

---

# Structural analysis of low latency audio coding schemes

Manfred Lutzky, Markus Schnell, Markus Schmidt and Ralf Geiger

Fraunhofer Institute for Integrated Circuits IIS, Am Wolfsmantel 33, 91058 Erlangen, Germany

Correspondence should be addressed to: [Manfred.Lutzky@iis.fraunhofer.de](mailto:Manfred.Lutzky@iis.fraunhofer.de)

## ABSTRACT

Low latency audio coding gains increasing importance among upcoming high quality communication applications like video conferencing and VoIP. This paper provides a comparison of two low latency audio codecs suitable for these tasks: MPEG-4 ER AAC-LD and ITU-T G.722.1 Annex C. Despite their similar coding strategies, both codecs show significant differences with respect to used tools and coding performance. A comparison of the coding tools is provided and the influence on different signal classes is discussed.

## 1. INTRODUCTION

Current VoIP and video conferencing systems are based on the usage of no longer up-to-date telephone codecs, e.g. ITU-T G.711, and low bitrate speech codecs, e.g. ITU-T G.728. Unfortunately, these codecs do not make the most of the available transmission bandwidth. Whereas telephone codecs prove to be too expensive with respect to bandwidth, speech codecs are not able to compete in coding tasks which more and more include multimedia data or even music.

Highly sophisticated audio codecs are necessary for encoding audio data at low bit-rates and to achieve natural sound quality which helps to improve the intelligibility in difficult environments such as understanding fast and inarticulate speakers in a foreign language. Application scenarios nowadays no longer include only pure speech transmission, but also the necessity of a transmission channel for music, e.g. for a multimedia business presentation broadcast over a video conferencing system. It is therefore necessary to equip such professional applications with audio codecs that are able to process ambient sound and even music at high quality. Additionally, in order to allow for a high quality interactive communication application in which

an effective echo cancellation remains possible, codec delay should not exceed the duration of about 50 ms.

This paper will conduct a scientific comparison of two current high quality audio communication codecs of moderate bitrate demand.

The subjects of our investigations are the state-of-the-art MPEG-4 ER AAC-LD coder (in the following "AAC-LD") and the upcoming ITU-T G.722.1 Annex C coder ("G.722.1-C"), also known as Siren14, which is a 14 kHz enhancement of ITU-T G.722.1. Though these two coders are in large parts very similar, they each present specialized tools of their own.

## 2. COMPARISON OF MPEG-4 ER AAC LD AND G.722.1 ANNEX C

### 2.1. Standardization issues

Two international standardization organisations, as there are the MPEG working group of the ISO/IEC and the telecommunication standardization sector ITU-T of the ITU, developed two procedures of standardization which are of a fundamentally different nature.

The ITU-T approach is to lay down *Recommendations* which comprise very specific, mandatory design rules for both encoder and decoder in an audio coding environment. This includes step-by-step instructions concerning computational details of the encoder's operations or the requirement of bit-exact identity of performed operations. So there is in the Recommendation on G.722.1-C [4], for example, provided a section which contains a 25-step flow chart formulation dealing with the categorization procedure used in both en- and decoder, see paragraph 2.2.3.2. The introduction to said Recommendation contains the clause that an encoding/decoding system has to "produce for any input signal the same output results" [4] as the reference program code included in the Recommendation. Only then the system is fully compliant. To this end, there are included in this reference program code for example reference implementations for basic mathematical operators like rounding, multiplication or addition which formulate required computational behaviour irrespective of the underlying system architecture.

This procedure ensures a reliable audio quality of all codec implementations regardless of the vendor or the used platform.

On the other hand, there is the MPEG procedure. In a very elaborate process institutions, research labs and involved industry come up with scientific methods which compete against each other and of which the most powerful is chosen as a reference model for further development work. In the following procedure the abilities of the standardized system are further improved. The most important point hereby is that usually only the decoder behaviour is standardized; on encoder side only the bitstream format is standardized. There exists an informative reference encoder which produces a correct bitstream but normally does not produce the best possible audio quality. This procedure enables all involved parties (and all future developers working on technical solutions based on MPEG standards) to further improve the performance of the encoding-decoding chain even after the standardization procedure has been completed, as long as the generated bitstreams adhere to these standards.

The compliance of audio decoding systems to an MPEG standard is ensured by conformance testing procedures [2]. MPEG provides a set of bitstreams, a reference decoder and conformance testing tools that compare the output of the reference decoder with the output of the decoder under test to affirm the compliance to all conformance criteria. Developers of hard- and software can thus verify the standard compatibility of their audio coding products. Bit exactness to the standard is not mandatory to be able to use special features of dedicated hardware (e.g. special rounding behaviour) in a native and effective way. This allows easy implementation on different hardware architectures without the burden of simulating a specific architecture on which the standard has been based.

This MPEG method not only encourages continuing technical development, but also enables all involved parties to build systems with exclusive knowledge and abilities and by doing that generate a by using and enhancing freely available and standardized technologies

### 2.2. Coding mechanisms

The coding workflow of both coders can be described in a similar way. All necessary modules at the encoder side can be identified as follows:

- Time – Frequency mapping
- Separation into frequency bands
- Psychoacoustic
- Quantization
- Noiseless coding

Each module is further discussed in the following paragraphs.

The block diagrams of both coders are shown in the Annex to this document. Figure 2 shows an AAC encoder from which, with some modifications, see [7], AAC-LD is derived; Figure 3 shows the structure of G.722.1-C.

### 2.2.1. Time-Frequency mapping

Both coders are block-based and use a real valued analysis filter bank on encoder side to convert the time domain input samples into a frequency domain representation. On decoder side, the frequency domain values are mapped back into time domain by a synthesis filter bank.

This includes a technique called *time domain aliasing cancellation* (TDAC) to guarantee the *perfect reconstruction* property. Each transform block is processed with an overlap of fifty percent with both previous and following one. It is then weighted by a window function. In order to ensure perfect reconstruction two constraints of the window function must be fulfilled [17], i.e. for TDAC:

- the sum of the weighted signal components must result in the input signal
- the window function has to be symmetric

Although the filter banks bear different labels, both the MDCT used by AAC-LD and the MLT implemented in G.722.1-C refer to the same cosine modulated filter bank [15]. Malvar defined, in [16], the MLT strictly connected to the use of a sine function as analysis and synthesis window. The MDCT has no restrictions concerning the window functions, except the TDAC property.

Admittedly, the coders utilize different implementations of the filter bank, including different lengths. AAC-LD employs a transform length of 512 and 480 samples respectively, while G.722.1-C uses 640. The transform window functions are identical when processing stationary signals. AAC-LD provides, apart from the sine window for the stationary mode, a specialized window for the non-stationary mode, called *Low Overlap Window*, which will be further discussed in paragraph 2.2.3.1.

The length of the transforms influences the algorithmic delay of a coder, see section 2.4.1, as well as the frequency resolution. A good frequency selectivity of the filter bank is desirable in order to be able to resolve complex harmonic spectral data for improving the coding gain [15].

### 2.2.2. Separation into frequency bands

While, in the following encoding process, both coders split the spectrum into bands, they do so in a different way. G.722.1-C divides the frequency domain into 28 equidistant bands. Each band contains a 500 Hz segment. In contrast to this segmentation, AAC-LD uses a method modeled closely to the nature of the human perceptual system. As explained in [21], a frequency-to-place transform takes place in the inner ear which can be interpreted, from a signal-processing view, as a bank of highly overlapping bandpass filters with increasing bandwidths towards higher frequencies. This classification into frequency groups is called *critical band rate scale*. AAC-LD uses a band allocation very close to the critical band rate scale. Further information about critical bands and the effectiveness for audio coding applications can be found for example in [15].

### 2.2.3. Psychoacoustic model

The aim of the psychoacoustic module is to increase the coding efficiency by exploiting the fact that "irrelevant" signal information is unnoticeable even for very sensitive listeners. Highly sophisticated mathematical models of the human auditory system therefore control the quantization of the input signal and distribute the quantization noise with respect to spectral and temporal masking effects.

Even though G.722.1-C does not calculate an exhaustive psychoacoustic model, there are several

mechanisms included in the coder which are able to exploit some psychoacoustic effects.

### 2.2.3.1. AAC-LD

First of all, masking thresholds are calculated in the frequency domain in order to estimate the necessary *signal-to-noise ratio* (SNR) to avoid noticeable distortions. These thresholds are not only determined inside each band, but also inter-band masking effects are estimated by spreading functions, as for example described in [15]. Unnoticeable bands can thus be detected and do not have to be coded at all. This SNR information is sent to the quantization module where suitable quantization step sizes are calculated in order to comply with the masking criteria.

Figure 4 illustrates the masking curve formed by the human hearing threshold in combination with three narrow-band noises [21]. The green bars indicate masked (right) and not masked (left) sound events.

Furthermore, AAC-LD features additional psychoacoustic tools for some dedicated audio scenarios:

- **Temporal Noise Shaping (TNS):** Basically, the TNS tool is an open-loop predictor operating in the frequency domain. Due to the interdependency of the *power spectral density* (PSD) with the squared Hilbert envelope, as described in [5], a prediction over spectral data does not only adapt the quantization error to the signal's PSD, but also to its temporal envelope. The combination of filter bank and prediction filter can also be interpreted as a continuously adaptive filter bank [6]. For dedicated signals with highly correlated spectral coefficients, the frequency resolution is decreased as a result of the combination (convolution) of these coefficients to calculate the prediction residual. The frequency and time resolution is therefore adapted to the characteristics of the input signal. As mentioned in [5] and [6], the TDAC property is affected by a prediction operation on the spectral data and therefore a special transform window was introduced in [7], the Low Overlap Window. The temporal aliasing artifacts, resulting from the disturbed TDAC,

are minimized by the lower overlap of both analysis and synthesis window.

- **Perceptual Noise Substitution (PNS):** The fact that "one noise sounds like the other" [11] provides an opportunity to represent noise-like bands in a very bit saving, parametric way. Noise-like bands are detected on encoder side, and only their energy level has to be transmitted to the decoder, where these bands are reconstructed by filling them with a signal constructed by a random noise generator.
- **Long Term Prediction (LTP):** Introduced in [14], the LTP is a forward-adaptive prediction tool working in time domain. Only the residual error signal in the frequency domain is further encoded. The LTP only affects those frequency bands in which predictable signals are detected.
- **Bit reservoir:** The bit demand is not equal for every type of signal. The use of a bit reservoir therefore provides the possibility to spend more bits on critical signal parts and save bits, in case of encoding non-critical input. However, the size of the bit reservoir does influence overall algorithmic delay which will be further discussed in section 2.4.1.
- **Mid-Side Stereo (M/S):** The M/S tool increases the coding gain for encoding a stereo channel pair compared to encoding two mono channels separately [12]. The left and right channel are mapped to a sum and difference representation which is a completely perfect reconstructable linear transform. That way, highly correlated stereo signals can be compacted into one "strong" mid channel and a "weak" side channel, codeable with small bit demand. Another advantage of using M/S is the possibility of controlling the quantization noise in the stereo panorama, see paragraph 2.3.

### 2.2.3.2. G.722.1-C

Even though G.722.1-C does not calculate a psychoacoustic model, it exploits some kind of in-band self-masking effects. The MLT coefficients are normalized by their *root mean square* (RMS) energy values of the corresponding band [4]. This means that

one part of the quantizer for each band is provided by its RMS value. That way, a higher level of quantization noise is added to bands containing higher energy.

The second part of the quantizer is determined by a fixed algorithm which also depends on the RMS values of each band. In order to find the optimal set of quantizers that are fulfilling a constant bit-rate constraint as exactly as possible, a *categorization procedure* is carried out; *category* denotes in this context a set of defined quantization and coding parameters. 32 sets of categories are determined. In each set only one band's category differs. Using a fixed bit demand estimation table, the algorithm is able to spend spare bits at lower frequency bands or to save bits at higher frequencies.

Furthermore, a noise filling mechanism is used on decoder side, which becomes effective for several dedicated categories. If one of these categories is chosen for a band, every MLT coefficient of value zero is replaced by a random value which takes into account the band's RMS value. That way, the spectrum is filled in order to avoid holes.

#### 2.2.4. Quantization

AAC-LD uses a non-uniform quantizer. Its advantage is a built-in noise shaping functionality which depends on the amplitude of the spectral coefficients. The increase of the signal-to-noise ratio with increasing signal energy is much lower than that of a linear quantizer [1]. Additionally, quantizer step sizes are used to distribute the quantization noise over the whole spectrum in an optimal way. The complete quantization is described in [1] as follows:

$$ix(i) = \text{sign}(xr(i)) \cdot n \text{int} \left( \left( \frac{|xr(i)|}{\sqrt[4]{2}^{\text{quant\_stepsize}}} \right)^{0.75} - 0.0946 \right),$$

where  $ix(i)$  is the quantized spectral line,  $xr(i)$  the unrounded spectral coefficient and the operation *nint* denotes 'rounding to nearest integer'. The quantizer can be changed in steps of 1.5 dB. The most suitable value for each band is estimated by the psychoacoustic model.

Inside G.722.1-C the RMS energy values are quantized using a log domain metric,  $2^{\binom{i+2}{2}}$ , where  $i$  is the *rms\_index* [4]. These quantized RMS values

constitute the first part of the quantizer. The second part, the quantization step size, is directly linked to the chosen category of each band which is constructed, as mentioned in section 2.2.3.2, by a fixed algorithm depending on the *rms\_indices* and a bit estimation table. The complete quantizer can be written as follows:

$$ix(i) = n \text{int} \left( \frac{\text{abs}(xr(i))}{\text{quant\_stepsize}(r) \cdot \text{quant\_rms}(r)} \right) - DR,$$

where  $r$  refers to the band index and  $DR$  denotes deadzone rounding which depends on the used category.

#### 2.2.5. Noiseless coding

AAC-LD transmits the differentially coded scale factor data utilizing Huffman codes. There exist eleven Huffman codebooks to represent the quantized spectral data. Each band can be coded with a different codebook. To minimize the side information needed to signalize a codebook, a sectioning mechanism is introduced. Several adjacent bands using the same codebook are combined into one section. A greedy-merge algorithm [3] is able to find the minimum in bit demand of the nearly uncountable variations of codebook distributions. After that, all necessary side information concerning the used psychoacoustic tools is written to the bitstream.

Inside G.722.1-C, the quantized RMS values are differentially and Huffman coded. The one chosen set of the 32 calculated sets is written to the bitstream in a 5-bit representation. This is sufficient, because all 32 sets can completely be reconstructed on decoder side from the RMS values and therefore only the index of the used set has to be transmitted. The quantized spectral data is Huffman coded using several codebooks, each linked to one category. The codebooks cannot be chosen independently, although another codebook might represent the spectral data with fewer bits.

The larger flexibility in choosing quantization step sizes for a better control of quantization noise inside AAC-LD unfortunately comes with a higher bit demand necessary for the signaling of the used quantization step sizes for each band.

### 2.3. Influence on signal classes

While audio signals can be divided into the classes speech, music and ambience, real world signals often

represent hybrids of these classifications. They comprise tonal, transient and noiselike signal characteristics. It is therefore both necessary and beneficial to equip an audio codec with a number of separate coding tools which respectively excel at the mentioned signal characteristics.

In this respect, AAC-LD provides several different tools which carry out highly specific tasks.

For transient or pitched signal parts there is TNS and the bit reservoir. TNS designs the introduced noise in accordance with the temporal shape of the original signal and is especially helpful in reducing pre-echo artifacts which occur at the onsets of attacks in an audio signal [5]. A plain example is the castanets with which coding transform based audio coders often have great difficulties. The introduced noise is smeared over such large portions of time, see Figure 6, that it becomes audible just before the onset of the castanets' clap. As an illustration of this effect see Figure 7 and Figure 8. Figure 7 shows the plot of the original castanets' clap, Figure 8 illustrates the resulting noise distribution with and without the use of TNS. If we look at Figure 5 which illustrates the process of temporal masking (with the green sound event on the right being masked, the one on the left being perceptible), we see why the smeared noise becomes audible. As it can be seen in Figure 4, pre-masking reaches, compared to post-masking, over a relatively short time-span of only a few milliseconds. If an analysis block, over which the introduced noise is distributed due to the use of a block transform, is however as long as the exemplary 25 ms, the above effects occur. The effectiveness of the TNS module can also be seen in the results of the listening test (Figure 1) in which AAC-LD performed profoundly better than G.722.1-C in coding the test item *si02* (castanets).

The bit reservoir technique offers an additional safety net for signal parts of peak bit demand. Bits which are saved while coding parts in the signal with low bit demand, can then be used coding critical parts (e.g. especially transient signal parts) while not violating any restraints of a constant rate audio coder.

Noiselike components are dealt with by the PNS tool. It achieves a coding gain using a parametric representation of these signal parts, see paragraph 2.2.3.1.

The inherent redundancy of tonal signal parts is exploited by the LTP tool, see paragraph 2.2.3.1. It aims

at stationary segments of the signal and works as an inter-frame prediction tool. The use of a non-uniform quantizer, see 2.2.4, also shows its advantageous potential, as it offers an improved noise distribution in comparison to a uniform quantizer dealing with this class of signal.

Additionally, AAC-LD is equipped with tools that deal with stereo signals, as the M/S algorithm provides both inter-channel redundancy reduction and noise shaping in the stereo panorama. Illustrative examples for these effects can be found on a tutorial CD-ROM [20]. There, in the chapter 'BMLD', it is shown how the phase relationship of masker and maskee influences the masking process and how noise in the stereo panorama can become 'unmasked' (i.e. audible) due to shifts in this phase relationship. This effect is also referred to as 'stereo unmasking'. As mentioned in paragraph 2.2.3.1, this effect can be prevented by the deliberate placing of the coding noise in the stereo panorama using M/S.

## 2.4. Delay and other parameters

### 2.4.1. Algorithmic delay determination

As explained in [7], the delay for transform based audio coders, like AAC-LD or G.722.1-C, results from the following factors:

- Framing: Due to the use of a block transform, a certain amount of time is needed to collect all samples belonging to one block.
- Filter bank: Due to the overlap-add operation of the filter bank with a 50 % overlap to previous and subsequent blocks, a delay of one frame is caused by the filter bank.

Each of the above operations produces a delay equaling the frame length. This results in a delay of 40 ms for G.722.1-C.

$$T = \frac{2 \cdot \text{frame\_size}}{\text{sampling\_rate}} = \frac{2 \cdot 640 \text{ samples}}{32000 \frac{\text{samples}}{\text{sec}}} = 40 \text{ ms}$$

G.722.1-C has a fixed algorithmic delay, whereas AAC-LD proves to be more flexible. AAC-LD supports different sampling rates (22.05, 24, 32, 44.1, 48 kHz)

and different frame sizes (480, 512 samples); thus, the coder's algorithmic delay ranges from 20 to 46.44 ms.

The above delay calculation only holds true if either no bit reservoir is used or the output bitstream is transmitted via a packet based transmission line, e.g. TCP/IP. If a continuous transmission is used, e.g. ISDN, the size of the bit reservoir has to be included in the calculation [22]. The additional delay in samples can be described as follows:

$$T_{bitres} = \frac{\text{size\_of\_bitres}}{\text{bitrate}}$$

The AAC-LD standard does only define a maximum bit reservoir which allows to reduce the actually used bit reservoir of constant rate systems to any value down to zero and with it make the bit reservoir delay insignificant.

#### 2.4.2. Application delay

As detailed in [19], the delay of a real-time implementation results from the algorithmic delay and real-time specific restrictions of limited calculation speed and necessary buffering. All these variables have to be taken into account in design processes for communication applications using audio coding.

#### 2.4.3. Bit rate, channel configuration

G.722.1-C provides three bit rate modes, 24, 32, 48 kbps, whereas any bitrate in the range from 12 to above 160 kbps/channel can be chosen for AAC-LD.

AAC-LD also provides a more flexible channel configuration. It is able to handle a single mono or a stereo channel pair, as well as a 5.1 channel set. In contrast, G.722.1-C supports one mono channel only. In case of encoding stereo data, the coder has to handle the channel pair as two mono channels, without any consideration of dependencies between the channels, see 2.2.3.1.

### 3. ERROR ROBUSTNESS

In real world applications the audio performance of the coding algorithm is only one parameter that influences the performance of the whole system. Due to bit errors,

loss of whole bitstream frames and late arrival of bitstream packages, the error robustness capabilities of the coding schemes becomes an important performance characteristic.

Four different measures can be combined to achieve adequate error robustness:

**Error Detection (ED):** allows to detect errors

**Error Concealment (EC):** synthesizes lost parts of the audio signal

**Error Protection (EP):** allows to recover corrupted data

**Error Resilience (ER):** makes the source coding algorithm more robust against transmission errors

The following table provides a compact comparison of the error robustness mechanisms:

	MPEG 4 ER AAC-LD	G.722.1-C
ED	- could be handled outside - exploitation of ER tools allow localisation of transmission errors	outside of codec
EC	not standardized; several frame concealment algorithms have been developed [8]; ER tools allow line concealment	frame repetition standardized
EP	unequal error protection adds protection to sensitive parts of the bitstream and avoids protection overhead compared to equal error protection	-
ER	Virtual Codebooks (VCB11) detect serious errors within spectral data [8]; Huffman Codeword Reordering (HCR) avoids error propagation within spectral data [8]; Reversible Variable Length Codes (RVLC) avoid error propagation within scale factor data [9]	-

### 4. SUBJECTIVE LISTENING TEST

In order to assess the performance of AAC-LD in comparison to G.722.1-C, an ITU-R BS.1534 MUSHRA (Multi Stimulus test with Hidden Reference

and Anchors) listening test was carried out. One main intention of this test was the evaluation of how important the aforementioned coding tools of AAC-LD are for producing adequately sounding coding results.

#### 4.1. MUSHRA setup

The items of the listening test include different types of signal classes, as discussed in section 2.3. In addition to these critical signals also speech and pop music have been included in this test. A listing of the used items can be found in Table 1. Two low pass filtered reference anchors (3.5 kHz, 7 kHz) were added to the test data pool. An mp3 codec has also been added to the test to provide a further anchor. Please note that the used 32kbps/ch is below the recommended bitrate range for the mp3 codec.

A group of 12 experienced listeners were asked to assess the test items on a subjective scale ranging from 'excellent' to 'bad'. Each test person was evaluating the test data on his own inside a dedicated listening room. The audio signals were presented to the listeners via Stax Lambda Pro headphones.

#### 4.2. Coder settings

In order to produce comparable results, both coders were run with the same algorithmic delay. Therefore, AAC-LD encoded the data using a frame length of 480 samples at a sampling rate of 24 kHz, resulting in a delay of 40 ms. Assuming a transmission of the encoded data via a packet based transport system, the use of a bit reservoir of slightly decreased size (2000 bits) was enabled for AAC-LD. G.722.1-C was working at its given sampling rate of 32 kHz. A constant bit rate of 32 kbps was chosen for both coders.

#### 4.3. Results and discussion

A plot of the MUSHRA test result can be found in Figure 1. AAC-LD clearly outperforms G.722.1-C for the critical test items tonal and transient signals si01, si02, and si03. For mixed signal classes, e.g. music, both coders produce comparable results. G.722.1-C excelled at one speech item, "Exp2d", which was also used for an ITU standardization test in which G.722.1-C was compared with an older version of AAC-LD. This item consists of reverberant speech which was mixed with interference and noise. The outperformance of

G.722.1-C over AAC-LD for this one item stems from the masking effect of the added noise over the coding artifacts. Over all items AAC-LD shows a better performance compared to G.722.1-C as the 95% confidence intervals do not overlap.

## 5. CONCLUSIONS

This paper presented an analysis and evaluation of two high quality, low latency audio coding schemes, AAC-LD and G.722.1-C by comparing the architecture and the available coding tools of the codecs and discussing the effects on different audio signal classes. It turned out that AAC-LD provides specialized tools and coding techniques for each signal class and should outperform G.722.1-C. A MUSHRA listening test verified these theoretical considerations. AAC-LD can be configured more flexibly in terms of bitrate, algorithmic delay as well as supported channel configurations.

The MPEG-4 framework in which AAC-LD resides, provides a powerful instrument of error handling, guaranteeing stability in a potentially very error prone transmission environment.

## 6. ACKNOWLEDGEMENTS

The authors wish to extend great many thanks to the colleagues at Fraunhofer IIS who helped in the writing of this paper by discussion and valuable advice. Special gratitude is due to all test listeners.

## 7. REFERENCES

- [1] ISO/IEC JTC1/SC29/WG11 (MPEG), International Standard ISO/IEC 14496-3: "Coding of audio-visual objects: Audio"
- [2] ISO/IEC JTC1/SC29/WG11 (MPEG), International Standard ISO/IEC 14496-4: "Coding of audio-visual objects: Conformance testing"
- [3] M. Bosi, K. Brandenburg, S. Quakenbush, "ISO/IEC MPEG-2 Advance Audio Coding", 101<sup>st</sup> AES Convention, November 1996, Los Angeles, California, USA, preprint 4382



- [4] ITU-T Recommendation G.722.1 (2005): "Low-complexity coding at 24 and 32 kbits/s for hands-free operation in systems with low frame loss"
- [5] J. Herre, J.D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)", 101<sup>st</sup> AES Convention, November 1996, Los Angeles, California, USA, preprint 4384
- [6] J. Herre, "Temporal noise shaping, quantization and coding methods in perceptual audio coding: A tutorial introduction", AES 17<sup>th</sup> International Conference on High Quality Audio Coding, September 1999, Florence, Italy
- [7] E. Allamanche, R. Geiger, J. Herre, T. Sporer, "MPEG-4 Low Delay Coding based on the AAC Codec", 106<sup>th</sup> AES Convention, May 1999, Munich, Germany
- [8] P. Lauber, R. Sperschneider, "Error Concealment for Compressed Digital Audio", 111<sup>th</sup> AES Convention, September 2001, New York, USA
- [9] R. Sperschneider, "Error Resilient Source Coding with Variable Length Codes and Its Application to MPEG Advanced Audio Coding", 109<sup>th</sup> AES Convention, preprint 5271, September 2000, Los Angeles, California, USA
- [10] R. Sperschneider, D. Homm, L. Chambat, "Error Resilient Source Coding with Differential Variable Length Codes and its Application to MPEG Advanced Audio Coding", 112<sup>th</sup> AES Convention, May 2002, Munich, Germany
- [11] J. Herre, D. Schulz, "Extending the MPEG-4 AAC Codec by Perceptual Noise Substitution", 104<sup>th</sup> AES Convention, May 1998, Amsterdam, Netherlands
- [12] J. D. Johnston, J. Herre, "MPEG-2 NBC Audio-Stereo and Multichannel Coding Methods", 101<sup>st</sup> AES Convention, November 1996, Los Angeles, California, USA
- [13] J.F. Johnston, A.J. Ferreira, "Sum-Difference Transform Coding", IEEE Proc. ICASSP, pp. 569 - 572, March 1992
- [14] J. Ojanperä, M. Väänänen, L. Yin, "Long Term Prediction for Transform Domain Perceptual Audio Coding", 107<sup>th</sup> AES Convention, September 1999, New York, USA
- [15] T. Painter, A. Spanias, "Perceptual Coding of Digital Audio", Proceedings of the IEEE, Vol. 88, No. 4, April 2000, pp. 451-512
- [16] H. Malvar, "Lapped transforms for efficient transform/subband coding", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, pp. 969-978, June 1990
- [17] J.P. Princen, A.B. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 34, 1986, pp. 1153 - 1161
- [18] B. Edler, "Codierung von Audiosignalen mit überlappender Transformation und adaptiven Fensterfunktionen", Frequenz, Vol. 43, 1989, pp. 252-256
- [19] J. Hilpert et al., "Real-Time Implementation of the MPEG-4 Low Delay Advanced Audio Coding Algorithm (AAC-LD) on Motorola DSP56300", 108<sup>th</sup> AES Convention, February 2000, Paris, France
- [20] M. Erne et al., "Perceptual Audio Coders 'What to listen for'", Tutorial CD-ROM and preprint 5489 at 111<sup>th</sup> AES Convention, September 2001, New York, NY, USA
- [21] E. Zwicker, H. Fastl, "Psychoacoustics – Facts and Models", Springer, Berlin, 1990
- [22] M. Lutzky et al., "A guideline to audio codec delay", 116<sup>th</sup> AES Convention, May 2004, Berlin, Germany
- [23] ITU-T, Study group 16, TD 13 (WP 3/16), "Qualification Listening and Processing Test Plan of the 14kHz Low-Complexity Audio Coding Algorithm at 24, 32 and 48 kbps Extension to ITU-T G.722.1", November 2004, Geneva, Switzerland

8. ANNEX

12 subjects - 2005-07-21

Average and 95% Confidence Intervals

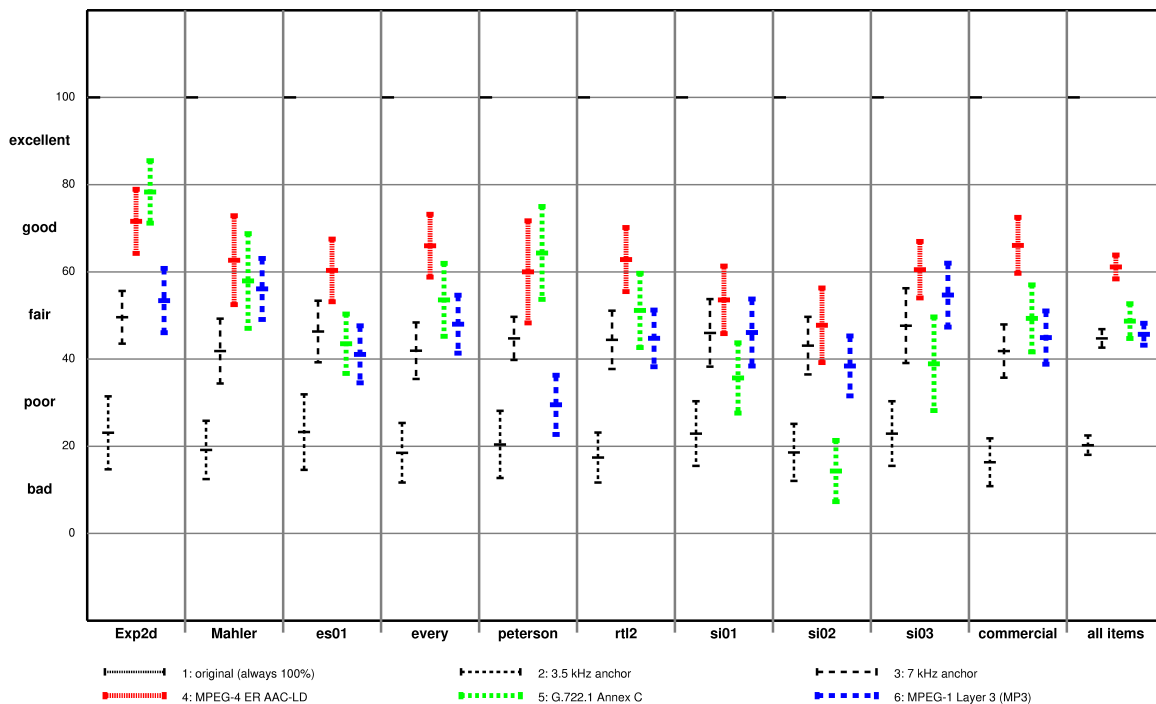


Figure 1: Result of MUSHRA listening test, 32 kbps mono

<b>Speech</b>	
Exp2d	Test item used for standardization of G.722.1 Annex C: reverberant speech + office noise + interference (Experiment 2d)
es01	Suzanne Vega, single singing voice
rtl2	radio news in French, with background music
commercial	Radio commercial spot: speaker + background music
<b>Music</b>	
every	Everything but the girl: "Missing", modern pop
peterson	Jazz music, Oscar Peterson
Mahler	classical music, Gustav Mahler
<b>MPEG items (single instruments)</b>	
si01	Harpichord, very complex harmonic spectrum with sharp attacks
si02	Castanets, very transient signal, sharp temporal attacks
si03	Pitch pipe, stationary and tonal

Table 1: Test items used in MUSHRA listening test

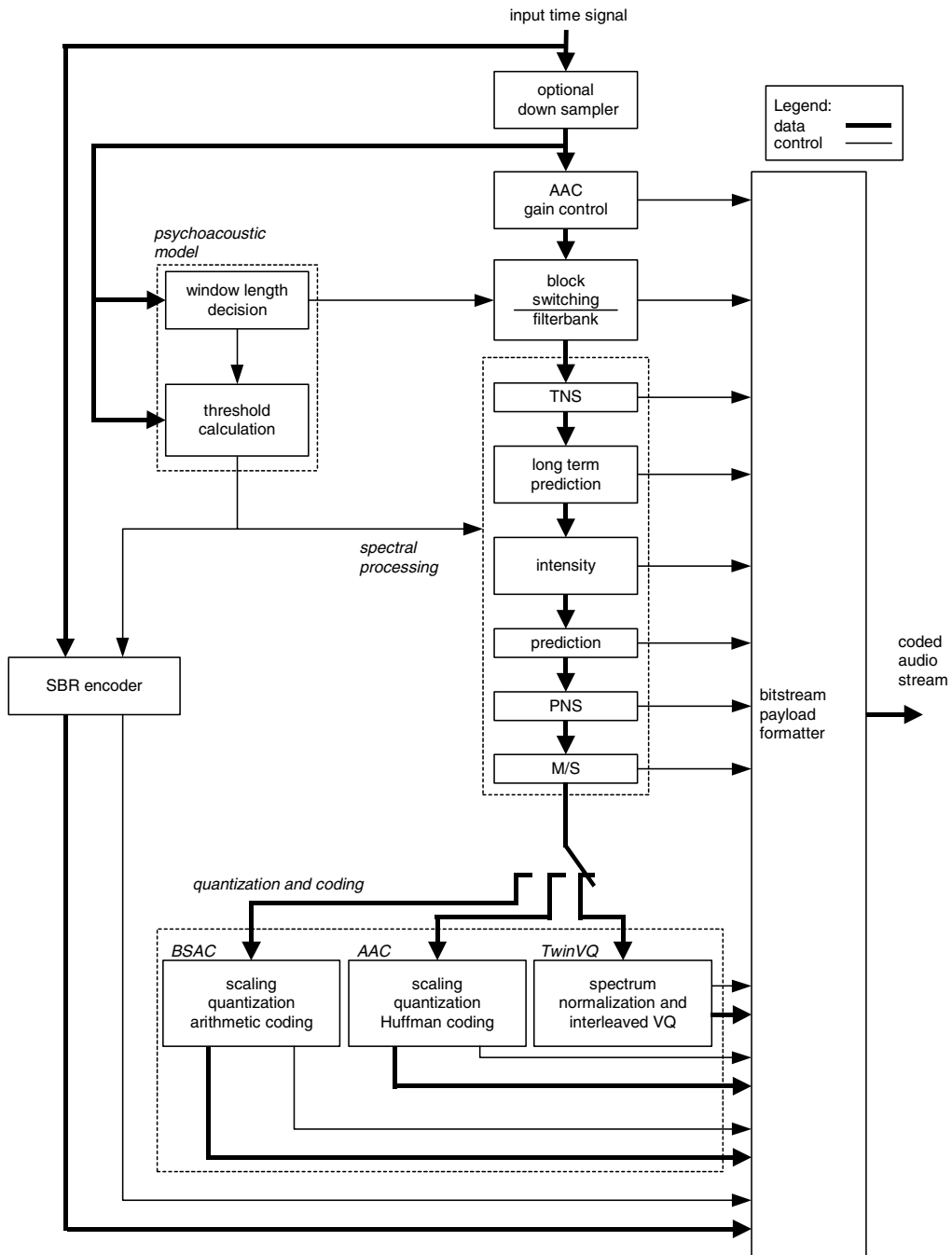
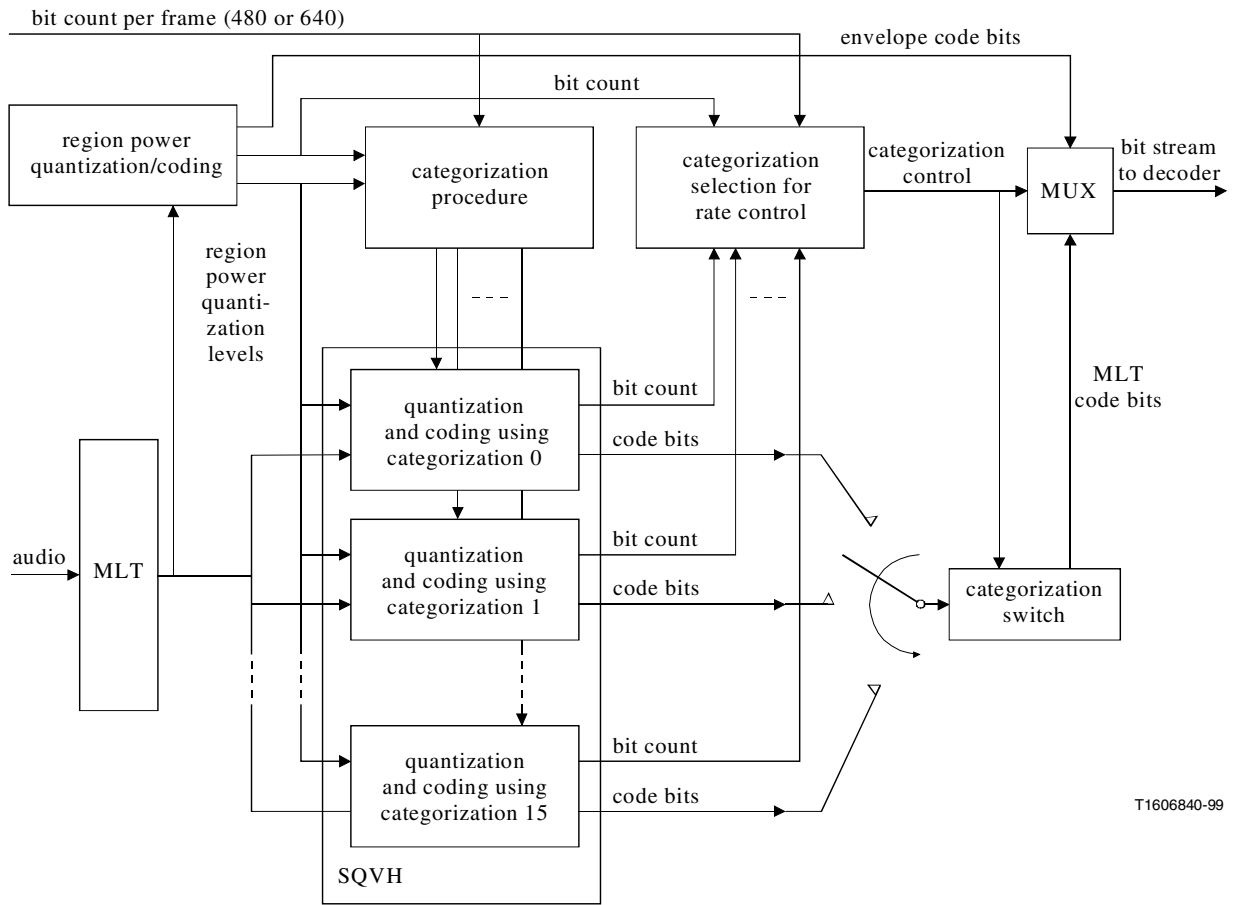


Figure 2: Block diagram encoder MPEG-4 AAC [1]



T1606840-99

Figure 3: Block diagram encoder ITU-T G.722.1 [4]

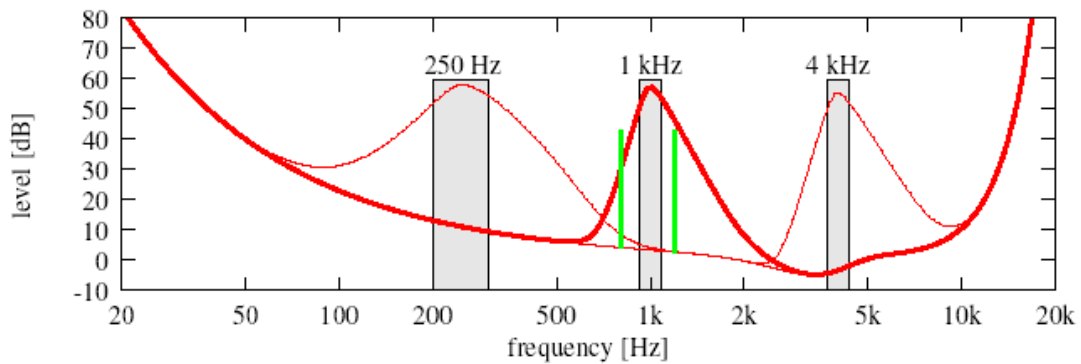


Figure 4: Frequency masking [21]

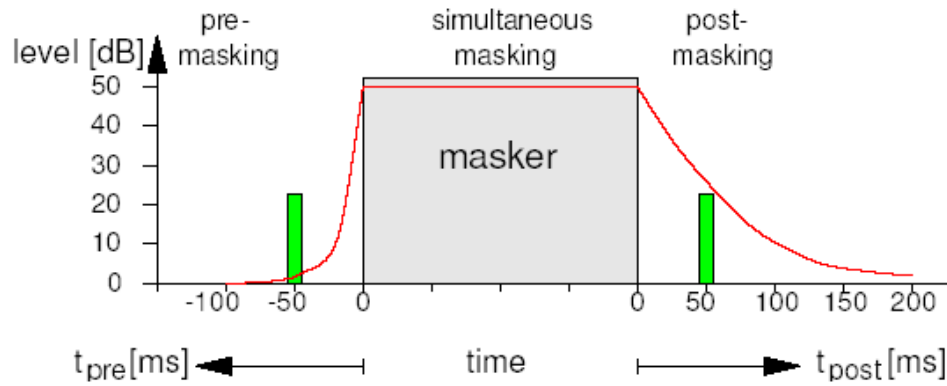


Figure 5: Temporal masking [21]

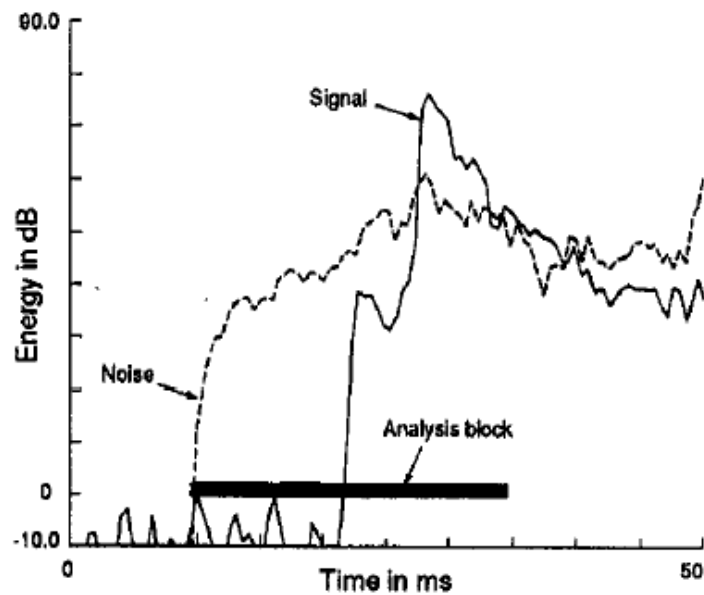


Figure 6: Problem of block-based transform coder with transient signal [5]

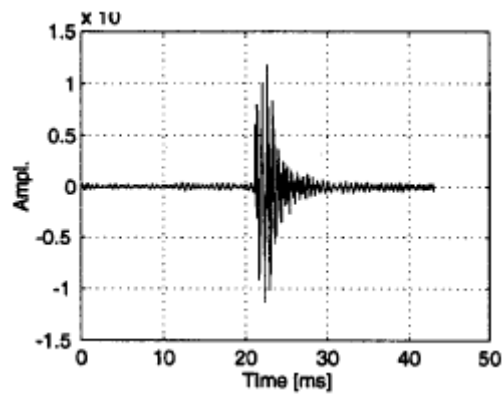


Figure 7: The castanets as an example of a transient signal, original [6]

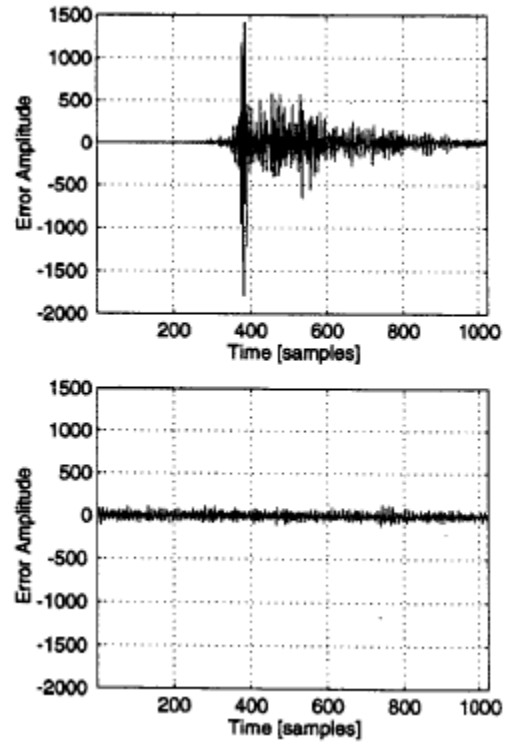


Figure 8: Coding noise of transient signal with (top) and without (bottom) TNS [6]