Delayless mixing – on the benefits of MPEG-4 AAC-ELD in high quality communication systems

Markus Schnell[1], Markus Schmidt[1], Per Ekstrand[2], Tobias Albert[1], Daniel Przioda[1], Manfred Lutzky[1], Ralf Geiger[1], Vesa Ruoppila[3], Fredrik Henn[2] and Erlend Tårnes[4]

[1]*Fraunhofer IIS, Erlangen, Germany*

[2]*Dolby Sweden, Stockholm, Sweden*

[3]*Dolby Germany, Nuremberg, Germany*

[4]*TANDBERG, Oslo, Norway*

Correspondence should be addressed to Markus Schnell (`Markus.Schnell@iis.fraunhofer.de`)

ABSTRACT

Tele- and video conferencing systems for modern business communication are managed by central hubs, so-called multipoint control units (MCU). One major task of these units is the mixing of audio streams from the participating sites. This is traditionally done by decoding the streams, mixing in time domain and then re-encoding of the mixed signals. This requires additional processing power, leads to increased delay and degraded audio quality. The paper demonstrates how the recently standardized MPEG-4 Enhanced Low Delay AAC (AAC-ELD) codec offers a solution to these problems by efficient and delayless mixing in the transform domain of the codec.

1 INTRODUCTION

In the context of a globalized environment, the demands on modern business communication systems keep increasing. Economical aspects and also increased environmental awareness predict a growing use of telecommunication systems for multipoint conferences, including both dedicated tele/video-conferencing systems and VoIP systems.

At the center of a communication system there is often a so-called MCU (multipoint control unit), a device coordinating the data transfer between the various participants in a conference call. As the MCU's may process

a large number of simultaneous calls, the computational workload in these devices is high. The MCU also introduces additional delay in the communication paths and the risk of degraded audio quality. All this is due to the occurring decoding and re-encoding, a process generally referred to as tandem coding or transcoding. There are several approaches to quantify the quality degradation caused by the cascading of coding steps. One of these approaches is the E-Model [1]. For wide-band codecs, an extension of the E-Model has been introduced in [2].

The reason for the above-mentioned disadvantages in traditional systems is the fact that the processing (mixing) of the audio streams takes place in the time domain. As a solution to this problem, this paper demonstrates the process of mixing in the transform domain. By means of the recently standardized MPEG-4 Enhanced Low Delay AAC (AAC-ELD) codec, it is shown how multi-point communication can be managed effectively: mixing audio content without adding delay, keeping complexity at a reasonable level while still maintaining the audio quality.

2 MODERN COMMUNICATION SYSTEMS

2.1 Communication codecs

The audio streams in modern communication systems, which adhere to standardized processes, are usually encoded using communication codecs such as MPEG-4 AAC-LD or a member of the ITU-T codec family. The used codec is agreed upon by, for example, the Session Initialization Protocol (SIP). In VoIP systems the G.729 and G.723.1 codecs are widely used, while AAC-LD and variants of the G.722.1 codecs are the most common in the video conferencing industry. The codecs have different properties like bit rates, complexity, audio bandwidth and audio quality, but common is the introduction of some amount of delay (see Table 1). Since the audio is usually processed in blocks, the amount of data that must be buffered before the processing can start introduces a framing delay(F). In addition, there is usually a look-ahead(L) or transform overlap delay(T) inherent in the codec. Some codecs also make use of a post processor(P) to conduct, for instance, a parametric bandwidth extension which might add further delay. The sum of these delay sources is called the algorithmic delay. With a full transcoding in the MCU, all the delay sources, except for the framing delay, are doubled in the communication chain.

MPEG-4 AAC-ELD excels in two operational areas:

- Low Delay: At a bit rate between 48-64 kbps the performance is comparable to the MPEG-4 AAC-LD codec but the delay is decreased further by 25%.
- Low Bit rate: Here, the delay of the MPEG-4 AAC-ELD is slightly increased but good audio quality is guaranteed even for low bit rates of 48 kbps down to 24 kbps.

2.2 Multi-point control units

High-quality interactive communication with three or more parties using an audio-only or an audio-visual connection (audio or audio-visual teleconferencing) requires a central communication hub, also known as *Multi-point control unit* (MCU). This unit sends to each participant downstream audio content which it creates by combining the audio streams from the other participants. This is shown in Figure 1.

Thus, the central functionality of the MCU is the combination or mixing of two or more, potentially low bit rate coded, streams into a single output stream. Apart from good audio quality, the two main requirements to this process are low processing delay and low computational complexity.

Usually, the mixing algorithm inside an MCU operates in a straightforward manner by decoding the received audio streams, creating each individual mix of all active streams in the time domain, followed by the encoding of the mixes for transmission to the corresponding participants. This cascading of encoding and decoding steps, i.e. tandem coding, increases algorithmic delay, degrades audio quality and causes high complexity.

In order to reduce the computational workload for the mixing operation itself, a algorithm also referred to as *mix minus mixer* is often implemented. The first step is to create a sum of all N mixer participants, requiring $N-1$ vector sums. Then, for each output, the corresponding input is subtracted to create the individual streams. The total number of vector sums is then $N-1+N$. The direct approach requires $N-2$ vector sums for each participant yielding a total of $N(N-2)$ vector sums meaning the break-even point is reached for an N of 4.

In a multipoint connection with N participants, N decoders have to run in parallel. The values for N can reach from 16 up to a maximum of 360. For the latter

codec	bitrate [kbps]	delay [ms]	audio bandwidth
AAC-LD	48-64	20 (10F+10T)	superwideband
AAC-ELD	**24-48**	**31.3 (2(10F+5T)+1.3P)**	**superwideband**
	48-64	**15 (10F+5T)**	
G.722	48, 56, 64	0.125 (L)	wideband
G.722.1 Annex C	24, 32, 48	40 (20F+20T)	superwideband
G.723.1	5.3, 6.3	37.5 (30F+7.5L)	narrowband
G.729 Annex A	8	15 (10F+5L)	narrowband

Table 1: Bit rates and delay values of communication codecs

value even a linear increase of the above-mentioned complexity is a challenging task. The number of encoders running in parallel is usually lower due the fact that the maximum number of active channels is limited to a reasonable value M. In consequence, the following operations are needed: N decoders $+(2M-1)$ vector sums $+(M+1)$ encoders. Nevertheless, M might be increased significantly if each participant requests a bitstream with special parameters, e.g. bit rate. Thus, it is very important that the complexity is kept as low as possible.

2.2.1 Jitter buffer

Today's network delay is not a stable parameter. To pay attention to this fact, a jitter buffer has to be included in MCU devices. This buffer potentially causes delay. Packets sent over IP networks should ideally arrive at equally spaced time intervals, but in real networks the packet transmission delay varies. Jitter is the variation of the packet arrival time at the receiver and therefore a jitter buffer is introduced. Packets arriving later than the jitter buffer delay are considered lost, and error concealment measures have to be taken.

As the network characteristics change dynamically, so must the jitter buffer. The dynamic behavior of the jitter buffer must be a trade off between how well the receiver is able to conceal the effect of lost packets and the amount of delay introduced. Keeping the total delay in the communication path at a comfortable level is an important goal.

2.2.2 Preselector

The MCU is responsible for creating and sending a conference mix with the highest possible quality to its participants. In order to achieve this, it is usually necessary to do more than simply add up the signals. The higher the number of conference participants gets, the higher the overall noise level will be if all signals are added. The MCU may prevent this by using different mixer approaches or noise control algorithms. The simplest approach would be to add a selection stage in front of the mixer to include only a subset of the audio streams in the mix. The selection may be based on simple level calculations (referred to as N-loudest mix) or more advanced algorithms like speech detection.

3 MIXING APPROACHES

The mixing process may be accomplished either in the time domain, as is done for current systems, or in the transform domain, as presented by this paper.

3.1 State of the art - Time domain mixing

The trivial solution for mixing two or more signals is to decode them, mix them together and encode this downmix again. For a transform-based perceptual codec, this is illustrated in Figure 2.

Obviously, this approach has some drawbacks, namely:

- High additional delay

  Utilizing a full chain of decoders and encoders introduces a delay from both processing steps, i.e. the algorithmic delay of the utilized codec is added to the transmission chain.

- High complexity

  In this trivial approach all the input streams have to be fully decoded and the downmix has to be encoded again. This causes a considerable amount of complexity, especially for a high number of input streams.

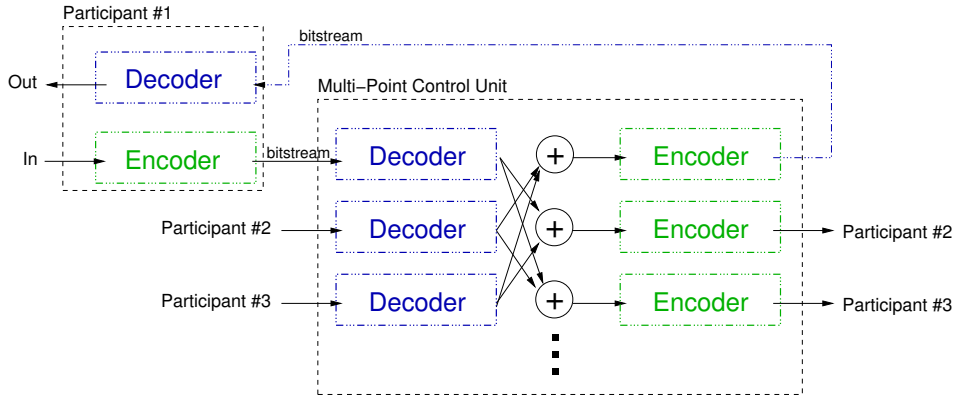

Fig. 1: Mixing of bitstreams in an MCU

- Tandem coding

  The process of decoding and re-encoding is also referred to as tandem coding. Due to the re-quantization of the audio content the quantization noise from the re-encoding step is added to the quantization noise of the first encoding step. This leads to a decreased audio quality.

3.2 Transform domain mixing

In order to eliminate the drawbacks mentioned above, mixing of signals may be realised in the transform domain. Transform codecs that use a window of fixed length and shape, enable the implementation of the mixing process directly without transforming the audio stream back into the time domain. This approach can prevent quality degradation and does not introduce any additional algorithmic delay. Moreover, the complexity is decreased due to the absence of the inverse transform steps in the decoder and the forward transform steps in the encoder. In the following, it is assumed that the signals are encoded using the same sampling frequency and frame length. Then, the resulting process can be illustrated in Figure 3.

Summing up the advantages of this approach:

- No additional delay

  By applying the mixing in the transform domain, no additional delay is introduced by the mixing process. All filter bank operations involving delay in a decoder/encoder chain are avoided.

- Low complexity

  As no filter bank operations are involved, N inverse and M forward transformations are avoided.

- Partial tandem coding

  Applying the mixing operation in the transform or parameter domain, offers the direct access to quantization and coding tool parameters. The parameters can be re-used to generate the outgoing bitstreams. In the case that one of the mixed signals is dominating, all bitstream information of the dominating stream can be copied to the outgoing stream and the remaining bitstreams can be discarded, which means that no tandem coding occurs. For several active incoming streams, it is possible to copy bitstream parts, e.g. all spectral coefficients from a scale factor band, of one stream in order to avoid re-quantization and thus tandem coding locally.

4 AAC-ELD - ENHANCED LOW DELAY

The MPEG work item on the MPEG-4 Enhanced Low Delay AAC codec was finalized in January 2008 [3]. The codec is derived from the MPEG-4 Low Delay AAC codec [4] but utilizes a low delay MDCT (LD-MDCT)[5] instead of the commonly used symmetric MDCT transform. The main feature of the LD-MDCT is the low delay window, which reduces the algorithmic delay compared to an orthogonal transform by 25%. Furthermore, the asymmetric shape of the window aligns to the shape of the temporal masking curve of the human auditory system. Concretely, the small overlap towards future

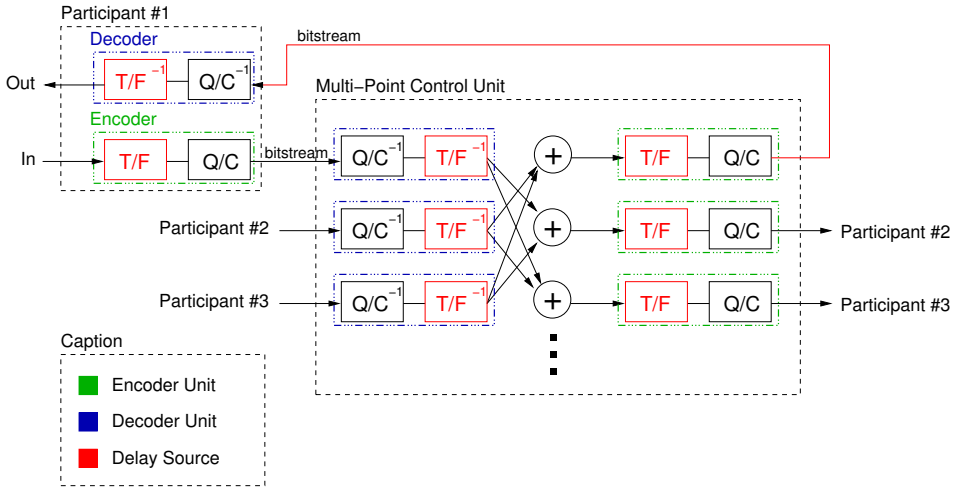

Fig. 2: Trivial solution for mixing of bitstreams

samples of the LD-MDCT fits to the low pre-masking capability of the human auditory system which helps to avoid annoying pre-echo artifacts. In combination with Temporal Noise Shaping [6], this filterbank makes window switching techniques, traditionally used to decrease time domain artifacts, obsolete.

Additionally, the codec utilizes a low delay version of the Spectral Band Replication (SBR) tool, known from High Efficiency AAC (HE-AAC) as standardized in MPEG-4. SBR is a semi-parametric so-called bandwidth extension technique, where a large part of the signal's bandwidth is reconstructed from the core coded low band signal in the SBR decoding process [7]. The SBR tool in AAC-ELD is optimized regarding delay by removing the overlap delay and exchanging the QMF bank with a Complex Low Delay Filter Bank (CLDFB) [8]. The usage of the low delay SBR tool increases slightly the algorithmic delay of the codec but allows it to perform at very low bit rates while maintaining high audio quality and full audio bandwidth.

Furthermore, the ELD codec can make use of the most important AAC tools which are listed below:

- Temporal Noise Shaping (TNS): TNS allows to control the temporal shape of the quantization noise using a prediction filter working in the frequency domain [6].
- Perceptual Noise Substitution (PNS): PNS offers a representation of noise-like signals in a compact, parametric way rather than encoding the spectral coefficients. [9]
- Mid-Side Stereo (M/S): The M/S-Stereo tool exploits inter-channel redundancies and avoids binaural unmasking effects [10].
- Error Resilience Tools (ER): For transmission over error-prone channels the ER tools provide strategies to detect transmission errors and recover the audio data [11].

5 MIXING WITH ELD

For the advantageous mixing in the frequency domain as outlined in 3.2, the coded streams to be combined need to use a unified spectral representation. Since the low delay MDCT (see Section 4) does not make use of window switching techniques, efficient mixing in frequency can be realized using AAC-ELD.

The frequency domain mixing scheme applied to AAC-ELD can be divided into two major parts: one for the transform based codec and one for the low delay SBR tool. The steps outlined below assume that no automatic compression or leveling is performed for the incoming streams, the outgoing stream is just the sum of all incoming.

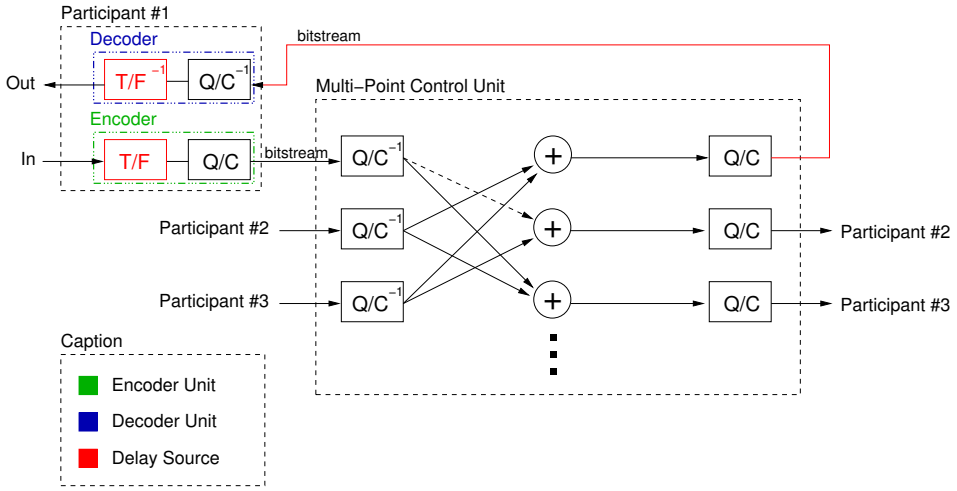

Fig. 3: Mixing of bitstreams in transform domain

5.1 ELD core codec mixing

As every step of repeated quantization adds additional quantization noise and thus degrades audio quality, a prime goal of mixing has to be a considerable reduction of the number of re-quantization steps. In the case under scrutiny, this means that the mixed audio content should be generated by re-using as many quantization and bitstream parameters from the incoming streams as possible. This would mean merely copying the encoded coefficients of parts of the dominant bitstream and discarding the coefficients of the remaining bitstreams whenever possible.

5.2 Inter-stream masking

The basic rules of psychoacoustics regarding masking effects can be applied to inter-stream masking as well. This means that a dominant stream is able to mask other streams and thus, these masked inputs can be neglected. This algorithms can be processed in a frame-wise manner as well as per scalefactor band. The preselector of an MCU (see Section 2.2.2) can be viewed as a frame based algorithm using a simple energy estimation in order to find the M most dominant streams. Another approach is to calculate the masking threshold inside each scale factor band according to the dominant band. All bands containing signal energy below this threshold are considered as masked and discarded. The possibility of copying bitstream elements is therefore also enabled for the granularity of a scalefactor band.

5.2.1 Mixing of the spectrum

The linear transformation property of the MDCT permit the super-positioning, or mixing, of data directly in the transform domain. As no block switching or window shape adaption is utilized in AAC-ELD, the mixing of spectral data becomes a straightforward linear operation.

5.2.2 Mixing of the tools

- TNS

  As the TNS tool alters the shape of the spectrum, the inverse filtering of the spectrum with the extracted TNS coefficients from the bitstream becomes necessary in order to be able to mix the different signals correctly. If one of the bitstreams is dominant in terms of signal strength for the region of the spectrum where a TNS filter is active, the coefficients of the respective TNS filter can be copied directly to the resulting bitstream and no TNS filtering will be necessary.

- PNS

  Concerning the utilization of the PNS tool, one has to distinguish between two cases: whether signals are to be mixed of which some use PNS and some do not, or whether both signals make use of the PNS

tool. In the first case, the signals have to be decoded by the PNS decoder in order to be mixable. In the second case, the mixing is a simple addition of the respective PNS factors.

- L/R and M/S stereo

  If both signals are coded with the same stereo coding tool, it is possible to mix the signals without de-matrixing them. If they, however, use different modes for stereo coding, the signals have to be dematrixed so that the mixing of the respective channel signals becomes possible.

5.3 SBR mixing

Although an optimal AAC-ELD mixer with SBR included is not yet implemented, the main processing steps will be outlined in this section. Currently, extensive studies are made regarding the combined mixing process and further results are expected in the near future.

As is known from MPEG-4 SBR decoding, all SBR parameters, several of which have an adaptive time and frequency resolution, are mapped to the common time/frequency grid given by the resolution of the QMF bank for subsequent high frequency generation and envelope adjustment. Hence, when mixing several incoming streams, all the parameters are decoded and combined using this time/frequency grid. Instead of performing the usual decoding steps: high frequency generation, envelope adjustment and synthesis filtering of the QMF subband samples, the combined data in the time/frequency grid is instead encoded into a new set of SBR parameters.

The following outlines a general concept for mixing SBR data streams in the parameter domain.

- Envelope and noise floor data

  The SBR envelope data is transmitted as encoded energy values. The noise floor data is the ratio of the noise energy compared to the envelope energy. The transmitted energy data constitutes the averaged energy values over a time/frequency grid being a coarser subset of the resolution given by the QMF bank. The resolution in time is referred to as an envelope, while the resolution in frequency is derived from the currently used frequency band table. The noise floor data has an even more coarse distribution, where the resolution in time and frequency are referred to as noise time borders and noise band tables respectively.

  When mixing, the envelope and noise floor data values of a frame are decoded and distributed as energy values over the common time/frequency grid of the QMF bank, where all contributions from the input streams are added together. New envelope and noise floor data values are formed by averaging values in the grid according to a new resolution for the current frame, where the envelope and noise time borders are determined by the energy distribution in the QMF grid and where the most energy dominant input stream determines the current frequency and noise band tables.

- Inverse filtering data

  The inverse filtering data is given as one of four possible levels: off, low, mid and high. The frequency resolution is common to the resolution of the noise floor values, while the time resolution is given by the framelength, i.e. only one value for the inverse filtering level is transmitted per noise band and frame. When, in a decoder, the lowband core signal is transposed to the SBR band, the corresponding amount of inverse filtering is applied to the transposed signal.

  When mixing, a similar approach as for the envelope and noise data is used. The inverse filtering levels of the incoming SBR streams are mapped to a common frequency grid, scaled according to the envelope energy levels previously calculated and subsequently added. The new data is averaged over the noise bands, given by the current noise band table.

- Additional sinusoids

  Additional sinusoids are synthesised in an SBR decoder at a level corresponding to the energy transmitted by the envelope data. They are located in frequency in the middle of the frequency band in which they were signaled.

  When combining additional sinusoids from several streams, it is important to know that they will be reproduced, or synthesised, with a level being the sum of the energy of all incoming streams for that frequency band. Hence, only the sinusoids present in bitstreams where the envelope energy is dominant in that particular frequency region should be signaled as additional sinusoids in the combined bit-

stream. In this way, the sinusoids will be reproduced correctly in the decoder processing the mixed stream.

6 PERFORMANCE EVALUATION

6.1 Methodology

In order to assess the quality of the mixer in a common scenario, we use two streams (A and B) timed as outlined in Figure 4 and mix them together. Each mixed item has thus three segments such that both streams are active only in the middle of the mixed item (segment II). We use two sets of mixed items: Set 1 comprising the mixes of the speech items in Table 2 and Set 2 including the mixes of music items listed in Table 3. All these speech and music items have an average playtime of 9.6 seconds. To balance the overlap of each item combination, Stream B is delayed by 4.5 seconds.

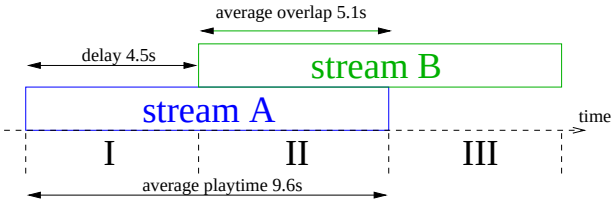

Fig. 4: Experiment setup

The experiment aims at assessing the audio quality of the transform domain mixer ($freq$), presented earlier in this paper, in comparison to the time domain mixing method ($time$). Therefore, we encode the streams using AAC-ELD, mix these streams using the two mixing algorithms and then lastly decode the mixed streams to simulate the entire processing chain illustrated in Figures 2 and 3. The uncoded reference is obtained by mixing A and B directly in time domain without any further processing. Additionally, the mixing algorithms are compared against the theoretical quality ceiling (ref) obtained by encoding/decoding the reference. During the experiment, all signals (mono with 48 kHz sampling frequency) were encoded with AAC-ELD at 64 kbps.

We use the Perceptual Evaluation of Audio Quality (PEAQ) method [12] to evaluate the audio quality associated with the two mixing algorithms. This method allows the objective assessment of the audio quality for non-parametric audio codecs and can be used for the AAC-ELD core codec, but not for SBR. All three coded versions (ref, $time$, $freq$) of the mixed items are compared to the original mix and the Objective Difference Grade (ODG) is calculated. The ODG is defined in ITU Recommendation ITU-R BS.1387-1 [13] as *the objectively measured parameter that corresponds to the subjectively perceived quality*. Its value ranges from 0, for no noticeable difference, to -4 for very annoying. For calculating the ODG value, the PEAQ implementation from McGill University [14] is used.

Item	Description
es02	German male
es03	English female
nadib07	Japanese male
nadib13	French female

Table 2: Speech items used in the test (Set 1)

Item	Description
es01	singing voice
sc03	pop music
sc02	classic music
nadib17	jingle, voice over music

Table 3: Music items used in the test (Set 2)

6.2 Results

The ODG values measured by the PEAQ method are listed in Tables 4 and 5. The first two columns in these tables show the input items used for streams A and B. The subsequent columns list the ODG values of the reference and the two mixing algorithms. The quality degradation caused by the time and frequency domain mixing is shown in columns ($ODG_{ref-time}$) and ($ODG_{ref-freq}$), respectively. Finally, the last column ($ODG_{freq-time}$) shows the quality improvement of frequency domain mixing compared to simple time domain mixing.

The quality degradation caused by tandem coding in time domain mixing can clearly be observed with all test items. All items which are encoded more than once show consistently a lower level of quality compared to the reference items. Frequency domain mixing shows a consistent quality improvement over time domain mixing across all 32 test cases shown in Tables 4 and 5. The tandem coding effect can thus be prevented at least partially

by using the frequency domain mixer. For some items such as es03 and nadib13, the audio quality remains close to the original.

It has to be kept in mind that the scenario used in this paper can be considered as extreme. In conferencing scenarios, usually only one stream is active and thus the loss of audio quality is completely avoided by the frequency domain mixing whereas the loss due to tandem coding remains for the time domain mixing method.

7 CONCLUSIONS

This paper presents an essential proof of concept regarding the advantanges of mixing signals in the transform domain. It is shown how the recently standardized codec MPEG-4 Enhanced Low Delay AAC can be utilized to mix both the core and the SBR part to generate full bandwidth audio signals. Considerable reduction of complexity can be achieved while no additional delay is introduced by the mixing process. Performance evaluations have shown quality improvements of mixing in the transform domain over mixing in the time domain.

8 REFERENCES

[1] International Telecommunication Union, "The E-model, a computational model for use in transmission planning", ITU-T Recommendation G.107, 2005

[2] S. Möller, A. Raake, N. Kitawaki, A. Takahashi, M. Wältermann, "Impairment Factor Framework for Wide-Band Speech Codecs", *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 1969–1976, Nov. 2006

[3] ISO/IEC 14496-3:2008, "Coding of Audio-Visual Objects, Part 3: Audio - Amendment 9", 2008

[4] ISO/IEC 14496-3:2005, "Coding of Audio-Visual Objects, Part 3: Audio", 2005

[5] M. Schnell, R. Geiger, M. Schmidt, M. Jander, M. Multrus, G. Schuller, J. Herre, "Enhanced MPEG-4 Low Delay AAC - Low Bitrate High Quality Communication", *122nd AES Convention*, Vienna, Austria, preprint 6998, May 2007

[6] J. Herre, J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)", *101st AES Convention*, Los Angeles, USA, preprint 4384, Nov. 1996

[7] P. Ekstrand, "Bandwidth Extension of Audio Signals by Spectral Band Replication", *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio*, Leuven, Belgium, Nov. 2002

[8] M. Schnell, R. Geiger, M. Schmidt, M. Multrus, M. Mellar, J. Herre, G. Schuller, "Low Delay Filterbanks for Enhanced Low Delay Audio Coding", *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, pp. 235–238, Oct. 2007

[9] J. Herre, D. Schulz, "Extending the MPEG-4 AAC Codec by Perceptual Noise Substitution", *104th Convention of the AES*, Amsterdam, preprint 4720, May 1998

[10] J. D. Johnston, A. J. Ferreira, "Sum-Difference Stereo Transform Coding", *IEEE ICASSP*, pp. 569–571, 1992

[11] R. Sperschneider, B. Grill, H. Sanae, "Optimization of MPEG-2 / MPEG-4 AAC for error prone transmission channels", *Proc. of the AES 18th International Conference - Audio for Information Appliances*, Burlingame, California, USA, Mar. 2001

[12] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerens, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, B. Feiten, "PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality", *Journal of the AES*, vol. 48(1/2), pp. 3–29, Jan./Feb. 2000

[13] International Telecommunication Union, "Method for Objective Measurements of Perceived Audio Quality", Recommendation ITU-R BS.1387-1, 1998

[14] Peter Kabal, "TSP Lab - Audio File Programs and Routines - AFsp", McGill University - Telecommunications & Signal Processing Laboratory, 2003, URL `http://www-mmsp.ece.mcgill.ca/Documents/Software/Packages/AFsp/AFsp.html`

Stream A	Stream B	ODG_{ref}	ODG_{time}	ODG_{freq}	$ODG_{ref-time}$	$ODG_{ref-freq}$	$ODG_{freq-time}$
es02	es02	-1.420	-2.068	-1.891	0.648	0.471	0.177
es02	es03	-1.338	-1.935	-1.762	0.597	0.424	0.173
es02	nadib07	-1.007	-1.548	-1.316	0.541	0.309	0.232
es02	nadib13	-1.197	-1.973	-1.573	0.776	0.376	0.400
es03	es02	-1.314	-1.868	-1.685	0.554	0.371	0.183
es03	es03	-1.189	-1.832	-1.467	0.643	0.278	0.365
es03	nadib07	-0.853	-1.350	-1.033	0.497	0.180	0.317
es03	nadib13	-1.142	-1.839	-1.370	0.697	0.228	0.469
nadib07	es02	-0.963	-1.447	-1.429	0.484	0.466	0.018
nadib07	es03	-0.809	-1.363	-1.201	0.554	0.392	0.162
nadib07	nadib07	-0.676	-1.114	-0.997	0.438	0.321	0.117
nadib07	nadib13	-0.920	-1.452	-1.318	0.532	0.398	0.134
nadib13	es02	-1.209	-1.811	-1.617	0.602	0.408	0.194
nadib13	es03	-1.060	-1.585	-1.513	0.525	0.453	0.072
nadib13	nadib07	-0.890	-1.370	-1.145	0.480	0.255	0.225
nadib13	nadib13	-1.072	-1.771	-1.439	0.699	0.367	0.332
AVG					0.579	0.356	0.223
MIN					0.438	0.180	0.018
MAX					0.776	0.471	0.469

Table 4: Results for ODG values for set 1, mixing of speech content

Stream A	Stream B	ODG_{ref}	ODG_{time}	ODG_{freq}	$ODG_{ref-time}$	$ODG_{ref-freq}$	$ODG_{freq-time}$
es01	es01	-1.665	-2.471	-2.111	0.806	0.446	0.360
es01	nadib17	-1.331	-2.076	-1.679	0.745	0.348	0.397
es01	sc02	-1.574	-2.280	-2.054	0.706	0.480	0.226
es01	sc03	-1.602	-2.374	-2.003	0.772	0.401	0.371
nadib17	es01	-1.313	-2.092	-1.810	0.779	0.497	0.282
nadib17	nadib17	-1.165	-1.833	-1.588	0.668	0.423	0.245
nadib17	sc02	-1.336	-2.031	-1.837	0.695	0.501	0.194
nadib17	sc03	-1.385	-2.173	-1.884	0.788	0.499	0.289
sc02	es01	-1.535	-2.251	-2.166	0.716	0.631	0.085
sc02	nadib17	-1.348	-2.023	-1.838	0.675	0.490	0.185
sc02	sc02	-1.470	-2.151	-2.096	0.681	0.626	0.055
sc02	sc03	-1.556	-2.287	-2.113	0.731	0.557	0.174
sc03	es01	-1.610	-2.328	-2.051	0.718	0.441	0.277
sc03	nadib17	-1.374	-2.150	-1.768	0.776	0.394	0.382
sc03	sc02	-1.545	-2.175	-1.944	0.630	0.399	0.231
sc03	sc03	-1.502	-2.400	-2.045	0.898	0.543	0.355
AVG					0.737	0.480	0.257
MIN					0.630	0.348	0.282
MAX					0.898	0.631	0.267

Table 5: Results for ODG values for set 2, mixing of music content and speech-over-music content