| 1 ISM | 2 ISMs | 3 ISMs | 4 ISMs |
| Object-based audio (ISM) | 13.2 – 128 | 16.4 – 256 | 24.4 – 384 | 24.4 – 512 |
| Combined ISM and MASA (OMASA) | 24.4 – 512 | 32 – 512 | 64 – 512 | 64 – 512 |
| Combined ISM and SBA (OSBA) | 96 – 512 | 128 – 512 | 128 – 512 | 128 – 512 |

Table 7.4-7: Object editing gain ranges per input format and coding mode

|  |  |  |  |
| --- | --- | --- | --- |
| Input audio format | Object editing gain ranges [dB] | | |
| Minimum gain | | Maximum gain |
| Parametric | Discrete |
| Object-based audio (ISM) | -24 | -Inf | 12 |
| Combined ISM and MASA (OMASA) | -24 | -Inf | 12 |
| Combined ISM and SBA (OSBA) | - | -Inf | 12 |

#### 7.4.10.2 ISM object editing

ISM object editing capability is used for the ISM format and for the objects of the combined formats, OSBA and OMASA (with certain modifications for OMASA). Object editing consists of position editing and gain editing. Object editing input information, for object , thus comprises the following elements:

* object position editing input information, i.e., object azimuth and object elevation for the current frame, and
* object gain editing input information for the current frame.

When an object is edited, the decoded object azimuth and elevation metadata values are replaced by the corresponding editing input values. For rendering of the object(s), the gain(s) of the separated ISM audio signal(s) are modified using the object gain editing input as follows:

In addition to the editing of azimuth, elevation and gain, the editing of extended metadata is supported for Discrete ISM, OMASA Discrete ISM and OSBA Discrete ISM modes. The extended metadata consists of radius, yaw and pitch parameters. Radius describes the distance of the object from the origin of the IVAS coordinates whereas yaw and pitch parameters describe the orientation of the objects. The editing of extended metadata follows the same logic of object editing where the decoded metadata values are replaced by corresponding editing values.

The ISM rendering is then performed based on the updated signals and metadata values. EXT processing output is also supported, where the edited values are used.

#### 7.4.10.3 OSBA object editing

Object editing for the OSBA format supports ISM object position and gain manipulation. In addition, the gain of the SBA part can be controlled as part of OSBA object editing.

OSBA object editing utilizes the ISM object editing in clause 7.4.10.2 for manipulation of the separated object(s).

#### 7.4.10.4 OMASA object editing

Object editing for the OMASA format supports ISM object position and gain manipulation. In addition, the gain of the MASA part can be controlled as part of OMASA object editing.

OMASA object editing is supported both for Param OMASA (see clause 6.9.4) and Disc OMASA (see clause 6.9.5) decoding. As part of the decoding operations, a conditional modification of the object editing (and MASA gain editing) input information is carried out. While the general approach of ISM object editing in clause 7.4.10.2 for manipulation of the separated object(s) is used, the OMASA object editing operations are described in more detail in clause 6.9.12.

CHANGE 25

#### 7.6.2.2 Supported Split Rendering bitrates with LCLD or LC3plus codec

NOTE: Operation of the LC3plus and LCLD codecs as part of IVAS codec operations is limited to the configurations defined in Tables 7.6‑3 and 7.6-4 including operation 48\_6 of LC3 basic audio profile according to clause 7.6.5.7 of this specification. This results in the supported bitrate range of 240 to 512 kbps for the transport codecs. The presence of other modes of operation in the reference source code [7], [12] or the LC3plus specification [28] does not imply their applicability for IVAS split rendering or any other IVAS codec operations.

Table 7.6-3: Supported Split Rendering bitrates with the LC3plus codec

|  |  |  |  |
| --- | --- | --- | --- |
| LC3plus Configuration | DoF | Split rendering total bitrate (kbps) | Split rendering frame size (ms) |
| 5ms or 10ms frame duration | 0 | 256, 384, 512 | 5 or 10 |
| 5ms or 10ms frame duration | 1-3 | 384, 512, 768 | 20 |

Table 7.6-4: Supported Split Rendering bitrates with the LCLD codec

|  |  |  |  |
| --- | --- | --- | --- |
| LCLD configuration | DoF | Split rendering total bitrate (kbps) | Split rendering frame size (ms) |
| 5ms or 10ms or 20ms frame duration | 0 | 256, 384, 512 | 5 or 10 or 20 |
| 20ms frame duration | 1-3 | 384, 512, 768 | 20 |

END OF CHANGES