UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Master en Ingeniería Acústica

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

"Headroom and precision requirements of fixed point audio processing in different data domains for real world content"

TRABAJO FINAL DE MASTER

Autor/a:
Damià Carbonell Tena

Tutor/a:
Miguel Ferrer Contreras
Holger Hoerig (Dolby Laboratories)

GANDIA, 2018

Abstract

In modern audio processing technology, the audio signal is converted to various data domains for processing by using, e.g. FFT, MDCT, or Complex QMF transforms. Each domain has different properties resulting in different requirements in headroom and precision to manage distortion and noise in such systems. The aim of the work is to investigate such requirements for immersive and multichannel real-world content and to develop guidelines for headroom and data precision requirements for processing such content. Processing blocks to study are the ones currently used in Dolby systems that can include smart implementations of data domain transforms, reduction of audio channels like rendering to a speaker configuration and down-mixing and signal loudness processing.

Resumen

En las técnicas de procesado de audio actuales, la señal temporal de audio se transforma en otros dominios de datos para su procesado, usando, por ejemplo, la FFT, MDCT o la QMF compleja. Cada uno de estos dominios tiene sus propiedades diferentes, lo que implica tener que satisfacer diferentes requisitos para gestionar correctamente la distorsión y el ruido en cada sistema. El objetivo del trabajo propuesto es profundizar sobre estos requisitos cuando codificamos señales reales usando técnicas de sonido inmersivo o sonido multicanal, y poder proponer pautas para conseguir el adecuado margen dinámico de protección frente a la distorsión (*headroom*) y la precisión necesaria en los sistemas de codificación de punto fijo. Los procesos que se aplican a las señales que van a estudiarse serán los mismo que se utilizan actualmente en los sistemas Dolby, y que incluyen implementaciones que mejoran la eficiencia tales como el uso de transformaciones de los datos en otros dominios, la reducción del número de canales de audio (como en la transformación y mezcla de las señales para la reproducción en una configuración de un altavoz), y el procesado de la sonoridad de la señal.

Key words

Headroom, Precision, MDCT, QMF, Real World Signals.

Table of contents

Table of figures

Table of tables

1. Introduction

Audio codecs have evolved dramatically in the last decades, making the coding and decoding fairly complex processes. The codecs and formats created and used at Dolby are no exception, as the processing chain of encoding and decoding of the new formats involve processing a signal in several data domains. To ensure the maximum quality, and the minimum distortion and loss of information, headroom and precision requirements must be predicted before designing the whole processing chain.

Because of the complexity of the processes involved, the headroom and precision requirements cannot be predicted using worst-case scenarios, as the requirements would be too expensive and inefficient. Therefore, a deep understanding of the processes involved and a better knowledge of the nature of the real-world signals to process are needed.

1.1. Objective

The aim of this work is to provide insides over the theoretical functioning of some processes and transforms, together with the effects of quantization in different data domains, involved in the processing chain used in some Dolby products, combined with their relation with the loudness standards. Also, it is intended to provide information about the functioning and the distortion introduced by the real fixed-point implementations of those processes for their better understanding.

Besides the processes, this work also aims to provide information about the nature of real world signals and its behavior in the different data domains for a better understanding of the real precision and headroom requirements needed to process those signals.

2. Methodology

This work has been developed during a stay at Dolby Germany GmbH in the city of Nuremberg (Bayern), Germany. The work was thought by Holger Hörig, a Dolby engineer, who felt the necessity to better know and understand the behavior of certain processes involved in many Dolby products, to better determine the headroom and the precision required in every stage of the processing chain.

The interest was focused in three main topics: the effects of quantization in several data domains, the precision loss and behavior of the error introduced by the real implementations of some processing blocks, and the analysis of statistical data of the characteristics of real world signals.

The first topic mentioned above was the first to be approached, as it consisted in theoretical tests that were helpful to familiarize with the concepts, transforms and processing blocks that would have to be used throughout the whole work. The theoretical tests done were experiments to test theoretical results to build a recompilation of concrete and informative examples that may be useful for further developments in the processing chain.

The second topic mentioned, was then approached to familiarize with the algorithms and implementations used at Dolby. The implementations of the processing blocks are highly optimized processes for different configurations and processors. The tests here made were basically comparisons between the results obtained with maximum resolution and the results obtained for several fixed-point implementations that introduce a certain amount of error. This way, the precision loss and the behavior of the error introduced in those processing blocks could be studied, providing useful information for deeper understanding of the origin of the error introduced.

For the last main topic here mentioned, real world content had to be analyzed. To do so, an analysis tool has been programmed in C using the necessary components from the Dolby libraries to read different file types, process the data, and store the statistical values of the results. To develop the tool, it was first necessary to familiarize with all the tools and libraries needed to be able to make the several parts and tools work together. When the results were obtained, an interpretation was made to find the best way to visualize them so useful comparisons could be made, and conclusions could be drawn. This final process has been made with MATLAB where several codes have also been programmed to visualize the results and to calculate the statistical parameters needed.

The result of the process is a useful resource for engineers to have more tools to better develop further applications and increase the processing chain, ensuring enough headroom and precision requirements to deliver the best signal quality at the end of each process.

3. Theory

This part of the work is focused in showing the effects of the quantization of a signal in different data domains from a theoretical point of view. The domains here explained are PCM (Pulse Code Modulation), MDCT (Modified Discrete Cosine Transform) and QMF (Quadrature Mirrored Filter).

The transforms used in this part, are 64 bits floating point implementations that ensure great accuracy, and in the case of the MDCT, perfect reconstruction when recovering the original signal. This way, the effects of the quantization in those domains can be studied without considering any other distortions in the signal.

When a signal is discretized to a fixed-point domain it is sampled in time and amplitude, in the time domain, and frequency and amplitude, in other frequency domains. This sampling will cause a loss of resolution of the signal in both dimensions which will be translated in a loss of precision and dynamic range [1].

Therefore, in this section, the main properties under test are the amount of error introduced by the discretization, and how this error introduced affects the maximal and minimal values in time and frequency domain.

3.1. PCM domain

The PCM domain is the most commonly known domain as it is the one in which most of the signals have been historically represented. There has been a lot of research on the effects of discretization in this domain and therefore, this part will only focus in some special interesting cases.

3.1.1. SNR approximation of a discretized sinusoid

By modeling the quantization error, the SNR (Signal to Noise Ratio) of a discretized sinusoid signal can be approximated as:

$$SNR_{dB} = 1.761 + 6.02 \cdot w$$

Where *w* is the bit word length used for quantization [2].

And therefore, the SNR values for different word lengths are:

N° of bits	SNR for FS sinusoid (dB)
8	49.92
16	98.08
24	146.24
32	194.40

Table 1. SNR for Full-Scale sinusoids depending on the bit-depth

This approximation is introduced, because a sinusoid is an interesting case when studying the effects of discretization with loudness normalized signals, as the loudness level of a stereo or multichannel file containing a pure sinusoid in each channel is easy to predict. The loudness normalization followed in this

part of the paper is the EBU (European Broadcasting Union) r128 recommendation, where it is specified that the target level of a signal should be -23.0 LUFS (or -23 LKFS).[1]

3.1.2. Stereo sinusoid signal at a reference level

The EBU recommendation states that a 1000 Hz sinusoid at -23 dBFS peak level (per channel), with 20 seconds of duration, should have a measured loudness of -23±0.1 LUFS in all three scales defined in the recommendation: Momentary (M), Short-term (S), and Integrated (I). [3]

Knowing the necessary amplitude of the wave to reach the target level, the SNR for such a loudness normalized signal, with different quantization word lengths, can be easily calculated with the previous formula. SNR for different number of quantization bits for a stereo 1000 Hz sinusoid at an EBU reference level of -23 LUFS:

N° of bits	Approximated SNR (dB)
8	26.92
16	75.08
24	123.24
32	171.4

Table 2. SNR approximation for loudness normalized stereo file containing 1000Hz sinusoids at -23 LUFS depending on the bit-depth

This is a rough approximation, because the quantization error for such a tonal signal is just approximated as a saw-tooth wave with uniform distribution. Even though, a quick simulation in MATLAB has been done, where a pure sinusoid is generated with the necessary amplitude to fulfill the required reference level, and then it has been quantized with different bit depths. The results in the next table show that the approximation of the SNR done by the previous calculations is a rough, but also a quick good approximation as in most of the cases the difference was smaller than 1 dB.

N° of bits	Approximated SNR (dB)	SNR of the real signal (dB)
8	26.92	27.39
16	75.08	74.83
24	123.24	123.29
32	171.40	171.03

Table 3. SNR approximation compared to real calculation

3.1.3. 5.1 multichannel sinusoid signal at a reference level

The EBU recommendation also indicates the necessary levels to reach the target loudness level of the channels for a 5.1 multichannel signal, each of them containing a 1000 Hz sinusoid signal. The levels are:

- −28.0 dBFS in L and R.
- −24.0 dBFS in C.
- −30.0 dBFS in Ls and Rs.
- LFE is not taken into account in the EBU recommendation.

[1] For more information about the EBU recommendation r128 and the loudness measurement process see: [4] [3]

As in the previous example, the SNR for each of the channels can be calculated depending on the bit depth of the quantization:

	L, R (-28 dBFS)	C (-24 dBFS)	Ls, Rs (-30 dBFS)
N° of bits	**Approximated SNR (dB)**	**Approximated SNR (dB)**	**Approximated SNR (dB)**
8	21.92	25.92	19.92
16	70.08	74.08	68.08
24	118.24	122.24	116.24
32	166.4	170.4	164.4

Table 4. SNR for loudness normalized 5.1 multichannel file containing 1000Hz sinusoids at -23 LUFS depending on the bit-depth

3.1.4. SNR for different types of signals at reference level

The loudness level of a signal is very frequency dependent, as in the loudness measurement process the signal is filtered with a weighting curve (K-weighting curve)[2] [4]. This can produce a loudness level difference of up to 24 dB for two signals with the same amplitude, but with different frequency content.

Because of this frequency dependency, a sinusoid is not the most representative signal to measure. Therefore, the SNR has been calculated for different types of signals with different bit depths, and the results can be seen in the next figure. The signals used here are mono signals, normalized to -23 LUFS taking the signals as the center channel when measuring its loudness level.

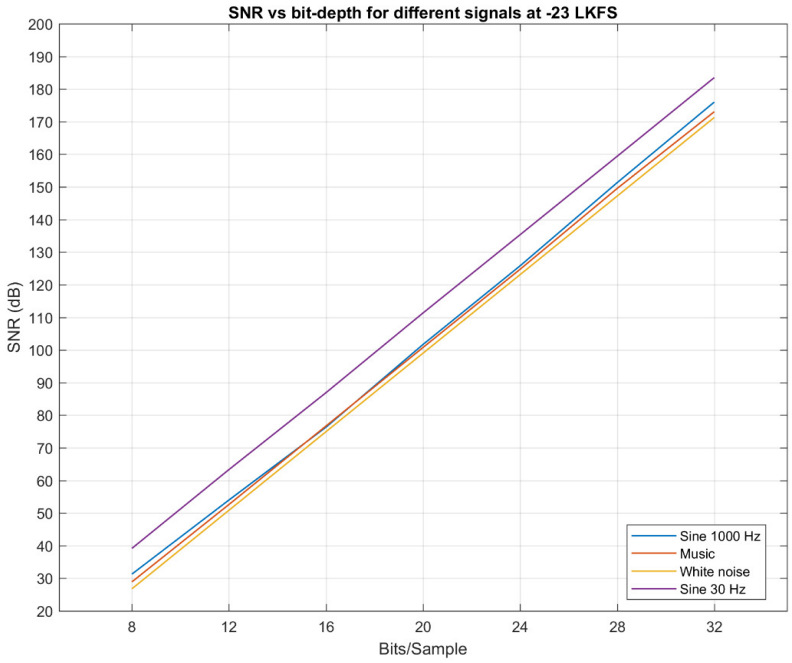

Figure 1. SNR for different bit depths and signal types normalized to -23 LUFS

[2] For more information about the K-weighting curve and the loudness level calculation process, consult the ITU recommendation BS.1770-4 [4].

SNR in dB	8 bits	12 bits	16 bits	20 bits	24 bits	28 bits	32 bits
Sine 1000 Hz	31.28	54.00	76.33	101.70	125.82	151.31	176.01
Music	28.90	52.62	76.70	100.78	124.75	149.53	173.05
White Noise	26.77	50.86	74.95	99.03	123.11	147.20	171.28
Sine 30 Hz	39.16	63.33	86.91	111.36	135.35	159.40	183.51

Table 5. Numeric results of SNR for different bit depths and signal types normalized to -23 LUFS

As it can be seen, the SNR for different signals at the target level varies a lot depending on the frequency content of a signal.

Considering that the dynamic range of a human hearing is between 110 and 130 dB of dynamic range, we could state, that 24 quantization bits would never produce audible quantization noise, as they provide enough SNR to fit the whole perceivable dynamic range, even with signal at a reference level.

3.1.5. Headroom

By quantizing a signal, its amplitude is represented with a limited number of quantization steps. This will round the real amplitude of the signal to the nearest quantization step, and therefore, it will change the original amplitude of the signal.

Also, the time resolution used when discretizing the signal may affect the amplitude of the discretized signal, as some of the original peak values may lay between discrete samples. Those peaks are called inter-sample peaks, and they can appear when recovering the original signal to the analog domain or when doing some kinds of processing, like re- or over-sampling. One way to detect inter-sample peaks is using a True-Peak meter, which will up sample the discretized signal, typically by a factor of 4, to give a more accurate peak level, called True Peak level[3].

Because of the previously explained reasons, leaving free headroom is a good practice before discretizing a signal, but setting enough headroom for any case it is a difficult task, as it will depend on the nature of the signal.

To set enough headroom possible bad or worst-case scenarios, like the next one explained, are studied:

Inter-sample peaks bad case scenario

A typical bad case scenario will occur when the signal has a frequency of ¼ of the sampling frequency. If so, all the peaks of the signal may not be correctly sampled. Example below:

Signal: pure 12000 Hz sinus, sampled at 48000 Hz, and with a phase shift of 45°

[3] To know more about the True-Peak calculation process, see the ITU BS. 1770 recommendation Annex 2. [4]

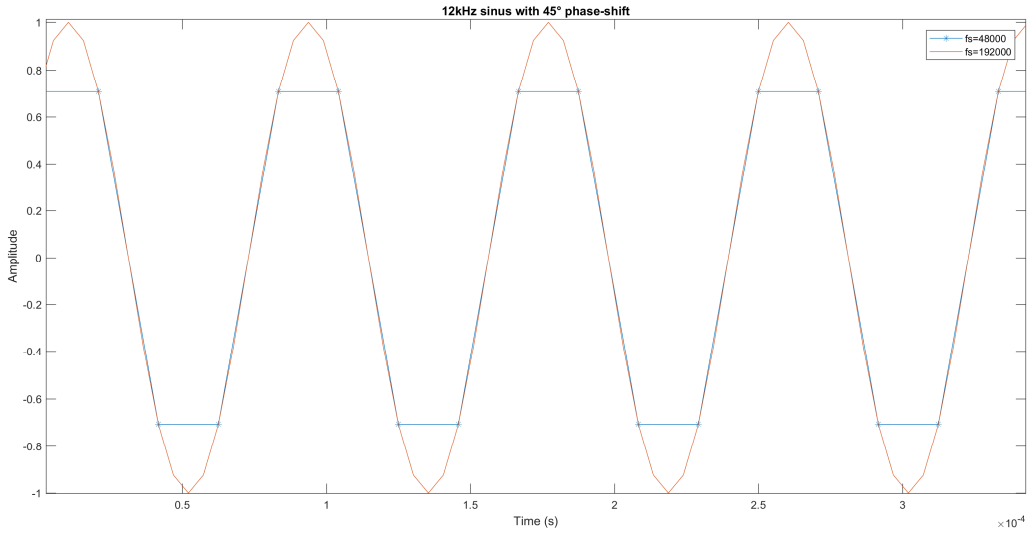

Figure 2. Signal with high inter-sample peaks

The true peaks of the signal are 3 dB higher than the ones detected at the sampled signal.

In this case the signal with the high peaks is normalized to 0 dBFS, and therefore, the quantized signal has a lower peak level of -3 dB. If the quantized signal would have been normalized to 0 dBFS, then the inter-sample peaks would have reached 3dBFS, creating possible errors when processing the signal or distortions at the reproduction state.

The EBU recommends a maximum True Peak level of -1dBTP. In the next figure we can see the previous signal adjusted at the recommended level and quantized:

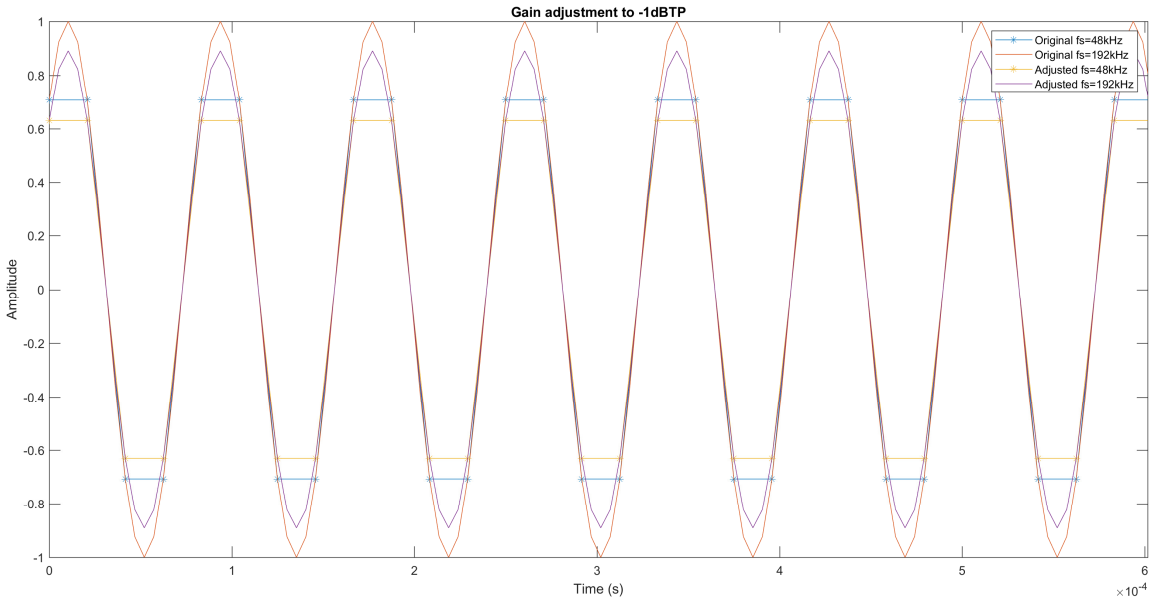

Figure 3. Signal with high inter-sample peaks normalized to -1dBTP

In this case an adjustment of -1dB has been made, but in the case that the samples were quantized at Full-Scale level, an adjustment of -4dB would have had to be done.

Inter-sample peaks worst case scenario

The worst case scenario would be an infinite sinusoid critically sampled with a phase shift at the middle. This signal, when going to the analog domain, will cause a peak far beyond full scale (in a theoretical case, even to the infinite), as when the samples are multiplied by a *sinc* function a summation of the lobes of the *sinc* functions will occur at the center. In the next image we present an example with finite duration:

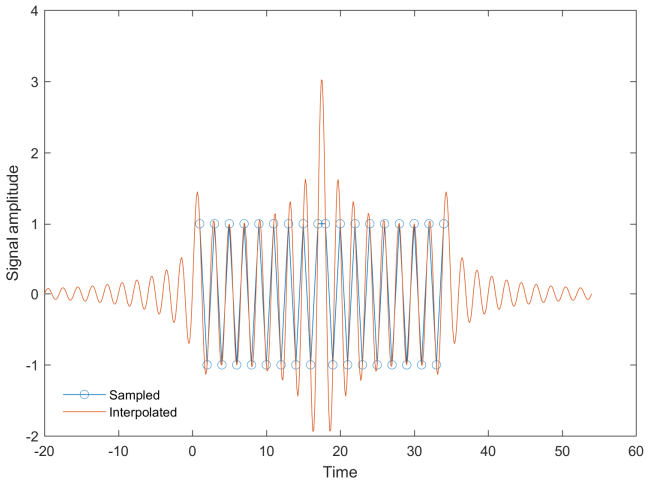

Figure 4. Signal with high peak when converted to the analog domain

Headroom conclusion

Of course, this last case will probably never occur in real world signals, and therefore it should not be considered when searching for a sufficient headroom requirement. Looking at the first example of inter-sample peak detection, is easy to understand the EBU recommendation of -1 dBTP. The typical true peak meter works at 4 times the original sampling frequency and can also miss the peaks sometimes.

According to *True peak metering – a tutorial review by Ian Dash,* this method reduces significantly the peak detection error, but it can still have 0.17 dB of error [5]. Therefore, the -1 dBTP maximum true peak level recommended by the EBU seems to be enough, but only when it is sure that the true peak detection process has enough precision. If not, -2 or -3 dBTP would be preferable.

Having that in mind and considering that true peaks may be 3 dB higher than the sample peaks, the actual headroom needed that inter sample peaks do not cause problems in further stages would be between 4 and 6 dB headroom.

			Error using ideal interpolation filter, dB				
			Oversample factor				
f_s/	Phase, degrees	Maximum sample value	1 (base rate)	2	4	8	16
3	30	0.866	-1.25	0.00	-0.30	-0.07	-0.02
4	45	0.707	-3.01	-0.69	-0.17	-0.04	-0.01
5	18	0.951	-0.44	0.00	-0.11	-0.03	-0.01
6	30	0.866	-1.25	-0.30	-0.07	-0.02	0.00

Table 6. True peak error on oversampled sine waves from sample timing [5]

3.2. MDCT domain

In this section, some theoretical concepts about the Modified Discrete Cosine Transform and the discretization of the signal in such a domain will be studied. To do so, a theoretical implementation of the transform has been implemented in MATLAB, and some tests have been made.

The Modified Discrete Cosine Transform (MDCT) is a transform widely used for audio signal compression [6]. It is a modification of the Discrete Cosine Transform that uses 50% overlapped windows to avoid blocking artifacts. Even with the 50% overlapping, no extra data is generated because of a following 50% decimation of the data [7]. Thanks to the property of Time Domain Aliasing Cancelation (TDAC) the aliasing introduced by the decimation is cancelled when adding together the resulting adjacent blocks of the inverse transform [8] [9].

To better understand the implementation here used, the mathematical expression of the implementation for the transform should be introduced:

- Forward MDCT: $$\alpha_r = \frac{1}{N}\sum_{k=0}^{2N-1} \tilde{a}_k \cdot \cos\left[\pi \frac{\left(k+\frac{N+1}{2}\right)\left(r+\frac{1}{2}\right)}{N}\right] \qquad r = 0, \dots, N-1$$

- Inverse MDCT: $$\hat{a}_k = 2\sum_{r=0}^{N-1} \alpha_r \cdot \cos\left[\pi \frac{\left(k+\frac{N+1}{2}\right)\left(r+\frac{1}{2}\right)}{N}\right] \qquad k = 0, \dots, 2N-1$$

Where $\tilde{a}_k = h_k a_k$ is the input signal multiplied by the window function. In this case the window used is a sine window which assures perfect reconstruction as it fulfills the next constrains for perfect reconstruction [10]:

$$h_k = h_{2N-1-k}$$
$$h_k^2 + h_{k+N}^2 = 1$$

And the sine window is defined as:

$$h_k = \sin\left[\frac{\pi\left(k+\frac{1}{2}\right)}{2N}\right] \qquad k = 0, \dots, 2N-1$$

There are many combinations of the gain coefficients in the literature. In this case, gains used in the transforms are 1/N in the analysis one, and 2 in the synthesis one. This way it is very unlikely to obtain coefficients in the MDCT domain that are beyond the ±1 range [11].

Apart of the formulas, a block diagram provides a good perspective of the whole process of forward and inverse transform:

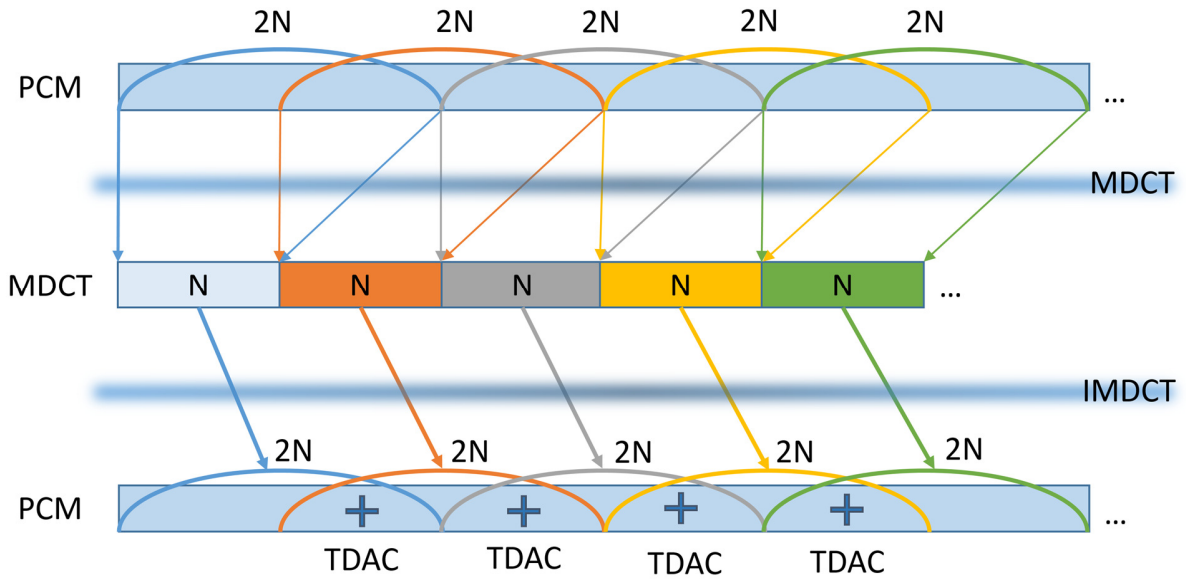

Figure 5. MDCT and IMDCT block processing

3.2.1. Quantization effects in the MDCT domain

The MDCT transform ensures perfect reconstruction, so it is a transparent process for the signal, so no error is introduced. This environment is perfect to study the effects of the quantization of a signal in the MDCT domain with different window lengths and quantization bit depths. The process followed for the purpose is the next one:

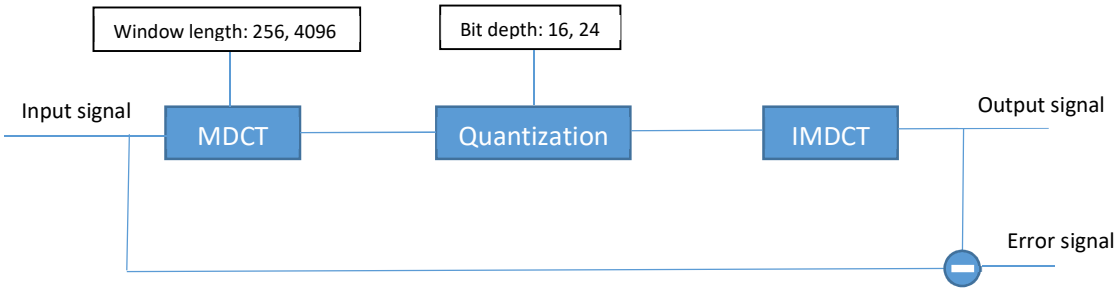

Figure 6. Block diagram of the analysis of the quantization effects in the MDCT domain

As the simulation is made under MATLAB environment with 64 bits floating point number representation, the error introduced by the calculation process is negligible.

The first signals used to test the effects of quantization are pure sinusoids. In particular, a sinusoid with a frequency multiple of the frequency resolution of the transform, which depends on the window length and the sampling frequency used. The other signal is a sinusoid with no direct relation with the frequency resolution of the transform.

3.2.1.1. Input signal multiple of the transform resolution

In the first case, the frequency used is 1125 Hz, as the window length used is 256 and the sampling frequency is 48000 Hz. When this signal is transformed to the MDCT domain, the energy is concentrated in very few coefficients, so the rest will have very little amplitudes. In the next image we can see the MDCT coefficients of one windowed input signal block in dBFS:

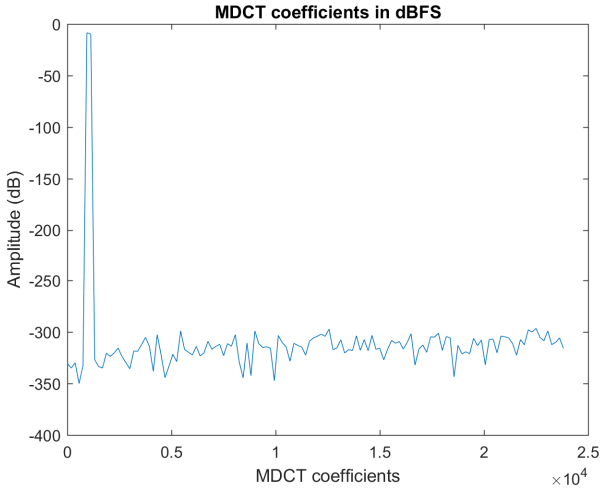

Figure 7. 256 MDCT coefficients in dBFS for an 1125 Hz sinusoid

It can already be seen that the smallest coefficients will be all quantized to the smallest quantizable value when using any number of bits for the quantization. This will produce a very small error, as the coefficients with the most energy can be quantized, but the error will be much input dependent, as the error produced is proportional to the amplitude of the smallest coefficients.

When using a larger window, the coefficients have smaller values, as the energy is distributed thorough out more bands, so the small coefficients are even smaller, and therefore, the conditions remain equal to the previous example.

In the next image, the spectrum of the original signal and the error signal when using 16 bits for the quantization can be seen:

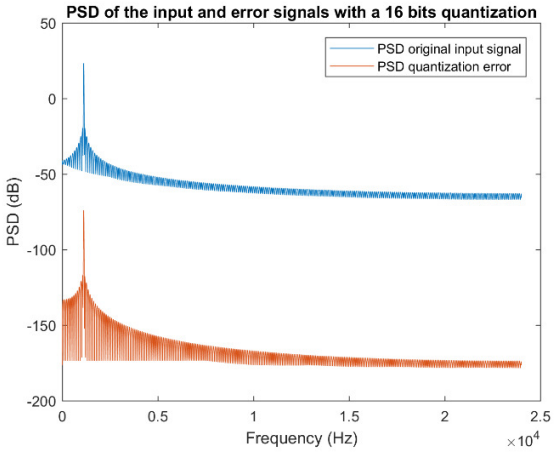

Figure 8. PSD of 1125 Hz input and error signal after 16 bits quantization in the MDCT domain[4][5]

3.2.1.2. Input signal not multiple of the transform resolution

When the input signal has no simple relation with the frequency resolution of the transform, as for example with an 1142 Hz sinusoid in our case, the results after the transform are much different, as the energy is not concentrated in such few bands. In the next figure, the MDCT coefficients for an 1142 Hz sinusoid can be seen:

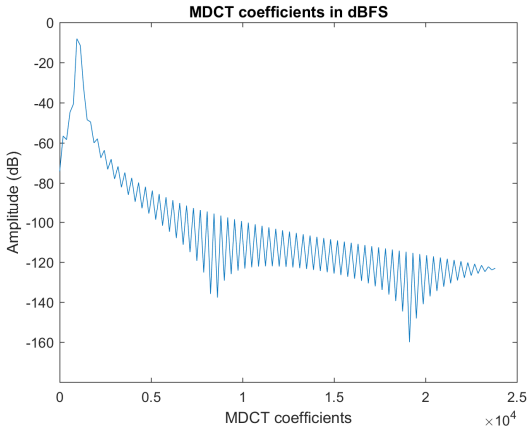

Figure 9. 256 MDCT coefficients in dBFS for an 1142 Hz sinusoid

In this case, it can be seen that more coefficients are in the achievable dynamic range with a 16 bit quantization. This will cause a more signal-independent quantization error. The more quantization bits used, the more coefficients are in the reachable dynamic range, and the resulting error has a more broadly distributed frequency content. The same way, the longer the window length, the smaller the coefficients, and therefore, the more signal dependent becomes the quantization error.

[4] PSD (Power Spectral Density)

[5] The spectrum has been obtained by performing a 4096 samples long FFT with a rectangular window by using the default build in function (`fft()`) of MATLAB.

To better see the quantization effects on such a signal, a small amplitude 12715.75 Hz sinusoid is used, as it will appear in the middle of the spectrum. In the next figures, the quantization errors for 16 bits quantization and for 24-bit quantization can be seen.

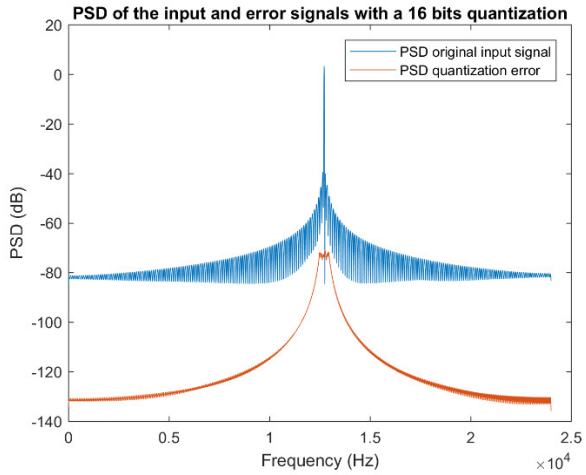

Figure 10. PSD of 12715.75 Hz sinus input and error signal after 16 bits quantization in the MDCT domain

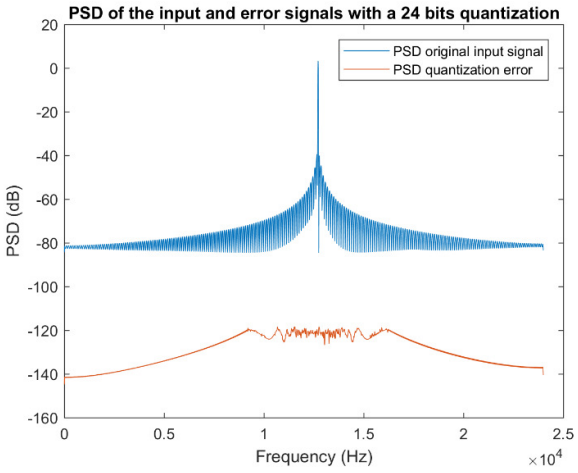

Figure 11. PSD of 12715.75 Hz sinus input and error signal after 24 bits quantization in the MDCT domain

As it can be seen, the error is spread out through the spectrum when using more bits, and therefore, the error is more signal independent.

3.2.1.3. Dithering

To make the quantization error more independent of the input signal, dithering can be used, but when using dithering, the total SNR decreases. The effects of dithering in the MDCT domain are similar to the effects in the PCM domain, as the quantization error becomes a random noise spread throughout the whole spectrum of the signal. To see the effects, a triangular dither of one quantization step has been applied to the last example:

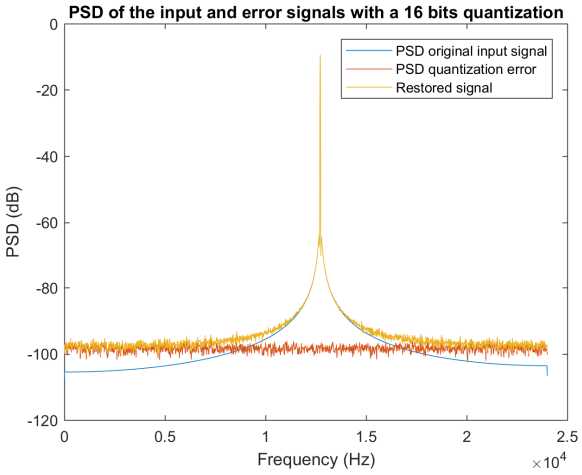

Figure 12. PSD of 12715.75 Hz input, error and recovered signal after 16 bits quantization in the MDCT domain with dithering

As it can be seen, now the error is higher in some regions, but much more independent from the input signal.

In the next sum up table, some SNR and THD values for the previous used signals can be seen:

		16 bits		24 bits	
	MDCT coefficients	**SNR (dB)**	**THD (dBC)[6]**	**SNR (dB)**	**THD (dBC)[6]**
1125 Hz	256	91.61	-312	138.87	-312
	4096	91.61	-301	138.87	-301
1142 Hz	256	75.75	-95.4	122.17	-147.66
	4096	71.09	-92.64	110.41	-137.23
1142 Hz with dithering	256	69.24	-∞	117.38	-∞
	4096	57.21	-∞	105.36	-∞

Table 7. SNR and THD for different bit depths and signals for quantization in the MDCT domain

3.2.1.4. Clipping and wraparound in MDCT domain

For some signals, energy can be concentrated in very few coefficients in the MDCT domain. If gain adjustments are not carefully made, signals with high true peak levels could generate values beyond the ±1 range when transformed to the MDCT domain. When quantizing such a signal, the MDCT values will be clipped or wrapped around and this will cause distortion when performing the inverse MDCT.

In the next examples, the signal used is a Full Scale square wave, with a fundamental frequency perfectly centered in a MDCT line. This will concentrate the maximum amount of energy in very few coefficients and values in the MDCT domain beyond ±1 will be obtained. This signal is used, because with the gains used in the simulation, a lot of energy is needed to obtain such values, but with other gain factors in the formula, values beyond ±1 could appear easily with other signals.

The signal used is a square wave with a fundamental frequency of 843.75, as a 256 window-length (128 MDCT length) is used. The result of the transform is the next one:

[6] This THD (Total Harmonic Distortion) values have been obtained by using the default configuration of the `thd()` function from Matlab.

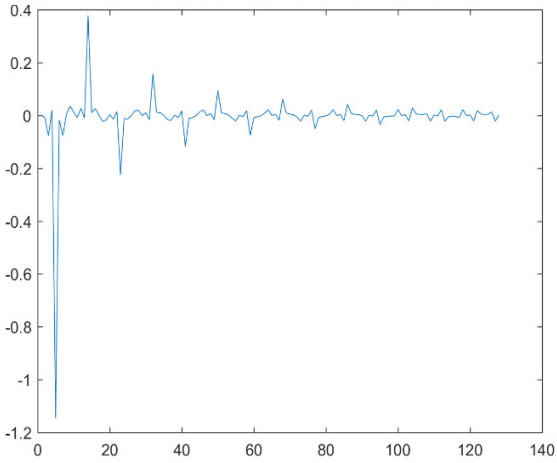

Figure 13. MDCT coefficients beyond full scale for a square wave of 843.75 Hz fundamental frequency

As it can be seen, one of the bands of the MDCT has a value beyond FS. When quantizing such a signal, these values would be clipped or wrapped around, causing unwanted distortions when performing the inverse transform.

Clipping

The effect of clipping, in this example, is very visual, as it can be seen in the next figure. It is well known that a square wave is formed of an infinite sum of the odd harmonics of its fundamental frequency. In this case, the fundamental frequency will be clipped, and therefore, will be interpreted by the inverse transform as it had less amplitude than it actually should have. The only coefficient clipped is the one representing the fundamental frequency, so when all the other harmonics are added together, the result will not look like a square wave anymore because of the lack of amplitude of the fundamental. In the next image the result can be seen:

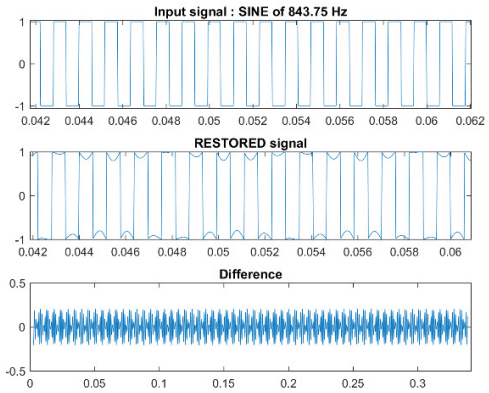

Figure 14. Input, restored and difference signal after clipping in the MDCT domain

This of course causes a very tonal distortion, because all the information lost belongs to one single band (the one that has been clipped).

Wrapping around

To visualize the effect of the wraparound, the previous example has been taken, but this time, the coefficients will not be clipped to ±1, but instead they will be wrapped around. The coefficients now look like this:

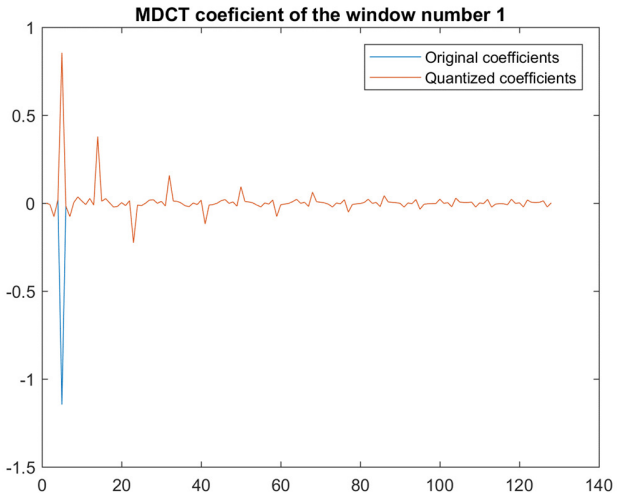

Figure 15. MDCT wrapped around coefficients

As in the previous example, there is a change in the MDCT bin corresponding to the fundamental frequency of the signal, but in this case, there is also a phase shift. When adding the phasors that form the signal, this phase shift will produce a much more different result that could be very different than the original one. In the next figure, the result of the process can be seen:

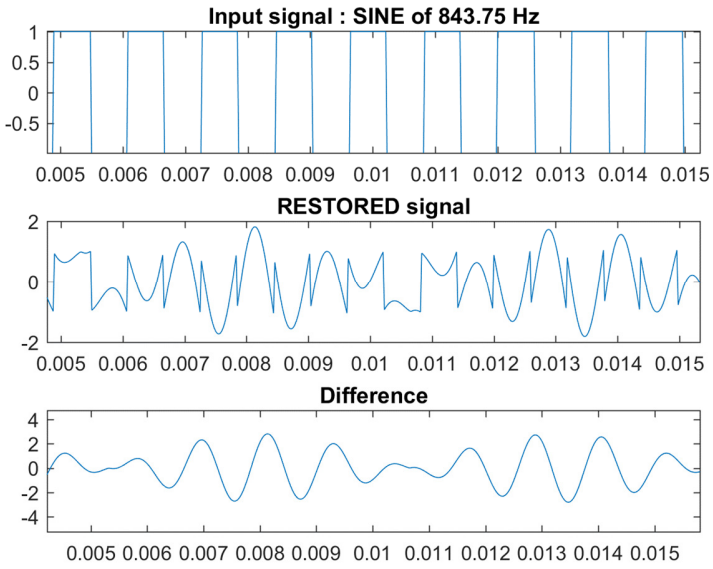

Figure 16. Input, restored and difference signal after wrapping around in the MDCT domain

Clipping and wrap around conclusion

By seeing the resulting signals of both cases, is easy to distinguish that wrapping around causes more distortion to the signal. Clipping by saturation, will just change the amplitude of some of the components, but wrapping around would change the amplitude and the phase of the components. Also, informal hearing tests have been performed, and the signals that have been wrapped around sounded much more distorted than the ones that have been just clipped.

3.2.2. True peak level

To see how the quantization in the MDCT domain affects the true peak level of a signal, we will measure the true peak level of a signal at the input and at the output of the transform.

The signal used is a classic example for true peak measurement examples, and it has been previously used in section *3.1.5*. It is a sinusoid with an oscillation frequency of a quarter of the sampling frequency, with a 45° phase shift. In this case, since the sampling frequency used is 48000 Hz, the input signal will be a 12000 Hz sinusoid.

This kind of signal, when sampled and normalized to 1, has a true peak level of +3dBFS, as it has high inter-sample peaks. With the simulation used, the signal has 3.02 dBTP, and after the transform and inverse transform, the signal still has a true-peak of 3.02 dBTP, so it has not experienced any change in the true peak level.

Some other tests with other signals with different true peak levels have been performed and the true peak value of the output signal has been always very similar to the input signal. Therefore, it can be concluded that the true peak level of a signal is maintained, independently of the sample peaks values.

3.3. QMF domain

Quadrature Mirrored Filters (QMF) are known to be a filter bank that divides the signal in two or more lapped sub-bands [12] [13]. As the frequency content of the original signal is split out, down sampling in the sub band signals can be performed without causing aliasing. When the sub-band signal is sub-sampled by a factor equal to the number of sub-bands, the system is known as a critically sampled system. The split, sub-sampled signal is usually processed to apply some changes to the original signal. Afterwards, in the synthesis part, the sub-band signals are again up-sampled and added together [14]. This kind of filter banks are extensively used in many fields, and very often used in audio, where they are used to perform any kind of frequency band processing, such as perceptual audio coding or multi-band equalizers [15]. In the next figure, a block diagram of an M band critically sampled QMF bank can be seen:

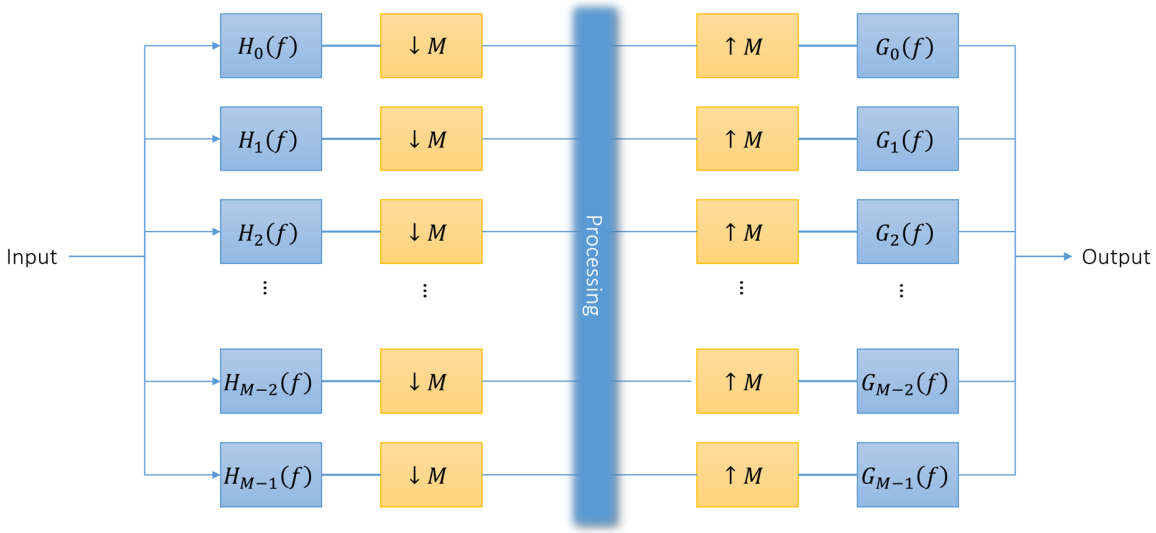

Figure 17. Critically sampled analysis and synthesis QMF filter bank

The QMF filter bank used at Dolby is a hybrid lapped filter bank as it divides the signal first in 64 lapped bands, and then it divides again the 3 lower sub-bands in 8, 4, and 4 smaller bands, respectively, for more frequency resolution in the low frequencies. Also, the filter used, is the same one for all the bands. This is possible because the signal itself is first multiplied by a complex factor that moves the frequency content of the signal to fit the corresponding band. Therefore, the transform used at Dolby components is called HCQMF (Hybrid Complex Quadrature Mirrored Filter). Because of its complex representation, the QMF transform used by Dolby is not a critically sampled transform, as there are two times the minimum values needed to avoid aliasing, as there is a real and an imaginary value for every coefficient.

In this section, the filter bank (or transform) used is just a complex QMF with 64 bands, so the lower bands are not further sub-divided. Even though, this 64 band CQMF is the actual primary transform used in the Dolby components, as it uses the same methodology of using the same FIR filter for all bands. The filter used is a result of an optimization process, to reduce distortion and leaking to adjacent bands. The filter used is the next one:

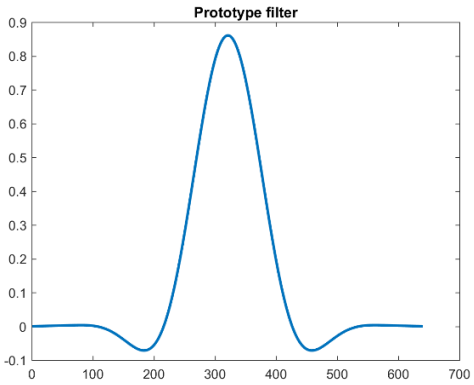

Figure 18. Prototype filter used in the QMF analysis

Even after the optimization process, the transform does not provide perfect reconstruction, and it introduces distortion to the signal. In the next figure, the frequency response of the filter bank can be seen, and the noise floor can be estimated by seeing the side lobes of the filters. In this image, the lower bands are indeed sub-divided.

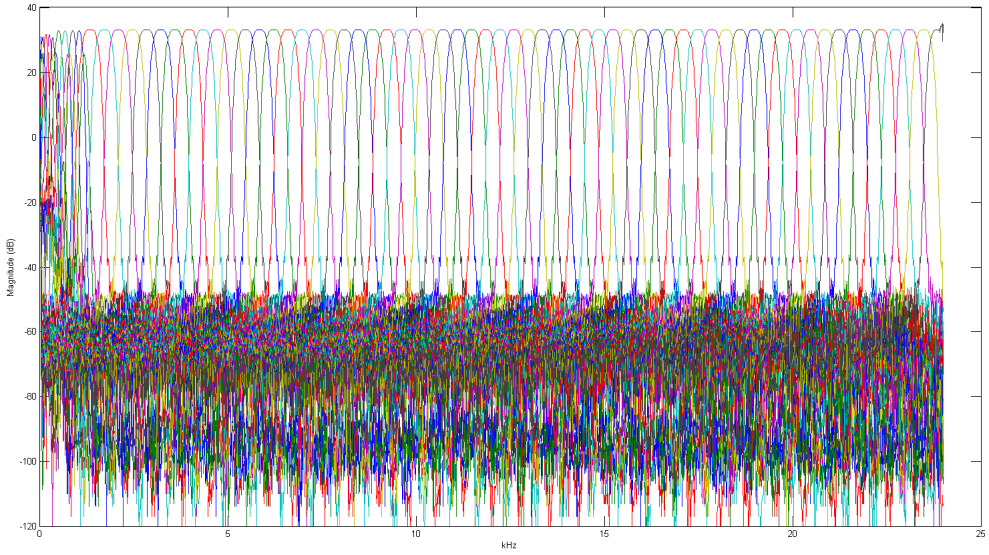

Figure 19. All filters of the HCQMF used in Dolby components[7]

The default gain factor of the filter is $\frac{1}{\sqrt{L}}$, but to avoid overflows, in this section the filter has again factor of $\frac{1}{L}$, where L is the length of the filter. This will be compensated at the inverse transform.

[7] Image provided by Dolby Laboratories

3.3.1. Distortion of the transform

To study the native distortion of the transform, several tests with three different signals have been made. The first signal is a sinusoid which frequency lies perfectly in the center of the band width of the filter, in this case 562.5 Hz. The second one, a 3000 Hz sinusoid with no relation with the filter, and the last one a white noise signal.

For the first signal, the transform presents high harmonic distortion throughout the whole spectrum. For the second one, the resulting error is similar to the first one, with high harmonic components, but much lower than in the first case. And for the white noise signal, the error introduced, is very uniform in the whole spectrum, so has no longer harmonic components, and it is lower than in the last two cases. In the next figures, the error for the 562.5 Hz sinusoid and the noise signal can be compared:

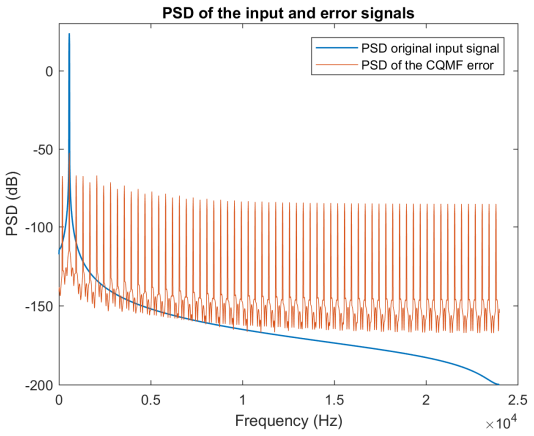

Figure 20. PSD of input and error signal for 562.5 Hz sinusoid after QMF transform

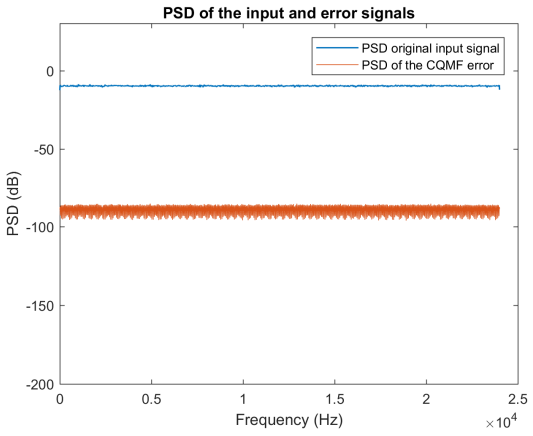

Figure 21. PSD of input and error signal for white noise after QMF transform

In the next table, the values for SNR and THD can be seen:

Signal	SNR (dB)	THD (dBC)
Sinus 562.5 Hz	75.33	-91.89
Sinus 3000 Hz	75.53	-107.31
White noise	78.29	-∞

Table 8. SNR and THD values for different signal types for the QMF transform

From all these three examples we can conclude that the transform works better when the energy of the signal is distributed in several bands, and it is not very focused in just one of them, as this would cause harmonic distortion that would spread throughout the spectrum.

To further check the harmonic distortion of the transform, another signal has been used. The signal is a typical test signal used in Dolby. As it can be seen in the next figure, the signal has several parts, all of them with different characteristics:

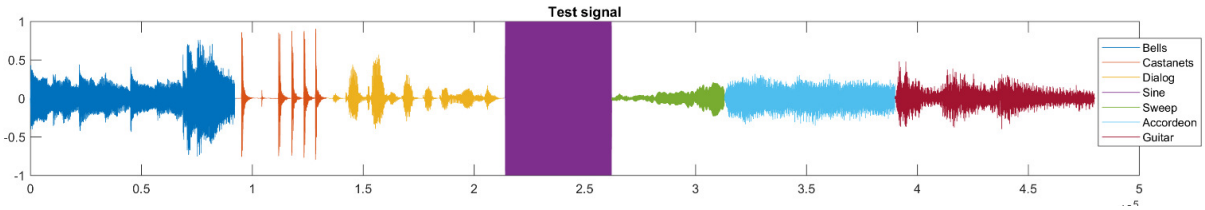

Figure 22. Test signal with different parts

The middle part of the signal is a full scale 1000 Hz sinusoid. To test the harmonic distortion, this signal will be given to the transform, and then the full-scale sinusoid will be subtracted of the original signal, and the results will be compared.

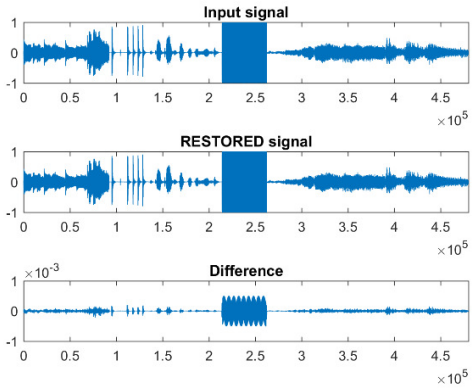

Figure 23. Input, restored and difference signal after QMF transform for test signal

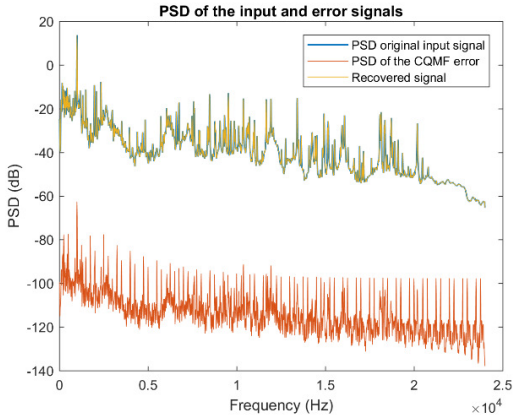

Figure 24. PSD of the input, error and restored signal after QMF transform for test signal

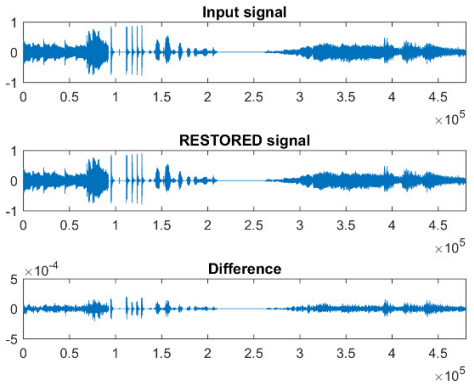

Figure 25. Input, restored and difference signal after QMF transform for test signal without full scale sinus

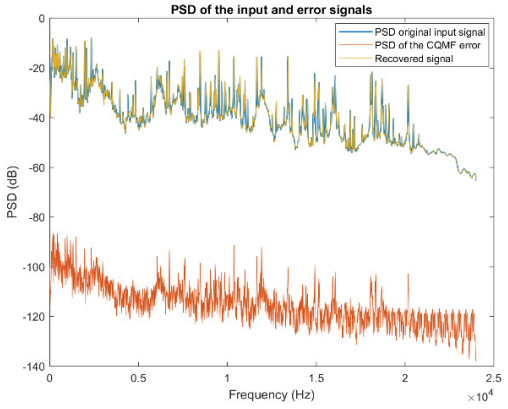

Figure 26. PSD of the input, error and restored signal after QMF transform for test signal without full scale sinus

As it can be seen, the maximal error is caused by the sinusoid, and this can be seen in the time and frequency domain error signals, where the harmonic components are very present. This can also be seen

in the resulting SNR values of both transforms. The first, with the sinusoid part has an SNR of 75.84 dB and the second one, without the sinusoid, has a SNR 3 dB better than the last one, 78.56 dB.

3.3.2. Frequency response of the transform

When looking at the frequency response of the transform, some lobes caused by the filter bank can be seen. In this section the difference between different regions of the spectrum will be presented in order to determine how this irregular frequency response of the transform could affect the result.

In the next image, the frequency response of the simulation used can be seen. Also, a detail of the same frequency response is drawn. The amplitude difference between specific frequencies can be seen.

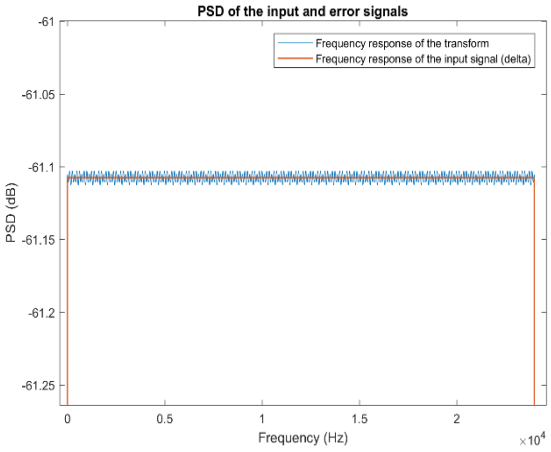

Figure 27. Flat frequency response and QMF filter bank frequency response

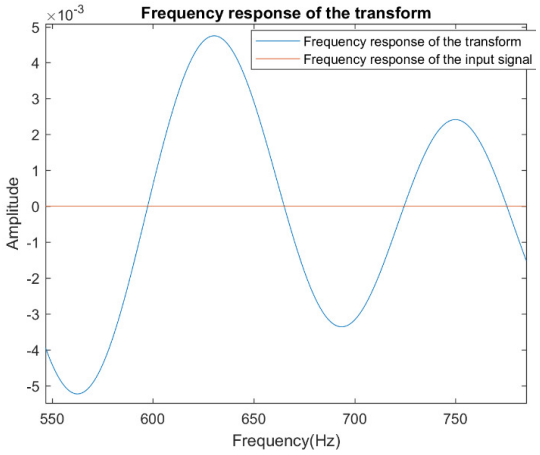

Figure 28. Detail of the QMF filter bank frequency response

At 630 Hz the transform has a positive lobe, but in 664.8 Hz the transform response is equal to flat response. Therefore, two sinusoids with the previous mentioned frequencies have been used to test the amplitude difference between spectral regions. In the next image, the spectrum of the error signal of both signals used can be seen.

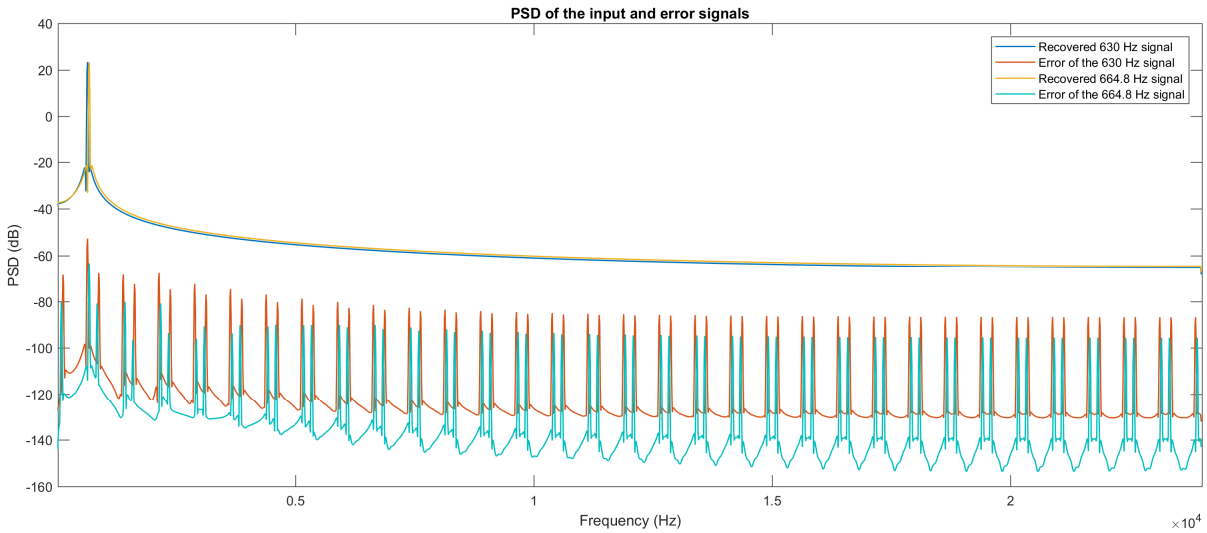

Figure 29. PSD of the recovered and error signals for two frequencies with diffrent amplitude in the QMF

As it can be seen, the recovered signals have almost the same amplitude, they defer only 0.1 dB from each other, but there is 10 dB of variation between amplitude of the harmonics of the error signals, as they have very small amplitudes. This error introduced by the frequency response variations of the transform is so small that can be neglected.

3.3.3. Quantization in the QMF domain

In this section the effects of discretization in the QMF domain will be presented. The procedure to test the effects will be the same done in the MDCT domain.

First a sinusoid input will be given to the transform, and then it will be quantized in the QMF domain with 16 and 24 bits. The sinusoid frequencies will be the same as in section *3.3.1*.

First the 3000 Hz is used, and it will be quantized with 16 bits. With such a quantization, some error is introduced as some coefficients of the QMF are smaller than a quantization step. In the next plot the effects of the quantization can be seen:

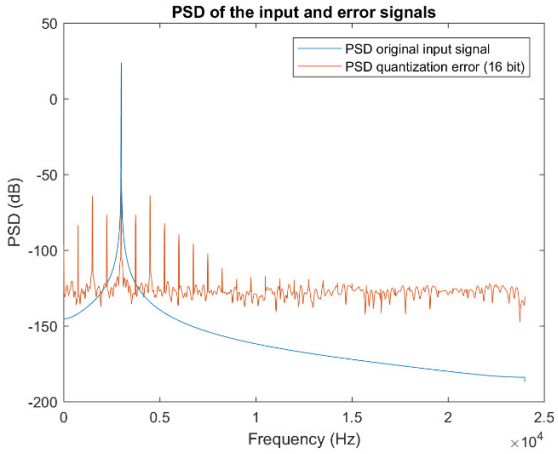

Figure 30. PSD of the input signal and error signal after quantizing with 16 bits in the QMF domain

Beyond a certain threshold the PSD of the error becomes flatter and looks like a PSD of a noise signal. The SNR is almost maintained in this case, but the THD is better in the quantized signal than in the directly recovered from the transform. This is because the harmonic behavior of the error introduced by the transform for such an input signal. When quantizing, the small harmonic components are lost, and the energy is distributed throughout the spectrum and therefore the THD is improved.

With a 24 bits quantization, there is enough precision for all the QMF coefficients to fit in the available dynamic range, and therefore the spectrum of the error will be very similar to the original one.

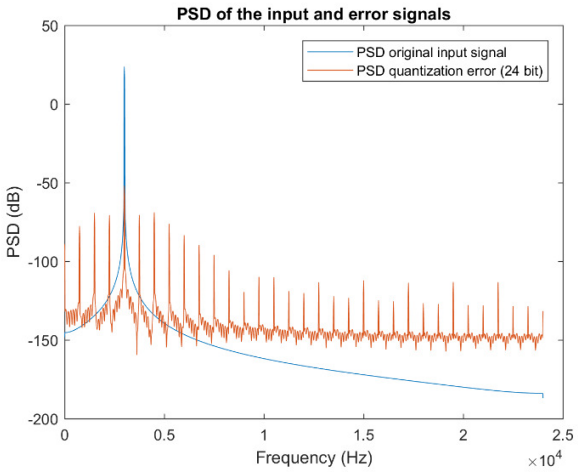

Figure 31. PSD of the input signal and error signal after quantizing with 24 bits in the QMF domain

In this case the SNR and the THD are also very similar to the original case without quantization, as the transform introduces more error than the quantization.

There is a very similar behavior when quantizing a signal with its frequency centered in one of the bands of the transform and when the input signal is noise. When using a 16 bit quantization, the quantization error introduced is bigger than the distortion of the transform, and when using 24 bits, the quantization error is smaller than the distortion introduced by the transform.

3.3.3.1. Clipping and wrap around in the QMF domain

The aim of this section is to study the effects of clipping and wraparounds in the QMF domain, therefore, values beyond the ±1 range should be obtained. This is easily achieved if the gain coefficients of the filter are adjusted, so in this section, the gain adjustment of the filter will be $\frac{1}{L/2}$ and it will be therefore afterwards corrected with $\frac{L}{4}$.

The effects of clipping and wrap around in the QMF domain are very similar to the effects in the MDCT domain, as in both cases, clipping or wrapping around is a change in the frequency domain, that will only affect the components of the bands affected in every case. These changes in the frequency domain, will change the result of synthesis process. Like in the MDCT domain, when a signal is clipped by saturation in the QMF domain there will be just an amplitude change in the frequency components of those affected bands, but when wraparound happens, the change is not only in amplitude, but also in phase. This could be not critical in some signals, but in others could change the nature of the signal completely.

The example used in the MDCT domain section used a square wave to exemplify these effects, as the results are very visual and easy to understand. Therefore, in the QMF domain, a square wave is also used.

In the next figures, the input and the output signals and the resulting QMF coefficients when clipped and wrapped around can be seen:

- Clipping

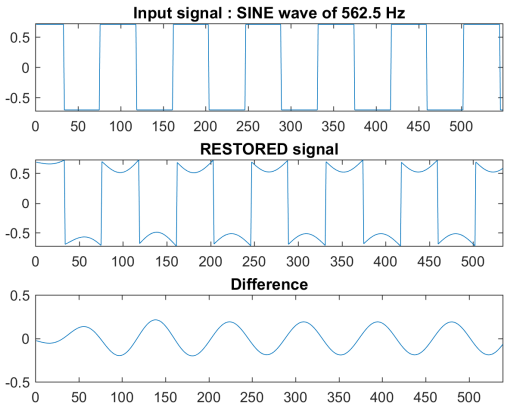

Figure 32. Input, restored and difference signal after clipping in the QMF domain

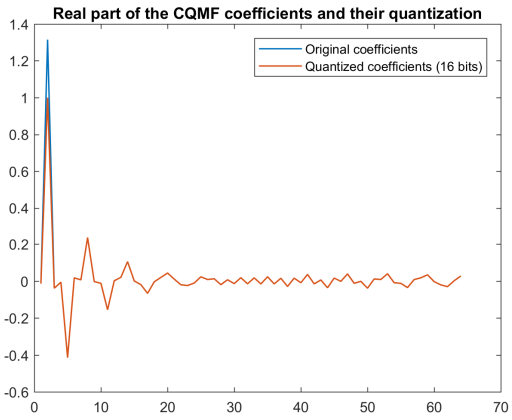

Figure 33. Original and clipped coefficients of the real part of the CQMF transform

- Wraparround

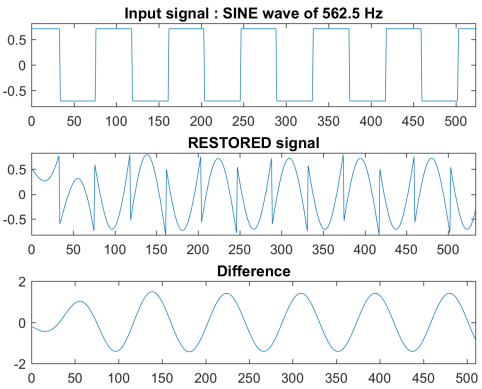

Figure 34. Input, restored and difference signal after wrapping around in the QMF domain

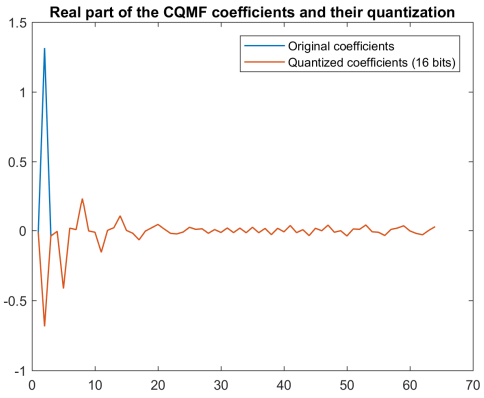

Figure 35. Original and wrapped around coefficients of the real part of the CQMF transform

As it can be seen, the effects are very similar as in the MDCT

3.3.3.2. Dithering

Now the effects of dithering in the QMF domain will be presented. The input signal used in this case has a small amplitude (0.1) and it is a sinusoid of 3000 Hz quantized with 24 bits. The results with and without dithering, will be compared. The dither used is a triangular dither of one quantization step.

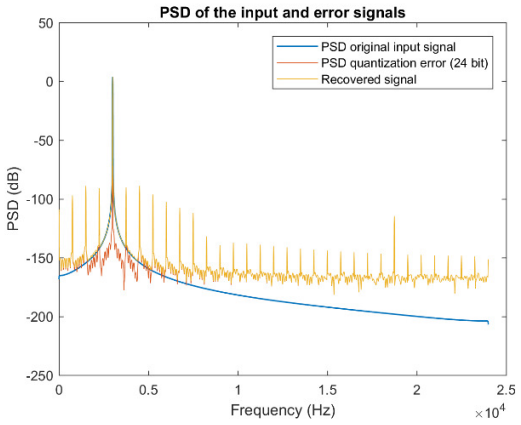

Figure 36. Original, recovered and error signal after quantizing in the QMF domain without dithering

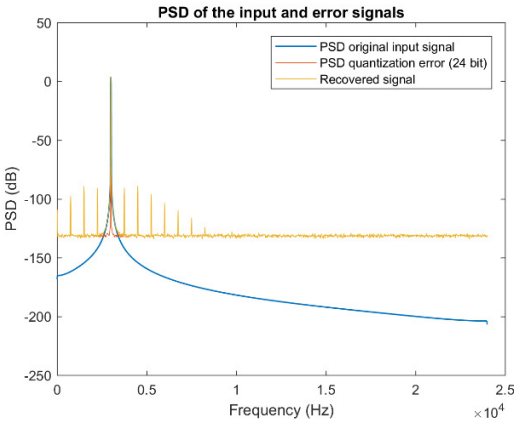

Figure 37. Original, recovered and error signal after quantizing in the QMF domain using triangular dithering

SNR = 75,42 dB

THD = -106,34 dBC

SNR = 75,14 dB

THD = -107,41 dBC

The SNR stays almost equal, because of the amplitude of the harmonic distortion of the recovered signal. When dithering, a lot of the harmonic components disappear under the added noise. The THD decreases when dithering as the error of quantization with dithering has a much flatter spectrum, and less harmonic components. Even though, those values could change depending on the calculation process of THD. In this case just the 5 first harmonics are taken into account for the calculation.

3.3.3.3. *Reference levels, quantization in QMF domain and SNR*

The next figure sums up the results previously presented as it shows the relationship between the quantization bit-depth in the QMF domain and the SNR of the final recovered signal for three different signal types, a sinusoid of 555 Hz, a white noise signal and a music sample.

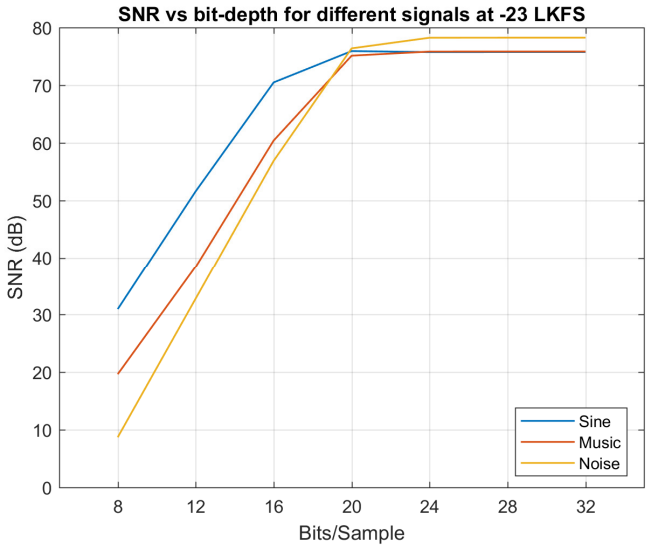

Figure 38. SNR for different bit-depths and signals at a reference level (-23 LKFS/LUFS)

As it can be seen, the SNR increases when the bit-depth is increased, but just until a certain threshold where it reaches the maximum SNR that the transform offers. This happens with all the different signals.

It is also important to mention that the signals used in this plot are loudness normalized signals to match the EBU target level of -23 LUFS as center (C) channels. This means that a music piece that matches the reference loudness level, will have approximately 60 dB SNR when quantized with 16 bit. When the bit depth is increased to 24 bits, the maximum SNR achieved is 75 dB. Any of these cases is not enough to cover all the human perception dynamic range, but it is a reasonable value that could provide good dynamic ranges for most production purposes.

It is also worth mentioning that since the distortion created by the transform is very tonal, and input related, when this kind of signal is processed by a perceptual encoder this kind of distortion will be almost certainly cancelled [15].

3.3.4. True peak level

To study the behavior of the true-peaks of a signal when going through the QMF transform an experiment has been planned. The transform will be fed with a sinusoid signal of a quarter of the sampling frequency, in this case 12000 Hz as 48000 Hz is used as sampling frequency. The signal will only have samples at full scale and at 0, as it will only have 4 samples per cycle. When measuring the true peak of such a signal, a true peak value of 0 dBTP is obtained.

When this signal goes through the analysis filter, a phase shift of 45° in the complex QMF domain, by simply multiplying with the complex value $\frac{\sqrt{2}}{2} + \frac{\sqrt{2}i}{2}$, is applied. Then the signal is synthesized back to the time domain and the true-peak level is again analyzed. The result showed that the true peak level of the signal is maintained, as the recovered signal has high inter-sample peaks, but the real samples of the signal are not at dBFS as in the input signal. The result can be visualized in the next figure:

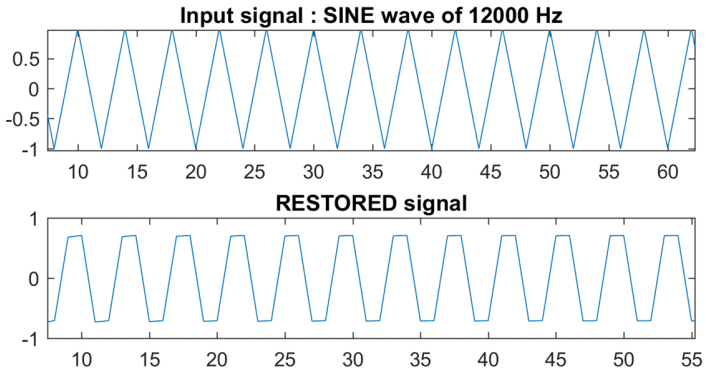

Figure 39. 12000Hz sinusoid and recovered signal after phase shift in the QMF domain

To better prove that the true-peak level is maintained when going through the QMF transform, the inverse of the previous test is performed. Now the transform is fed with a 12000 Hz sinus with a 45° phase shift and normalized to full scale. The true peak level of such signal is at 3 dBTP. In the transform a phase shift in the complex frequency domain is also introduced, and the signal is transformed back to the time domain. The resulting signal is a sinus signal with peaks beyond full scale, and a true peak level of 3 dBTP, so it is also maintained after the transform. The signals can be seen in the next figure:

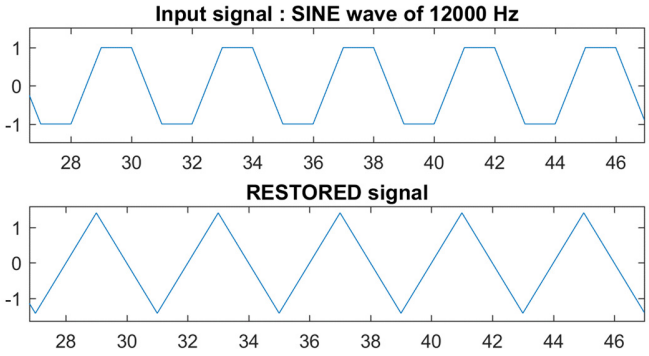

Figure 40. 12000Hz sinusoid with 45° phase shift and recovered signal after phase shift in the QMF domain

4. Real world fixed-point processing blocks

4.1. Fixed point basic operations

In this section different fixed-point operations will be presented, together with their headroom and precision requirements. The operations are:

- Addition
- Multiplication by scalar
- Multiplication of two signals

In this section the signals are assumed to be represented in the time domain as a wave form and in the range between 1 and -1.

Addition:

The addition requires extra headroom available to fit the result of the operation, the maximum amplitude of the two signals added should meet the condition $A_{Max1} + A_{Max2} \leq 1$ (where A_{Maxi} is the maximum amplitude of the i^{th} signal) to ensure that the resulting signal fits in the available range.

There are no extra precision requirements for the addition if the previous condition is met. Even though, the output signal is normally scaled when adding two signals. This process is made by multiplying the signal by a scalar factor, and this does require extra precision.

If the headroom condition is not met, the necessary bits to represent the resulting signal are $B_c = \max(B_a + B_b) + 1$ where B_a and B_b are the number of bits to represent the original signals [16].

Multiplication by a scalar:

The multiplication operation does not require extra headroom when the scalar factor or gain factor (G_f) is smaller than 1. If the gain factor is greater than 1, the signal should meet the condition $|A_{Max}| \cdot G_f \leq 1$ (where A_{Max} is the maximum amplitude of the signal) to ensure that the resulting signal fits in the available range.

Multiplying by a scalar requires more precision in order to represent the content with the same exactitude as in the original signal, as the resulting amplitudes of the samples may not be a multiple of the quantization step. If so, a maximum error of half a quantization step (or 3dB) will be introduced. If the gain factor is smaller than 1, the smallest values of the original signal would be then smaller than a quantization step, so their information would be lost. The number of bits required to represent the exact result of the operation is equal to the sum of the bits used to represent the original signal and factor [16].

Multiplication of two signals:

The multiplication of two signals is often used in many processes such as windowing, for example. This operation does not require extra headroom when both signals are expressed in the same range of values.

As in the previous case, multiplying two signals requires extra precision. To be exact, the number of bits required in the resulting signal must be equal to the sum of the bits used to represent the original signals [16]. If not, a maximum error of a half quantization step (or 3 dB) will be introduced to the signal and the SNR will be therefore decreased, and the dynamic range will be limited to the maximum representable range with the number of bits used to represent the result.

- Square of a signal:

As a special case for the multiplication, the square of a signal is now also presented. It is often used to calculate the energy. The operation requirements are very similar to the previously explained, as it is mainly a multiplication of two signals, so there are no headroom requirements, and the resulting signal requires as many bits as the sum of the bits used to represent the original signal.

In this case, though, if the range that is being represented includes negative numbers, the result of the operation will return always a positive number, leaving one free bit as the representation of the sign is no longer needed. This bit could be used to increase the SNR of the square of the signal, but this means that the binary number type must be changed, and this could cause interpretation problems.

4.2. Quantization error propagation theory

For some fixed-point operations there is a precision loss in the process due to the quantization with limited precision. This error can be seen as an added noise to the signal. This noise can be modeled depending on the quantizer used as a uniformly distributed white noise with different means and variances depending on the quantization step *q*, the number of bits eliminated during the quantization *k* and the type of quantization mode: truncation, conventional rounding or convergent rounding [17]. The white noise mean and variance are modeled as:

Quantization mode:	**Truncation**	**Conventional rounding**	**Convergent rounding**
Mean	$\frac{q}{2}(1-2^{-k})$	$\frac{q}{2}(2^{-k})$	0
Variance	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1-2^{-2k+1})$

Table 9. Mean and variance of quantization error for different types of quantization [17]

Depending on the operation done, this added noise will propagate differently. If two scalar noises are defined, b_x and b_y, which are associated to two inputs (X and Y) of the operator, the operator will generate an output noise b_z defined as:

$$b_z = \alpha_1 b_x + \alpha_2 b_y$$

Where $\alpha_1$and α_2 are defined depending on the operation type as:

Operation	**Value of α_1**	**Value of α_2**
$z = x \pm y$	1	±1
$z = x \cdot y$	y	x
$z = \frac{x}{y}$	$\frac{1}{y}$	$-\frac{x}{y^2}$

Table 10. Propagation of quantization error for different operations [17]

It is easy to understand the previous formula just by studding the multiplication of two signals. If we multiply two quantized signals (X, Y), which are composed by the signals (x, y), plus the quantization noise (b_x, b_y), we can model the multiplication as:

$$X \cdot Y = (x + b_x) \cdot (y + b_y) = xy + b_x y + b_y x + b_x b_y$$

If we divide the result between signal and noise parts, we can identify as the resulting noise (b_z):

$$b_z = y b_x + x b_y + b_x b_y$$

As the term $b_x b_y$ is much smaller than the others, it can be neglected, and the result is the same as in the previous formula:

$$b_z = y b_x + x b_y$$

To prove the previous theory, the next experiment is done:

Two white noise signals are generated and quantized with 16 bits and re-quantized with 8 bits, losing 8 bits precision, and therefore, generating a quantization error that will propagate through the operations. The quantization uses conventional rounding, so the expected mean and variance values can be calculated. Also, the real mean and variance obtain in the experiment can be calculated and compared to the expected value:

	Mean	**Variance**
Signal 1	$1.5819 \cdot 10^{-5}$	$5.1463 \cdot 10^{-5}$
Signal 2	$1.5128 \cdot 10^{-5}$	$5.1458 \cdot 10^{-5}$
Expected values	$1.5259 \cdot 10^{-5}$	$5.0862 \cdot 10^{-5}$

Table 11. Mean and variance of the quantization errors of two noise signals and their expected values

To prove the error propagation through the operations, the two quantized signals will be added, and the result will be compared with the addition of the double precision signals. By doing this, the error of the signal can be extracted and should match the addition of the first two quantization errors. The same process has been made for the product of those signals.

The errors obtained are compared to the expected error according to the error propagation theory. The results can be seen in the next table:

		Error propagation theory	**Actual error of the signals**
Addition	**Mean**	$3.1034 \cdot 10^{-5}$	$3.1034 \cdot 10^{-5}$
	Variance	$1.0290 \cdot 10^{-5}$	$1.0291 \cdot 10^{-5}$
Multiplication	**Mean**	$3.4034 \cdot 10^{-8}$	$3.4216 \cdot 10^{-8}$
	Variance	$3.4303 \cdot 10^{-6}$	$3.4302 \cdot 10^{-6}$

Table 12. Mean and variance values of the propagated error after addition and multiplication and the theoretical values

As it can be seen, the results match the predictions made by the propagation model.

When doing the square of a signal this may change a little, as the errors and signals that are being multiplied are the same, so the formula would be:

$$X \cdot X = (x + b_x) \cdot (x + b_x) = x^2 + 2 b_x x + b_x^2$$

As the last term will be much smaller than the other, the error can be approximated as:

$$b_z = 2b_x x$$

In the next two plots the propagation error of the square of a signal that has been quantized with 16 bits can be seen. The maximal error of such a signal would be $\frac{\Delta}{2} = 1.5 \cdot 10^{-5}$, but when squared, the maximal error will be twice that value. Then the theoretical propagation with the actual propagation of the square of a real signal will be compared:

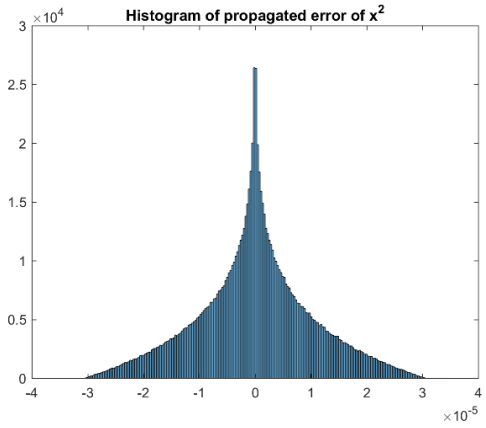

Figure 41. Histogram of the propagated error when squaring the signal

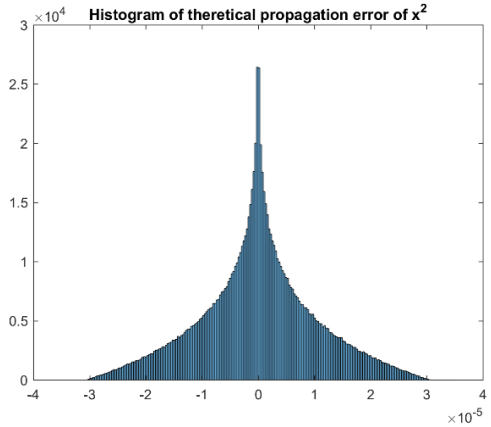

Figure 42. Histogram of the propagated error according to the propagation theory

The previous figures show the histogram of the errors that approximate the PDF of the noise resulting from the multiplication of two uniformly distributed noises. This distribution is known as the multiplication distribution, and it also confirms the noise propagation theory through basic operations.

4.3. Mean square

With the theory introduced in the previous sections, a practical fixed-point and more complex example, such the mean square calculation, is here introduced. The mean square operation is very often used to calculate the signals energy. It is based in a multiplication of a signal by itself, an addition of all the squared numbers and a division by its length, obtaining the mean of the squared of a signal. The error propagation of such a signal is a little bit more complex than the previous examples, but conclusions can be drawn by looking at the probability distribution of the resulting error at each stage.

Square:

As it has already seen, the quantization noise of a signal can be approximated as a uniformly distributed noise with a mean value near to 0. When multiplying such noise by itself, in theory, the resulting noise will have a multiplication distribution with mean value also near 0 and a maximum value of two times the quantization step used for the input signals. But the fixed-point multiplication ads more noise to the result. This noise has a maximum value of half a quantization step, and it will be added to the previous one.

Mean:

When adding the squared signals, the previous error will be accumulated for every addition. If the addition is performed with an accumulator there is no need to scale the signal down first, and no extra error will be accumulated, as the accumulator provides enough headroom to contain the result. When performing the last division to calculate the mean, an error will be introduced with a maximum value of half a quantization step

If the addition is performed without accumulator, a scaling factor should be applied and will introduce more noise to the calculation. Since a division by the length of the signal is needed to calculate its mean, the division will be performed before the addition and will act also as a scaling factor for the addition. The error of the fixed-point division will be accumulated when adding all the values of the signal, but the addition itself will not introduce any extra error.

Now two block diagrams of both processes are presented, one with accumulator in the addition process and the other without it:

- With accumulator:

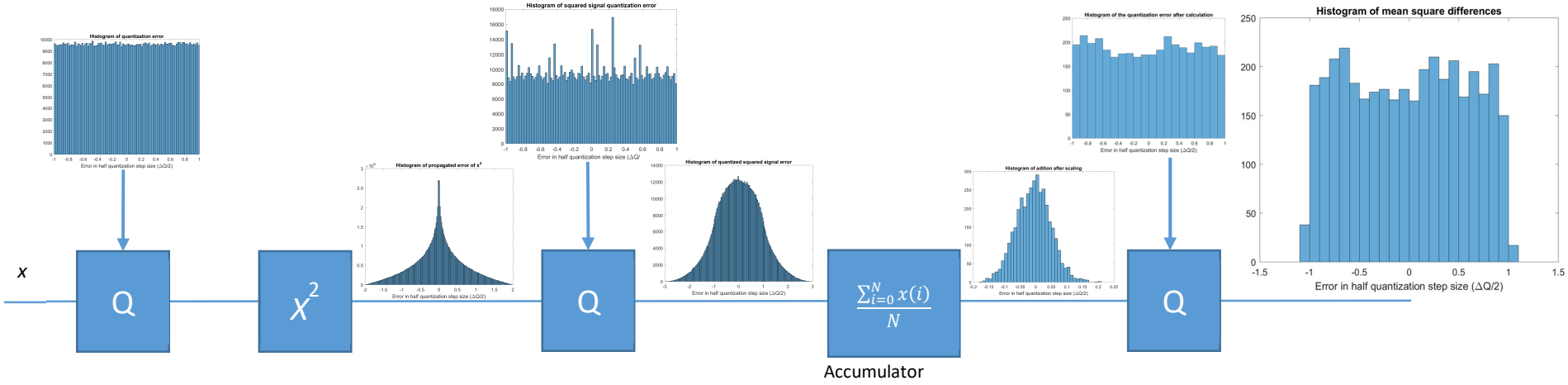

Figure 43. Distributions of the error through the process of mean square calculation with accumulator

- Without accumulator:

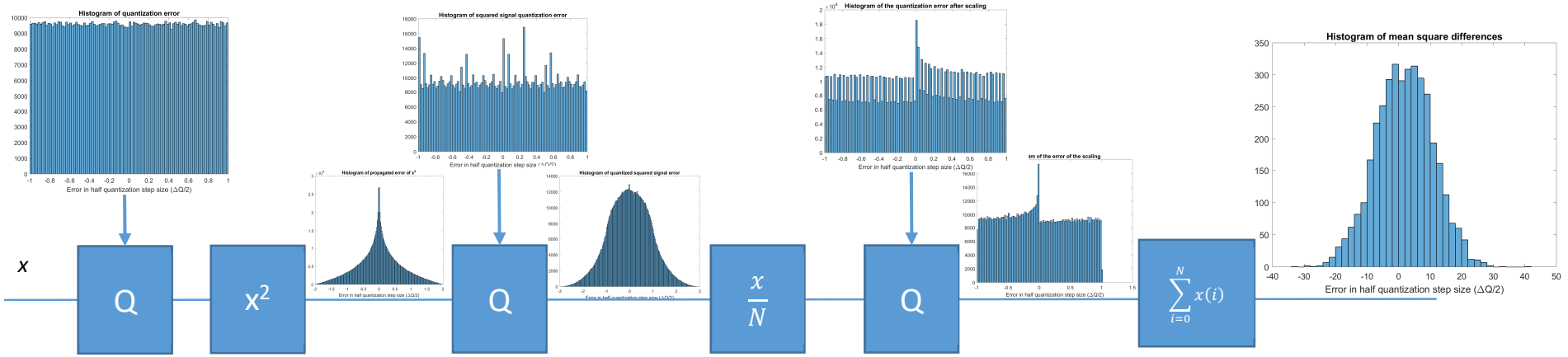

Figure 44. Distributions of the error through the process of mean square calculation without accumulator

It can be seen that the results are very different for both cases. In the first one, the error of the scaling is not accumulated when adding all the values of the signal, and therefore the maximum error is still half of a quantization step. In the second case, though, the error of the scaling is accumulated, and therefore, the resulting error is bigger, and it will depend on the size of the block in which the mean square is being calculated.

It is also worth mentioning, that the distribution of the error signals changes depending on the case. When the calculation without accumulator is done, all the uniformly distributed errors introduced by the fixed-point scaling are added, and therefore, a sum distributed random signal is obtained. When using an accumulator, the resulting error still has a uniform distribution, as the accumulated error is scaled down, and a new bigger and uniform error is added at the last stage.

RMS error for 16 bit quantization	**Block size = 128**	**Block size = 640**	**Block size = 4096**
Accumulator	8.8026e-06	8.8567e-06	8.8008e-06
No accumulator	9.8292e-05	2.6154e-04	0.0024

Table 13. RMS error for 16 bit quantization values for the calculation of the mean square value for different block sizes

To obtain the previous values, the mean square of a white noise signal has been calculated with both methods. It can be seen that without accumulator the error grows together with the block size of the calculation, as the error introduced by the scaling will also be added. In the case of the accumulator, the resulting error is independent of the block size as no error is accumulated in the addition.

4.4. Fixed point implementation of the transforms

In this section, the results of testing the real fixed-point implementations of the MDCT and QMF transforms used at Dolby will be presented. The results are highly implementation dependend as differently optimized implementations for different processors lead to different results [18]. Therefore some results presented in the next sections could not be explained with detail because of the complexity of the processes involved in the transforms combined with the time constrains faced during the project.

This part will focus on the error introduced by the transforms, it source, and the distribution of the error throughout the bands. In section *5* more will be explained about the maximum reachable values with real world signals, and therefore, more information about the headroom requirements of the transforms can be obtained.

4.4.1. MDCT

To be able to test the precision of the MDCT transform, it has been isolated from its inverse (IMDCT) and a spectral analysis has been done to study the error distribution throughout the frequency bands.

The error has been obtained by subtracting the result obtained using the 64 bits floating-point implementation and the fixed-point 16 bit. This procedure has been made for different signals and different window lengths (4096 and 256).

Long signals had been given as input, and therefore the output is a series of transforms, in our case, of length 2048 or 128 samples. The results shown are the average of all the samples that correspond to the same band of the MDCT transform. The error is shown, not as an average, but as a histogram per band, where the more likely errors per band are displayed with lighter colors. This way the maximal and minimal errors can be taken into account, but also the more likely cases are shown.

In the next images, examples of some results can be seen. They have been obtained by using a generic fixed-point build configuration, implemented with standard C operators, that is able to run on different processors (generic_risc16x32), where 32 bit word length is used for the data, and 16 bit word length is used for the coefficients. The result has therefore a SNR around 90 dB, as the coefficients set the maximum precision of the calculations.

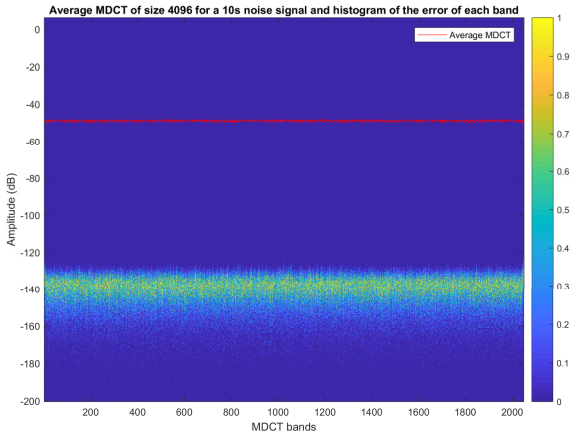

Figure 45. Average MDCT of size 4096 for a 10s noise signal and histogram of the error for 16 bit coefficient precision of each band

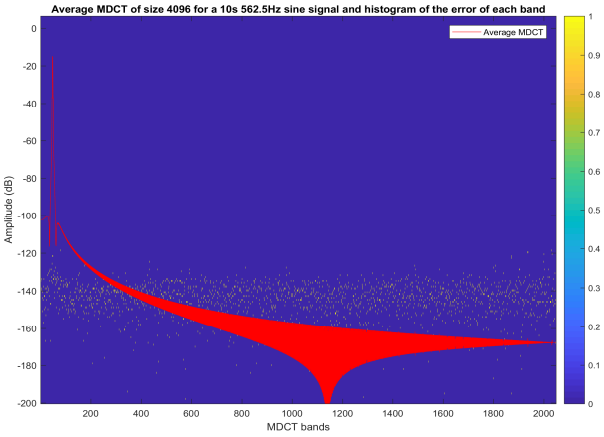

Figure 46. Average MDCT of size 4096 for a 10s 562.5Hs sine signal and histogram of the error for 16 bit coefficient precision of each band

It can be seen in the histograms of the error that for the noise signal, the error can have different values, but in the case of the 562.5 Hz sinusoid, the error is always the same, as the frequency of the sinus is perfectly centered in a MDCT band.

To see how the calculation process affects the results a similar configuration to the previously used has been modified (from model_risc16x32 converted to model_risc32x32) so that 32 bits are used for both, the data and the coefficients. In this case the precision of the result will be determined by the number of operations that produce a loss of precision. Therefore, the SNR of the result should be predictable if the number of operations is exactly known.

This particular implementation of the MDCT is fairly complex, and because of time constrains a deeper analysis of the process could not be reached. Even though, it is known that the core of the transform is a radix4 complex Fast Fourier Transform[8].

Knowing that, the shape of the resulting error can be better understood, as the main part of the error is coming from the operations involved in the FFT. In the case of a 256 MDCT window length, a radix 4 scaled 64 FFT is used and in the case of a 4096 MDCT window length, a radix4 scales 1024 FFT is used. The results obtained for a white noise signal are the following:

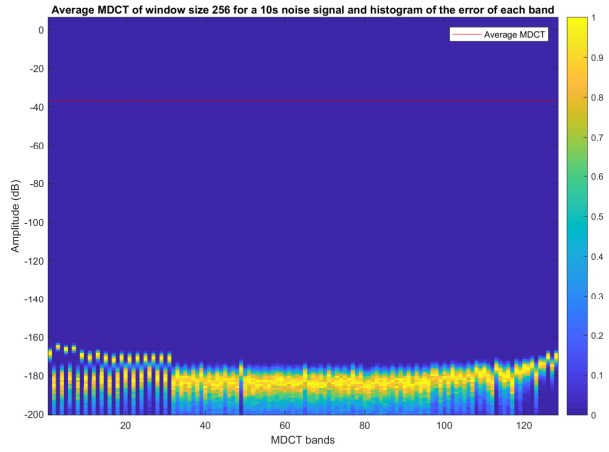

Figure 47. Average MDCT of size 256 for a 10s noise signal and histogram of the error for 32 bit coefficient precision of each band

SNR = 141.4 dB

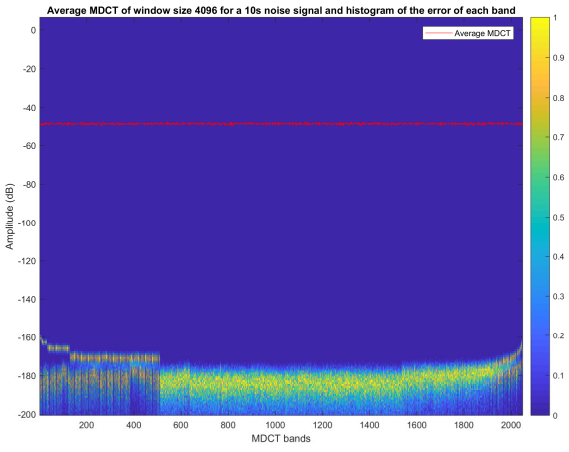

Figure 48. Average MDCT of size 4096 for a 10s noise signal and histogram of the error for 32 bit coefficient precision of each band

SNR = 128.6 dB

As it can be seen, the lower bands show an unusual behavior as some bands have more error than others. This may be an effect caused by the order of the samples in the complex FFT implementation, as not every sample follows the same path of the radix4 algorithm, and different twiddling factors are used [19] [20]. Therefore, different errors may be obtained in different regions of the transform. The same error shape is obtained in the QMF transform, and some tests have been made to test the source of this error (*4.4.2. QMF*).

This phenomenon could not be explained with detail, but it shows the impact of the FFT in this MDCT implementation. Therefore, to better understand the source and amount of error introduced by the calculations, a better understanding of the FFT algorithm used is necessary.

Another important fact is the amplitude difference between both transforms. The shorter transform has a bigger amplitude as the longer transform, as the energy of the signal is divided in less bands. This effect is more visible when the signals energy is spread through all the bands.

[8] To exactly know how the MDCT can be implemented by using an FFT, the mathematical approach can be checked at *Efficient Implementation of the Complex Modulated Filter Bank* by *Per Ekstrand* [14]. The chapter of interest is annexed in Annex 3: *Fast DCT type IV transform* by Per Ekstrand (Coding Technologies).

To see the impact of the bits used to represent the coefficients of the transform, in the next images the evolution of the SNR depending on the coefficient bits and the transform length can be seen. The plot has been obtained by performing the transform with the same input signal but changing the number of bits of the transform coefficients.

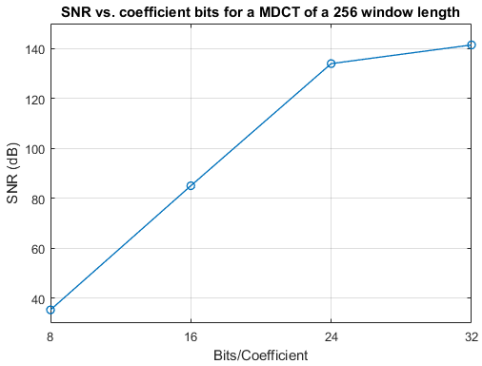

Figure 49. SNR vs. coefficient bits for a MDCT of a 256 window length

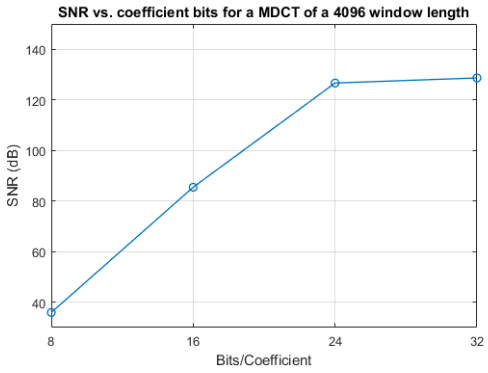

Figure 50. SNR vs. coefficient bits for a MDCT of a 4096 window length

4.4.2. QMF

To test the precision of the QMF transform, almost the same procedure as in the MDCT test has been followed. The figures below represent the same as in the previous section, an average of several transforms, and a histogram for every band of the QMF.

First, the 64 bit transform has been performed, in order to get the results with the maximum precision. Then, the results with maximum precision will be compared to the results of other configurations with less precision to extract their error. As the QMF used is in fact a CQMF the values represented here are the absolute values of the results.

As the QMF has less bands than the MDCT, statistical box-plots can be presented. In these plots, the central red line represents the median, and the edges of the box the 25^{th} and 75^{th} percentiles. The lines extend to the last samples that are not considered outliers, and the outliers are represented with a red cross. Finally, the continuous red line represents the mean of each band and the blue line the average of all the QMF transforms obtained through the duration of the signal.

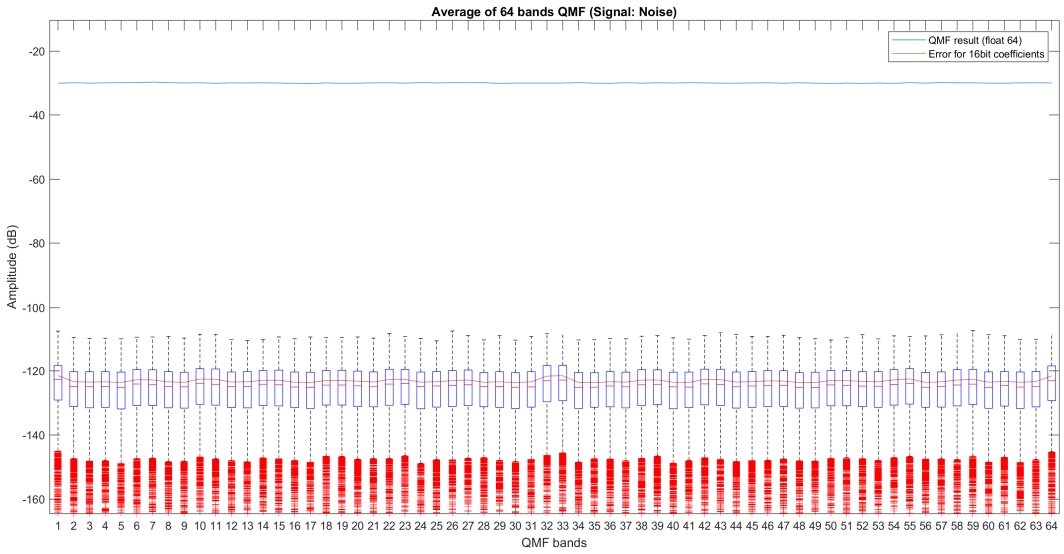

Figure 51. Average QMF result and distribution of the error of each band with 16 bit coefficient precision for a noise signal

In the previous image the result obtained with 16 bit coefficient can be seen. As seen in other cases, the SNR is around 90 dB as the coefficients set the maximum precision of the result. An important fact to remark is the amplitude difference between the QMF and the MDCT transform. The QMF show an average result around 10 dB higher than the MDCT. In section *5,* more comparisons between the results of the MDCT and QMF will be made and commented.

In the next figure, the plot presented is the result obtained for music.

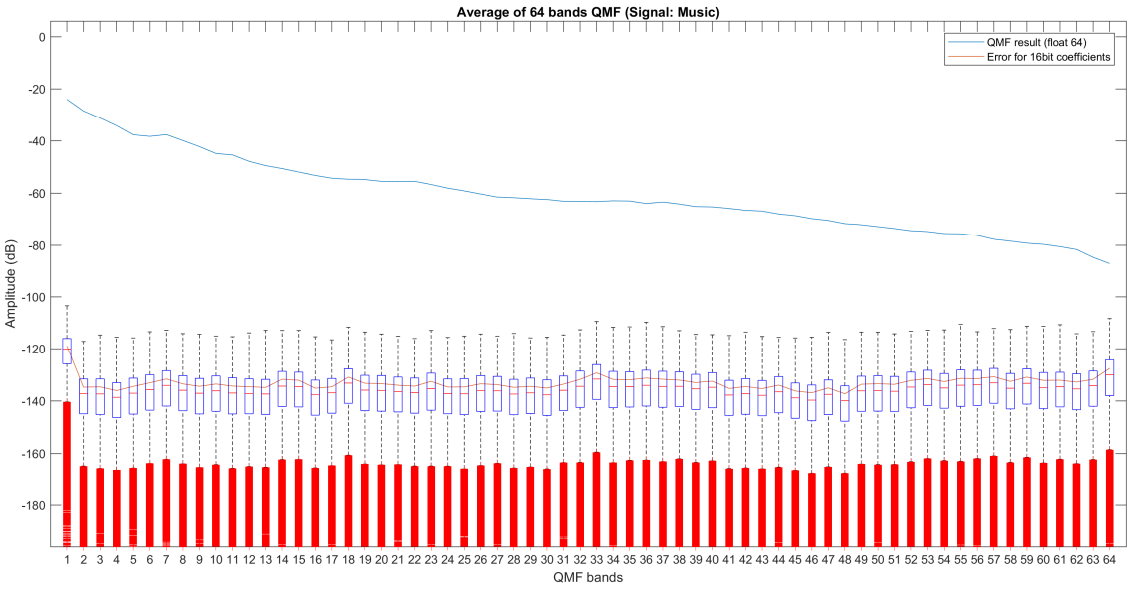

Figure 52. Average QMF result and distribution of the error of each band with 16 bit coefficient precision for music

As it can be seen, the distribution of the error in every band is very similar, as they are broad band signals with frequency components throughout all the spectrum. But in the case of music, the lower band

presents higher amplitude error than the rest, probably caused by the fact that low frequencies are very present in music.

This distribution of error is different when the signal has its energy in just few bands. In the next case, the frequency chosen is perfectly in the center of a filter response. This causes that the errors obtained are always the same and therefore the distribution of the error of every band is very narrow.

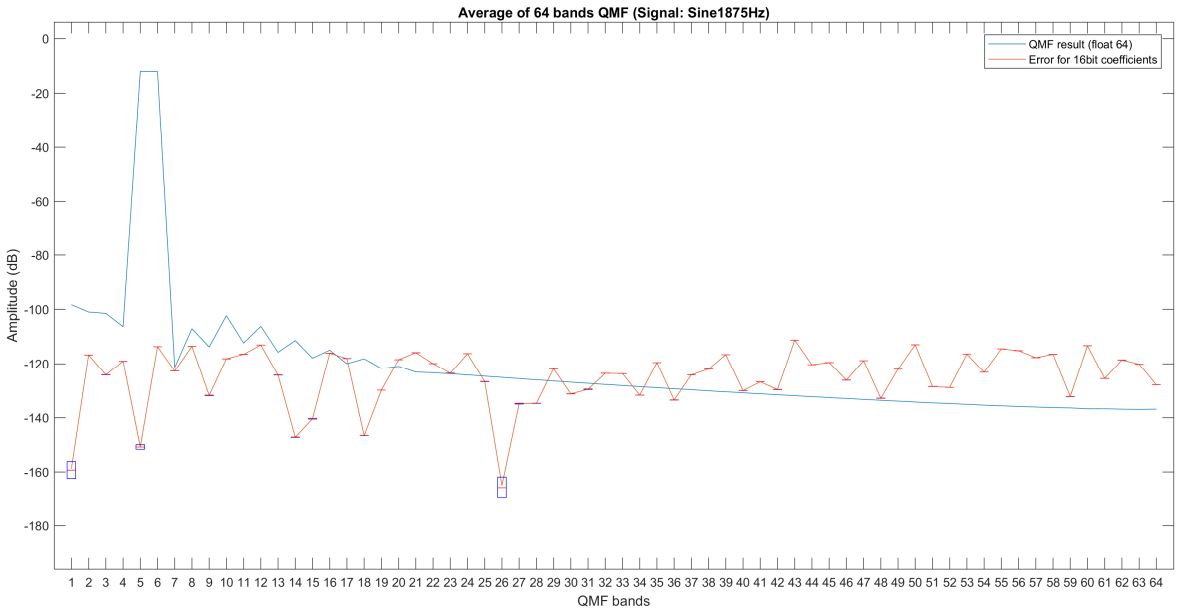

Figure 53. Average QMF result and distribution of the error of each band with 16 bit coefficient precision for an 1875 Hz sine

The following result has been obtained, as in the previous section (*4.4.1. MDCT*), by modifying the configuration used so that data and coefficients have the same precision: 32 bits. The error in the next figures is again presented as histograms, in order to be comparable with the results from the MDCT.

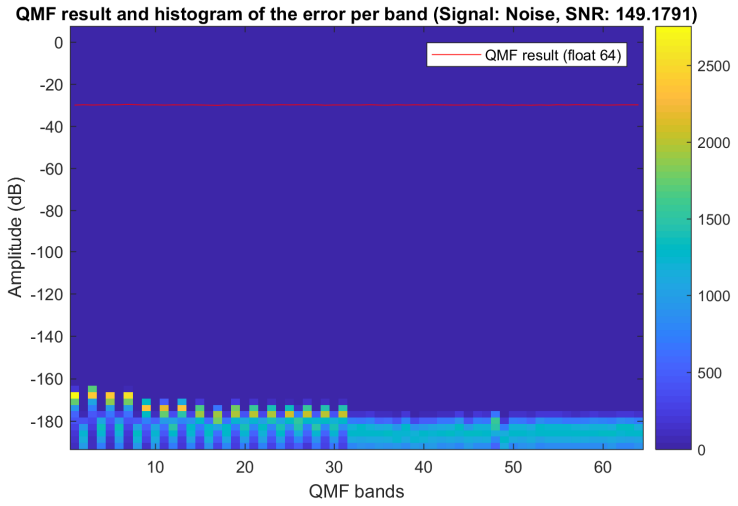

Figure 54. Average QMF result and histogram of the error of each band with 32 bit coefficient precision

Here, the same phenomenon as in the error obtained in the MDCT can be seen. The QMF implementation has a non-uniform error distribution, where the odd bands of the lowest half of the transform have a greater error than the others. To prove that this error is coming from the FFT, different tests have been made:

First, the effects when changing the transform length had been checked. A 32 band QMF has been used, and the result was similar to the 64 band QMF, as the odd bands of the lower half of the transform had more error than the rest.

To discard overflows in the process, a signal with 2 bits (12 dB) of headroom has been used. The result has been the same as in the previous case.

Once the overflows have been discarded, the possibility that this effect may come from the type of filter used in the transform has been also checked, and the result with different filters has always been very similar.

The last theory is that the effect could be caused by the twiddling factors of the FFT used. When changing the twiddling factors, a change can be seen in the distribution of the error histograms:

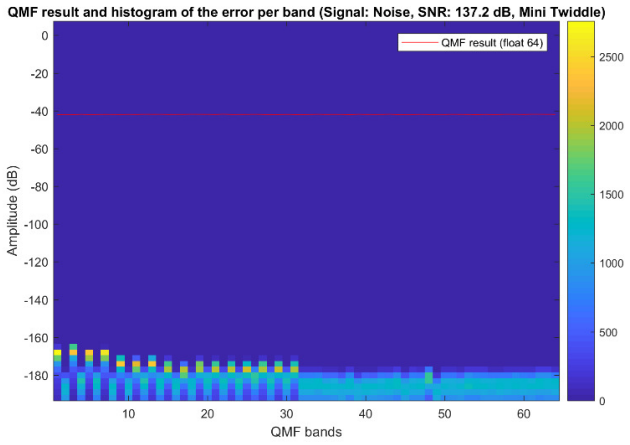

Figure 55. Average QMF result and histogram of the error of each band with 16 bit coefficient precision with mini twiddle coefficients

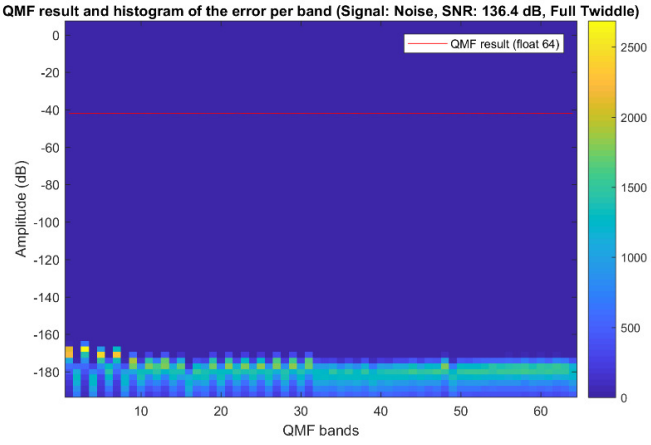

Figure 56. Average QMF result and histogram of the error of each band with 16 bit coefficient precision with full twiddle coefficients

The difference between the figures is subtle, but the overall error is higher in the right figure, because the full twiddle factors are used. The distributions of the histograms of the lower bands have changed, proving that this effect is caused by the operations in the FFT.

To see the importance of the precision of the coefficients used in the transform, several tests have been performed by changing the data bit-depth (Lfract bits) and the coefficient bit depth (Sfract bits) used by the implementation to obtain the SNR of all the possible combinations of Lfract and Sfract bits. This will give a good overview of what bits act as a bottle neck and why. The results obtained can be seen in the next figure:

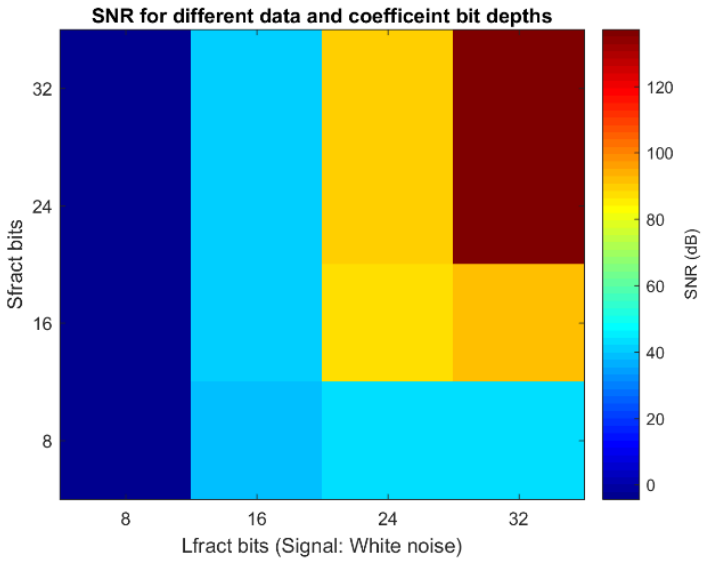

Figure 57. SNR for different combinations of data and coefficient bit depths

This matrix has been obtained using a white noise signal at -12 dBFS to ensure no overflows. As it can be seen in the figure above, the Lfract bits are the most critical as the result of the operations will be represented with this number of bits, and therefore, they limit the SNR and the dynamic range of the result. Even though, if the coefficients are represented with less bits than the data, the result of the operations will also be inaccurate because of the error introduced by the coefficients.

5. Real world signal analysis

When designing some complex processes, it is sometimes difficult to stablish sufficient safety measures such as headroom or precision requirements of the processes. Those predictions can be based in rule of thumb thoughts about the theoretical functioning of the processes involved, but, if doing so, errors along the processes are still probable. Also, they can be based in worst case scenarios, but in very complex processes, these measures would over-dimension the requirements, as worst-case scenarios are usually unrealistic and improbable in the real world.

The purpose of this section is to give some real-world signal statistics in different data domains and through different processes involved in the coding and decoding of different Dolby components such ac-4 or the delivery of Dolby Atmos content. To do so, an analysis tool has been programmed, together with some help of some Dolby engineers to be sure all the different components used work properly together.

After the tool has been successfully programmed and functioning, it is able to read any audio delivery configuration such as: mono, stereo, multichannel and object-based immersive audio files. It analyzes every channel or object given as input and, depending on the input configurations, it performs down-mixes or renderings to smaller channel configurations and it analyzes the down-mixed content again. By doing this, the behavior of the processes involved can be tracked and, as the signals used are real content, the requirements of this processes could be analyzed and adjusted if necessary. In the next figure a block diagram of the tool can be seen.

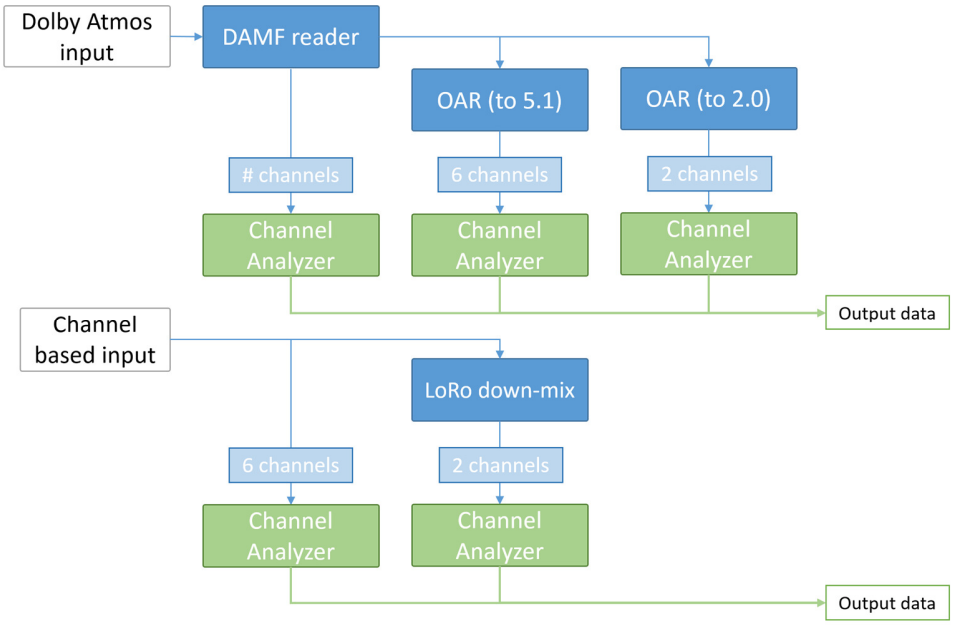

Figure 58. Block diagram of the multichannel analysis tool

As it can be seen, the tool reads different input configurations and, in the case of Atmos content, it analyzes every object and channel of the original content. Then, it performs a rendering to a 5.1 and 2.0 channel configuration, using the Dolby Object Audio Renderer (OAR) as specified in *ETSI TS 103 448 V1.1.1 (2016-09)* [21], and analyzes every channel of the renderings.

If the content is channel based, a LoRo down-is performed and the tool analyzes the two final channels. The LoRo down-mixing process is made using the next coefficients:

LoRo	L	R	C	LFE	Ls	Rs
L	1	0	$\sqrt{2}$	0	$\sqrt{2}$	0
R	0	1	$\sqrt{2}$	0	0	$\sqrt{2}$

Table 14. Coefficients for the LoRo down-mix

The down-mix performed with those coefficients is sometimes also called “power preserving, which means that the result of the down-mix is not normalized to the maximum possible value.

The channel analyzer, analyses every input channel the same way. It reads two different block sizes, 2048 samples long, and 3 seconds (or 144000 samples long) blocks[9]. For each block the mean square value, the maximum peak value, and the maximum true peak value are calculated. Also, two domain transforms are also performed, the MDCT and the QMF. For the MDCT, two blocks of the input signal are used so the output of the transform is 2048 complex samples long, and the QMF is performed in sub-blocks of 64 samples long. The QMF is used to calculate the energy of the signal in each band.

The results of each calculation are sent to a histogram object that creates a histogram for each of the parameters, and in the case of the transforms it performs a histogram for each line of the transforms (2048 for MDCT and 64 for QMF). With these histograms, statistical parameters such as percentiles can very easily calculated. In this case, all the plotting and statistical calculations have been obtained with MATLAB. In the next figure, a block diagram of the channel analyzer can be seen.

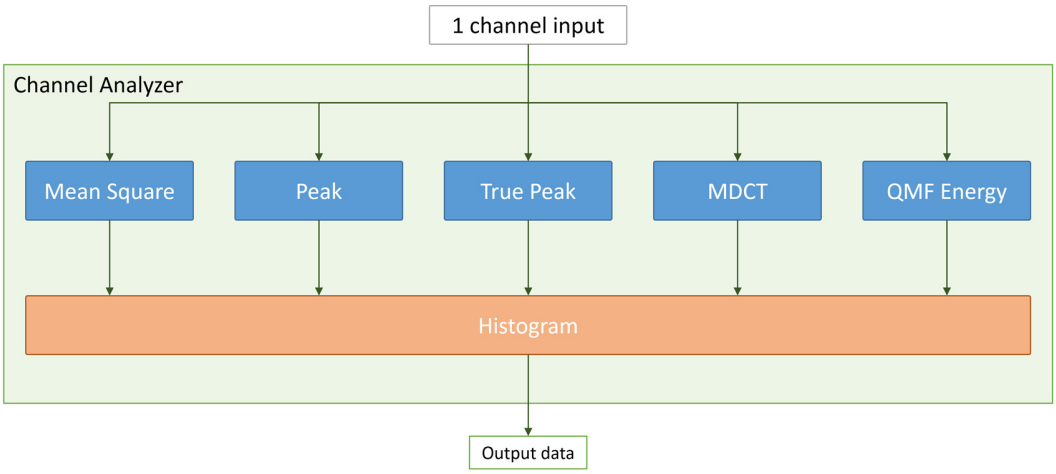

Figure 59. Block diagram of the channel analyzer

The content that has been analyzed can be basically grouped in three groups: Original stereophonic music content from CDs, original multichannel cinematic content from DVDs, and original object-based immersive content from Dolby Atmos Master Files. In the next sections the results of such analysis will be presented.

[9] The results for the long blocks have been neglected as they were redundant.

5.1. Stereophonic music content

In this category of content, four sub-groups have been differentiated. The sub-groups categories are based on their music genre. The sub-groups are: Classica, Jazz, Rock and Pop.

It is difficult to make such a rough classification with music, as it is sometimes too diverse, and categories can always be sub divided. But these four categories can good represent the technical differences between music types.

To have enough data, a relatively big sample for each category has been analyzed[10]. The content used, has always been original content directly obtained from original CDs to make sure that no "lossy" encoding processing has been previously made to the signals. The parameters analyzed are the ones explained in the previous section: mean square values of 2048 samples long blocks, peak values, true peak values, amplitude of MDCT bands and energy of QMF bands.

Now a comparison between genres will be presented:

These four genres have very different dynamic behavior, and this can be seen in the results obtained. In the next table, the maximum values, and the 95th percentile of each parameter is presented, as these two values combined give a good idea of the distribution of the histogram obtained after the analysis.

When looking only at the maximum values, it could be thought that classical music uses as much compression as rock. But when looking at the difference between the maximum value and the 95th percentile, it can be easily seen, not only that classical music is much more dynamic than rock, but also that rock hits the full-scale value very often, as the maximum value and the 95th percentile are very close or are even the same value.

Music content		**Mean square**	**Peak**	**True Peak**	**MDCT**	**QMF**
Classical	**95th percentile**	-19 dBFS	-9.5 dBFS	-9.5 dBFS	-35 dBFS	-29 dBFS
	Highest value[11]	-4 dBFS	0 dBFS	1.5 dBFS	-1 dBFS	-7.5 dBFS
Jazz	**95th percentile**	-12.5 dBFS	-2.5 dBFS	-2 dBFS	-21 dBFS	-20 dBFS
	Highest value	-3.5 dBFS	0 dBFS	1.5 dBFS	-1.5 dBFS	-6.5 dBFS
Rock	**95th percentile**	-7 dBFS	0 dBFS	0 dBFS	-15.5 dBFS	-13.5 dBFS
	Highest value	-2 dBFS	0 dBFS	2.5 dBFS	0.5 dBFS	-6.5 dBFS
Pop	**95th percentile**	-7.5 dBFS	0 dBFS	0 dBFS	-13 dBFS	-13.5 dBFS
	Highest value	-2 dBFS	0 dBFS	2.5 dBFS	1 dBFS	-6.5 dBFS

Table 15. Highest and 95th percentile values for mean square, peak, true peak, MDCT and QMF for all genres

With this sum up table, it can already be seen that such high levels, as the ones achieved by rock or pop, can be problematic [22] as some values beyond full scale are appearing in the transforms. These values, if being processed with fixed point processors, would have been clipped or wrapped around, causing errors when synthetizing the signal.

[10] For a complete list of the music content that has been analyzed, see: *Annex 1: List of content used for analysis of stereophonic music.*

[11] The highest values shown in the entire document are normally outliers of the histogram distribution, and therefore, they do not represent the behavior of the average. Nevertheless, they are important to determine the possibility, even if remote, that some samples may have values beyond the full-scale range.

This is a concerning fact, as the signals here analyzed are only original content, and no down mixing, dynamic processing or any other process that could concentrate the energy in one channel has been performed.

In the next figures, a comparison between the distributions obtained for each genre can be seen. These figures, show the normalized histograms of the mean square, peak and true peak values for the different genres. The histograms just show the relative amount of times that a value has appeared in a music category, so it shows the average nature of the signals for each genre.

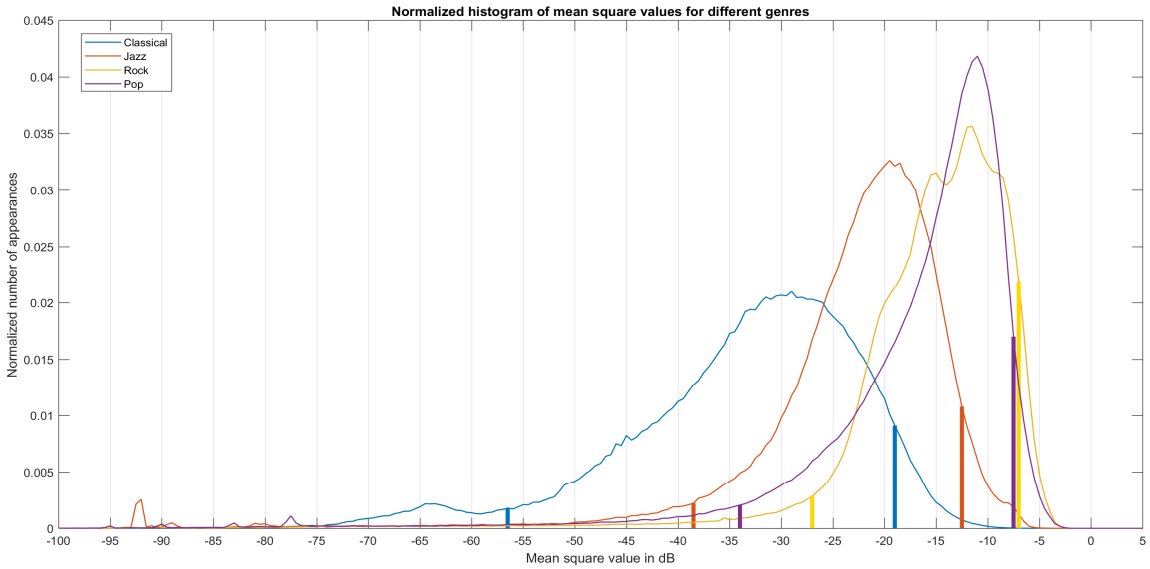

Figure 60. Normalized histograms of mean square values and their 5th and 95th percentiles for different genres

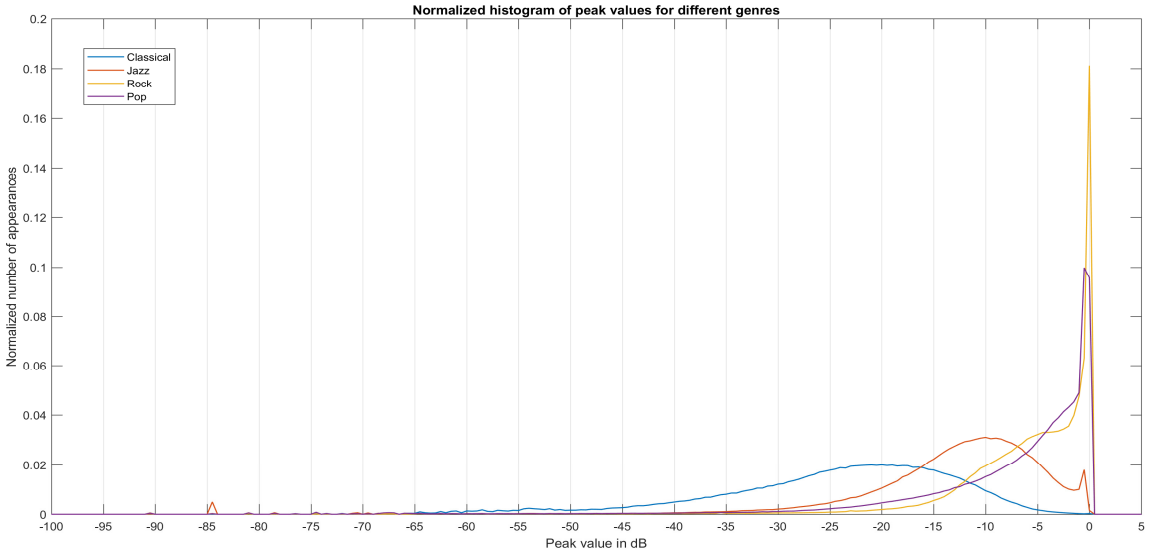

Figure 61. Normalized histogram of peak values for different genres

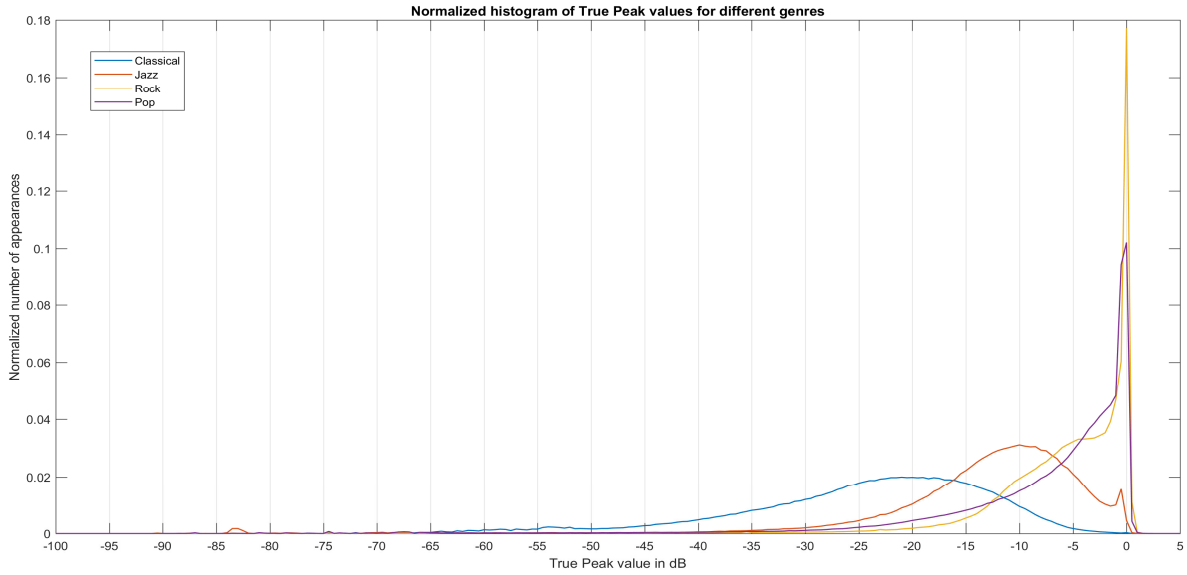

Figure 62. Normalized histogram of true peak values for different genres

It is interesting to see the different distributions between genres. Classical music is the most dynamic and it has almost a normal distribution shape in all three cases.

The mean square value, gives an idea of the amount of energy that the signal has in time. It can be seen, that pop or rock are genres where the signal has a lot of energy in an extended amount of time, as their distributions are narrow and have high mean values.

By looking at the peak and true peak distributions, it can be seen in which genres more amount of limiting is used, as the amount of values at full scale is much higher in pop and rock than in jazz or classical. Also, the kind of sounds involved in every music genre can give some information, as in the classical music there are almost no values at 0 dBFS, but in jazz 0 dBFS values start to appear. This could be caused by the presence of percussion instruments, such as drums, in jazz music. Those instruments, when recorded, normally generate high amplitude signals and can hit full scale easily.

5.1.1. Transforms

As mentioned before, the transforms used are the MDCT and the QMF. In these sections the results of the transforms for each genre will be presented. In *Figure 63* to *Figure 70*, the results can be seen. The figures represent a color code histogram for each band of the transform, where the values that appeared more often in each band are colored with lighter colors. The color code is normalized through all the bands, so there can be some color changes between figures, as a specific histogram bin of a band may have higher values than in other figures. Also, the lines for the 5^{th} and 95^{th} percentile in red, and the maximum and minimum values in green can be seen.

5.1.1.1. MDCT

By looking at the Table 15, together with the figures below, some interesting observations can be made, and differences between genres can be detected.

First of all, it is clear that the results for pop and rock music are similar as they present high values and their spectral content is also similar. Also, both genres present a maximum value in the MDCT domain that is beyond the full-scale range. Also, the dynamics presented in the MDCT domain are similar as the values of the mean 5th and 95th percentile respectively are -97 and -51.3 for rock and -104.7 and -52 for pop music.

Pop and rock are different than the results for classical music, as this genre presents much lower values. The percentiles of classical music are lower than any other genre presented here, with a mean value for the 5th and 95th percentiles of -125.9 dB and -77.9 dB respectively. It is interesting to note that the harmonic content of classical music can be observed in the form of the MDCT. Another interesting fact is that the use of dither and noise shaping techniques can be seen in the MDCT transform for classical music, where the high frequency bands show a higher level than it would be expected.

The results for jazz are in between the previous two cases, where higher values than classical music are achieved, but not as high as pop or rock. The mean percentile values obtained are -105.3 dB and -60.1 dB for the 5th and 95th percentiles respectively.

The common facts for all genres are that their general spectral shape is similar, as the lower bands present always the maximum amplitude, and it decays as the frequency increases. Also, it is interesting to observe the dynamics in the MDCT domain, because, even though in the PCM domain the signals had very different dynamic ranges, by looking at the difference between the mean percentiles, it can be seen that in all genres almost the same value is obtained. Even more, the genre with less dynamic range in the PCM domain is the one that presents the maximum difference between percentiles in the MDCT domain.

Genre	Mean 5th percentile (dBFS)	Mean 95th percentile (dBFS)	Difference (dB)
Classical	-125.9	-77.9	48
Jazz	-105.3	-60.1	45.2
Rock	-104.7	-52	52.7
Pop	-97	-51.3	45.7

Table 16. Mean values of the 5th and 95th percentiles and its difference for all genres

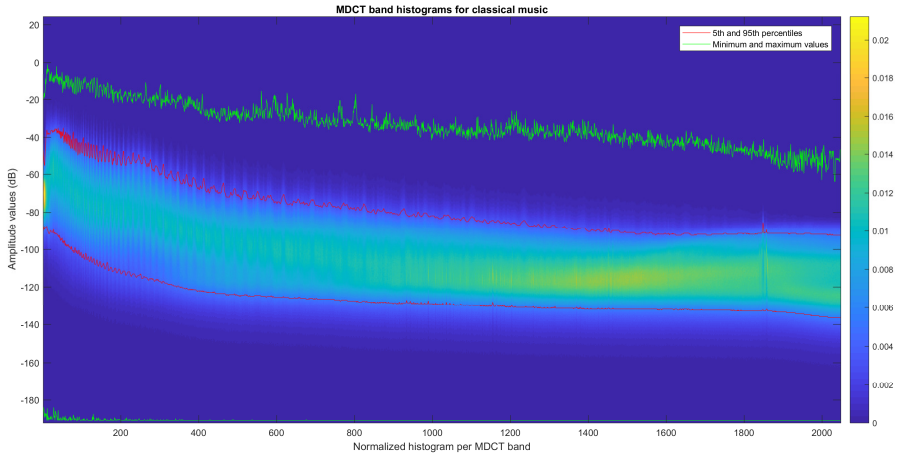

Figure 63. MDCT band histograms for classical music

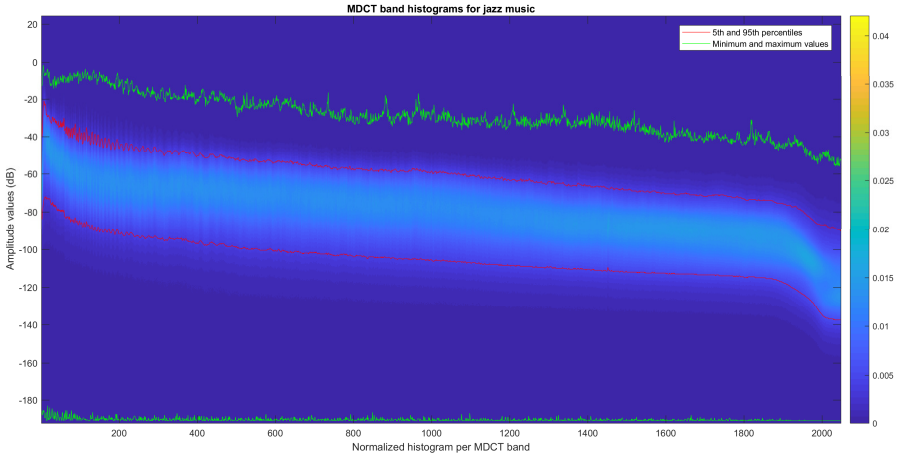

Figure 64. MDCT band histograms for jazz music

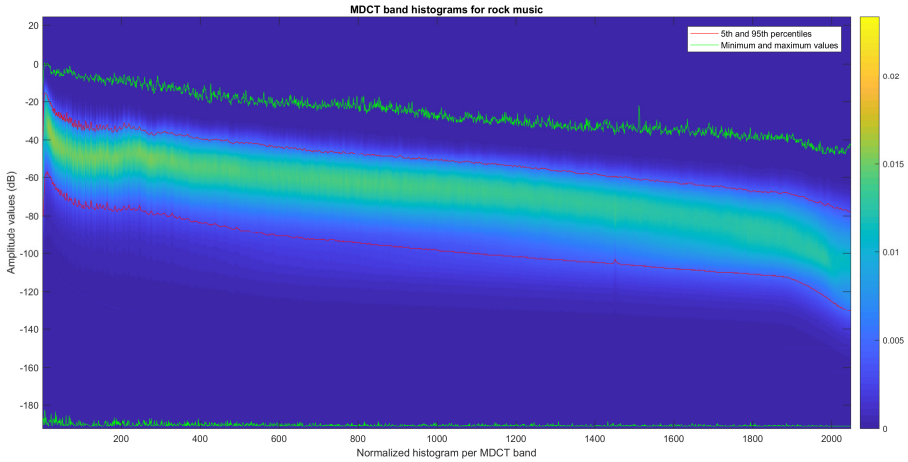

Figure 65. MDCT band histograms for rock music

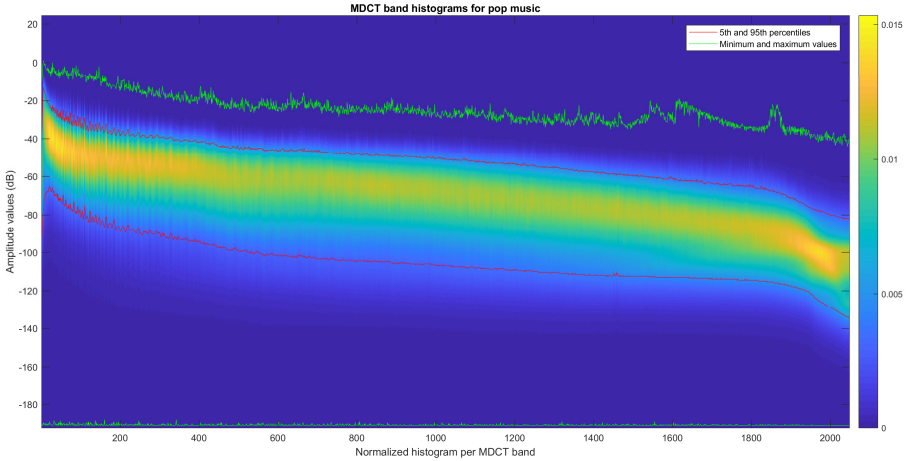

Figure 66. MDCT band histograms for pop music

5.1.1.2. QMF

By looking at the results in Table 15, the QMF have surprisingly low values in comparison with the MDCT transform. This is because one extra bit of headroom is saved in the QMF transform to avoid overflows with signals that may have a DC offset, or that may act as one for a QMF block length such as a square wave with a low fundamental frequency. This amplitude difference is present in all the results shown in this work, as these are the actual transforms used at Dolby.

This decrease of the maximum values obtained in the QMF transform is shown in Table 15, but even though, the 95th percentile is almost equal to the values obtained for the MDCT. The mean values for the 5th and 95th percentiles for each genre have been also calculated for the QMF results, and they show that even though the maximum level is lower than in the MDCT, the percentiles are higher. Also, that the difference between percentiles, stays almost equal in comparison with the MDCT. In the next table the numeric results can be seen:

Genre	Mean 5th percentile	Mean 95th percentile	Difference
Classical	-117.7	-68.7	49.0
Jazz	-98.3	-52.4	45.9
Rock	-92.1	-44.6	47.5
Pop	-99.9	-45.2	54.7

Table 17. Mean 5th and 95th percentile values and their differences

This shows that the transforms distribute the energy differently because of their different block sizes and therefore number of bands, and because of the energy calculation in the QMF. In the next images the resulting plots for the QMF results can be seen:

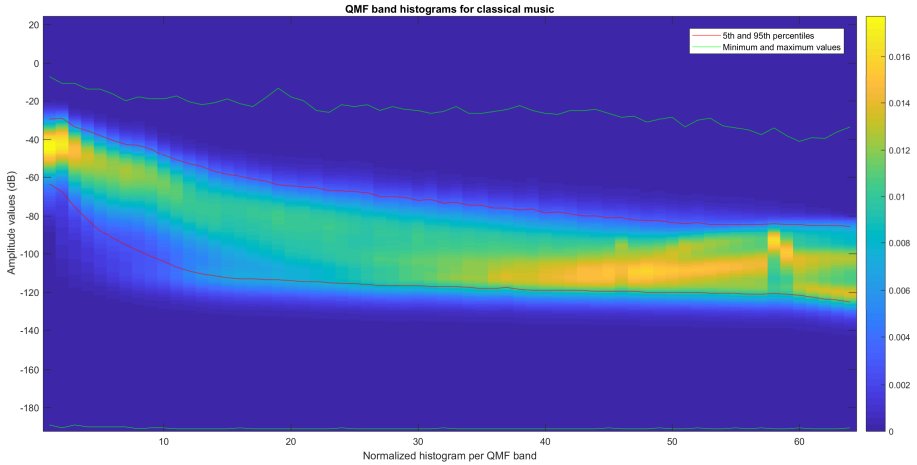

Figure 67. QMF band histograms for classical music

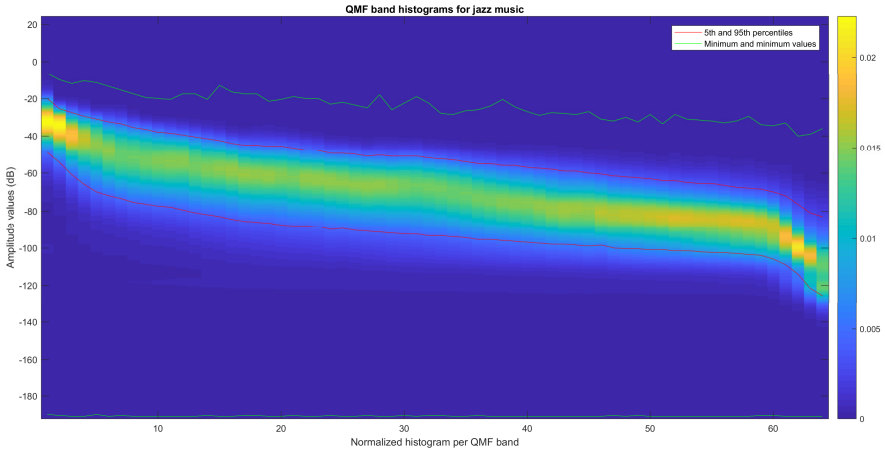

Figure 68. QMF band histograms for jazz music

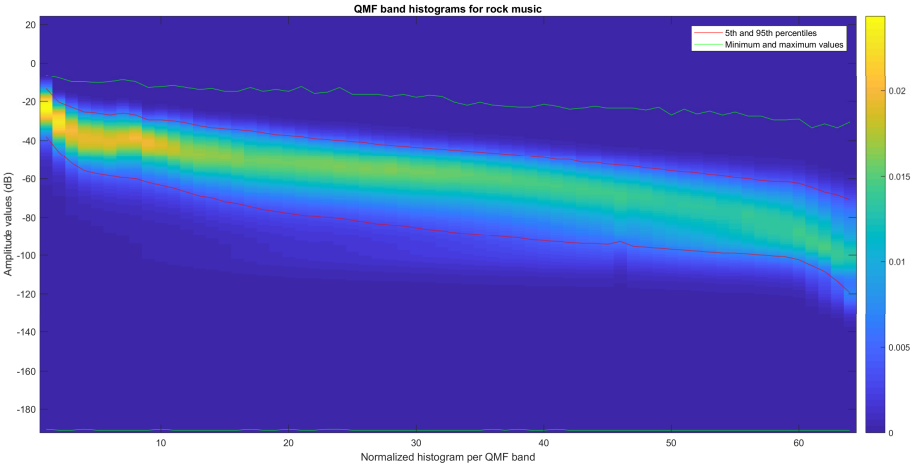

Figure 69. QMF band histograms for rock music

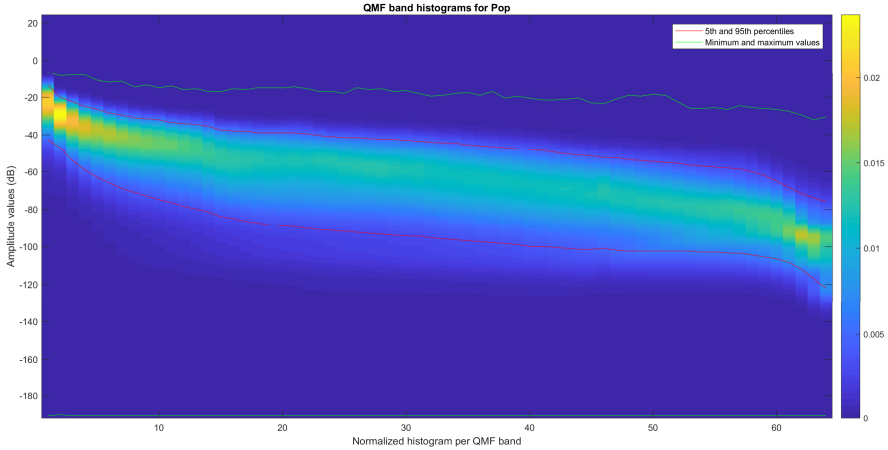

Figure 70. QMF band histograms for pop music

5.2. Multichannel cinematic content

In this section, the analysis results for 5.1 cinematic content will be presented. The content used is specifically cinematic content, so multichannel music and TV programs have been left apart. The list of content used can be seen in the *Annex 2: List of content used for analysis of cinematic multichannel content*.

First of all, the content of every channel will be compared, to better understand the nature of the signals. By looking at the true peak level of the signals some conclusions can be drawn, so in the next figure, the true peak value for every channel can be seen:

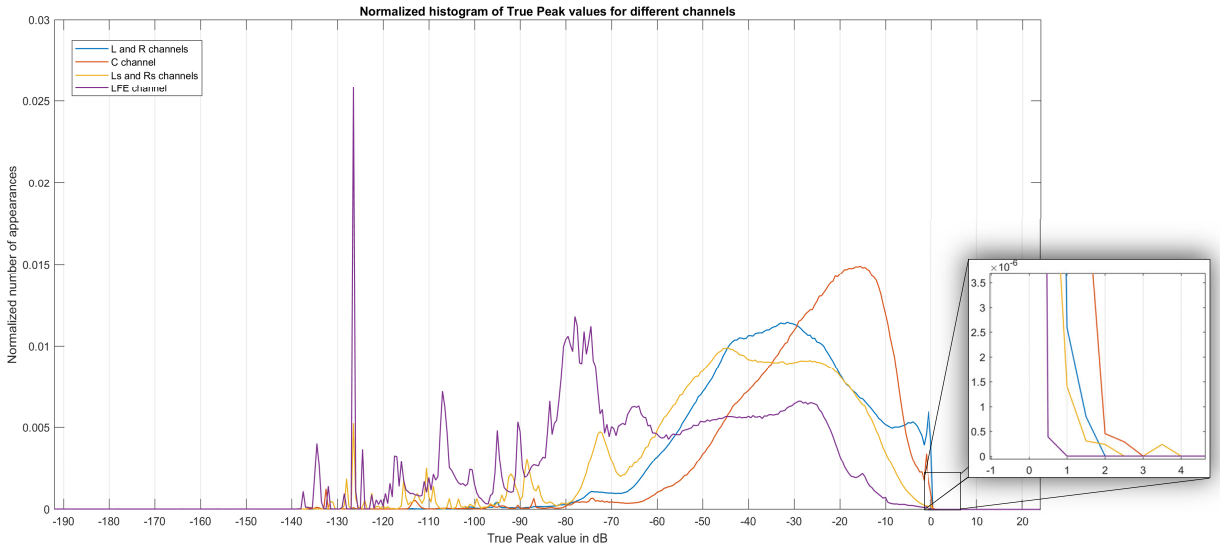

Figure 71. Normalized histograms of true peak values for different channels

The plot above shows that some of the channels present more true peak values at full scale level, or beyond, than others. The channels that present higher true peaks are the Left and Right and the Center channels. Also, limiting can be appreciated in the plot in those channels, as a high amount of true peak values are at 0 dBFS. This is not a strange fact, as in these channels usually concentrates all the dialog of a film, and compression and relatively high levels are usually used when mixing to increase intelligibility.

Now, the mean square values of the signals will be presented, as it is a good approximation of the energy in each channel:

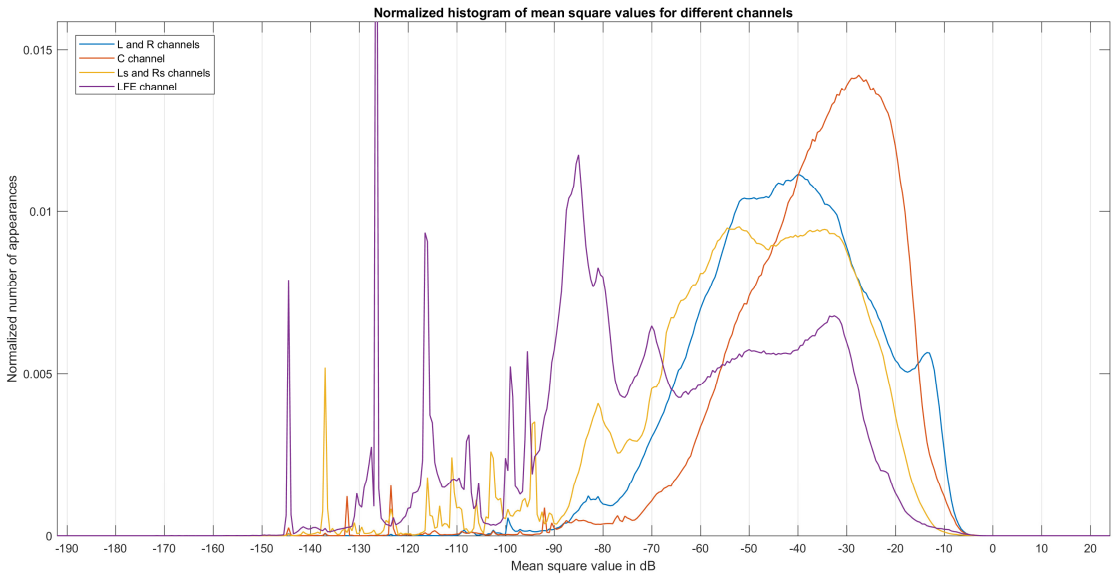

Figure 72. Normalized histogram of mean square values for different channels

With the mean square values of the channels, it can be seen again, that the channel with the higher amount of energy are the Center and Left and Right channels, as the most important information of the film, such as dialog and music, is normally in those channels. Even though, the Center channel out stands the others and shows a narrower distribution than any other channel. This is because, the Center channel carries most of the dialog of a film, and with this plot can be seen how the dialog level is used as a reference for the level of all the other elements of a film. This is described in the EBU r128 recommendation as “anchor signal” [23].

Also, it can be seen that the LFE channels has much less energy than the other channels, but it is important to know that a 10 dB gain to the LFE channel will be applied in the play back stage.

Note: the high peak in the LFE channel is caused by the high amount of silence in this channel, combined with the dithering used. In the analysis, the last bin of the histograms is considered as silence, and therefore is never considered. But when dithering is used leaving silence parts out of the histograms is more complicated due to the different types of dithering and quantization methods.

5.2.1. Down mixing

The down mixing results in this section are obtained when analyzing the LoRo down mix of the original 5.1 content. The LoRo coefficients used when down mixing are the ones presented in *Table 14*.

It can be easily seen by looking at the coefficients that the energy of the signals will be concentrated in fewer channels. Taking in mind that in some channels there was many peak values at 0 dBFS, it is very probable that, when down mixing, values beyond full scale will be obtained. In the next figures the comparison between the original content and the down mixed content, for mean square and peak values, can be seen. In this plot the LFE channel is left out, as it is not taken into account for the LoRo down-mix.

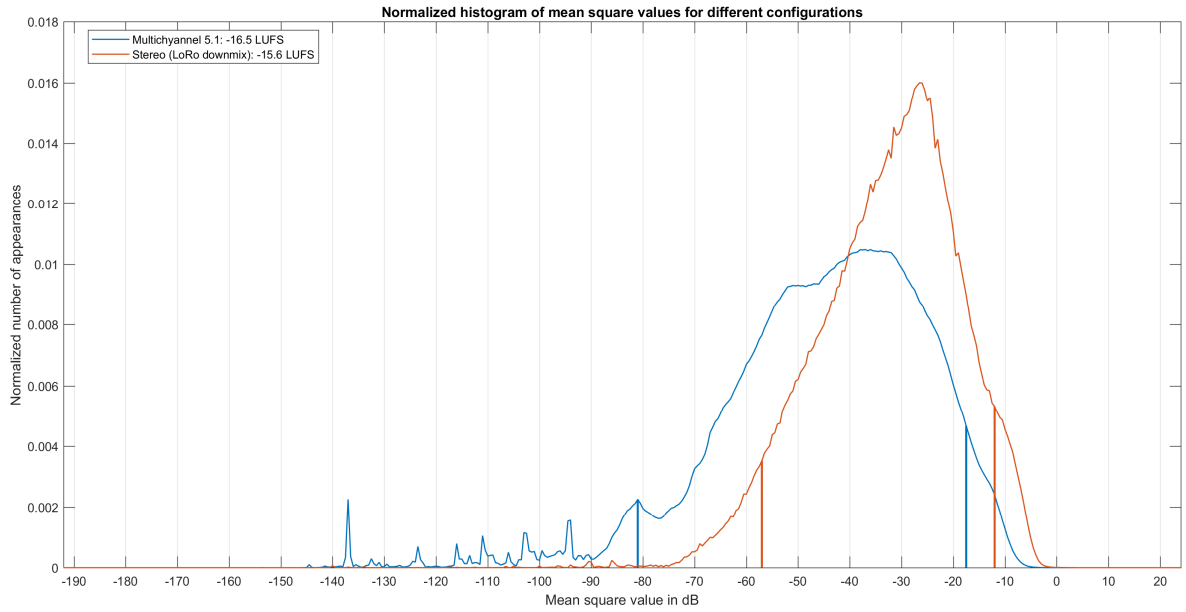

Figure 73. Normalized histogram of mean square values and its 5th and 95th percentiles for 5.1 and its LoRo down-mix

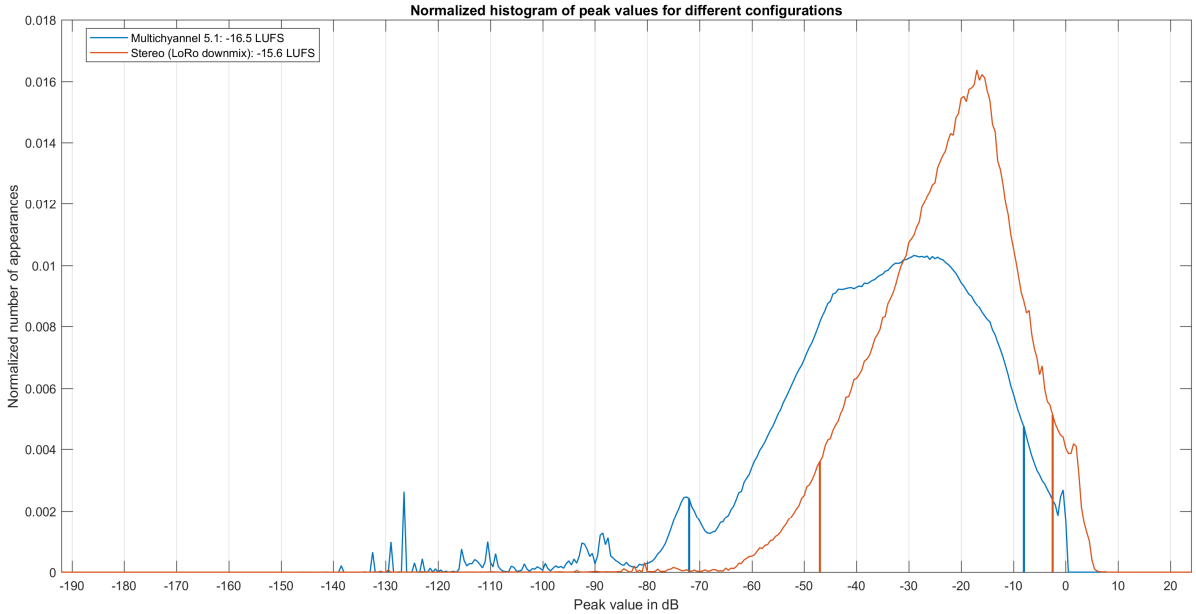

Figure 74. Normalized histogram of peak values and its 5th and 95th percentiles for 5.1 and its LoRo down-mix

As it can be seen in the images, the energy is higher in the down mixed channels than in the multichannel ones, as the energy has been just concentrated in fewer channels. The levels have increased around 6 dB, in both cases (mean square and peak values). This causes that many peak values are beyond full scale range and therefore, those values would be clipped or wrapped around in a real fixed-point system. This can also be potentially dangerous, as it could cause overflows also when transforming the signal to other domains. To prove it, in the next table, a list of the maximum value and the 95th percentile of each parameter analyzed is presented:

Multichannel 5.1		Mean square	Peak	True Peak	MDCT	QMF
Multichannel	95th percentile	-18.5 dBFS	-9 dBFS	-9 dBFS	-33.5 dBFS	-28 dBFS
	Highest value	-1 dBFS	0 dBFS	2.5 dBFS	2 dBFS	-6 dBFS
LoRo down mix	95th percentile	-12 dBFS	-2.5 dBFS	-2.5 dBFS	-24.5 dBFS	-21 dBFS
	Highest value	4.5 dBFS	7.5 dBFS	8.5 dBFS	6.5 dBFS	1.5 dBFS
	Δ 95th percentile	+6.5 dB	+6.5 dB	+6.5 dB	+9 dB	+7 dB
	Δ Highest value	+5.5 dB	+7.5 dB	+6 dB	+5 dB	+7.5 dB

Table 18. Mean square, peak, true peak, MDCT and QMF 95th percentile and maximum value for multichannel content and its LoRo down-mix and their increments

As it can be seen, the values of all the parameters increase when down mixing, showing an average increase of 6.7 dB. It can also be seen, that values beyond full scale in the MDCT domain have been already obtained with the original content, and therefore, when down mixed, those values are even higher causing more potential error in the MDCT domain of a hypothetical real fixed-point system. The down mixed content generates values over full scale in the QMF domain too.

5.2.2. Transforms

In this section the results of the MDCT and QMF transforms will be again presented. This time, the comments on the results will focus on the differences between channels, as the transforms can give a good representation of the content of each channel. As it is mentioned, the signals used in this section are 5.1 signals, so the channels used are Left (L), Right (R), Center (C), Low Frequency Effect channel (LFE), Left surround (Ls) and Right surround (Rs). As the content for L and R channels are very similar, the results of these channels will be presented together, the same happens with the Ls and Rs channels.

First, the results for the L and R channel:

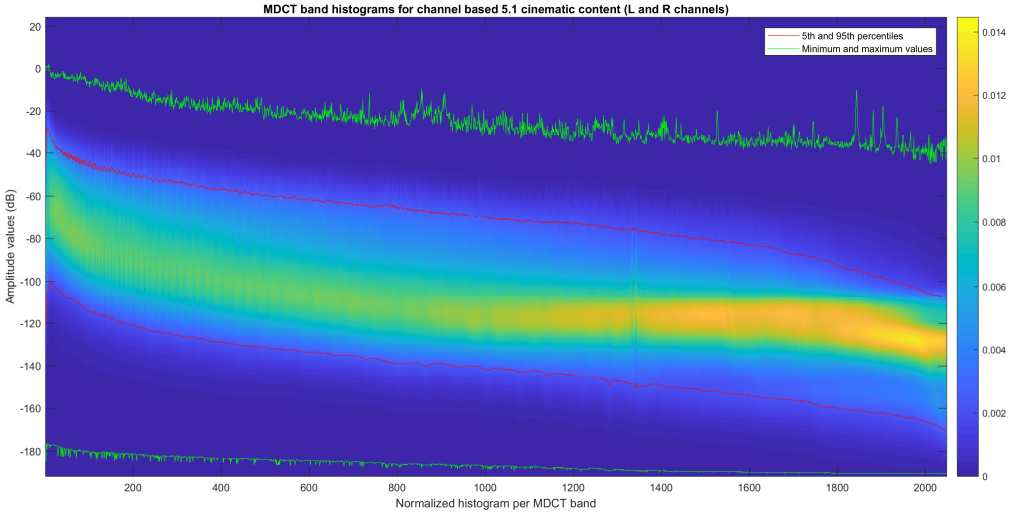

Figure 75. MDCT band histograms for channel based 5.1 cinematic content (L and R channels)

The difference between the percentiles for the L and R channel is 70.3 dB. It can also be seen, that the signal has a harmonic nature, with higher amplitude in the low and mid frequencies, as these channels are often used for background music. Also, the use of noise shaping techniques can be seen, as the high frequency bands show a high concentration of low amplitude values. The maximum values obtained in the transform are beyond full scale, presenting up to +2 dBFS in the low frequency bands, so there are high energy signals in the L and R channels.

Now the results for the Centre channel will be presented:

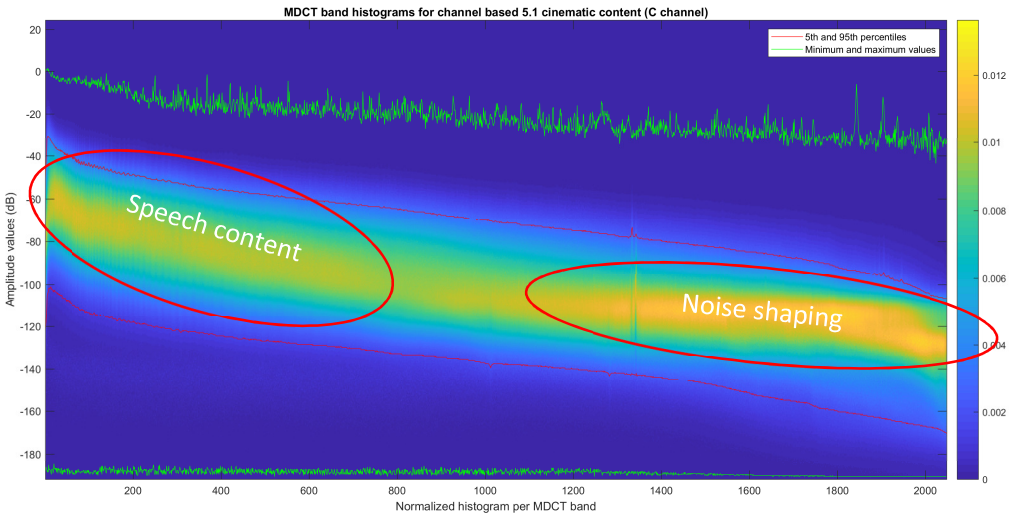

Figure 76. MDCT band histograms for channel based 5.1 cinematic content (C channel)

The Centre channel is often used for speech content, and this can be seen in the previous figure, where the first area corresponds to the speech frequency range, and the second one, is the result of noise shaping techniques. Here the dynamic range is very similar to the previous one with an average difference between the percentiles of 67.14 dB and a maximum value of +1.5 dBFS in a low frequency band. Now the results for the Ls and Rs channels will be presented:

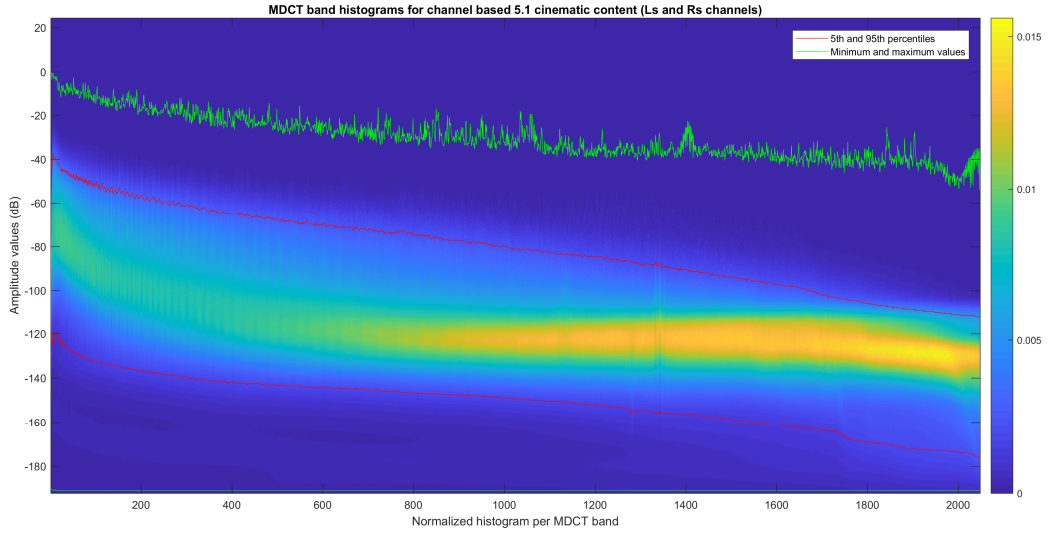

Figure 77. MDCT band histograms for channel based 5.1 cinematic content (Ls and Rs channels)

In this case, the signals have less energy than in the two previous examples, as both percentiles are lower, but the difference between them is very similar as in the previous cases, 70.1 dB. The maximum value is also lower than in the previous examples with a value of 0 dBFS. The 5th percentile is approximately 20 dB lower than in the previous case, causing the high dynamic range on Figure 80. Also, the noise shaping techniques can be seen in the high frequency bands. Now the LFE channel results will be presented:

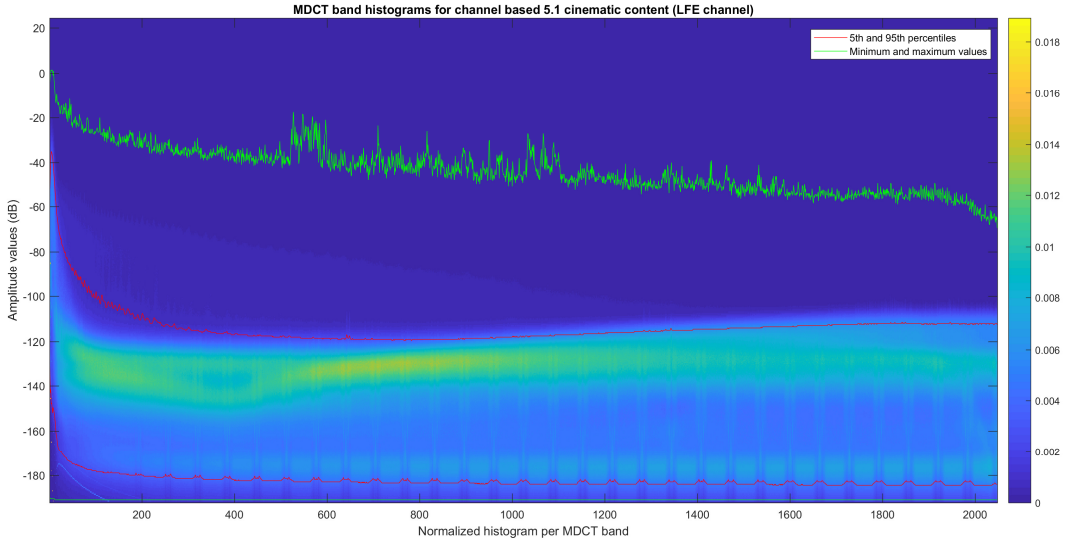

Figure 78. MDCT band histograms for channel based 5.1 cinematic content (LFE channel)

In this case it is clear that the results are from the LFE channel, as there is a clear difference between the low frequency bands and the rest, where the low bands present high energy and all the rest have much lower values. The maximum value of the transform for the LFE channel is +1.5dBFS and it is, of course, in a low frequency band. The difference between the percentiles, is in this case, 68.95 dB. It is surprising, though, that the maximum values are so high for all the transform bands. This is caused by errors during

the rendering from the mixing session to a certain channel configuration, as some elements that do not belong to the LFE channel have been detected in this channel, like in the next example:

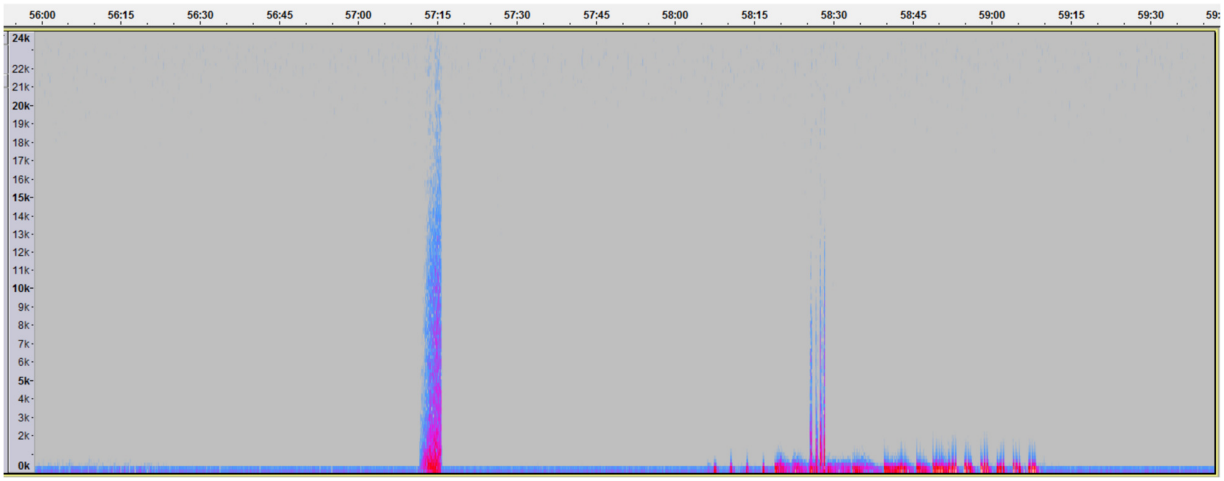

Figure 79. Spectrogram from a clip of the LFE channel from a film with multichannel 5.1 audio

These unfiltered parts in the LFE channel could cause some problems in some cases, as the level of content of the LFE channel will be increased 10 dB and, therefore, could prorogue unpleasantly high levels if it is not filtered before the processing. This can be very problematic for example in the case of Headphone rendering, as LFE channel is also taken into account.

Another surprising thing of the results is the periodic shape of the transform. This could be caused by the side lobes of the window used for the transform, together with the fact that there is very little energy in the mid and high frequency bands.

Finally, it can be also observed that dithering and noise shaping techniques have been used in some files as an increase of the amplitude in the high frequency bands. When combining all the channels, the resulting results are the following:

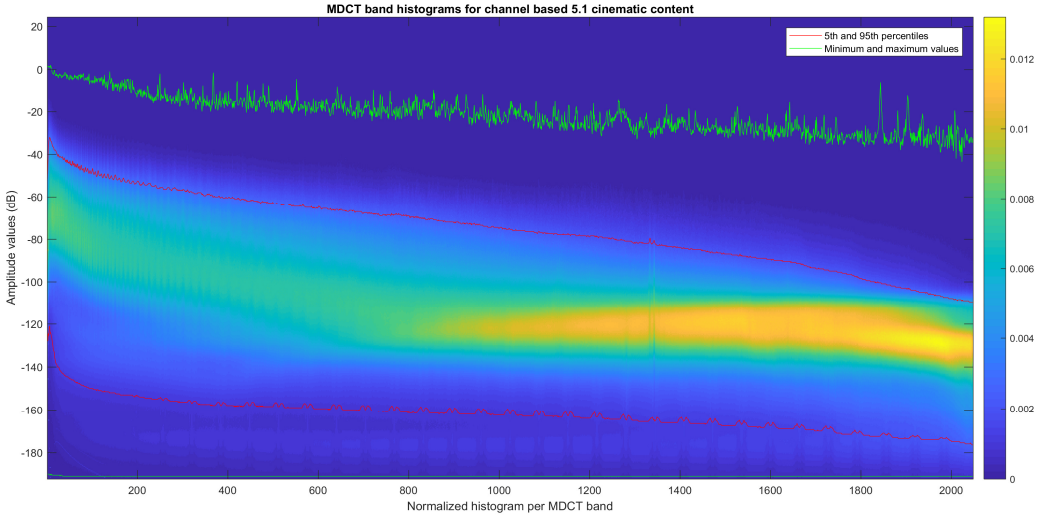

Figure 80. MDCT band histograms for channel based 5.1 cinematic content

As it can be seen, all the previously described characteristics are present in the figure. The mean difference between the percentiles if all the channels are taken into account is 86.5 dB, but each channel separately has a mean difference between percentiles of 69.21 dB. This will be now compared with the results for the LoRo down mix.

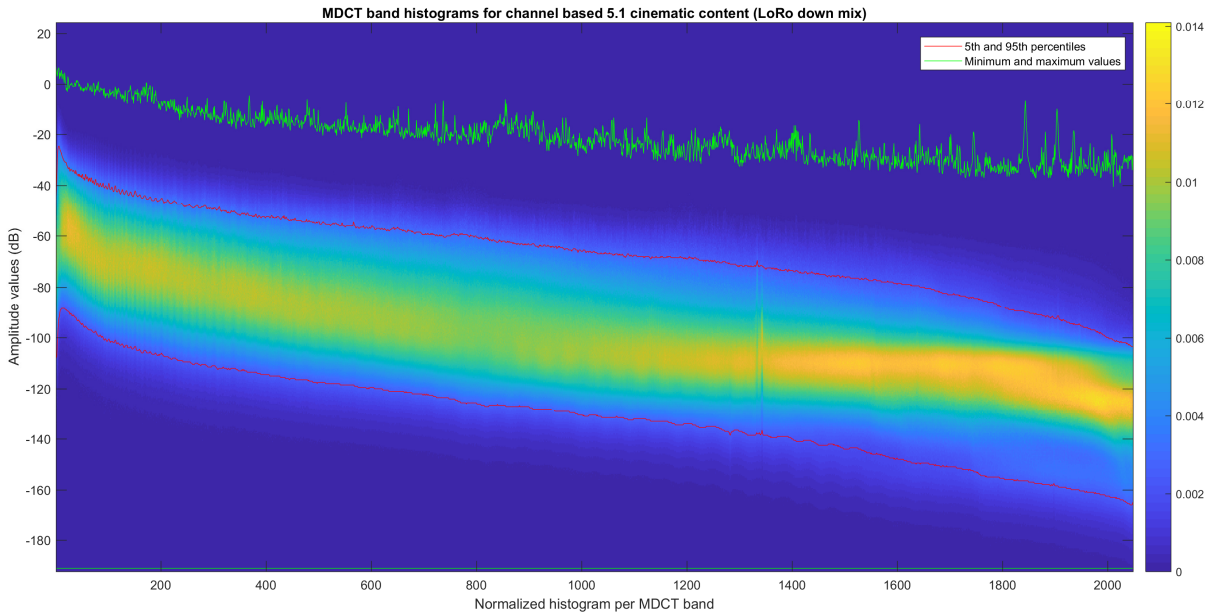

Figure 81. MDCT band histograms for channel based 5.1 cinematic content (LoRo down-mix)

The overall shape of the transform is similar to the previous one, but there are some key characteristics that are different this time. First of all, the difference between the percentiles has decreased from mean per channel of 69.21 dB to 64.5 dB, and the maximum value has also increased from +2 dBFS to +6.5 dBFS. The 5th percentile is now much higher than in the previous example, as there are fewer channels and therefore less silence or small signals.

All the previously explained characteristics are very similar for the QMF domain, but with a lower maximum amplitude and fewer bands. One big difference between the results from MDCT and QMF, is that for the MDCT, just the amplitude of each band is shown, but for the QMF the results represent the energy contained in each band. To show the comparison between transforms and the evolution of the results for both transforms, the results for the MDCT and QMF for 5.1 and LoRo are presented in the next figures:

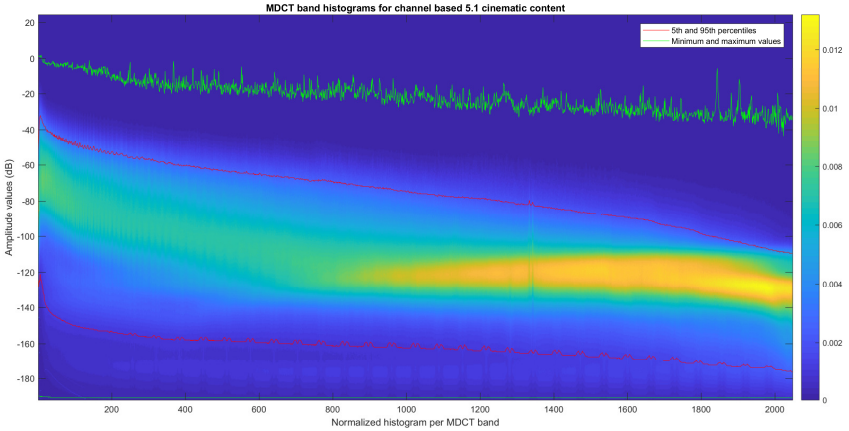

Figure 82. MDCT band histograms for channel based 5.1 cinematic content

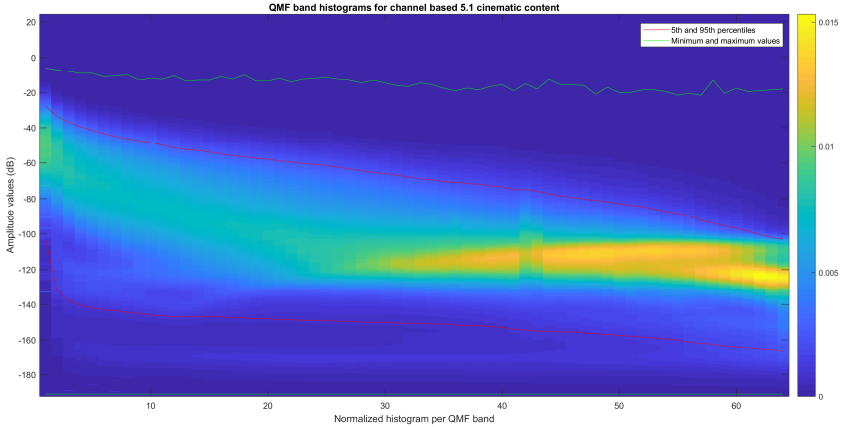

Figure 83. QMF band histograms for channel based 5.1 cinematic content

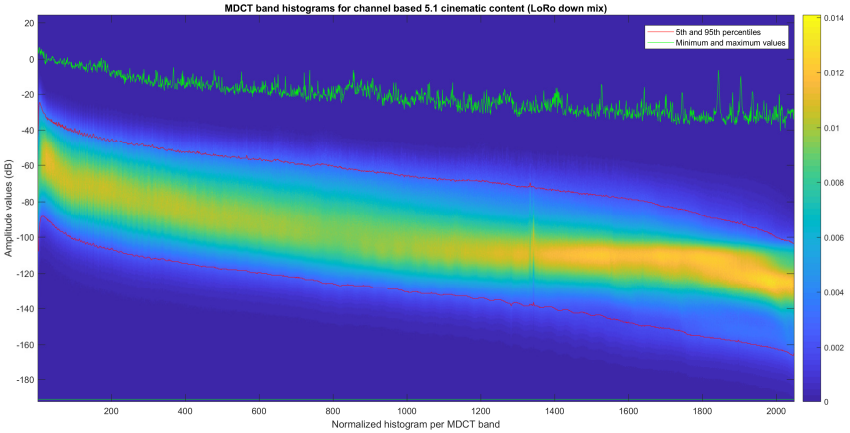

Figure 84. MDCT band histograms for channel based 5.1 cinematic content (LoRo down-mix)

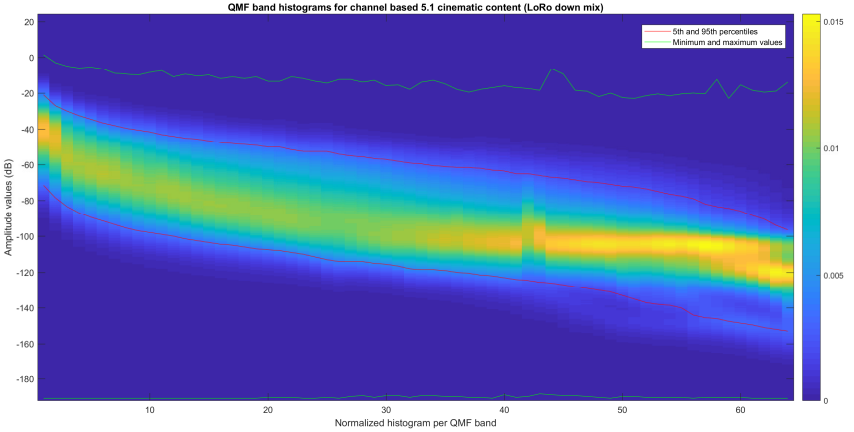

Figure 85. QMF band histograms for channel based 5.1 cinematic content (LoRo down-mix)

5.2.3. Loudness

Another important parameter of the analyzed files is its loudness level, as normally the signals will be loudness normalized when played back. The analyzed cinematic content is not mixed to reach the recommended loudness target (nor the one recommended by EBU (-23 LUFS), neither the one recommended by ATSC (-24 LKFS)), as the analysis shows an average loudness level of -16.5 LUFS. The level of such a signal, therefore, would be adjusted at the play back environment, if the loudness normalization is activated, or at the ingest or broadcasting stage of any broadcaster that follows the recommendations.

Also, the Loudness Range of a signal is also important, as some play back environment will apply dynamic processing to the signal at the playback stage if the signal shows too high dynamic range. The analysis shows an average loudness range of 37.5 LU for cinematic multichannel content. This could be good at the theater, but it would be challenging for any living room if playing the signal without any kind of dynamic processing. So, the signal will go through dynamic processing at the playback stage. [24]

In the next image, the histogram for momentary loudness level of the files analyzed, the 10th and 95th percentiles (representing the loudness range), the average loudness range value, and the average loudness level value, can be seen:

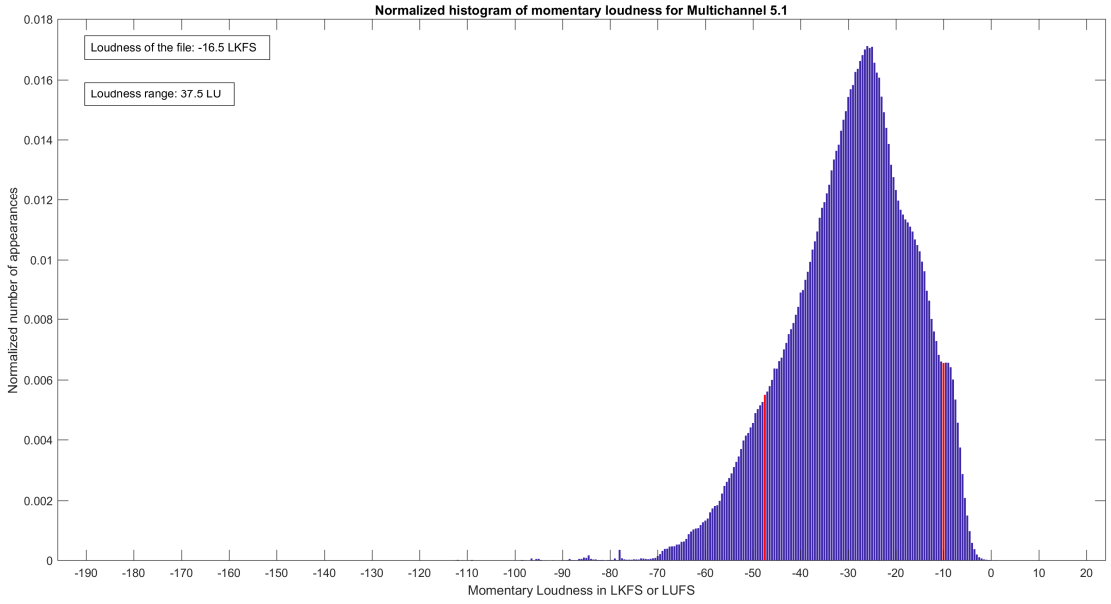

Figure 86. Normalized histogram of momentary loudness for multichannel 5.1 cinematic content and its 10th and 95th percentiles

When down mixing the signal to a stereo configuration, the Loudness Range value even increases one LU, as the 95th percentile increases in the process. Another interesting fact about the stereo down mixing is that, even though the mean square and peak values increase approximately 5 to 10 dB, the loudness measure of the resulting signal does not show such a big increase. The original multichannel content loudness level is -16.5 LUFS and the stereo down mix has a loudness level of -15.6 LUFS. This gives a good idea to understand how important the surround channels are for the loudness level calculation.

5.3. Object-based cinematic content

In this section, the content analyzed is no longer based in a fixed number of channels, instead, it is based in objects that are always accompanied by metadata that determines their position in a three-dimensional space. The data sample used in this section is formed by 11 pieces formed by films and trailers. This sample is much smaller than the used in the previous as it is difficult to obtain access to this content because of its sensibility.

As explained in the beginning of the section *5,* the audio content of every object has been analyzed first. Then a rendering to a fixed loudspeaker configuration is performed and, as in the previous sections, every channel for all loudspeaker configurations will be analyzed and compared. So, firstly, the results for the original object-based content are presented.

Analyzing object-based content presented some extra difficulties, in comparison with channel based, because of the big amount of silence that is present in the audio tracks of the objects. This silence would not have been important, if dithering would not have been used, or if the metadata for active or inactive objects would had been set. This presented a problem, as all histograms where biased by the big amount of low values caused by dithering. To solve that, the files with dither have been identified, and eliminated from the sample used to obtain the data.

The results obtained show that the content has greater dynamic than other previously analyzed content types. They also show relatively low 95th percentiles, but surprisingly high maximum values. The maximum value and the 95th percentiles of each parameter can be seen in the next table:

Atmos		**Mean square**	**Peak**	**True Peak**	**MDCT**	**QMF**
Atmos Channels	**95th percentile**	-22 dBFS	-13 dBFS	-13 dBFS	-33.5 dBFS	-30 dBFS
	Highest value	-0.5 dBFS	0 dBFS	1 dBFS	2.5 dBFS	-5.5 dBFS

Table 19. Highest and 95th percentile values for mean square, peak, true peak, MDCT and QMF for object-based cinematic content

The maximum values, are surprisingly high considering that object based content will always be rendered to a specific channel configuration, and therefore, the final channels will always have the same or more energy than the original signals. Those high levels in the original content cause levels over full scale for the true peak value and in the MDCT. This is problematic when rendering, as more and higher values over full scale will be obtained.

It has been observed, that the bed channels have higher values than the objects, and those channels are the ones causing high true peaks. In the next table the 95th percentile and the maximum value for objects and beds is compared:

Atmos		**Mean square**	**Peak**	**True Peak**	**MDCT**	**QMF**
Atmos Beds	**95th percentile**	-19 dBFS	-10 dBFS	-10 dBFS	-29.5 dBFS	-27 dBFS
	Highest value	-0.5 dBFS	0 dBFS	1 dBFS	1.5 dBFS	-5.5 dBFS
Atmos Objects	**95th percentile**	-25.5 dBFS	-16.5 dBFS	-16.5 dBFS	-37.5 dBFS	-34 dBFS
	Highest value	-1 dBFS	0 dBFS	0 dBFS	2.5 dBFS	-6 dBFS
	Δ 95th percentile	-6.5 dB	-6.5 dB	-6.5 dB	-8 dB	-7 dB
	Δ Highest value	-0.5 dB	0 dB	-1 dB	-1 dB	-0.5 dB

Table 20. Mean square, peak, true peak, MDCT and QMF 95th percentile and maximum value for beds and objects and their increments

As it can be seen, the beds present higher values in both, maximum value and 95th percentile, but the increase in the percentile is much bigger than the maximum value. This indicates that the objects have a higher dynamic range than the beds. It must be also considered that beds would never be silent, while objects in the original master file are silent most of the time. Even though the silence has been filtered out of the histograms, if dithering has been used in the object tracks, the silence will still be shown in the plots. In the next figures the histograms for the beds and the objects are compared, for the mean square and true peak value:

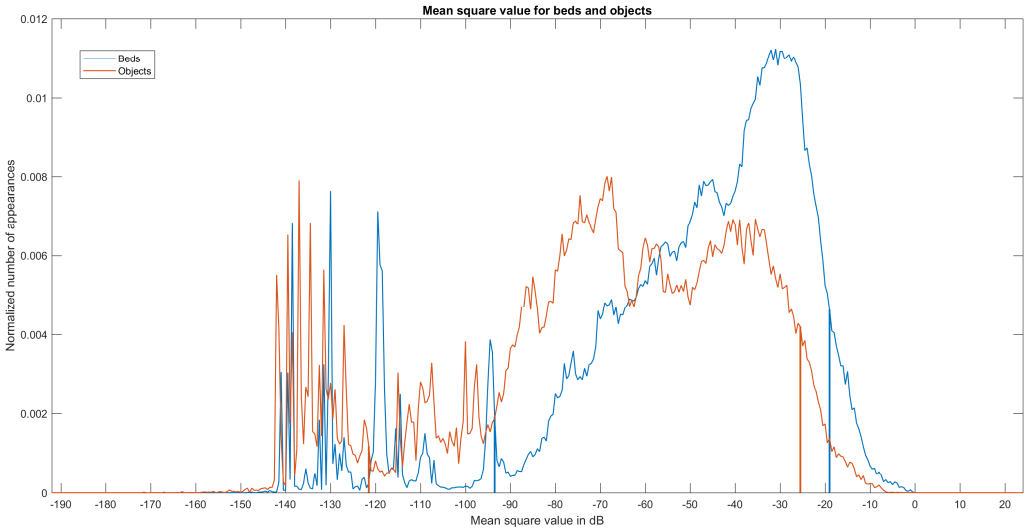

Figure 87. Normalized histogram of mean square values and its 5th and 95th percentiles for bed and object channels

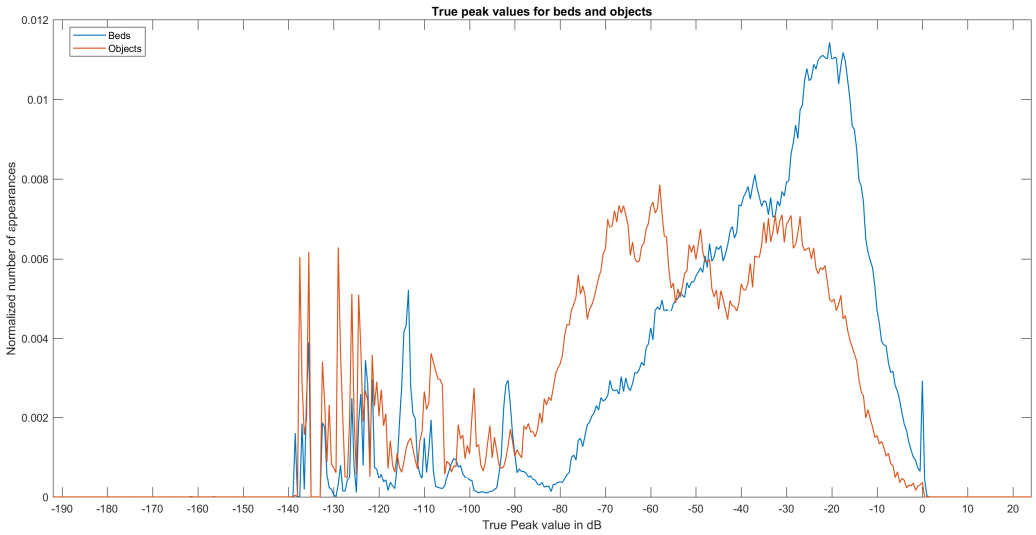

Figure 88. Normalized histogram of peak values for bed and object channels

5.3.1. Rendering

In the case of object based content, no down mix has been performed, like in the previous section. Instead, the Object Audio Renderer, as specified in ETSI TS 103 448 V1.1.1 (2016-09), used for Dolby Atmos has been used. The process is similar to a down mix, but more complex as it takes other variables into account like all three spatial dimensions, sizes of the audio objects, gains, etc. The renderings used in this section are a rendering to 5.1 and to stereo 2.0, to be comparable to other cinematic content.

In the next table the maximum values and the 95th percentile of every parameter analyzed for the original object-based input and the 5.1 channel rendering can be seen:

Atmos		Mean square	Peak	True Peak	MDCT	QMF
Atmos Channels	95th percentile	-22 dBFS	-13 dBFS	-13 dBFS	-33.5 dBFS	-30 dBFS
	Highest value	-0.5 dBFS	0 dBFS	1 dBFS	2.5 dBFS	-5.5 dBFS
5.1 rendering	95th percentile	-18 dBFS	-9 dBFS	-9 dBFS	-29 dBFS	-26 dBFS
	Highest value	-0.5 dBFS	6.5 dBFS	6.5 dBFS	1.5 dBFS	-1.5 dBFS
	Δ 95th percentile	+4 dB	+4 dB	+4 dB	+4.5 dB	+4 dB
	Δ Highest value	0 dB	+6.5 dB	+5.5 dB	-1 dB	+4 dB

Table 21. Mean square, peak, true peak, MDCT and QMF 95th percentile and maximum value and their increments for object-based content and its 5.1 rendering

As predicted, the values have increased when rendering to a 5.1 channel configuration. The values obtained after rendering are very similar to the values of the channel-based content presented in the previous section. The peak and true peak values are the exception as they present much higher values, that when rendering to 2.0 will become even higher.

The increment of the maximum values is not as high as expected in some cases and even a curious case can be seen in the case of the maximum value of the MDCT. The maximum value obtained in the MDCT has decreased after the rendering. This can be caused by the trims set in the object audio metadata that causes an attenuation to the surround and height channels.

Now, the results for 2.0 rendering are presented:

Atmos		Mean square	Peak	True Peak	MDCT	QMF
Atmos Channels	95th percentile	-22 dBFS	-13 dBFS	-13 dBFS	-33.5 dBFS	-30 dBFS
	Highest value	-0.5 dBFS	0 dBFS	1 dBFS	2.5 dBFS	-5.5 dBFS
2.0 rendering	95th percentile	-12.5 dBFS	-3 dBFS	-3 dBFS	-22 dBFS	-20 dBFS
	Highest value	3 dBFS	8.5 dBFS	8.5 dBFS	3.5 dBFS	1.5 dBFS
	Δ 95th percentile	+9.5 dB	+10 dB	+10 dB	+11.5 dB	+10 dB
	Δ Highest value	+3.5 dB	+8.5 dB	+7.5 dB	+1 dB	+7 dB

Table 22. Mean square, peak, true peak, MDCT and QMF 95th percentile and maximum value and their increments for object-based content and its 2.0 rendering

As it can be seen, the values have grown higher than in the last rendering. An important value to remark here is the increase of the 95th percentile, as it has experienced a mean growth of 10 dB for all the parameters measured. This must be considered for planning headroom requirements as it is a substantial increase that has been measured with the 95th percentile, which means that this is not a single outlier case, but a significant percentage of all samples.

Also, the maximum values have experienced a substantial increase, even though not as big as the 95th percentile values. Now the peak and true peak values are far beyond the full-scale level and would be, therefore, severely clipped in a hypothetical fixed-point system.

All these effects can be graphically seen in the next three figures, where the mean square value, the peak value and the true peak value for the original content, the 5.1 down mix, and the 2.0 down mix are presented:

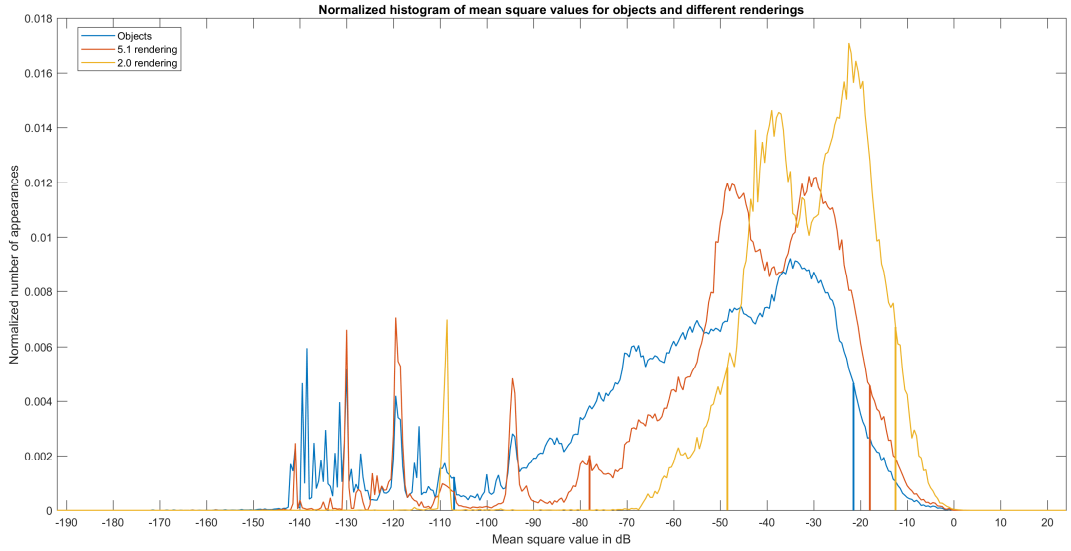

Figure 89. Normalized histogram of mean square values and its 5th and 95th percentiles for object-based content and its 5.1 and 2.0 rendering

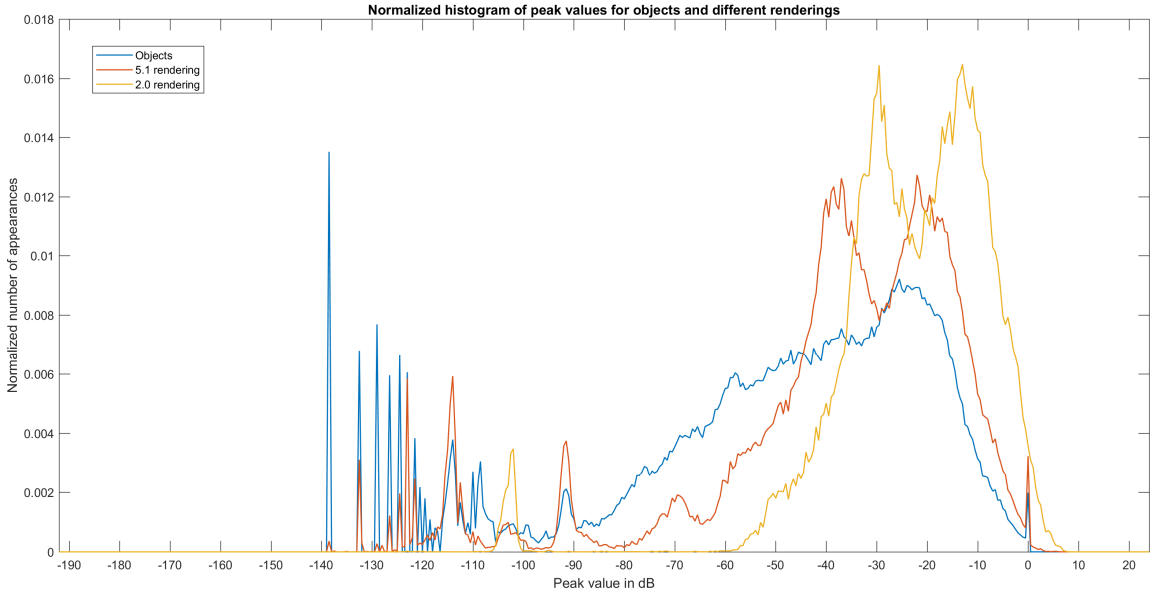

Figure 90. Normalized histogram of peak values and its 5th and 95th percentiles for object-based content and its 5.1 and 2.0 rendering

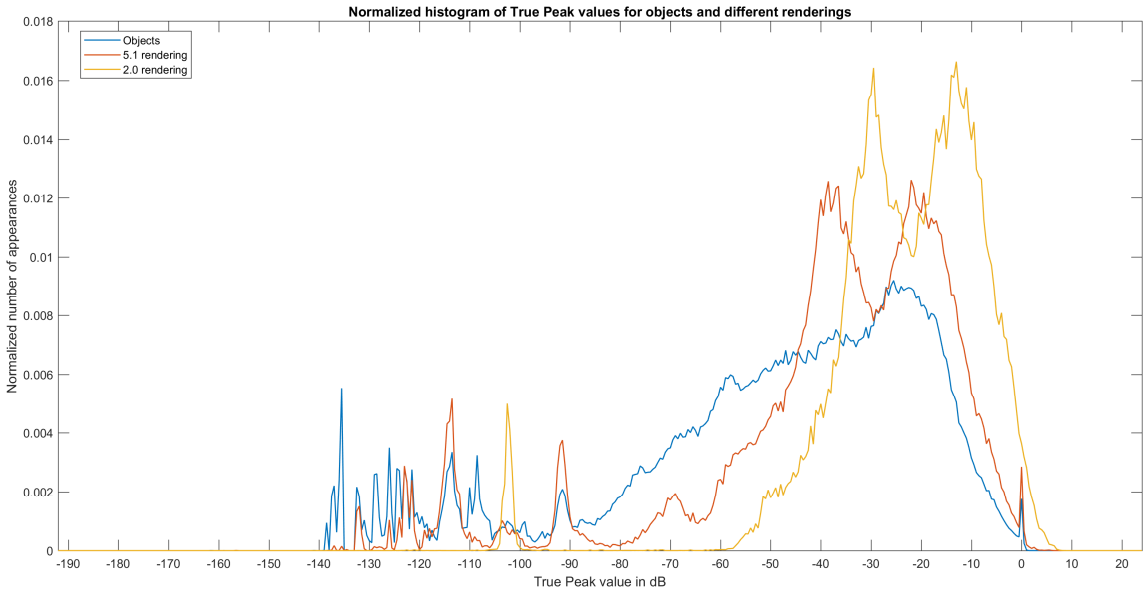

Figure 91. Normalized histogram of true peak values and its 5th and 95th percentiles for object-based content and its 5.1 and 2.0 rendering

5.3.2. Transforms

In this section, the results for the MDCT and QMF transforms for the resulting rendered to 5.1 and 2.0 signals are presented. In this part the comments will be focused in the amplitude and dynamic range differences when rendering to different number of channels. The results for both transforms are very similar, and therefore they can be commented together. The results for object based audio are not presented here, as because of the great amount of silence in the objects, it is difficult to visualize properly the results of the transforms.

First, when looking at the results for the 5.1 rendering below, it is noticeable how dynamic the transforms are in comparison with the transforms presented for the music content. In this case, every band has a broad area colored with lighter colors that indicates a relative high concentration of values. Also, an area with very low values can be seen throughout all the bands, probably caused by very small amplitude signals in some of the channels, like the surrounds or LFE, as seen in the previous section *5.2*.

Then, when comparing the results with the 2.0 rendering, it is clear how this low energy region on the transforms is gone, because the signals are now concentrated in fewer channels and therefore, there is no channel with only low amplitude signals. Also, it can be seen how the overall level has increased, and that the light-colored areas have become a little narrower because of this addition of the signals in fewer channels.

By looking at the maximum values, it can be seen how the values that are beyond 0 dBFS are always in the lower bands of the transform. In the rendering for 5.1 there are fewer bands, and with lower levels beyond 0 dBFS than in the 2.0 rendering.

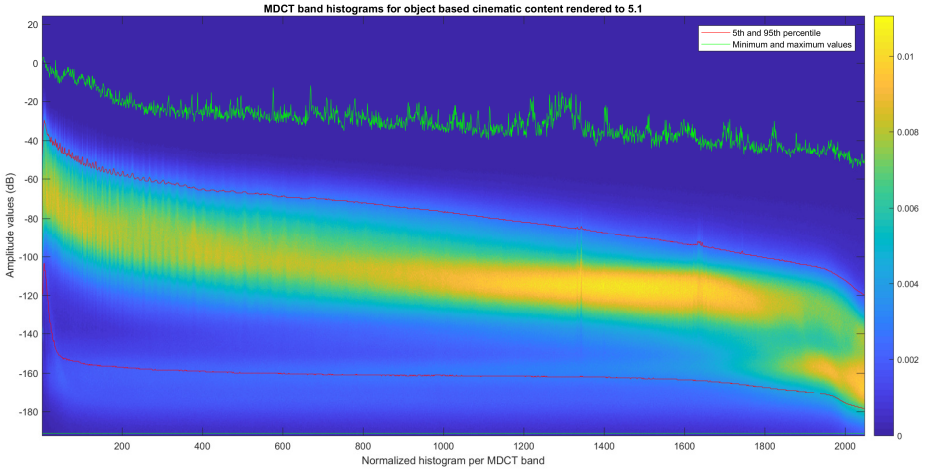

Figure 92. MDCT band histograms for object-based cinematic content rendered to 5.1

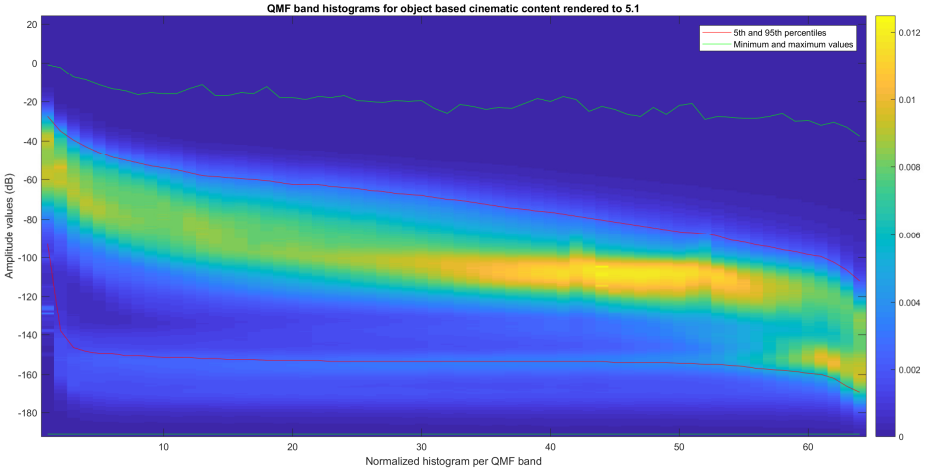

Figure 93. QMF band histograms for object-based cinematic content rendered to 5.1

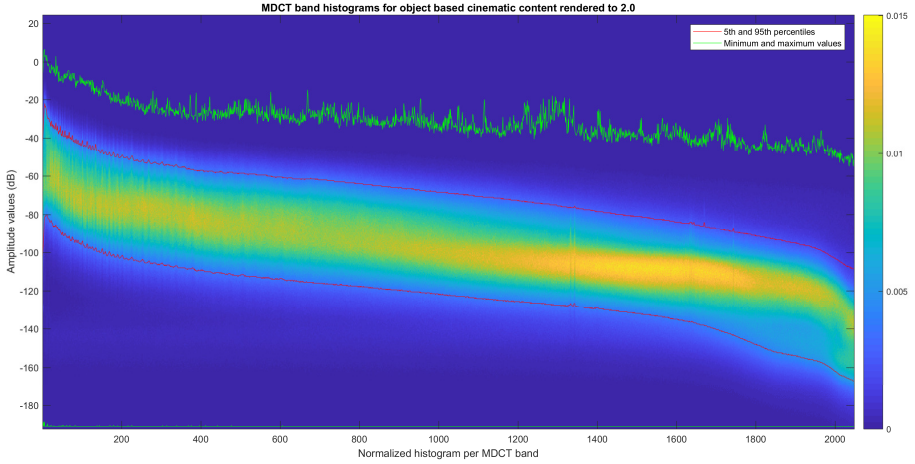

Figure 94. MDCT band histograms for object-based cinematic content rendered to 2.0

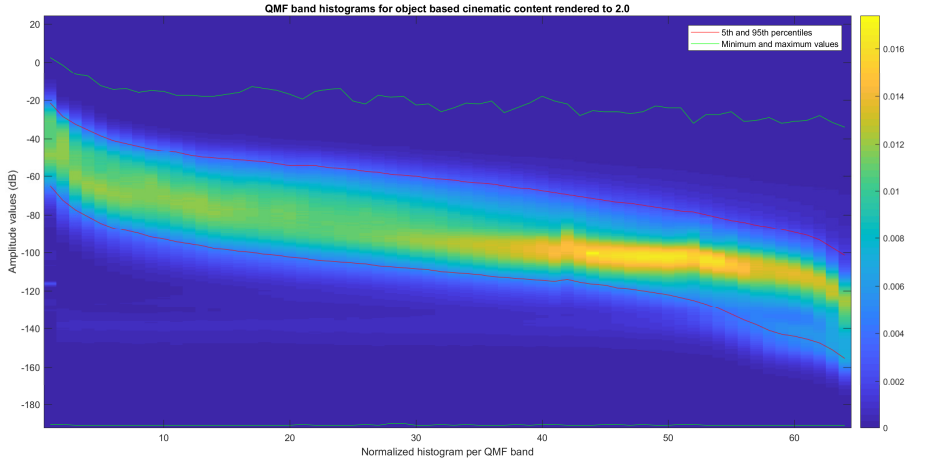

Figure 95. QMF band histograms for object-based cinematic content rendered to 2.0

6. Conclusion

The aim of this thesis has been to provide information and documentation about the behavior of several data domains and computation processes together with the nature of real world signals. This has been partially, but successfully accomplished.

Chapter 2 provides an overview of the relation between the quality of a quantized signal (SNR) with the recommended reference loudness levels in the time domain. Also, it provides information about the behavior of MDCT and QMF in a theoretical environment, together with the effects of quantization in those domains. The objectives for this section were successfully accomplished, as all the planed topics have been covered and meaningful results have been obtained.

In chapter number 3, an analysis of fixed point-processing blocks has been provided. The fixed point basic operations have been reviewed, and the implementations for the MDCT and QMF transforms have been tested. This gives some important insides of the error introduced by those processes and its distortion.

Even though the results of this part are not deceiving, the information provided is not detailed enough because of the complexity of the processes involved in the transforms, together with the time constrains of the project. There are also many other processes that could be analyzed, as they are also critical for the quality of the result of some processes. Those processing blocks are, for example: OAR for different channel configurations, Dolby volume, headphone rendering, up-mixing, etc.

Finally, in chapter number 4, the results for the real world content analysis have been presented. The analysis was performed for different contents such as music, channel based cinematic and object-based cinematic. The behavior of these signals in the three main data domains has been also presented, together with the effects of the Object Audio Renderer (OAR) for 5.1 and 2.0 configurations and the down-mixing process from 5.1 to 2.0. These results showed have been obtained without using any kind of limiting in order to provide information about potential overflows in the fixed-point domain.

The results obtained in chapter 4 provide important statistics about the nature of the signals, but there are more channel configurations, content types or genres that could be analyzed. For example: Channel based immersive cinematic, sports and TV content, immersive music content, 7.1.4 and 5.1.4 OAR configurations, 7.1.4 or 5.1.4 to 5.1 or 2.0 down-mixing, etc.

Even with the limitations of the work, with all the information provided that has been mentioned above, a better understanding of the functioning of the transforms and processing blocks and of the nature of the signals should be achieved. Providing, therefore, important knowledge to better plan and meet he optimal headroom and precision requirements for every domain in a signal processing chain.

Bibliography

[1] U. Zolzer, Digital Audio Signal Processing, Chichester, West Sussex: John Wilet and Sons Ltd, 1999.

[2] Ching Man; Analog Devices, Inc, "Quantization Noise: An Expanded Derivation of the Equation, SNR = 6.02 N + 1.76 dB," Norwood, MA, USA, 2012.

[3] EBU, "Tech 3341 - 'EBU Mode' metering to supplement Loudness normalisation," January 2016c. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3341.pdf.

[4] ITU, *Recommendation ITU-R BS.1770-4,* 2015.

[5] I. Dash, "True peak metering – a tutorial review," April 2014. [Online].

[6] Y. Wang and M. Vilermo, "The Modified Discrete Cosine Transform: Its Implications For Audio Coding And Error Concealment," June 2002. [Online].

[7] Y. Wang, L. Yaroslavsky, M. Vilermo and M. Vaananen, "Some Peculiar Properties of the MDCT," 2000. [Online].

[8] V. Arun Raj, M. Davidson Kamala Dhas and D. Gnanadurai, "An Overview of MDCT for Time Domain Aliasing Cancellation," in *International Conference on Communication and Network Technologies (ICCNT)*, 2014.

[9] J. P. Princen and A. B. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation," 5 October 1986. [Online].

[10] H. S. Malvar, Signal Processign with Lapped Transforms, Norwood, MA: Artech House, INC., 1992.

[11] B. Edler, "Codierung von Audiosignalen mit überlappender Transformation und adaptativen Fensterfunktionen," Universität Hannover, 7 July 1989. [Online].

[12] P. Vaidyanathan, Multirate Systems and Filter Banks, Upper Saddle River, NJ: Prentice-Hall, INC., 1993.

[13] S. K. Agrawal and O. P. Sahu, "Two-Channel Quadrature Mirror Filter Bank: An Overview," 2013. [Online].

[14] P. Ekstrand, "Efficient Implementation of the Complex Modulated Filter Bank," Coding Technologies, 2003.

[15] M. Bosi, "Filter banks in perceptual audio coding," in *AES 17th International Conference*, Los Angeles, California, USA.

[16] W. T. Padgett and D. V. Anderson, Fixed-Point Signal Processing, Morgan and Claypool, 2009.

[17] R. Rocher, D. Menard, O. Sentieys and P. Scalart, "Analytical Accuracy Evaluation of Fixed-Point Systems," *EUSIPCO,* pp. 999-1003, September 2007.

[18] V. Britanak and H. J. L. Arriens, "Fast computational structures for an efficient implementation of the complete TDAC analysis/synthesis MDCT/MDST filter banks," *Signal Processing, Elsevier,* 2009.

[19] M. Soni and P. Kunthe, "A General Comparison Of Fft Algorithms," Pioneer Journal Of IT & Management, 2011.

[20] W.-H. Chang and T. Q. Nguyen, "On the Fixed-Point Accuracy Analysis of FFT Algorithms," October 2008. [Online].

[21] ETSI, "ETSI TS 103 448 V1.1.1 (2016-09) - AC-4 Object Audio Renderer for Consumer Use," ETSI, 2016.

[22] S. H. Nielsen and T. Lund, "0dBFS+ Levels in Digital Mastering," 2000. [Online].

[23] EBU, "Tech 3343 - Practical guidelines for production and implementation in accordance with EBU R 128," August 2011. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3343.pdf.

[24] EBU, "Tech 3342 - Loudness Range measure to supplement loudness normalisation," January 2016a. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3342.pdf.

Annexes

Annex 1: List of content used for analysis of stereophonic music

Classical: 106 songs from 10 CDs

- CD list:
    1. Wolfgang Amadeus Mozart - Klaviekonzerte Nr. 20 & 24 (Martin Stadtfeld)
    2. Anne-Sophie Mutter - Beethoven Violin Concerto; Romances
    3. Christian Gerhaher - Franz Schubert Winterreise
    4. Fretwork - Purcell The Fantazias & In Nomines
    5. Günter Wand NDR Symphony Orchestra Hamburg - Bruckner Symphony #5
    6. Heinz Holliger & Friends - Britten - Mozart
    7. Johann Sebastian Bach - Oratorios BWV 249 & BWV 11 [Suzuki]
    8. Kronos Quartet - Winter Was Hard
    9. Marin Alsop - John Adams Shaker Loops; The Wound-Dresser; Short Ride in a Fast Machine
    10. Royal Concertgebouw Orchestra - Shostakovich Symphony No. 8

Jazz: 45 songs from 6 CDs

- CD list:
    1. Steve Coleman - On the Rising of the 64 Paths
    2. Andy Sheppard - Soft on the Inside
    3. Carla Bley - Fleur Carnivore
    4. Intergalactic Maiden Ballet - Intergalactic Maiden Ballet
    5. John Coltrane - Giant Steps [Deluxe Edition]
    6. SNO - Sunday Night Orchestra - Music Without Words

Rock: 176 songs from 14 CDs

- CD list:
    1. Arctic Monkeys - Whatever People Say I Am, That's What I'm Not
    2. Blink 182 - Enema Of The State
    3. Die Toten Hosen - Laune der Natur
    4. Kitchens of Distinction - Cowboys and Aliens
    5. Metallica - Death Magnetic
    6. New Model Army - Between Dog and Wolf
    7. Nirvana - Nevermind
    8. OK Kid - Zwei
    9. Radiohead - The Bends
    10. Rage Against The Machine - Evil Empire
    11. Ramones - Too Tough to Die
    12. Red Hot Chili Peppers - Californication
    13. Red Hot Chili Peppers - Mother's Milk
    14. The Cranberries - No Need To Argue

Pop: 126 songs from 11 CDs

- CD list:
    1. The Unthanks - Mount the Air
    2. ABBA - Super Trouper
    3. Austra - Feel It Break
    4. Falco - Out of the Dark (Into the Light)
    5. Goldfrapp - Silver Eye
    6. K.Flay - Every Where Is Some Where
    7. Madonna - Music
    8. Michael Wollny - Nachtfahrten
    9. Seiler & Speer - Ham Kummst
    10. Sleigh Bells - Reign of Terror
    11. The National - Alligator

Annex 2: List of content used for analysis of cinematic multichannel content

1. 2014 World Series Film
2. Akira
3. Amelie
4. Apocalypse Now Disc
5. Batman Begins
6. Battle for the Planet of the Apes
7. Beneath The Planet Of The Apes
8. Black Hawk Down
9. Conquest of the Planet of the Apes
10. Das Boot
11. Escape From The Planet Of The Apes
12. Finding Mr. Right
13. Flying Swords of Dragon Gate
14. Fury
15. House Of Flying Daggers
16. Iceage
17. In The Mood for Love
18. Inception
19. Letters From Iwo Jima
20. Master And Commander
21. Pacific Rim
22. Pulp Fiction
23. Punch Drunk Love
24. Ray
25. Reservoir Dogs
26. Saving Private Ryan
27. Spirited Away
28. Stop Making Sense
29. Superman Returns
30. The Bourne Identity
31. The Dark Knight
32. The Dark Knight Rises
33. The English Patient
34. The Lord Of The Rings The Two Towers
35. The Matrix
36. The Planet of the Apes (1968)
37. The Roxy
38. Titanic
39. Up
40. Whiplash

Annex 3: *Fast DCT type IV transform* by Per Ekstrand (Coding Technologies)

The M-point DCT type IV matrix can be computed by an half-size FFT core with pre- and post-twiddling.

The DCT type IV transform is defined

$$y(n) = \sum_{l=0}^{M-1} x(l) \cos\left\{\frac{\pi}{M}(n+\frac{1}{2})(l+\frac{1}{2})\right\}, \quad n = 0 \ldots M-1 \tag{1}$$

Split the summation into two sums with even and odd-indexed samples as

$$\begin{aligned} y(n) = & \sum_{l=0}^{M/2-1} x(2l) \cos\left\{\frac{\pi}{M}(n+\frac{1}{2})(2l+\frac{1}{2})\right\} + \\ & + \sum_{l=0}^{M/2-1} x(M-1-2l) \cos\left\{\frac{\pi}{M}(n+\frac{1}{2})(2l+\frac{1}{2}-M)\right\} \quad , n = 0 \ldots M-1 \end{aligned} \tag{2}$$

In the same way, split the output samples into an even-indexed and odd-indexed sequence as

$$\begin{aligned} y(2n) = & \sum_{l=0}^{M/2-1} x(2l) \cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\} + \\ & + \sum_{l=0}^{M/2-1} x(M-1-2l) \cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2}-M)\right\}, \quad n = 0 \ldots M/2-1 \end{aligned} \tag{3}$$

and

$$\begin{aligned} y(M-1-2n) = & \sum_{l=0}^{M/2-1} x(2l) \cos\left\{\frac{\pi}{M}(2n+\frac{1}{2}-M)(2l+\frac{1}{2})\right\}+, \quad n = 0 \ldots M/2-1 \\ & + \sum_{l=0}^{M/2-1} x(M-1-2l) \cos\left\{\frac{\pi}{M}(2n+\frac{1}{2}-M)(2l+\frac{1}{2}-M)\right\} \end{aligned} \tag{4}$$

Using ordinary trigonometric reductions, Eq. 3 and Eq. 4 equal

$$y(2n) = \sum_{l=0}^{M/2-1} x(2l)\cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\}+ \\ + \sum_{l=0}^{M/2-1} x(M-1-2l)\sin\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\}, \quad n=0\ldots M/2-1 \tag{5}$$

And

$$y(M-1-2n) = \sum_{l=0}^{M/2-1} x(2l)\sin\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\}+, \quad n=0\ldots M/2-1 \\ - \sum_{l=0}^{M/2-1} x(M-1-2l)\cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\} \tag{6}$$

If we define a new complex sequence z(n) by

$$z(l) = x(2l) + i\, x(M-1-2l), \quad l=0\ldots M/2-1 \tag{7}$$

the two output sequences (Eq. 5 and 6) can be expressed

$$y(2n) = \sum_{l=0}^{M/2-1} \mathrm{Re}\{z(l)\}\cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\}+ \\ + \sum_{l=0}^{M/2-1} \mathrm{Im}\{z(l)\}\sin\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\}= \\ = \mathrm{Re}\left\{\sum_{l=0}^{M/2-1} z(l)\,\mathrm{e}^{-i\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})}\right\}, \quad n=0\ldots M/2-1 \tag{8}$$

and

$$
\begin{aligned}
y(M-1-2n) &= \sum_{l=0}^{M/2-1} \mathrm{Re}\{z(l)\} \sin\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\} + \\
&\quad - \sum_{l=0}^{M/2-1} \mathrm{Im}\{z(l)\} \cos\left\{\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})\right\} = \\
&= -\mathrm{Im}\left\{\sum_{l=0}^{M/2-1} z(l)\, \mathrm{e}^{-i\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})}\right\}, \quad n = 0 \ldots M/2-1
\end{aligned}
\tag{9}
$$

Thus

$$
\begin{aligned}
y(2n) &= \mathrm{Re}\{Z(n)\} \\
y(M-1-2n) &= -\mathrm{Im}\{Z(n)\}
\end{aligned}, \quad n = 0 \ldots M/2-1
\tag{10}
$$

where

$$
\begin{aligned}
Z(n) &= \sum_{l=0}^{M/2-1} z(l)\, \mathrm{e}^{-i\frac{\pi}{M}(2n+\frac{1}{2})(2l+\frac{1}{2})} = \\
&= \sum_{l=0}^{M/2-1} z(l)\, \mathrm{e}^{-i\frac{\pi}{M}n}\, \mathrm{e}^{-i\frac{\pi}{M}(l+\frac{1}{4})}\, \mathrm{e}^{-i\frac{\pi}{M}4nl} = \\
&= \mathrm{e}^{-i\frac{\pi}{M}n} \sum_{l=0}^{M/2-1} \left\{ z(l)\, \mathrm{e}^{-i\frac{\pi}{M}(l+\frac{1}{4})} \right\} \mathrm{e}^{-i\frac{2\pi}{M/2}nl}, \quad n = 0 \ldots M/2-1
\end{aligned}
\tag{11}
$$

This is a M/2-point DFT of a twiddled version of the signal z(l) followed by post-twiddling. The DST type IV computation can be deduced analogously.