LOW-DELAY ERROR CONCEALMENT WITH LOW COMPUTATIONAL OVERHEAD FOR AUDIO OVER IP APPLICATIONS

Marco Fink, Udo Zölzer

Dept. of Signal Processing and Communications
Helmut-Schmidt-Universität
Hamburg, Germany
marco.fink@hsu-hamburg.de

ABSTRACT

A major problem in low-latency Audio over IP transmission is the unpredictable impact of the underlying network, leading to jitter and packet loss. Typically, error concealment strategies are employed at the receiver to counteract audible artifacts produced by missing audio data resulting from the mentioned network characteristics. Known concealment methods tend to achieve only unsatisfactory audio quality or cause high computational costs. Hence, this study aims at finding a new low-cost concealment strategy using simplest algorithms. The proposed system basically consists of an period extraction and alignment module to synthesize concealment signals from previous data. The audio quality is evaluated in form of automated measurements using PEAQ. Furthermore, the system's complexity is analyzed by drawing the computational costs of all required modules in all operating modes and comparing its computational load versus another concealment method based on auto-regressive modeling.

1. INTRODUCTION

The transmission of audio material over packet-switched networks experienced increasing popularity in many application areas over the last decade. To comply with the bound of available data rates it was necessary to develop audio codecs to decrease the data rate of audio material without affecting the audible quality in a severe way. In other words, the massive distribution of digital music content was initiated by the development of codecs.

Nowadays, the usage of the internet tends to change from file-based multimedia exchange to streaming-based and interactive scenarios, like *Networked Music Performances* (NMP) [1, 2]. To allow the experience of a real-time system the overall delay between sender and receiver should be minimal. Unfortunately, there are many delay contributors between source and sink. Besides the obvious non-deterministic network delay, the blocking delay of the audio hardware, the delay introduced by receiver buffers, and the algorithmic delay of underlying codecs have to be considered. Apparently, only the blocking delay and the algorithmic delay can be reduced with the help of signal processing. Therefore, the focus of optimization in codecs has changed from data rate to delay. State of the art audio codecs [3, 4] feature algorithmic delays less than 5 ms to allow the aforementioned interactive scenarios.

A major problem in interactive online applications is the diminished audio quality caused by non-optimal network conditions leading to jitter and packet loss. Commonly, receivers apply error concealment techniques to reduce the impact of audible artifacts by replacing the gap, resulting from packet loss, in various ways. Many different concealment strategies are known [5, 6] but the majority of them requires an unpractical amount of processing power or can not satisfy a certain quality level. Popular strategies are often based on Overlap and Add techniques, like *Waveform Similarity Overlap and Add* (WSOLA) [7], or model-based extrapolation [8, 9, 10].

For *Audio over IP* (AoIP) implementations on embedded devices [11] like the Raspberry Pi [12], the concealment is even the systems's limiting module. Hence, this study aims at finding a low-cost alternative and still allow a reasonable audio quality. Since the concealment is to be applied in low-delay AoIP applications, it is optimized for very short blocks.

The proposed concealment strategy aims at extracting periods from past data blocks and utilize them to fill the gap produced by packet loss. In contrast to the before mentioned techniques, the computation of an auto-regressive model and/or autocorrelation shall be avoided to save computations.

A crucial property of high-quality audio error concealment strategies is to guarantee correct phase alignments between frames. Hence, the fade-in and fade-out of the concealed block to the original data receives much attention. Several methods to realize this important property are proposed and analyzed. The actual period extraction is accomplished with the help of zero-crossings and matched pre-processing. The quality of the proposed concealment strategy was ranked using large-scale automated measurements with the *Perceptual Evaluation of Audio Quality* (PEAQ) [13] algorithm. Additionally, the complexity was compared to a high-quality concealment strategy [8], based on extrapolation using auto-regressive models [14].

The paper is structured as follows. The overall system and all its sub-modules are explained in Section 2. The result of the quality measurements can be found in Section 3, whereas the complexity analysis is located in Section 4. Section 5 concludes this paper.

2. SYSTEM

The concealment system is structured module-wise as shown in Fig. 1. Previous data $x_p(n)$ is optionally enhanced in the Pre-Processing block to improve the performance of the Zero-Crossing Analysis module, which generates the zero-crossings vector $z_c(n)$. The resulting $z_c(n)$ is then used to identify extraction boundaries, allowing the cutting out of multiple periods of the unaltered previous data $x_p(n)$ within the Extraction block. Subsequently, the alignment module is supposed to identify the phase offset between the last data block and the extracted concealment signal $x_e(n)$. The actual phase alignment can then be performed within the same

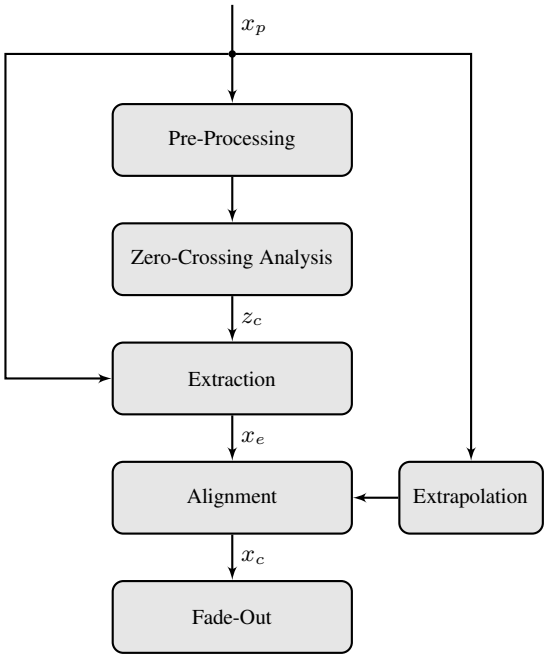

Figure 1: System overview

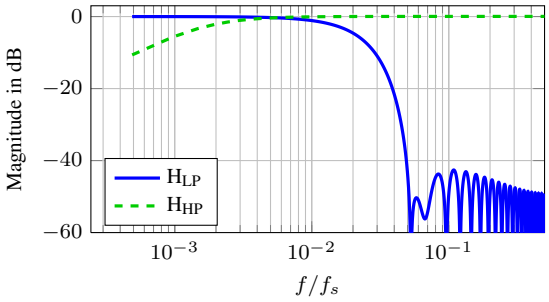

Figure 2: Pre-processing filters

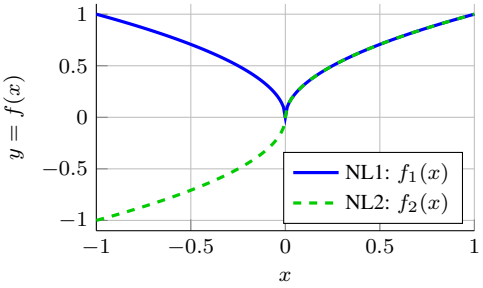

Figure 3: Odd and even pre-processing non-linearities

module. To allow smooth transitions from the previous data block into the concealment data the extrapolation module is applied. It computes several continuative samples and cross-fades them with the concealment data resulting in x_c. Lastly, the Fade-Out module guarantees a certain continuity from the concealed data x_c into the next data block. Hence, the length of x_c should exceed a single block length.

2.1. Pre-processing

Two different pre-processing steps are considered to improve the system's overall performance. Only the fundamental harmonic content of the input signal should be suspect to the zero-crossing analysis. Therefore, a FIR lowpass H_{LP} with a normalized frequency of $\omega_n = .01 \cdot 2\pi$ and order 20 is utilized. In order to guarantee a DC-free signal a first-order recursive filter $H_{\text{HP}} = \frac{1-z^{-1}}{1-0.99z^{-1}}$ is additionally applied. The corresponding transfer functions are illustrated in Fig. 2. To conserve the position of the zero-crossings, it is crucial to apply zero-phase filtering. Therefore, the FIR lowpass is applied in forward and reverse direction [15]. The IIR filter phase response is close to zero in the passband and hence, is not affecting the signal.

Besides restricting the bandwidth of the input, the signal was subject to non-linear processing to enhance the first harmonic as proposed in [16]. The characteristic curves of the considered non-linear functions

$$f_1(x) = \sqrt{|x|} \tag{1}$$

$$f_2(x) = \sqrt{|x|} \cdot \text{sgn}(x) \tag{2}$$

are depicted in Fig. 3. Apparently, $f_1(x)$ and $f_2(x)$ are odd and even non-linear functions, respectively.

2.2. Zero-Crossing Detection

Zero-crossing positions are a meaningful low-level feature, mostly utilized in speech processing. For example, the detection of voiced and unvoiced parts in speech recordings using the zero-crossing rate (ZCR) is very common [17]. It can also be beneficial in the context of AoIP since received audio streams primarily consist of single instruments or speakers and therefore, should mainly contain harmonic signals featuring a strong periodicity. Zero-crossings can be stored in a binary vector

$$z_c(n) = \begin{cases} 1 & \text{if} \quad \text{sgn}(x(n) \cdot x(n-1)) < 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

of length N_i, indicating a zero-crossing at samples $[n_1, \ldots, n_{N_i}]$.

A zero-crossing is similar to sign changes between two consequent samples $x(n-1)$ and $x(n)$. Whenever their product turns negative a zero-crossing can be assumed. Applying the XOR operation to the sign bits of $x(n-1)$ and $x(n)$ is an efficient way to achieve the same result.

The following processing steps require the actual position index vector i_{zc} containing the N_i sample indexes $[n_1, \ldots, n_{N_i}]$, where $z_c(n)$ is non-zero. The resulting vector i_{zc} can be refined by defining a lower bound δ_{zc} for the distance between two zero crossings. This value should be set according to the maximal expected fundamental frequency f_{max} of the input signal and the sampling frequency f_s

$$\delta_{\text{zc}} = \left\lceil \frac{f_s}{2 \cdot f_{\text{max}}} \right\rceil . \tag{4}$$

Whenever the inter-zero-crossing-interval falls below δ_{zc}, the corresponding zero-crossing candidate index is removed from i_{zc}.

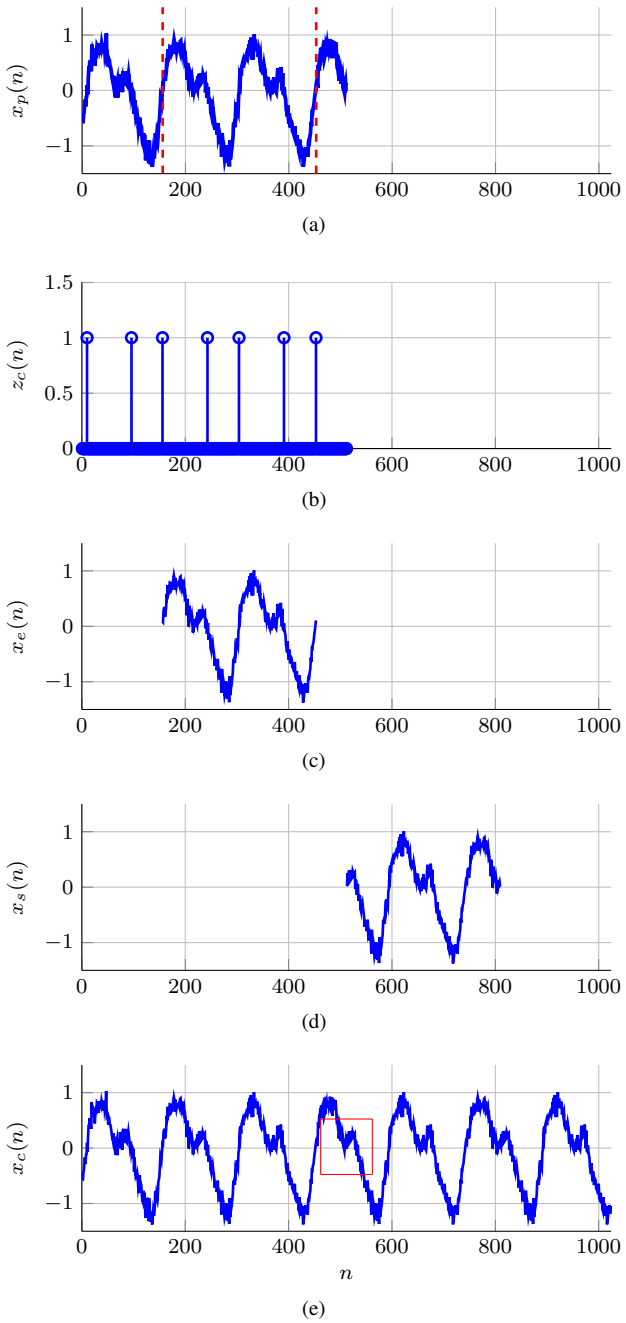

Figure 4: Relationship between x_p(a), z_c(b), x_e(c), x_s(d), and x_c(e)

2.3. Period Extraction and Alignment

The period extraction module receives a block of previous data x_p and a vector i_{zc}, containing the indexes of zero crossings in x_p, to extract one or P periods. The process is illustrated in Fig. 4. Fig. 4a) shows a periodic waveform $x_p(n)$, while Fig. 4b) depicts the corresponding zero-crossings, which are used to define the extraction boundaries, defined by the zero crossing indexes n_{N_i-2P} and n_{N_i}, which are marked with red stripes in the Fig. 4a). The extracted signal

$$x_e(n) = x_p(n), \quad \text{for} \quad n = [n_{N_i-2P}, \ldots, n_{N_i}] \tag{5}$$

of length N_e is shown in Fig. 4c). The quantity of extractable periods depends on the amount of identified zero crossings. To extract at least a single period one has to allow a certain length N_p for x_p. N_p can be restricted when a minimal frequency f_{min} for the concealment process is defined. From this it follows that

$$N_p = \left\lceil \frac{\beta \cdot f_s}{f_{min}} \right\rceil, \tag{6}$$

where β is an optional safety margin.

Since the extracted periods are concatenated until they show a certain length N_c, the beginning and end have to be matched to allow smooth transitions between the repeated periods. In the implementation this feature was realized by replacing n_m samples at the front and the end with a linear series between $x_e(N_i - n_m + 1)$ and $x_e(n_m)$ of length $2 \cdot n_m$.

The block $x_e(n)$ is zero-phased due to its cutting at the zero crossings. To align the phase of the extracted periods to the end of the last block x_p, a circular shift by l samples is applied to obtain extracted and shifted sequence

$$x_s(n) = x_e(n - l \bmod N_e), \tag{7}$$

which is plotted in Fig. 4d).

Several ways to estimate the shifting offset l are possible. In this study three methods are chosen and analyzed:

1. Zero-crossing distance: The simplest reviewed method is to compute the offset of the last zero-crossing in i_{zc} and the length of the past data block $x_p(n)$, which is determined by the amount of blocks in the buffer M and the block size N

   $$l_1 = NM - n_{N_i}.$$

   Apparently, the estimation accuracy mainly depends on the zero-crossing's precision.

2. Slope and amplitude matching: If the slope and amplitude of two periodic signals with the same frequency are similar, one can assume a similar phase at that point. Hence, it is necessary to obtain the slope at the end of $x_p(n)$ and the slope of $x_e(n)$ by differentiating both. The ten closest values are assumed as candidates. In a second step, the candidate showing the smallest deviation in amplitude to the last sample of $x_p(n)$ is determined and its index is used for the shift operation.

3. Cross-correlation: It is well-known that the delay between two signals can be computed using the cross-correlation [18]. The time delay is the offset between the location of the cross correlation's maximum and the zero lag index. Since the alignment of $x_e(n)$ to the end of $x_p(n)$ is desired, the cross correlation from the end of the signals in inverse direction is computed.

Subsequently, the concealment signal $x_c(n)$ is synthesized by concatenating $x_s(n)$ until it exceeds a certain length $N_c > N$ to allow the replacement of a complete data block of length N. The phase-matched concatenation of $x_p(n)$ and $x_c(n)$ is illustrated in Fig. 4e). The actual block transition is highlighted with the red box.

2.4. Extrapolation and Fade-In

The last section exposed a strategy to produce the actual concealment waveform $x_c(n)$ that is supposed to show similar characteristics as the waveform of previous blocks. However, the concealment quality can be improved by extrapolation of previous data and fading the resulting N_f samples into $x_c(n)$. Alike the estimation of the offset l, multiple strategies of extrapolation were compared and shall be described in the following.

1. Reflection: Applying the point reflection to the signal's tail can be performed by reverse indexing, inversion, and an offset with the last value times 2

$$x_{f1}(n) = 2 \cdot x_p(N_p) - x_p(N_p - n).$$

2. Weighted Slope Continuation: The slope of previous data is linearly weighted to avoid a constant slope in the extrapolation

$$x_{f2}(n) = x_p(N_p) + (x_p(N_p - n) - x_p(N_p - 1 - n)) \cdot n.$$

3. Linear: A linear extrapolation can be achieved by determining the slope at the end of x_p and accumulating it N_f times to the last value of x_p.

$$x_{f3}(n) = x_p(N_p) + (x_p(N_p) - x_p(N_p - 1)) \cdot n.$$

4. Polynomial: The extrapolation with a polynomial can be described as the attempt to find a function $f(x)$ describing previous data and feed it with following x values to obtain new data. The polynomial p can be found in the least-square sense by solving

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^{N_f-1} \\ 1 & x_2 & \cdots & x_2^{N_f-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N_e} & \cdots & x_{N_f}^{N_f-1} \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{N_f-1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{Nf}) \end{pmatrix},$$

where $[x_1, \ldots, x_{N_f}]$ is assumed to be a linear series from 1 to N_f. The actual extrapolated signal can then be computed the following way

$$x_{f4}(n) = p_0 + p_1(n+N_f)^1 + \cdots + p_{N_f-1}(n+N_f)^{N_f-1}$$

or using the Horner's method.

An overview of the utilized extrapolation is illustrated in Fig. 5. It shows an arbitrary waveform which is extrapolated at $n = 8$ using the four presented methods.

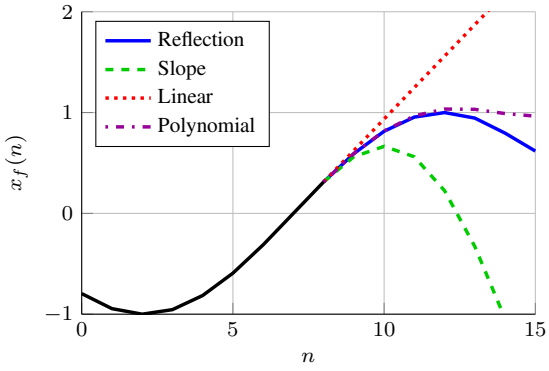

Figure 5: Different simple extrapolation algorithms for $N_f = 8$

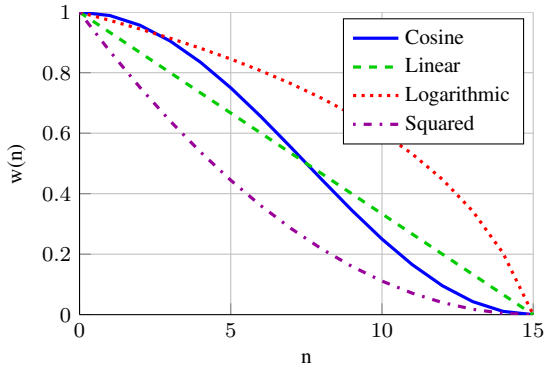

Figure 6: Fading windows for $N_w = 16$

2.5. Fade-Out

As already mentioned, the signal quality of error concealment strategies depends crucially on smooth transitions between extrapolated and actual received data. Hence, the transition from conceal data to the subsequent intact audio block has to be assured. As already denoted in [8] it is beneficial to choose the extrapolated block's length N_c longer than the actual block size N and apply a cross-fade with the next block. Several window forms $w(n)$ and lengths N_w were investigated and it turned out that only cross-fades leading to a constant amplitude

$$w(n) + w(N_w - n) = 1 \tag{8}$$

are of benefit. Although, constant power windows

$$w^2(n) + w^2(N_w - n) = 1 \tag{9}$$

are widely used in audio processing, and especially mixing, they can only be optimally used in the case of uncorrelated signals, like when mixing different tracks. In this study, four different constant amplitude cross-fade windows are applied. Namely, a cosine, a linear, a logarithmic, and a squared. All those windows are illustrated in Fig. 6. The influence of the cross-fade window length and form shall be evaluated in the following sections.

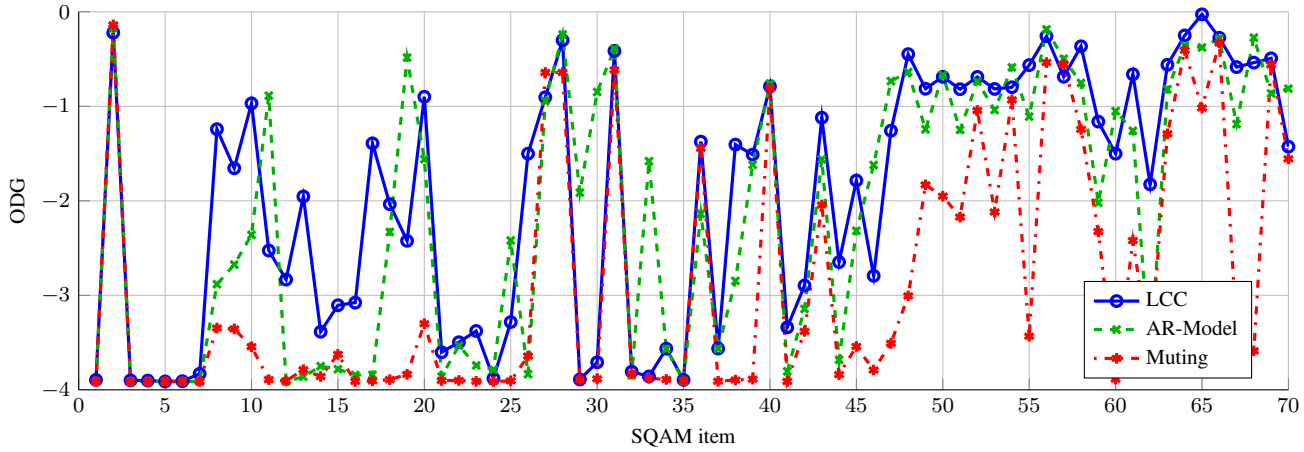

Figure 7: ODG score over SQAM test files

3. EXPERIMENTS

3.1. PEAQ

The quality of the proposed system was initially evaluated objectively using automated measurements in MATLAB, similar to the experiments in [8]. Although the *Perceptual Evaluation of Audio Quality* (PEAQ) method [13] was actually developed to rate audible artifacts caused by audio codecs, it turned out to be an advantageous tool to judge the quality of audio error concealment. The experiments in [8] revealed a significant correlation between the automated PEAQ measurements and a listening test. The result of the PEAQ algorithm is the so-called *Objective Difference Grade* (ODG). It ranges from -4 to 0, covering the identified audio quality impairment from "very annoying" to "imperceptible". The sound data base used for the measurement is the *Sound Quality Assessment Material* (SQAM) [19], consisting of 70 high-quality test items featuring a large variance of sound sources and tonal characteristics.

The actual test procedure was designed as follows: Initially, a SQAM item is loaded and down-mixed to mono. Using the test item's amount of samples and the current block size N allows to calculate the corresponding amount of frames. For every frame a random value is computed, that indicates a lost frame if its larger than the currently simulated packet loss rate assuming that one network packet contains a single audio frame. To have an error reference for following comparisons the input signal is copied once and all erroneous frames are set to 0. In other words, muting is applied as a worst-case concealment for the comparison.

The concealment routine was then called for the erroneous frame with N_P previous samples, the amount of required samples N_c, and the modes for the pre-processing, alignment, and extrapolation modules, respectively. The first N samples of its result are used to replace the erroneous frame and the remaining samples are cross-faded with the following audio frame. The overall result is written to a wave file. To obtain the ODG score, the self implemented PEAQ tool is fed with the resulting wave file and the corresponding original one. This test procedure is repeated for different parameters of the block size N, packet loss rate e, pre-processing mode, alignment mode, extrapolation mode, cross-fade length N_w, the cross-fade window function w_{cf}, and for every SQAM test item. Note, that the sample rate f_s of all test items is 44.1 kHz, the minimal frequency f_{min} is chosen to be 80 Hz, and the safety margin β is set to 1.2. Hence, the search window length N_p, which is used to find zero-crossings, is restricted to 662 samples.

Note, that the authors verified the correctness of the self implemented PEAQ tool by comparing it with two different freely available implementations peaqb and PQevalAudio [20, 21]. A test run over the SQAM data set using the three implementations and using a random concealment setting resulted in a mean absolute deviation of 0.09 for peaqb and 0.07 for PQevalaudio in comparison to the ODG scores of the own implementation.

First of all, it shall be shown how well the presented concealment strategy, denoted as Low-Cost Concealment (LCC) in the following, performs for the different SQAM items. Fig. 7 shows the ODG score over the SQAM items for a block size of 64 samples, packet loss rate $e = 0.01$, a cosine-shaped cross-fade window of length 32. Additionally, the corresponding ODG score of another concealment method [8], based on auto-regressive modeling (AR-Model) using the Burg method, and the muting error reference are shown. The AR-Model produces very good result if the models are computed using a large amount of previous data. Nevertheless, the quality of LCC and AR-Model are similar for very short blocks, although the AR-model requires much more computational effort. LCC even shows a slightly improved average score $\mu_{\text{LCC}} = -1.93$ vs. $\mu_{\text{AR}} = -2.11$ for the explained simulation parameters. Both proposed methods outperform the muting concealment clearly ($\mu_{\text{Mute}} = -2.88$). It comes apparent that the results are strongly signal-dependent. Interestingly, the overall trend is quite similar for LCC and AR-Model. For example, the SQAM items $1 - 7$, which are different synthetic signals, don't show any improvement while the string instruments, voice, and singing items work very well. Also items $26 - 34$ lead to very bad results since these are percussive and bell-like sounds without strong periodicity. As mentioned in the introduction, the system is designed to conceal natural, harmonic sounds. Hence, the previously described items are removed from the test set to evaluate the proposed system in the context of its potential application.

As a next step, the measurement was repeated using the limited SQAM data set but with varying block length N and error rate e. The results of the individual SQAM test items were averaged.

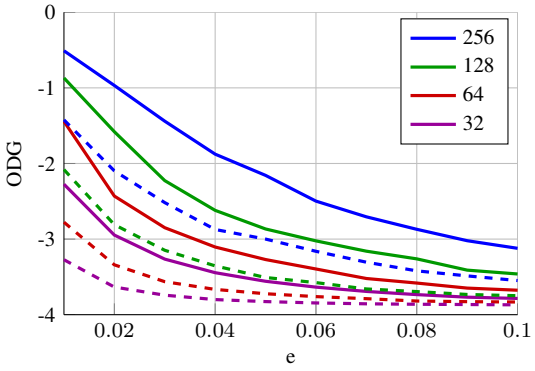

Figure 8: ODG score over error rate using different block lengths for the proposed method (solid) and the AR reference method (dashed)

As expected, the quality decreases for increased e and hence, the ODG curves in Fig. 8 decrease monotonically. The solid curves represent the LCC results, whereas the dashed curves show the results for the error reference. The ODG scores are degraded more severely for shorter blocks even when the same amount of previous data is used for the concealment. This behavior allows two conclusions. On the one hand the replacement and fading using longer blocks is smoother and causes less abrupt signal changes or on the other hand, short signal deviations have stronger negative impact on the ODG score.

In the following, the influence of the combination of the different operating modes are analyzed. Therefore, the block size N and error rate e were set to 64 and 0.01, respectively. Only the operating modes are varied and the average ODG score using the limited SQAM data set was computed. First of all it should be noticed that all LCC modes, besides the combination of the first non-linear function and the filter with arbitrary alignment and extrapolation, improved the average ODG score significantly. Most combinations show an improvement of more than 1 ODG score in comparison to the muting error reference, which was rated $\mu_{\text{Mute}} = -2.8361$ by PEAQ. The mode combinations, yielding the ten best results, are shown in Table 1. Apparently, the pre-processing improves the performance of the zero-crossing analysis since the four best combinations use the filter and/or the second non-linear function. The alignment, based on matching of slopes and amplitudes, outperforms the cross correlation and zero-crossing distance estimations to estimate the phase offset. The simple linear extrapolation seems to be the favorable method for the fade-in process since the six best results are obtained using the linear extrapolation.

As mentioned before, also the effect of the cross-fade window's form and length on the PEAQ results shall be demonstrated. The previous test setup was repeated with fixed parameters ($N = 64$, $e = 0.01$) and the best-performing concealment mode (see Table 1), presented in the previous section. The limited SQAM set was used for this experiment again. As it is apparent from Fig. 9, showing the ODG score for the windows described in Sec. 2.5 and relative window lengths, the centered cross fades (Cosine and Linear) clearly outperform the non-centered ones (Logarithmic and Squared). This implies that neither extrapolated data nor the next intact block should be emphasized in the cross-fade process. The linear cross-fade window is slightly beneficial, especially for larger N_w. There is only minor improvement for window lengths $N_w > 0.5N$ but if it falls below, strong quality degradation occurs.

Table 1: 10 best-performing LCC modes

Pre-Processing	Alignment	Extrapolation	ODG score
NL2+F	SA	L	−1.5924
F	SA	L	−1.5948
F	ZC	L	−1.6171
NL2+F	ZC	L	−1.6247
None	SA	L	−1.6452
NL2	SA	L	−1.6452
NL2+F	SA	P	−1.6648
F	SA	P	−1.6661
NL2+F	SA	S	−1.6740
F	SA	S	−1.6744
Muting			−2.8631

F: Filter, NL1: Non-linearity 1, NL2: Non-linearity 2,
SA: Slope and Amplitude Matching, ZC: Zero-Crossing Distance,
L: Linear Extr., P: Polynomial Extr., S: Slope Extr.

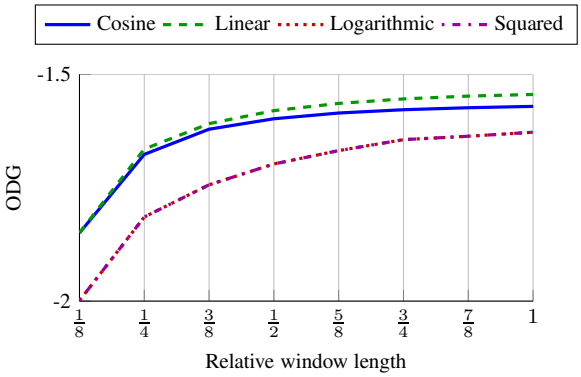

Figure 9: ODG score over different window lengths and forms

4. COMPLEXITY

Since the LCC complexity significantly depends on the operation mode and parametrization it is non-trivial to express an overall complexity. Hence, the complexity of the included modules in different configurations shall be discussed. The following expressions (see Table 2) represent the complexity to conceal a block of length N. Non-complex multiplications (MULs) and additions (ADDs) are listed in the corresponding MUL and ADD column, whereas important conditionals, memory operations, and functions, which significantly depend on the implementation and target platform like the root square function, are listed in the misc column.

1. Preprocessing: The twice-executed FIR filter of order 20 (21 taps) consumes $2 \cdot 21 \cdot N$ MULs and ADDs, whereas the IIR filter requires 4 MULs and 5 ADDs. Both non-linear functions consist of the abs function (e.g. conditional and assignment) and the square root function, which is strongly implementation-dependent and hence can not be described generically.
2. Zero-Crossing Analysis: The zero-crossing analysis can be implemented by comparing the sign of two consecutive sam-

Table 2: Complexity Overview

Module	MUL	ADD	misc
Preprocessing			
FIR	$2 \cdot 21 \cdot N$	$2 \cdot 21 \cdot N$	
IIR	$4 \cdot N$	$5 \cdot N$	
NL1	0	0	$N\cdot$ sqrt, $N\cdot$ abs
NL2	N	0	$N\cdot$ sqrt, $N\cdot$ abs
Zero-Crossing	N	0	$N\cdot$ sign
Extraction	0	0	memcpy(N_e)
Alignment			
ZC	1	0	
SA	0	$2 \cdot N_e + 11$	sort(N_e) + sort(10)
XC	0	0	conv(N_e) + sort($2 \cdot N_e - 1$)
Extrapolation			
Mirror	0	5	
Slope	4	8	
Linear	5	8	
Polynomial	9	12	solve(4)
Fading	8	4	

ples by multiplying them and comparing the resulting sign bit.

3. Extraction: The actual extraction is a simple copy operation (memcpy) of previous data in a new processing buffer.
4. Alignment: The alignment itself is simply a rearrangement of the extracted buffer which can be realized by adapted indexing, e.g. with a pointer offset. If the alignment offset index l is computed using the zero-crossing distance one can subtract the last zero-crossing index from the length of the buffer used for the zero-crossing analysis.

   For the case of the Slope and amplitude matching, the derivative of the extracted concealment signal (N_e ADDs) needs to be computed. Then, the derivative of the last two samples (1 ADD) of the last block is subtracted from the derived extracted block (N_e ADDs). The result is sorted (sort(N_e)) to find the 10 smallest values, corresponding to the closest matches. The closest amplitude is obtained by subtracting the amplitude of the last sample of the last block (1 ADD) from the 10 candidates and again searching the maximum value by sorting (sort(10)).

   Finding the offset index l using the cross correlation (XC) is the most complex method since it requires to convolve the extracted periods with the last samples of previous data (conv(N_e)) and then find the maximum in the result by sorting (sort($2 \cdot N_e - 1$)).
5. Extrapolation: The result of the extrapolation is always cross faded with the aligned extracted block to allow smooth transitions. The fading window length is fixed and hence the actual fading consumes 8 MULs and 4 ADDs on top of every extrapolation method. The mirrored extrapolation is implemented by subtracting the inversely indexed previous samples (4 ADDs) from the last sample times 2 (1 ADD). The weighted slope continuation requires 4 ADDs to compute the slope, 4 MULs for the the weighting, and 4 ADDs for offsetting the result. The simple linear extrapolation only utilizes the last derivative of previous data (1 ADD), the linear weighting (4 MULs), plus the offset (4 ADDs). To compute the polynomial extrapolation a linear equation system with 4 unknown variables has to be solved (solve(4)). Computing the polynomial using the Horner's scheme requires $3 \cdot 3$ MULs and $3 \cdot 4$ ADDs.

Comparing the performance of the different LCC modes (Table 1) and their corresponding complexities (Table 2) reveals an extraordinary situation. The typical quality versus complexity trade-off doesn't hold true in these measurements. The best-performing LCC alignment mode was SA which clearly outperforms the XC mode but is clearly less complex. The same holds true for the extrapolation mode. The best performing linear mode only requires a fraction of the polynomial complexity but yields better results.

To roughly determine the computation time difference of the LCC in contrast to the AR-Model, a test run over the complete SQAM test set was performed and the execution times of the corresponding conceal function calls were measured. Both concealment strategies were fed with the same data, a common block size N of 64, and error rate of 0.01. LCC, in its best working configuration, required about 233 ms to conceal a complete file in average, whereas the AR-Model computed about 1269 ms. In other words, the current implementation of LCC requires only about 18 % of the AR-Model's execution time.

5. CONCLUSIONS

The goal of this study was to find a low-cost error concealment which is suited for AoIP applications requiring lowest latency, like Distributed Network Performances, on low-power platforms, like embedded devices. The proposed system basically consists of an period extraction and alignment module. The simple extraction, based on zero-crossings and matched pre-processing, is sufficient to extract periods, which can then be aligned to prior data and concatenated to synthesize a concealment waveform. For the purpose of smooth phase transitions from previous frames into the concealment frame and back into following audio frame, cross-fades are applied. The samples, extending the previous audio frame to allow the cross-fade, are obtained by extrapolation. The audio quality of

the presented concealment system, and all of its operating modes, is evaluated using a large-scale automated test using PEAQ and the SQAM data set. The improvement of the average ODG constitutes more than 1 ODG score in comparison to muting as the simplest concealment method. The best-performing operating modes were identified using the same test setup.

Apparently, the repetition of periods from prior audio frames, combined with proper fade-in and fade-out, allows a significant quality improvement in comparison to simplest methods like muting or frame repetition. The resulting audio quality was found to be even slightly superior then for computationally demanding concealment methods based on auto-regressive modeling. The initial goal of this study, the reduction of computational complexity of concealment strategies, was accomplished since the new proposed system only consumes 18 % computation time while offering at least comparable quality.

6. REFERENCES

[1] A. Carôt and C. Werner, "Network music performance-problems, approaches and perspectives," in *Proceedings of the Music in the Global Villag*, Budapest, Hungary, 2007.

[2] C. Chafe, S. Wilson, and R. Leistikow, "A simplified approach to high quality music and sound over IP," in *Proc. of the COST G6 Conference on Digital Audio Effects (DAFx-00)*, Verona, Italy, pp. 1–5.

[3] J.M. Valin, G. Maxwell, T. Terriberry, and K. Vos, "High-Quality, Low-Delay Music Coding in the Opus Codec," in *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.

[4] J.M. Valin, T. Terriberry, C. Montgomery, and G. Maxwell, "A High-Quality Speech and Audio Codec With Less Than 10-ms Delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, Jan. 2010.

[5] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *Network, IEEE*, pp. 40–48, 1998.

[6] B.W. Wah, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," *Proceedings International Symposium on Multimedia Software Engineering*, pp. 17–24, 2000.

[7] H. Sanneck, A. Stenger, K. Ben Younes, and B. Girod, "A new technique for audio packet loss concealment," in *Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, Nov 1996, pp. 48–52.

[8] M. Fink, M. Holters, and U. Zölzer, "Comparison of Various Predictors for Audio Extrapolation," *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, 2013.

[9] S. Preihs, F.R. Stöter, and J. Ostermann, "Low delay error concealment for audio signals," in *Audio Engineering Society Conference: 46th International Conference: Audio Forensics*, Jun 2012.

[10] J.M. Valin, K. Vos, and T. Terriberry, "RFC 6716: Definition of the Opus Audio Codec," Website, Available online at `http://tools.ietf.org/html/rfc6716` visited on May 20th 2014.

[11] F. Meier, M. Fink, and U. Zölzer, "The JamBerry - A Stand-Alone Device for Networked Music Performance Based on the Raspberry Pi," *Linux Audio Conference 2014*, 2014.

[12] The Raspberry Pi Foundation, "Raspberry Pi Homepage," `www.raspberrypi.org`.

[13] International Telecommunication Union, "BS.1387: Method for objective measurements of perceived audio quality," Website, Available online at `http://www.itu.int/rec/R-REC-BS.1387` visited on March 7th 2014.

[14] I. Kauppinen and K. Roth, "Audio signal extrapolation - theory and applications," in *Proc. of the 5th Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, 2002.

[15] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Pearson Education, Limited, 3rd edition, 2010.

[16] W. Hess, "Time-domain pitch period extraction of speech signals using three nonlinear digital filters," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, Apr 1979, vol. 4, pp. 773–776.

[17] A. Lerch, *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, A John Wiley & Sons, Inc., publication. Wiley, 2012.

[18] C. Knapp and G. Clifford Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, Aug 1976.

[19] European Broadcast Union, "EBU SQAM CD," Website, Available online at `tech.ebu.ch/publications/sqamcd` visited on March 7th 2014.

[20] Giuseppe Gottardi, " Perceptual Evaluation of Audio Quality - beta," `http://sourceforge.net/projects/peaqb/files/peaqb/1.0.beta/`.

[21] P. Kabal, "PQevalAudio," `http://www-mmsp.ece.mcgill.ca/Documents/Software/Packages/AFsp/PQevalAudio.html`.