Standard ECMA-416

1st Edition / June 2018

Scalable Sparse Spatial Sound System (S5) – Base S5 Coding in Frequency Domain

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

Contents

Page

Introduction

S5 denotes a scalable multichannel coding system for spatial audio data compression, which can be applied to provide 3D audio experience with little overhead. Such system may incorporate a wide range of state-of-the-art audio codecs and can be applied to provide 3D audio experience. By using an audio codec, which may offer encapsulation capacity for external data, S5 data may be carried within the audio coder stream with little overhead and maintain a compatible bit stream syntax.

This Standard specifies the base S5 encoder and decoder in terms of configuration data, downmix and upmix. In addition, it provides reference and guidance on how to incorporate further components to form a scalable multichannel coding system for audio data compression.

The base S5 encoder achieves data compression of multichannel audio data by reducing the number of channels by a downmix process. Due to the invariance of spatial audio information to the downmix process, the S5 decoder is able to retrieve the input to the downmix by a special upmix process, the so-called Inverse Coding. Whereas the related Standard ECMA-407 applies Inverse Coding in time domain, this Standard specifies the Inverse Coding in frequency domain. An important feature of Inverse Coding is the tuning of the upmix to the downmix process. For this purpose the S5 encoder conveys sets of algebraic expressions to instruct the S5 decoder how to perform the upmix for the received downmix data. This side information contributes only a neglectable overhead to the audio payload.

Compressing the downmix data by a state-of-the-art audio codec will further increase the coding efficiency of S5. The Standard provides advice on how the base S5 encoder/decoder may be extended by incorporation of an audio codec and other components; however, specific components and their interfaces are not specified.

The standard is fully based on ECMA-407 syntax. This is extended with specifications of reserved codes and compatible new code elements. Therefore a S5 codec may apply Inverse Coding in frequency domain for some audio channels and Inverse Coding in time domain (ECMA-407) for others.

This Ecma Standard was developed by Technical Committee 32 and was adopted by the General Assembly of June 2018.

Scalable Sparse Spatial Sound System (S5) – Base S5 Coding in Frequency Domain

1 Scope

This Standard specifies the base encoder and decoder for S5 Coding in Frequency Domain in terms of configuration data, downmix and upmix. In addition, it provides reference and guidance on how to incorporate further components to form a scalable multichannel coding system for audio data compression.

2 Conformance

Conformant base S5 encoders generate data flows as specified in Clauses 7, 8, and 9. Conformant base S5 decoders generate the upmix as specified in Clause 10 by processing the downmix and data flows as specified in Clauses 7, 8, and 9.

3 Normative references

ISO/IEC 23001-8: *Information technology -- MPEG systems technologies -- Part 8: Coding-independent code points*

IETF RFC 5234: *Augmented BNF for syntax specifications: ABNF*

ECMA-407: *Scalable Sparse Spatial Sound System (S5) – Base S5 Coding, 1st edition, June 2014*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
downmix
reduced number of audio channels from an input signal

4.2
upmix
increased number of audio channels from a downmix

4.3
base S5 encoder in frequency domain
encoding unit providing the downmix in frequency domain and the upmix configuration

4.4
base S5 decoder in frequency domain
decoding unit providing the upmix based on the downmix in frequency domain and the upmix configuration

4.5
base audio codec
audio codec component providing lossless or lossy compression and decompression of the downmix

4.6
loudness
perceived level of an audio programme

4.7
Q format
fixed point binary format for fractional numbers, where the number of fractional bits and the number of integer bits is specified

5 Acronyms

For the purposes of this document, the following acronyms apply.

uimsbf unsigned integer, most significant bit first

uqmsbf unsigned Q format most significant bit first

NOTE This Standard uses for **uqmsbf** the Q format notation Qm.n, where “m” designates the number of bits of the integer part and “n” denotes the number of bits of the fractional portion to the right of the binary point. The width “w” of the corresponding bit field is w = m + n bits. The value range covers 0 to 2^{m} - 2^{-n} with a constant resolution of $2^{-n.}$ To convert a number from unsigned Q format to a decimal number take the Q bit field as an integer and multiply it by 2^{-n}.

sgmsbf signed Q format most significant bit first

NOTE This Standard uses for **sqmsbf** the 2’s complement with Q format notation Qm.n, where “m” designates the number of bits of the integer part without the sign bit and “n” the number of bits of the fractional portion to the right of the binary point. The width “w” of the corresponding bit field is w = m + n + 1 bits, which includes the sign bit as most significant bit. The value range covers 2^{-m} to 2^{m} - 2^{-n} with a constant resolution of 2^{-n}. To convert a number from signed Q format to a decimal number take the Q bit field as a 2’s complement integer and multiply it by 2^{-n}.

6 S5 Overview

S5 denotes a scalable multichannel coding system for spatial audio data compression, which can be applied to provide 3D audio experience with little overhead. Such a system may incorporate a wide range of state-of-the-art audio codecs and can be applied to provide a 3D audio experience. By using an audio codec, which may offer encapsulation capacity for external data, S5 data may be carried within the audio coder stream with little overhead and maintain a compatible bit stream syntax.

The system of an S5 codec can be determined by the functional block diagrams of the S5 encoder as depicted in Figure 1, and of the S5 decoder, as depicted in Figure 2. An S5 encoder shall at least consist of a base S5 encoder and a S5 decoder shall at least consist of a base S5 decoder.

The base S5 encoder achieves data compression of multichannel audio data by reducing the number of channels to transmit by a downmix process. Due to the invariance of spatial audio information to the downmix, process the S5 decoder is able to retrieve the original channels by a special upmix process, so-called Inverse Coding. Whereas the related Standard ECMA-407 applies Inverse Coding in time domain, this Standard specifies the use of Inverse Coding in frequency domain. An important feature for Inverse Coding is the tuning of the upmix to the downmix process. For this purpose the S5 encoder conveys sets of algebraic expressions to instruct the S5 decoder how to perform the upmix for the received downmix data. This side information contributes only a neglectable overhead to the audio payload.

Compressing the downmix audio by a state-of-the-art base audio coder can further increase the coding efficiency of S5. The various bitstreams produced by the functional units of an S5 encoder may be encapsulated into a single bitstream by the functional unit 'Multiplexer' (see Annex F).

Ancillary data may be conveyed from the S5 encoder to the S5 decoder and may be used to encapsulate data other than Inverse Coding data. An example is loudness parameters, which may be used to adjust the perceived level of audio signals. For the loudness parameters, see Annex D.

This Standard specifies the base S5 encoder/decoder and their interfaces only. Extensions of the base S5 coding system with other components and their interfaces in order to set up specific S5 codecs may be subject to separate standards or other specifications. As the base S5 encoder/decoder is agnostic to the other system components, the base S5 coding standard shall represent the common base for all S5 specific standards.

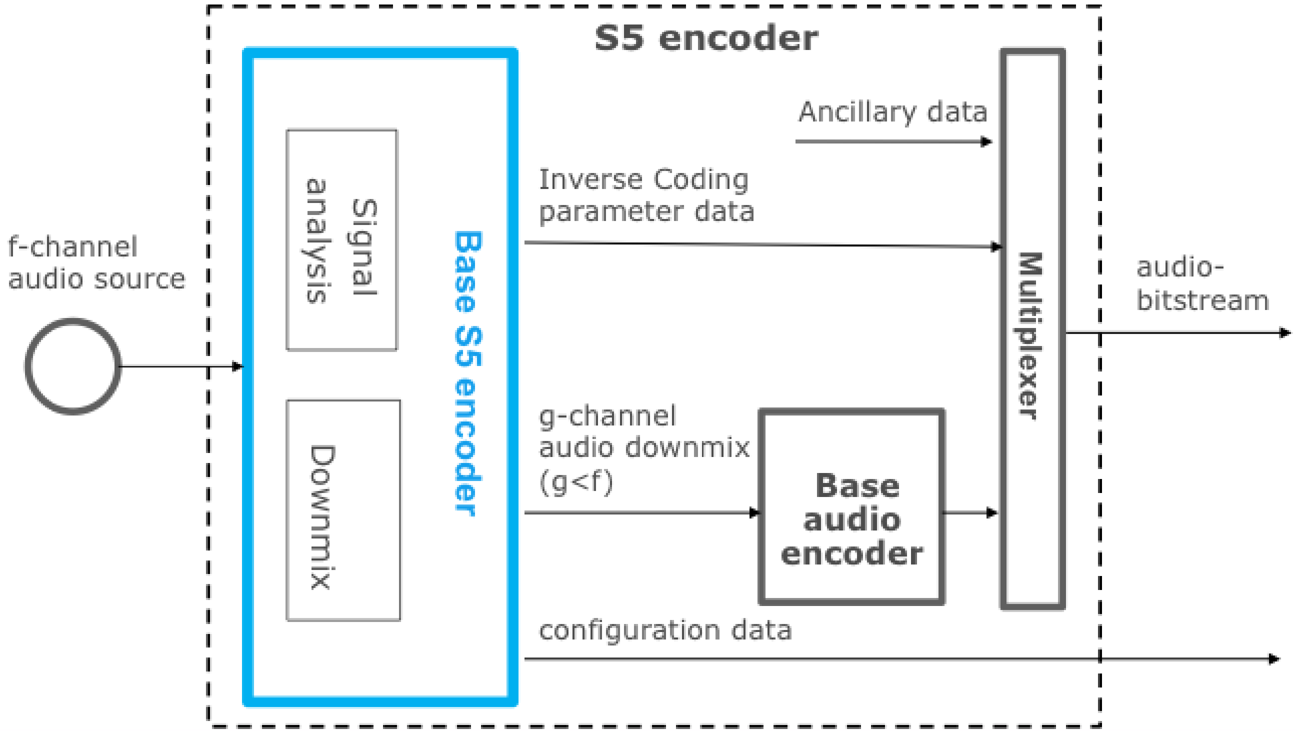

Figure 1 — Functional block diagram of the S5 encoder

The subsequent clauses of this Standard specify the syntax of data streams by using Augmented Backus Naur Form (ABNF) as is defined in IETF RFC 5234. In addition to this notation, the code of the data stream elements is denoted by the format and the length of their bit fields. Note, that syntax and final encoding of a data stream are strictly separated. For the same syntax of a data stream, an external encoding e.g. by a multiplexer may vary according to the constraints of the storage or transmission environment. Examples are byte alignment or error protection. However, external encoding details are beyond the scope of this Standard and are subject to specific S5 standards or other specifications.

This standard is fully based on the syntax specified in ECMA-407; however the subsequent clauses replicate only those elements which are relevant for Inverse Coding in the Frequency Domain. In addition to ECMA-407 Annex B specifies new compatible code elements (see Table B.4, Table B.5 and Table B.6) Therefore a S5 codec may apply Inverse Coding in frequency domain for some audio channels and Inverse Coding in time domain (ECMA-407) for others.

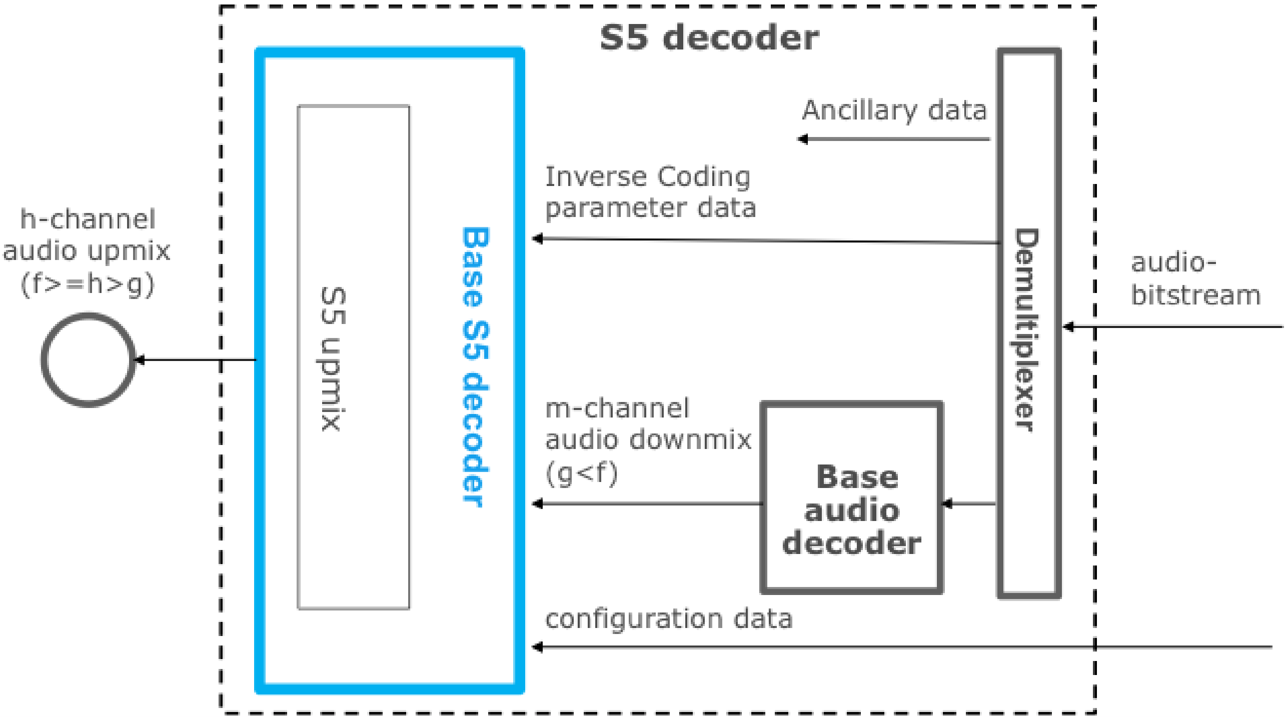

Figure 2 — Functional block-diagram of the S5 decoder

7 Inverse Coding in Frequency Domain

Considering that the ambiance information of multichannel audio signals is invariant to downmix processes, which reduces the number of channels, it should be possible to ‘rechannel’ a downmix by an inverse process, the so-called upmix. A well-known application is pseudo-stereophony, which synthesizes from a monophonic input channel a left and right output channel.

ECMA-407 takes a similar approach which is called ‘Inverse Coding’. The method is characterized by using a signalling model of the audio source in time domain to recover two output channels from one input channel. In contrast to prior art, upmix by Inverse Coding is not static but dynamically tuned to the downmix process using side information.

This Standard applies for upmix a different method which is called ‘Inverse Coding in Frequency Domain’. The upmix process takes place in frequency domain and reconstructs a third channel from two input channels. Figure 3 shows the functional block diagram of the upmix in frequency domain.

Figure 3 uses the following notations:

r’(n), l’(n)	audio samples of downmix output
L’(k), R’(k)	complex Fourier transform components of l’(n), r’(n)
L“(k), C”(k), R”(k)	complex Fourier components of Inverse Coding output
l”(n), c”(n), r”(n)	upmix of audio samples l’(n), r’(n).

This Standard does not specify a specific algorithm to use for Inverse Coding in Frequency Domain. In principle any algorithm may be applied which is able to derive the output components L“(k), C”(k), R”(k) from the downmix components L’(k), R’(k). As specified in Table B.6 of Annex B Inverse Coding in Frequency Domain is based on the following 3 functions:

S5FDLeft Inverse Coding function to derive L”(k) from L’(k), R’(k);
S5FDRight Inverse Coding function to derive R”(k) from L’(k), R’(k);
S5FDCenter Inverse Coding function to derive C”(k) from L’(k), R’(k);

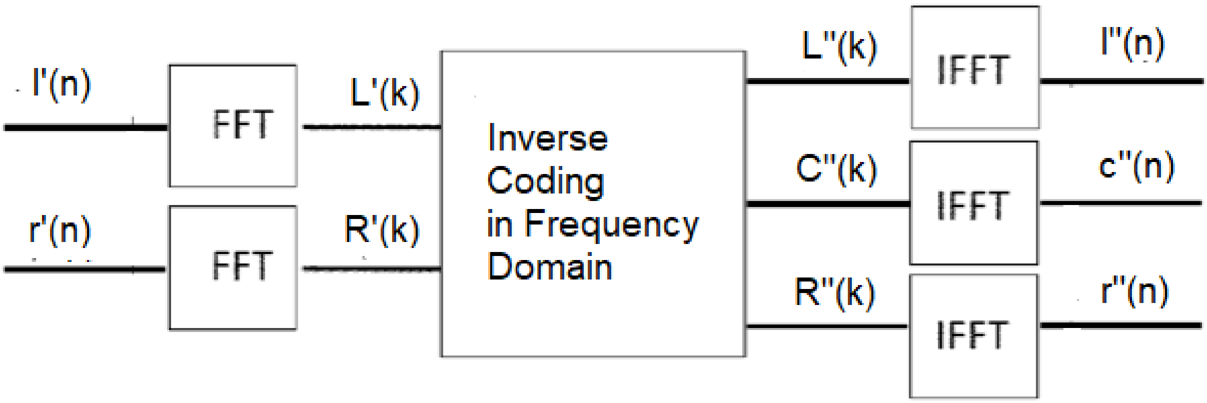

Figure 3 — Functional block diagram of upmix based on Inverse Coding in Frequency Domain

Inverse Coding assumes a downmix as defined in Clause 6.1. As an example Clause 6.2 introduces the ‘Correlation Comparison’ approach, being proposed by [2] and [7].

Inverse Coding cannot perfectly reconstruct the original audio inputs L(k), R(k), C(k) to the downmix. The following error distribution Δ (k) is assumed for the output of the upmix L”(k), R”(k) and C”(k):

L”(k) = L(k) + Δ (k)
R”(k) = R(k) + Δ (k)
C”(k) = C(k) – 2 * Δ(k)

For further information on upmix distortions and their cancellation see Annex F.

The figures of this Standard always show Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) used for spectral transform. But any other appropriate approach that fits to the Inverse Coding functions can be applied instead. However, it is generally recommended to use FFT/ IFFT of sufficient frame length.

7.1 Downmix Requirements

The underlying assumption for any solution of Inverse Coding in Frequency Domain is a downmix process as depicted in Figure 4.

Figure 4 uses the following notations:

l(n), c(n), r(n) samples of an audio source (time domain))
L(k), C(k), R(k) complex Fourier transform components of l(n), c(n), r(n)
L’(k), R’(k) complex Fourier components of the downmix
l’(n), r’(n) audio samples of the downmix (time domain)

With

L’(k) = L(k) + downmixfactor * C(k)
R’(k) = R(k) + downmixfactor * C(k)

the Fourier components R’(k) and L’(k) are correlated to each other.

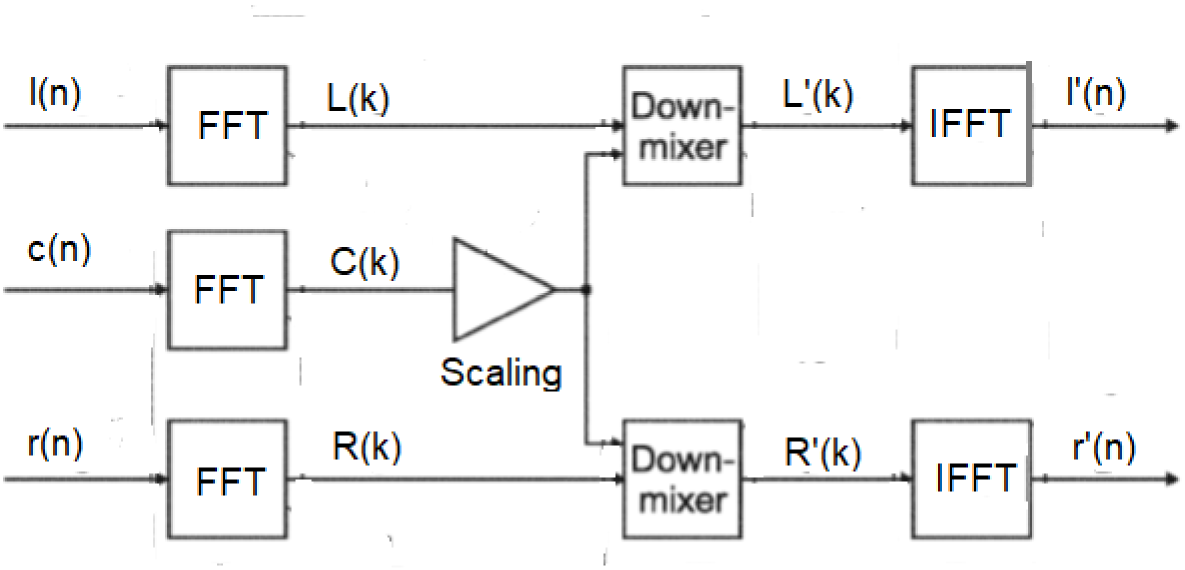

Figure 4 — Functional block diagram of downmix in frequency domain

7.2 Inverse Coding by Correlation Comparison

Inverse Coding in Frequency domain by ‘Correlation Comparison’ is based on pair-wise comparing the spectral components of corresponding frames R’(k) and L’(k) with k=0,1,...,N-1 where N denotes the number of components in a frame. The following rules apply for the determination of L”(k), R”(k) and C”(k):

Real parts

```
If sgn(Re(L’(k))) not equal sgn(Re(R’(k)))
   Re(L”(k)) = Re(L’(k))
   Re(R”(k)) = Re(R’(k))
   Re(C”(k)) = 0
Else
      If abs(L’(k)) < abs(R’(k))
            Re(L”(k)) = 0
            Re(R”(k)) = Re(R’(k)) – Re(L’(k))
            Re(C”(k)) = Re(L’(k))
      Else
Re(L”(k)) = Re(L’(k)) – Re(R’(k))
Re(R”(k)) = 0
Re(C”(k)) = Re(R’(k))
```

Imaginary parts

```
If sign(Im(L’(k))) not equal sign(Im(R’(k)))
      Im(L”(k)) = Im(L’(k))
      Im(R”(k)) = Im(R’(k))
      Im(C”(k)) = 0
Else
      If abs(L’(k)) < abs(R’(k))
            Im(L”(k)) = 0
            Im(R”(k)) = Im(R’(k)) – Im(L’(k))
            Im(C”(k)) = Im(L’(k))
      Else
```

Im(L"(k)) = Im(L'(k)) – Im(R'(k))
Im(R"(k)) = 0
Im(C"(k)) = Im(R'(k))

The Inverse Coding functions are specified as follows:

S5FDLeft	**compilation of all instructions to derive L"(k) from L'(k), R'(k)**
S5FDRight	**compilation of all instructions to derive R"(k) from L'(k), R'(k)**
S5FDCenter	**compilation of all instructions to derive C"(k) from L'(k), R'(k)**

Exploiting the symmetry of the N Fourier spectral components, the number of pair-wise comparisons can be reduced from N to N/2.

8 S5 Configuration Data

The configuration data shall consist of the downmix configuration ID (**S5DownmixConfig),** the output channel configuration ID (**S5ChannelConfig**) and a group of data elements which specify the upmix configuration (**S5UpmixConfig**).

8.1 Syntax of Configuration Data (S5Config)

Table 1 — Syntax of S5Config

Syntax	No. of bits	Format
S5Config =		
"S5ConfigID"	**4**	**uimsbf**
"S5SyncTagWindow"	**10**	**uimsbf**
"S5SyncTagAccuracy"	**2**	**uimsbf**
"S5DownmixConfig"	**7**	**uimsbf**
"S5ChannelConfig"	**7**	**uimsbf**
"S5UpmixConfig"		

8.2 Configuration Identifier (S5ConfigID)

Each instance of **S5Config** is represented by its unique identifier **S5ConfigID**.This identifier is referred by S5 data streams to link data elements or group of data elements to its corresponding configuration data.

The value range covers decimal numbers from 0 to 15.

8.3 Window Size for the Calculation of Synchronization Tags (S5SyncTagWindow)

The value of **S5SyncTagWindow** sets the number of consecutive audio samples used for the calculation of synchronisation tags as specified in 8.2.

The value range covers decimal numbers from 64 to 1024.

8.4 Accuracy of the Calculation of Synchronization Tags (S5SyncTagAccuracy)

The value of **S5SyncTagAccuracy** sets the number of bits used in 8.2 for the integer representation of the energy values.

Table 2 — Code assignment for S5SyncTagAccuracy

00: 16 bit
01: 24 bit
10: 32 bit
11: reserved

8.5 Downmix Configuration (S5DownmixConfig)

The **S5DownmixConfig** gives the channel configuration, which may be used to render the downmix channels on to loudspeaker positions whilst ignoring **S5ChannelConfig** and **S5UpmixConfig.**

The downmix configuration shall be identified by a ‘Channel Configuration’ value, as specified in Annex A.

8.6 Output Channel Configuration (S5ChannelConfig)

The **S5ChannelConfig** describes the reference loudspeaker arrangement for which the base S5 decoder upmix shall be intended.

The output channel configuration shall be identified by a ‘Channel Configuration’ value, as specified in Annex A.

8.7 Upmix Configuration (S5UpmixConfig)

The data elements of **S5UpmixConfig** instruct the base S5 decoder on how to combine the Inverse Coding functions for generating the upmix samples from the downmix. This is accomplished for each output channel by an algebraic expression.

The **S5UpmixConfig** syntax shall adhere to the specification in Annex B.

9 S5 Inverse Coding Data

The Inverse Coding data shall comprise the data elements **S5ConfigID**, **S5SyncTag, S5SyncTag-1, S5SyncTag-2,** and **S5ParameterSetCount.**

An Inverse Coding data stream starts always with a unique identifier value which links the parameter data to the corresponding configuration data. This element **S5ConfigID** is specified in Clause 7.2.

9.1 Syntax of Inverse Coding Data (S5InvCodeData)

The Inverse Coding data bitstream from the base S5 encoder to the base S5 decoder shall correspond to the syntax as specified in Table 3:

Table 3 — Syntax of S5InvCodeData

Syntax	No. of bits	Format
S5InvCodeData =		
“S5ConfigID”	**4**	**uimsbf**
“S5SyncTag”	**8**	**uimsbf**
“S5SyncTag-1”	**8**	**uimsbf**
“S5SyncTag-2”	**8**	**uimsbf**
“S5ParameterSetCount”	**8**	**uimsbf**

9.2 Synchronization Elements (S5SyncTag, S5SyncTag-1, S5SyncTag-2)

The value of the synchronization element **S5SyncTag** supports synchronization between the downmix stream and the upmix process inside the base S5 decoder. The bit field **S5SyncTag** shall denote the energy signature of the downmix in a time granularity corresponding to the duration of such non-overlapping window of **WindowLength** samples as set by the value of **S5SyncTagWindow** in the configuration data stream (7.3). The energy signature **S5SyncTag** shall be derived by the S5 encoder from the downmix by the following process:

For each of the downmix channels, the energy of its time domain signal over **WindowLength** samples is computed by accumulating the squared sample value of the DC-removed signal (can be calculated by subtracting the arithmetic mean of the signal).

The time domain values are integer values of type **uimsbf** where the number of bits is set in the configuration data stream by **S5SyncTagAccurancy** (7.4 Table 2).

The sum of the energy of all downmix channels **TotalEnergy** is normalized by **WindowLength** and converted into a logarithmic dB representation.

$$\mathbf{dBTotalEnergy} = 10 * \log10(\max \frac{\mathbf{TotalEnergy} * 1024}{\mathbf{WindowLength} + 1}, 1)$$

It is quantized into an 8 bit unsigned integer by determining min(**dBTotalEnergy**, 255).

Whereas **S5SyncTag** refers to the window where the parameter set data should be applied, **S5SyncTag-1 and S5SyncTag-2** relate to its past two windows and shall use the same rules as given for **S5SyncTag.**

The decoder shall apply the same procedure to determine synchronisation tags from the received downmix data. In case a synchronization tag matches with the conveyed **S5SyncTag**, the decoder is able to re-establish the correct temporal relationship between the received audio bitstream and the S5 side information. To verify the correctness of the synchronisation result, the preceding windows may be checked in addition by using the corresponding synchronization tags **S5SyncTag-1** or **S5SyncTag-2**.

9.3 Number of Parameter Sets (S5ParameterSetCount)

S5ParameterSetCount is set to **"0",** no parameter set data is conveyed.

10 S5 Downmix

The base S5 encoder in frequency domain shall generate the g-channel downmix from the f-channel input signal (g < f) according to the channel configuration number, which has been assigned to **S5DownmixConfig**. The basic downmix function shall applied in frequency domain to create the downmix signals L'(k) and R'(k), by adding up input signals L, R, C as defined in Clause 6.1:

Left downmix signal: L'(k) = L(k) + downmixfactor * C,(k)
Right downmix signal: R'(k) = R(k) + downmixfactor * C,(k)

For corresponding input signals L(k), R(k), C(k) the same value for downmixfactor shall be used. The value is assigned to **S5FDFactor** of the S5 configuration data (see Table B.5).

The downmix for Inverse Coding in frequency domain inside the base S5 **en**coder needs not to use the same paradigm of frequency domain processing as used in the base S5 **de**coder, and could even perform the downmix in the time domain (while preserving the correlation of L'(k), R'(k)). With certain base audio encoders, spectral performance can be boosted, e.g. by combining a downmix in time domain in the S5 **en**coder; with an upmix in frequency domain.

11 S5 Upmix

The upmix unit of the base S5 decoder shall create in the frequency domain the *h* channel upmix from the *g* channel downmix in accordance with the S5 configuration data **S5Config**. Figure 5 shows the functional block diagram of the S5 upmix.

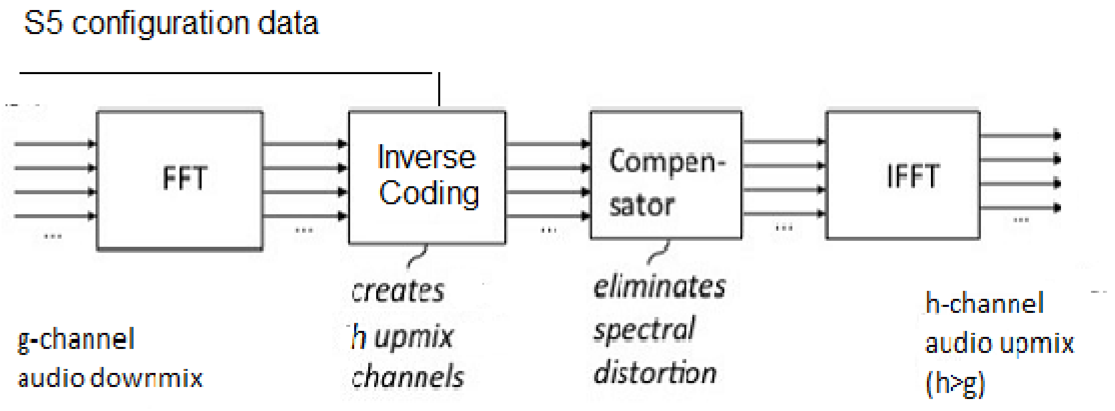

Figure 5 — Functional blocks of the S5 upmix unit

The figure shows Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) used for spectral transform, but any other appropriate approach that fits to the Inverse Coding functions can be applied instead. However, it is generally recommended to use FFT/ IFFT of sufficient frame length.

This standard does not specify a specific algorithm to use for Inverse Coding. Any algorithm may be applied for the three Inverse Coding functions which meets the following conditions:

S5FDLeft	shall derive L”(k) from downmix L’(k), R’(k);
S5FDRight	shall derive R”(k) from downmix L’(k), R’(k);
S5FDCenter	shall derive C”(k) from downmix L’(k), R’(k);

Setup and processing rules of Inverse Coding shall carry out in accordance with the S5 configuration data **S5Config** (see Clause 7 and Annex B)

Inverse Coding cannot perfectly reconstruct the audio inputs L(k), R(k), C(k) to the downmix. To minimize or compensate spectral distortions the output of Inverse Coding may be processed by an optional unit.

Annex A
(normative)

Channel Positions and Configurations

The knowledge of loudspeaker positions and their corresponding channel configurations is essential for correct reconstruction and presentation of spatial audio, at the S5 decoder site. This normative annex specifies the codes to convey this information to S5 decoder.

The output channel position, the loudspeaker position, the channel configurations from “1” to “31” and the channel to speaker mapping shall adhere to Tables 7 and 8 in ISO/IEC 23001-8 *Information technology -- MPEG systems technologies -- Part 8: Coding-independent code points*.

Table 7 of ISO/IEC 23001-8 indicates the loudspeaker position in the 3D environment of the listener as used in Annex B by **S5OutputChannelPos** and **S5InputChannel**. In order to ease the understanding of loudspeaker positions, Table 7 also contains loudspeaker positions according to IEC 62574.

Table 8 of ISO/IEC 23001-8 specifies the channel configuration, as is used in Clause 8 by **S5DownmixConfig** and **S5UpmixConfig**. This defines the number of audio channels and their associated loudspeaker positions. The name, abbreviation, and general position of each loudspeaker can be deduced from Table 7 (and Figure 9).

For the purpose of this Standard the following normative extensions apply:

- ‘Output Channel Position’ values from “32” to “127” are reserved.
- ‘Channel Configuration’ values from “15” to “63” are reserved.
- ‘Channel Configuration’ values from “64” to “126” are open for definition and are specified in the following Ecma Registry: http://www.ecma-international.org/standards/ecma-407/registry.htm
- ‘Channel Configuration’ value “127” indicates a non-renderable downmix.

Annex B
(normative)

Syntax for S5UpmixConfig

Considering that using a fixed configuration of the S5 upmix process for the various types and formats of audio sources, the upmix cannot be kept in a defined range of perceived audio quality. Therefore the S5 upmix configuration is provided as an algebraic expression, which can easily be adapted to various audio sources and as well incorporate future advanced S5 coding methods. This normative annex specifies the syntax which shall be applied.

Each upmix configuration shall be denoted by **S5UpmixConfig**. **S5UpmixConfig** instructs as part of the configuration data (see Clause 7) to the base S5 decoder on how to combine the inverse coding functions for generating the upmix values from the downmix. This is accomplished for each output channel by a mathematical expression represented in Polish Notation.

In the following, the syntax of the coded representation is specified. The syntax of **S5UpmixConfig** shall adhere to the specification in Table B.1.

Table B.1 — Syntax of S5UpmixConfig

Syntax	No. of bits	Format
S5UpmixConfig =		
“S5ChannelCount”	**7**	**uimsbf**
1* (		
“**S5OutputChannelPos”**	**7**	**uimsbf**
“**S5ExpressionCount”**	**8**	**uimsbf**
1* (		
“**S5OperatorType”**	**3**	**uimsbf**
/ **S5Operand**		
)		
)		

S5ChannelCount carries the total number of output channels needed for a channel configuration according to the value of **S5ChannelConfig.** Its integer value sets the number of occurrences of the subsequent group of data elements. . The values of **S5ChannelCount** cover the range from 1 to 127.

S5OutputChannelPos shall show the index number of the output channel as defined in ‘Channel Position’ of Annex A. The values of **S5OutputChannelPos** cover the range from 0 to 127.

S5ExpressionCount carries the total number of operators and operands required to specify the upmix expression. Its integer value indicates the number of occurrences of the subsequent group of data elements. . The values of **S5ExpressionCount** cover the range from 1 to 255.

S5OperatorType denotes the required mathematical operation in the upmix expression by a 3-bit code. The binary codes shall adhere to the specification in Table B.2.

Table B.2 — Codes used by S5OperatorType

“**000**”	indicates addition
“**001**”	indicates subtraction
”**010**”	indicates multiplication
”**011**”	indicates division

S5Operand specifies the different types of an operand. The syntax is given in Table B.3.

Table B.3 — Syntax of S5Operand

Syntax	No. of bits	Format
S5Operand =		
“**S5OperandType**”	**3**	**uimsbf**
(		
S5Function		
/ **S5Sample**		
/ S5Number		
/ S5FD		
)		

S5OperandType is a 3-bit prefix code to indicate the type of the operand. Table B.4 specifies the assignment.

Table B.4 — Codes used by S5OperandType

“**100**”	indicates type “Function” and use of ECMA-407
“**101**”	indicates type “Sample” and use of ECMA-407
”**110**”	indicates type “Number” and use of ECMA-407
“**111**”	indicates type “FD"

Table B.5 — Syntax of S5FD

Syntax	No. of bits	Format
S5FD =		
“**S5FDType**”	**2**	**uimsbf**
“**S5FDFactor**”	**Q3.13**	**uqmsbf**
“S5DownmixChannelLeft”	**7**	**uimsbf**
“S5DownmixChannelRight”	**7**	**uimsbf**

S5FDType shall show, which of input channels L(k), R(k), C(k) shall be reconstructed by means of Inverse Coding in frequency domain. Table B.6 specifies the assignment.

Table B.6 — Codes used by S5FDType

“**00**”	indicates Inverse coding function **S5FDLeft** for reconstructing *L(k)*
“**01**”	indicates Inverse coding function **S5FDRight** for reconstructing *R(k)*
“**10**”	indicates Inverse coding function **S5FDCenter** for reconstructing *C(k)*
“**11**”	reserved for use by external specifications

As stated in Clauses 6 and 10 the algorithm of the Inverse Coding functions **S5FDLeft, S5FDRight, S5FDCenter** are not specified by this standard. However Clause 6.2 gives an example which meets the requirements of Clause 10.

S5FDFactor shall specify the value of *downmixfactor* (see Clauses 6.1 and 9).

S5DownmixChannelLeft shall show the index number of the channel position of the left downmix channel L(k) according to the table specified in Annex A. The values of **S5DownmixChannelLeft** cover the range from 0 to 127.

S5DownmixChannelRight shall show the index number of channel position of the right downmix channel R(k) according to the table specified in Annex A. The values of **S5DownmixChannelRight** cover the range from 0 to 127.

Annex C
(informative)

Channel Configuration and Position Tables

Tables C.1, C.2 and Figure C.1 of this Annex replicate the information in Table 7 and 8 and Figure 9 of ISO/IEC 23001-8 *Information technology -- MPEG systems technologies -- Part 8: Coding-independent code points.* This is for the convenience of the reader, and is for information only. The normative reference is Annex A.

Table C.1 illustrates the 'Output Channel Position', the speaker abbreviation and the loudspeaker position - indicating the loudspeaker position in the 3D environment of the listener. In order to ease the understanding of loudspeaker positions, Table C.1 also contains loudspeaker positions according to IEC 62574, which are listed here for information. Table C.1 gives informative guidance; however, is no normative reference with regard to ISO/IEC 23001-8 *Information technology -- MPEG systems technologies -- Part 8: Coding independent code points.*

Table C.1 — Mapping of Channel position to loudspeaker position

Output Channel Position	Loudspeaker position		Loudspeaker position according to IEC 62574 :2011	
	Abbr.	**Name**	**Abbr.**	**Name**
0	L	Left front	FL	Front left
1	R	Right front	FR	Front right
2	C	Centre front	FC	Front centre
3	LFE	Low frequency enhancement	LFE1	Low frequency effects-1
4	Ls	Left surround	LS	Left surround
5	Rs	Right surround	RS	Right surround
6	Lc	Left front centre	FLc	Front left centre
7	Rc	Right front centre	FRc	Front right centre
8	Lsr	Rear surround left	BL	Back left
9	Rsr	Rear surround right	BR	Back right
10	Cs	Rear centre	BC	Back centre
11	Lsd	Left surround direct	LSd	Left surround direct
12	Rsd	Right surround direct	RSd	Right surround direct
13	Lss	Left side surround	SL	Side left
14	Rss	Right side surround	SR	Side right
15	Lw	Left wide front	FLw	Front left wide
16	Rw	Right wide front	FRw	Front right wide
17	Lv	Left front vertical height	TpFL	Top front left
18	Rv	Right front vertical height	TpFR	Top front right
19	Cv	Centre front vertical height	TpFC	Top front centre
20	Lvr	Left surround vertical height rear	TpBL	Top back left
21	Rvr	Right surround vertical height rear	TpBR	Top back right
22	Cvr	Centre vertical height rear	TpBC	Top back centre
23	Lvss	Left vertical height side surround	TpSiL	Top side left
24	Rvss	Right vertical height side surround	TpSiR	Top side right
25	Ts	Top centre surround	TpC	Top centre
26	LFE2	Low frequency enhancement 2	LFE2	Low frequency effects-2
27	Lb	Left front vertical bottom	BtFL	Bottom front left
28	Rb	Right front vertical bottom	BtFR	Bottom front right
29	Cob	Centre front vertical bottom	BtFC	Bottom front centre
30	Lvs	Left vertical height surround	TpLS	Top left surround
31	Rvs	Right vertical height surround	TpRS	Top right surround
32-127		Reserved		Reserved

Figure C.1 below informatively illustrates the loudspeaker position in the 3D environment relative to the listener, with each labelled with an abbreviation from Table C.1. Loudspeakers lying on the innermost box are in the bottom level, those on the middle box are in the middle level and those on the outermost box are in the top level. The circles labelled Ts represent the listener position:

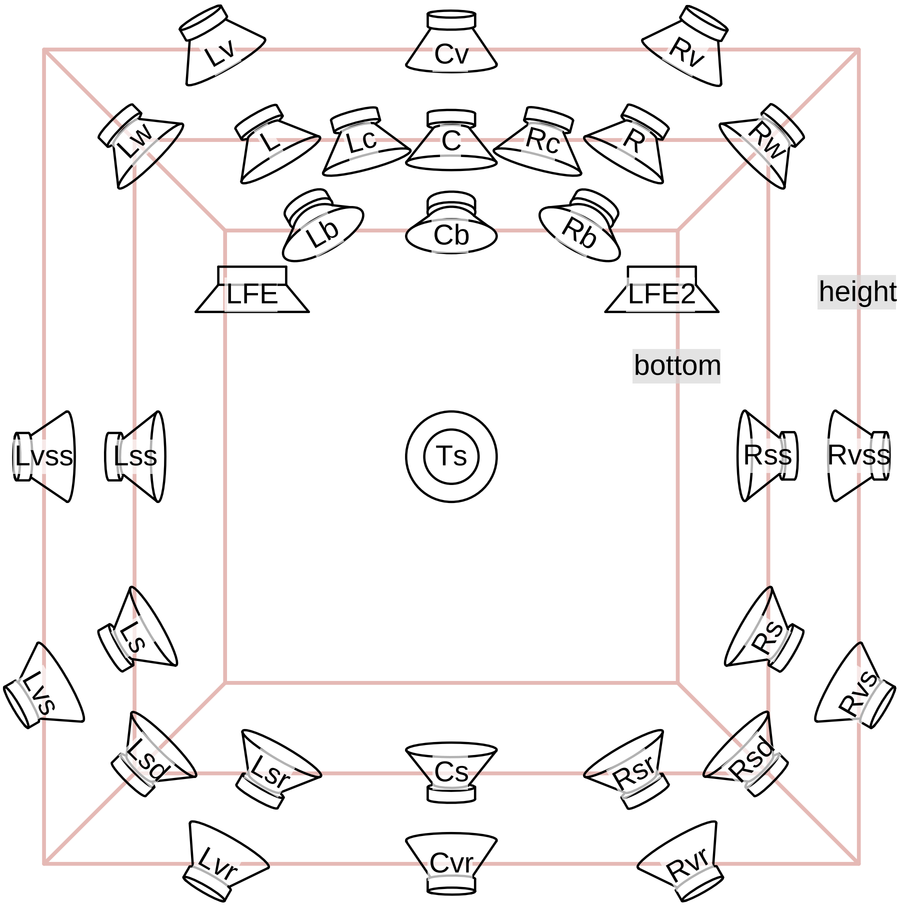

Figure C.1 — Loudspeaker position in the 3D environment

Table C.2 provides information about 'Channel Configuration', the assigned speaker positions, and their abbreviations in accordance with Figure C.1. In addition it includes the notation for indicating the involved number of audio channels and their associated loudspeaker positions. Note that Table C.2 gives informative guidance only and is no normative reference with regard to ISO/IEC 23001-8 *Information technology -- MPEG systems technologies -- Part 8: Coding-independent code points.*

Table C.2 — Channel configuration, channel to speaker mapping, and speaker abbreviation

Channel Configuration	Channel to speaker mapping	Speaker abbreviation	"Front/Surr. LFE" notation
1	centre front speaker	C	1/0.0
2	left, right front speakers	L, R	2/0.0
3	centre front speaker, left, right front speakers	C L, R	3/0.0
4	centre front speaker, left, right front speakers, rear centre speaker	C L, R Cs	3/1.0
5	centre front speaker, left, right front speakers, left surround, right surround speakers	C L, R Ls, Rs	3/2.0
6	centre front speaker, left, right front speakers, left surround, right surround speakers, low frequency enhancement speaker	C L, R Ls, Rs LFE	3/2.1
7	centre front speaker, left, right front centre speakers, left, right front speakers, left surround, right surround speakers, low frequency enhancement speaker	C Lc, Rc L, R Ls, Rs LFE	5/2.1
8	channel1 channel2	N.A. N.A.	1+1
9	left, right front speakers, rear centre speaker	L, R Cs	2/1.0
10	left, right front speaker, left surround, right surround speakers,	L, R Ls, Rs	2/2.0
11	centre front speaker, left, right front speakers, left surround, right surround speakers, rear centre speaker, low frequency enhancement speaker	C L, R Ls, Rs Cs LFE	3/3.1
12	centre front speaker, left, right front speakers, left surround, right surround speakers, rear surround left, right speakers, low frequency enhancement speaker	C L, R Ls, Rs Lsr, Rsr LFE	3/4.1
13	centre front speaker, left, right front centre speakers, left, right front speakers, left, right side surround speakers, rear left, right surround speakers, rear centre speaker, left front low frequency enhancement speaker, right front low frequency enhancement speaker, centre front vertical height speaker, left, right front vertical height speakers, left, right vertical height side surround speakers, top centre surround speaker, left, right surround vertical height rear speakers, centre vertical height rear speaker, centre front vertical bottom speaker, left, right front vertical bottom speakers	C Lc, Rc L, R Lss, Rss Lsr, Rsr Cs LFE LFE2 Cv Lv, Rv Lvss, Rvss Ts Lvr, Rvr Cvr Cb Lb, Rb	11/11.2
14	centre front speaker, left, right front speakers, left surround, right surround speakers, low frequency enhancement speaker, left, right front vertical height speakers	C L, R Ls, Rs LFE Lv, Rv	5/2.1
15-63	Reserved	Reserved	-
64-126		See NOTE	
127		Non-renderable downmix	

NOTE For specification see Ecma Registry: http://www.ecma-international.org/standards/ecma-407/registry.htm

Annex D
(informative)

Loudness Adjustment

Considering that listeners desire the subjective loudness of audio programmes to be uniform for different sources and programme type, this informative annex provides guidance on loudness adjustment by the base S5 decoder.

The subjective loudness of audio programmes, as perceived by the listener, should be described by four measurements according to ITU-R BS.1770-3:2012 *Algorithms to measure audio programme loudness and true-peak audio level,* namely:

Programme Loudness
the integrated loudness of the whole programme

Max True Peak
the maximum peak value measured after 4x oversampling

Max Momentary Loudness
the maximum loudness value for one block of 400ms

Max Short Term Loudness
the maximum loudness value for a 3 seconds sliding window

Loudness Range
the measure that reflects the overall dynamic range of the programme.

The last parameter Loudness Range is according to EBU (European Broadcast Union) TECH3347 (https://tech.ebu.ch/docs/tech/tech3342.pdf) a measure to supplement *EBU R-128* loudness normalisation*.*

The upmix generated by the base S5 decoder may be adjusted according to these measurements, which are conveyed as ancillary data from the S5 encoder to the S5 decoder, together with any descriptive data.

Annex E
(informative)

Multiplexing

Considering that the transmission format of the base S5 bitstreams is not specified in this standard, but subject of multiplex method used in a specific S5 system. This informative annex gives a brief introduction to suitable multiplexing approaches.

The various bitstreams produced by the functional elements of an S5 encoder may be either transmitted separately or be multiplexed into a single logical bitstream. The corresponding optional functional elements of the S5 encoder in Figure 1 and S5 decoder in Figure 2 are referred to as ‘Multiplexer’ and ‘Demultiplexer’.

A common way of multiplexing bitstreams (audio and video) with other data into a single transport stream may be the use of container formats, e.g. Ogg (see S. Pfeiffer. The Ogg Encapsulation Format Version 0, IETF *RFC 3533*, May 2003).

Such formats generate high-level media codec streams, which provide framing, error protection and random access structure.

If the S5 data should be transported within the data stream of an audio codec, another approach may be more appropriate: some audio-coding standards are built upon a ‘core and extension’ architecture, which provides a single, unified bit stream syntax for the core and extension technology to encapsulate external data. Examples are the MPEG coders HE-AAC (see ISO/IEC 14496-3 *Information technology – Coding of audio-visual objects – Part 3: Audio)* and USAC (see SO/IEC 23003-3 *Information technology -- MPEG audio technologies -- Part 3: Unified speech and audio coding)*. These mechanisms can be used as a container to carry S5 data within the data stream of the base audio codec with little overhead while maintaining compatible bit stream syntax.

Annex F
(informative)

Cancellation of upmix distortions

As mentioned in Clause 6, the output channels of the upmix differ from the input channels of the downmix. These deviations may have many sources:

1) Spectral leakage of the transformation:
Discrete Fourier transformations of audio signals are processed on a frame by frame basis, where each frame consists of a fixed finite number N of audio samples. However, this leads to distortions of the signal spectral components, which are referred as spectral leakage. To reduce spectral leakage effects, the frame length N should be as large as possible, but large frame length will increase the latency and the computational load of the system. Another measure is the application of a window function, e.g. Hamming window, Flat-Top window etc., to the frames either in time or in frequency domain. The selection of a window function should consider the properties of the Inverse Coding functions.

2) Mismatch of downmix and upmix:
Another source of distortions is the upmix. In general Inverse Coding is not the exact reversion of the downmix process. However, tuning the upmix process to the download process will minimize errors. Means are the right selection of the algorithm for Inverse Coding and dynamic adjustment of the formula Inverse Coding functions are applied to downmix data. S5 accomplish this feature by providing side information (see Clause 7 and Annex B) to instruct the S5 upmix process.

Upmix distortions get acoustically cancelled in the sweet-spot, if errors are distributed according to the definitions in clause 6 and if the center loudspeaker is exactly placed between the loudspeakers of the Left and Right channel, otherwise distortions may be audible. In this case there are several publications, which have addressed the problem and provide proposals on how to minimize or even to compensate distortions. Examples are [1] and [2].

Bibliography

[1] C. Par. Two Undiscovered Treasures for Ground-breaking 3D Audio Coding Technology at Lowest Bitrates: Inverse Problems and Invariant Theory. Proceedings of 27th VDT International Convention, 11/2012, ISBN 978-3-9812830-3-7.

[2] R. V. Ambartsumian, ed. A Life in Astrophysics: Selected Papers of Viktor Ambartsumian. Allerton Press, 1998.

[3] J. Hong, B. Leonard, C. Par, S. Quackenbush, W. Woszczyk ea. ECMA-407: New Approaches to 3D Audio Content Data Rate Reduction with RVC-CAL. 137th AES Convention Paper, 10/2014.

[4] J. Hong, B. Leonard, C. Par, S. Quackenbush, W. Woszczyk ea. ECMA-407: A New 3D Audio Codec Implementation up to NHK 22.2 with RVC-CAL. Proceedings of 28th VDT International Convention, 11/2014.

[5] C. Par. ECMA-407 - Internationaler Standard für modularen 3D-Audio-Transport. Teil 1. FKT, 4/2015.

[6] C. Par. ECMA-407 - Internationaler Standard für modularen 3D-Audio-Transport. Teil 2. FKT, 5/2015.

[7] C. Par. Rationalism versus Empirism. A Crash Course in Invariant Theory and a Tribute to Rudolf E. Kálmán. Intercomms, 10/2015.

[8] C. Par. ECMA-407 – *Instant HD to UHD Audio* White Paper. Intercomms, 04/2016