COMPARISON OF VARIOUS PREDICTORS FOR AUDIO EXTRAPOLATION

Marco Fink, Martin Holters, Udo Zölzer

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

ABSTRACT

In this study, receiver-based audio error concealment in the context of low-latency Audio over IP transmission is analyzed. Therefore, the well-known technique of audio extrapolation is investigated concerning its usability in real-time scenarios, its applied prediction techniques and various transmission parameters. A large-scale automated evaluation with PEAQ and a MUSHRA listening test reveal the performance of the various extrapolation setups. The results show the suitability of extrapolation to perform audio error concealment in real-time and the qualitative superiority of block based methods over sample based methods.

1. INTRODUCTION

The internet is increasingly utilized as the transport framework for nowadays communication. The common technique for the transmission of speech, Voice over IP (VoIP), has been used for about 20 years and replaced analog as well as ISDN telephony extensively. Also the spreading of musical content, called Audio over IP (AoIP), has been well-established. However, this trend mainly applies for broadcast scenarios but not for low-latency, bidirectional communication, which shall be the use case in the following.

To allow the transmission of continuous, analog signals it is necessary to convert the signal into a digital representation, fragment it into blocks, and encapsulate it into an IP packet. Thereafter, the actual transport over the IP network can occur before a receiver can extract the audio segment from the packet, and convert it to the analogue domain again for the purpose of replaying it. All these steps introduce a certain amount of delay, conflicting with the requirement of low latency in many AoIP applications. Especially, interactive scenarios like distributed musical performances [1, 2, 3, 4], which require very low latency, seem to be difficult to realize. Besides various system approaches, many specific problems of IP based musical interaction such as issues related to transmission delay [5] or the necessity of audio device synchronization has been analyzed [6]. Also, the severe problem of error concealment in case of packet loss due to non-optimum network conditions has been investigated [7, 8]. Simplest concealment schemes like block interpolation, repetition, silencing or noise substitution are known to perform only with a moderate quality. Techniques considering the actual signal and resynthesizing it are hence more promising. Possible systems are sinusoidal analysis/synthesis [9], adapted waveform similarity overlap-add [10], and model-based variants. In this study, Kauppinen's [11] approach of audio extrapolation, based on auto-regressive modeling, is used to substitute missing audio fragments. The quality of the concealment mainly depends on the used signal model, obtained with prediction. The focus of this survey lies on the exact evaluation of the prediction scheme's influence on the audio quality, considering its algorithmic complexity and usability in an AoIP scenario. Section 2 recaps Kauppinen's audio signal extrapolation algorithm and the application in an AoIP scenario whereas section 3 introduces different prediction techniques. The setup of the undertaken experiments and the corresponding results are explained in section 4. Section 5 depicts one way of integrating the analyzed concealment scheme into an existing AoIP framework. The summarized findings and further conclusive thoughts are pointed out in section 6.

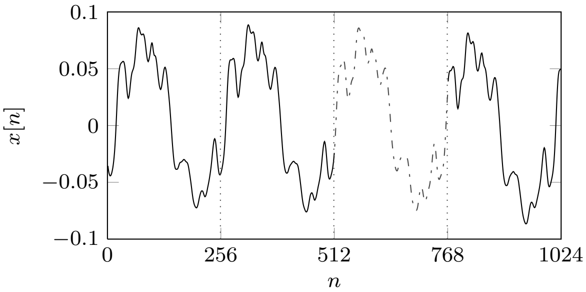


Figure 1: *Extrapolation of a missing audio fragment*

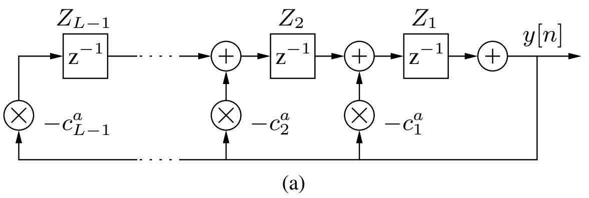
2. AUDIO SIGNAL EXTRAPOLATION

In the context of audio signals, extrapolation describes the process of extending a sequence of samples with estimated values $\hat{x}[n]$. This estimation is based on a previously determined model of the process which created the known signal parts $x[n]$. The quality of the estimated sequence essentially relies on the correctness of the model. The extrapolation result itself can be described as the superposition of previous weighted system output values and the current input value $x[n]$, characterizing this kind of model as an *autoregressive model* (AR), given by

$$y[n] = \frac{1}{a_0}\left(a_1 y[n-1] + \cdots + a_p y[n-p]\right) + x[n] = \\ = \frac{1}{a_0}\left(\sum_{i=1}^{p} a_i y[n-p]\right) + x[n]\,, \quad (1)$$

where a_i are the AR parameters and p describes the order of the model. This difference equation can be z-transformed to obtain the following transfer function

$$H(z) = \frac{1}{\sum\limits_{i=0}^{p} a_i z^{-i}}\,. \quad (2)$$



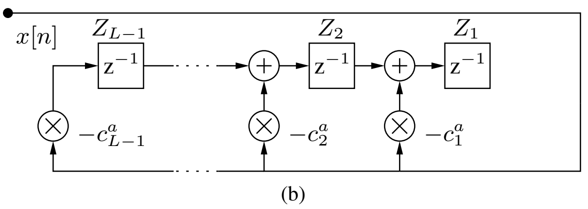


Figure 2: *Transposed direct form II implementation of purely recursive filter: Extrapolation (a) and initialization (b) mode*

Apparently, Eq. (2) also represents the transfer function of a purely recursive *infinite impulse response* (IIR) filter. Therefore, audio signal extrapolation can be implemented with a single recursive filter using AR parameters a_i as filter coefficients. Since the purpose of extrapolation is to prolong a sequence $x[n]$ the input of the recursive extrapolation filter is expected to be zero

$$x[n_0], \dots, x[n_0 + s - 1] = 0, \tag{3}$$

assuming the extrapolation process starts at sample n_0 and is applied for s samples. Obviously a recursive filter of length L, illustrated in Fig. 2a, can't emit non-zero output without excitation or prior output values $y[n_0 - 1], \dots, y[n_0 - L]$. Therefore, a proper filter initialization is essential. The initialization is depending on the used filter structure. For the case of non-transposed direct form filter implementations the copying of the last known samples in reverse order $x[n_0 - 1], \dots, x[n_0 - L]$ directly into the states $Z_1, \dots, Z_{L-1}$ is sufficient. The implementation considered in the following is based on transposed direct form II filters, as shown in Fig. 2a. The filter states are affected by input $x[n]$ already weighted with filter coefficients in this filter realization. Hence, the initialization can be achieved by feeding the last known samples $x[n_0 - L], \dots, x[n_0 - 1]$ directly in the feedback path and weighting them with AR parameters a_i as filter coefficients $c^a_{L-1}, \dots, c^a_1$, like illustrated in Fig. 2b.

With respect to AoIP transmission extrapolation can be used to compensate faulty or missing audio material caused by adverse network conditions, as demonstrated in Fig. 1. The solid line shows the waveform of a clarinet tone, segmented in 4 frames. Assuming the third frame (sample 512 to 767) is corrupted or missing completely, audio extrapolation can be used to approximate a waveform (dotted line) similar to the lost one. For that purpose, the M lastly received audio frames can be used to compute the signal model. The lastly received audio frames $\mathbf{x}_{m-1}$ to $\mathbf{x}_{m-M}$ are fed into a receiver buffer of size M used by the extrapolation module as shown in Fig. 3. If a network packet $\mathbf{x}_m$ has not arrived in time, the extrapolated audio frame $\hat{\mathbf{x}}_m$ can be used. To allow the application of $\hat{\mathbf{x}}_m$ it has to be computed in advance during the playback of frame $\mathbf{x}_{m-1}$. To guarantee smooth transitions to the next intact frame it is recommended to extrapolate sequences that are longer than a single audio frame size and apply a cross-fade with the next audio packet.

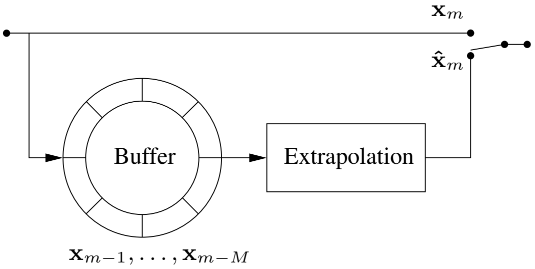


Figure 3: *Basic error concealment strategy*

The following section explains various approaches to obtain the mentioned model parameters a_i with the help of prediction.

3. PREDICTION TECHNIQUES

For the purpose of estimating an unknown future signal value of a given time series $x[n]$ *linear prediction* is typically applied. The predicted value $\hat{x}[n]$ is computed with the last known samples $x[n-1], x[n-2], \dots, x[n-p]$ and p predictor coefficients a_i as given by

$$\hat{x}[n] = \sum_{i=1}^{p} a_i x[n-i]\,. \tag{4}$$

The accuracy of the estimation using a predictor of order p can be assessed with the *forward prediction error* $e_p[n]$, also known as the residual

$$e[n] = x[n] - \hat{x}[n]\,. \tag{5}$$

The coefficients a_i should be chosen to achieve the minimum prediction error energy $e[n]^2$. Predictors can be implemented in several ways. In the following we classify different approaches in sample-based and block-based variants. Formulas and the pseudo code, used to compute the algorithmic complexity, are mainly based on [12].

3.1. Sample Based Approaches

3.1.1. Direct Form

The minimization of the prediction error energy can be achieved with the method of steepest descent. The expected value of the instantaneous prediction error energy is defined as the cost function $J = E\{e^2[n]\}$, which depends on the set of prediction coefficients $a_i[n]$. Those p coefficients

$$a_i[n+1] = a_i[n] - \frac{1}{2}\mu \frac{\partial J}{\partial a_i[n]},\ i = 1, ..., p. \tag{6}$$

are adapted by subtracting the negative derivative of this cost function, which is additionally weighted with the fixed step size $\frac{\mu}{2}$.

Applying the instantaneous error energy as the cost function $J = e^2[n]$ and solving the derivative leads to

$$a_i[n+1] = a_i[n] + \mu\, e[n]\, x[n-i],\ i = 1, ..., p. \tag{7}$$

Since the updated coefficient $a_i[n+1]$ depends on the cost function, describing the squared error, this algorithm is called Least

Mean Squares (LMS). A practical rule to guarantee the convergence of LMS is to choose μ between

$$0 < \mu < \frac{2}{\sum_{i=1}^{p} x^2[n-i]}, \tag{8}$$

where the numerator can be stated as the "tap-input power" [13]. Eq. (8) can also be computed continuously, leading to time-variant $\mu[n]$. This modification is called Power Normalized LMS (PNLMS) [14]. In addition to the well-known direct filter structure (Fig. 2), the sample-based prediction with lattice filters shall be shown in the next section.

3.1.2. Lattice Form

The lattice filter structure, illustrated in Fig. 4, contains a forward and backward branch with the corresponding forward and backward errors $f_i[n]$ and $b_i[n]$. These errors can be recursively computed in the following way

$$\begin{aligned} f_i[n] &= f_{i-1}[n] - k_i\, b_{i-1}[n-1] \\ b_i[n] &= b_{i-1}[n-1] - k_i\, f_{i-1}[n],\ i = 1, ..., p, \end{aligned} \tag{9}$$

where k_i are the so-called reflection coefficients comparable to the filter coefficients a_i of the direct form filter. The prediction error is minimized in every lattice stage. Therefore, the initial prediction errors are the input signal

$$f_0[n] = b_0[n] = x[n]. \tag{10}$$

Similar to the direct form filters, the method of steepest descent is applied

$$k_i[n+1] = k_i[n] - \frac{1}{2}\mu_i[n]\frac{\partial \hat{J}_i}{\partial k_i[n]},\ i = 1, ..., p. \tag{11}$$

Applying a modified cost function, describing the prediction error energy in i_{th} lattice stage, $\hat{J}_i = f_i^2[n] + b_i^2[n]$ yields

$$k_i[n+1] = k_i[n] + \mu_i[n]\ (f_i[n]\, b_{i-1}[n-1] + b_i[n]\, f_{i-1}[n]). \tag{12}$$

Similar to PNLMS, the state-dependent gradient weight can be replaced with a power normalized $\mu_i[n] = \frac{\alpha}{\sigma_i^2[n]}$, where α is a scaling factor and $\sigma_i^2[n]$ is the prediction error energy in the i_{th} lattice stage. It can be computed recursively like

$$\sigma_i^2[n] = \lambda\, \sigma_i^2[n-1] + (1-\lambda)\, (f_{i-1}^2[n-1] + b_{i-1}^2[n-1]), \tag{13}$$

where λ describes an attenuation factor controlling the influence of past values. This implementation of a predictor is called Gradient Adaptive Lattice (GAL). A noticeable advantage of GAL over LMS is the guaranteed stability for $|k_i| < 1$ and its fast convergence due to stage-dependent gradient weights $\mu_i[k]$. Additionally, it should be pointed out that the error is minimized using the forward and backward prediction error.

3.2. Block Based Approaches

The finding of optimal prediction coefficients for a block of length N requires the minimization of the prediction error using the whole block. Two common methods are presented in the following.

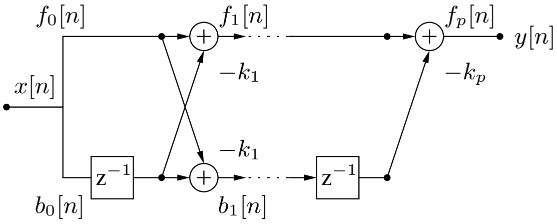


Figure 4: *Lattice filter implementation*

3.2.1. Autocorrelation-based

A main idea of the autocorrelation method (ACM) is to consider values outside the currently analyzed block to be zero by operating on a windowed block $u[n] = w[n]\, x[n]$. The prediction error energy of order p for this block can be described as

$$E = \sum_{n=0}^{N+p-1} e^2[n], \tag{14}$$

where the error $e[n]$ can be expressed with Eq. (4) and (5). The actual minimization of E is achieved by setting its derivative with respect to the desired prediction coefficients a_i to zero

$$\frac{\partial E}{\partial a_i} \overset{!}{=} 0. \tag{15}$$

Substituting (14) in (15) yields after several manipulations (see [13])

$$\begin{pmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(0) & \dots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \dots & r(0) \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = - \begin{pmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{pmatrix}, \tag{16}$$

where $r(i)$ are the autocorrelation coefficients, corresponding to $u[k]$, computed like

$$r(i) = \frac{1}{N}\sum_{l=i}^{N-1} u[l]u[l-i]. \tag{17}$$

These so-called Wiener-Hopf equations can be efficiently solved with the well-known Levinson-Durbin algorithm [15].

3.2.2. Burg Method

Global optimization within a block to compute an optimal set of model parameters can also be achieved with lattice filters. For that purpose, the sum of forward and backward error energy over all p lattice stages $J_i = \sum_{n=i}^{N-1} \left(f_i^2[n] + b_i^2[n]\right)$ is minimized in the same manner as for ACM. Therefore, the derivative of J_i with respect to k_i is set to zero and solved for k_i, yielding

$$k_i = \frac{2\sum_{n=i}^{N-1} (f_{i-1}[n]\, b_{i-1}[n-1])}{\sum_{n=i}^{N-1} \left(f_{i-1}^2[n] + b_{i-1}^2[n-1]\right)}. \tag{18}$$

This so-called "Burg formula" can recursively be used to compute the reflection coefficients k_i for all $i = 1, \dots, p$. After every computation of k_i all lattice stages have to be updated with the new reflection coefficients according to Eq. (9).

Table 1: *MUL's and ADD's of the proposed prediction algorithms*

	MUL	ADD
LMS	$3p + 2$	$3p + 1$
GAL	$10p$	$7p$
ACM	$p + \frac{p^2+5p+4N+2}{2N}$	$p + \frac{p^2-3p+2N-2}{2N}$
Burg	$5p - \frac{5p^2+p}{2N}$	$5p - \frac{5p^2+9p}{2N}$

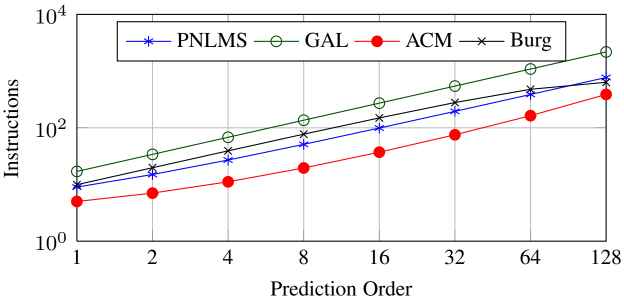


Figure 5: *Computational complexity in instructions per sample*

3.2.3. *Complexity*

The overall design of the desired AoIP system must consider the complexity of all involved modules. To evaluate the algorithmic complexity of the proposed predictors, the required instructions in the corresponding formulas were counted. The amount of operations for the instantaneously computed block-based methods was divided by N to be able to compare the averaged complexity per sample. Note, that divisions were treated like multiplications. Table 1 illustrates the amount of multiplications and additions per sample for the proposed predictors depending on the prediction order p. Plotting the overall sum of necessary operations over p and a fixed block length $N = 128$ reveals (see Fig. 5) that ACM is the cheapest method, exceeded by the similarly expensive PNLMS and Burg. Due to its expensive recursion, GAL is the most expensive of the analyzed predictors.

Implementing ACM and Burg on DSP-platforms, restricted to a certain amount of operations per sample, is not directly possible due to their unbalanced work load. A way of load balancing the necessary computations is shown in [12]. The unbalanced work load of the block-based methods is not an issue in most AoIP implementations since the audio data is fragmented in frames anyways and existing AoIP systems were implemented on platforms allowing the use of multi-threading for the purpose of load-balancing.

4. EXPERIMENTS

The proposed prediction schemes were extensively evaluated, following objective and subjective criteria. The measurement method and the listening test setup are explained in this section.

4.1. Measurements

The evaluation of audio quality with software tools instead of real listeners has always been a controversially discussed topic. Nevertheless, the psychoacoustically motivated Perceptual Evaluation of Audio Quality (PEAQ) [16] method is a widely-used measurement tool. It has to be fed with the signal under test and a reference signal. Multiple features like the noise-to-mask-ratio are computed within PEAQ to obtain an estimate for the perceptual quality of the signal under test. This estimate is expressed in the form of the so-called objective difference grade (ODG) score. This score reaches from -4, meaning very annoying, to 0, indicating an imperceptible difference between signal under test and reference. Although, the typical application of PEAQ is to assess the audio quality of lossy audio codecs, which introduce different artifacts than the analyzed extrapolation scheme, the authors assume that PEAQ is at least a suitable quality indicator for error concealment.

Table 2: *Used parameters in the evaluation with PEAQ*

Parameter	Minimum	Increment	Maximum
Packet Loss Rate	0.01	0.01	0.1
Block Length	64	2^{n+1}	512
Prediction Order	1	2^{i+1}	Block Length

The evaluation is implemented in Matlab as described below. A test item is loaded into the workspace and once copied as a reference. Then the test item is divided into frames of a fixed length and a fraction of those frames is deleted to imitate physical packet loss. Afterwards, the test item is processed frame-wise. Whenever an empty frame was detected, the $M = 3$ last frames were fed to the four proposed predictors and the extrapolation was computed with obtained model parameters a_i. The extrapolated block of length $\frac{3}{2}N$ is cross faded into the next frame to avoid discontinuities at the block edges. After completion of the test item the reference, an unconcealed error reference and the four concealed versions are written to wave files.

Finally, a self-implemented PEAQ tool is fed with a concealed version and the reference of the test item to obtain an ODG score. This procedure is repeated for various simulation parameters, listed in Table 2, and the complete SQAM [17] data set, which consists of various synthetic test signals, high-quality recordings of single instruments, speech and full mixes. The results are explained in the following section.

4.2. Results

First of all it shall be shown that extrapolation of audio data is highly signal-dependent in general. Figure 6 shows the ODG score for the various predictor schemes from Sec. 3 and an unprocessed error reference against every test item of the SQAM data set. The simulation parameters for this plot were a fixed block length as well as a prediction order of 128 and a packet loss rate of 0.01. Regarding Fig. 6 leads to an estimate for the following order of quality: Burg, ACM, GAL, PNLMS.

The mean ODG score of the error reference $\mu_{Err} = -2.33$ clearly characterizes the unconcealed audio material as annoyingly defective. It is noticeable that there is a large variance $\sigma^2_{Err} = 1.98$ in the error reference, indicating that the impact of the introduced error is highly signal-dependent. Certain test items seem to be nearly unaffected by the packet loss simulation. However, this effect can be explained by analyzing the actual test items and the rating criteria of PEAQ. For example, items $26 - 28, 36$ are recordings of instruments with a percussive characteristic with a very fast decay like castanets or a harpsichord. These items con-

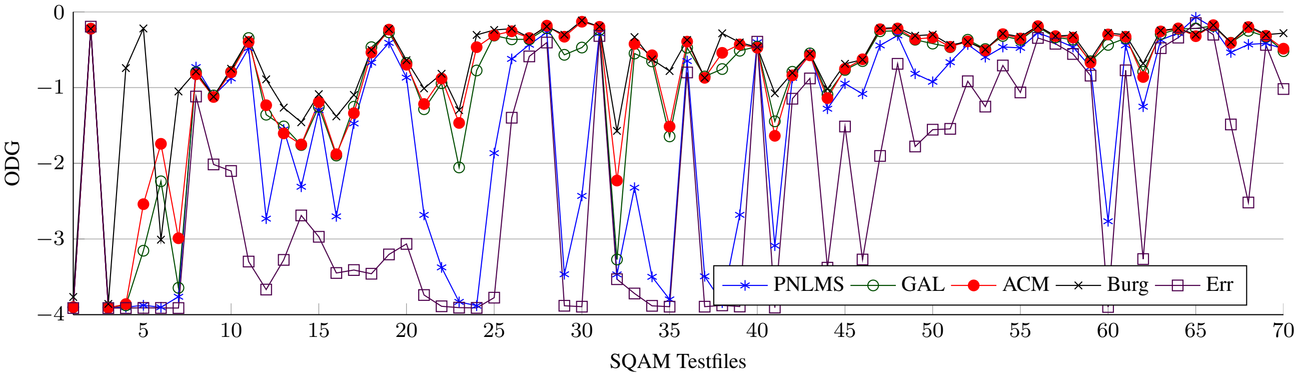
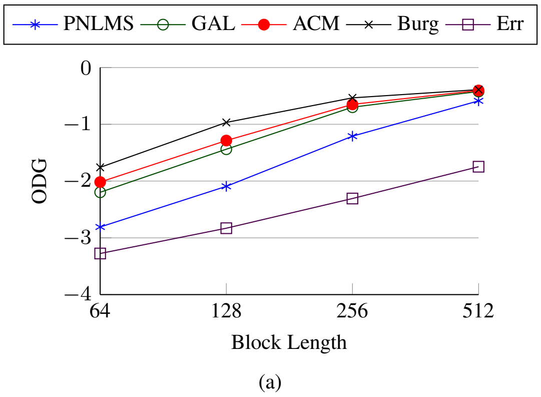
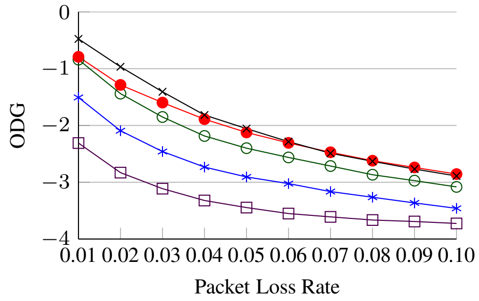


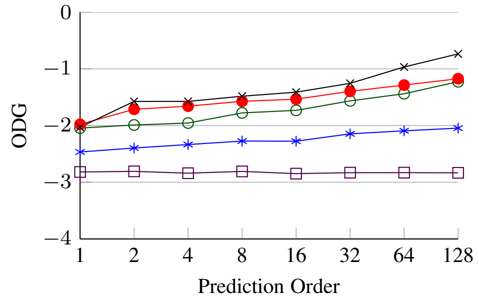
Figure 6: *ODG score over SQAM test files*



(a)



(b)



(c)

Figure 7: *ODG score over various simulation parameters*

tain much silence, increasing the probability of having the simulated packet loss within a silent section causing no audible artifact. Test items $44 - 54$ are speaking and singing voice examples, also featuring many calm passages. The SQAM data set includes some recordings consisting of many sound sources like orchestras or full mixes. The authors assume that PEAQ rates artifacts due to the packet loss less severe for wide-band signals than for narrow-band signal. The simulation of simultaneous and temporal masking effects within the PEAQ tool seem to be reasonable explanations. Indeed, artifacts in polyphonic sound material are known to be less noticeable than in monophonic material.

The mean and variance of the ODG scores decrease drastically when extrapolation is applied. The Burg-processed files only feature a variance of $\sigma^2_{Burg} = 0.32$ for instance. Both block-based methods deliver good results of $\mu_{ACM} = -0.74$ and $\mu_{Burg} = -0.44$ on average. Test items 3-7, which are synthetic test signals with a very fast amplitude modulation, seem to be very problematic for all predictors. The sample-based prediction schemes, especially the lattice predictor, still produce acceptable averaged results of $\mu_{PNLMS} = -1.38$ and $\mu_{GAL} = -0.75$. Nevertheless, the performance of the Burg predictor can not be achieved for most test items. In general, the lattice based predictors outperform the direct form predictors in almost every case. This is likely caused by the lattice predictor's property of minimizing the forward and backward prediction error and therefore, creating a more convenient signal model.

The influence of the used block length on the prediction performance is illustrated in Fig. 7a, where the ODG score, averaged over all SQAM test items, is plotted against it. As expected, an increased block length increases the prediction performance. In other words, a prolonged input sequence leads to a better model for the current signal. The sample-based methods seem to benefit even more from the increased input signal length since the corresponding curves are steeper, and therefore the relative improvement is larger, although the performance of the block-based methods are not reached.

The impact of the packet loss rate appears to be comparable for the analyzed prediction schemes. The curves in Fig. 7b, illustrating the ODG score for different packet loss rates, develop similarly. Certainly, the curves are clearly shifted depending on the already stated trend of quality. For instance, the scores for the Burg method are offset by about 0.5 against the GAL scores.

The ODG score gain achieved by increasing the prediction order is visualized in Fig. 7c. The Burg predictor seems to benefit most from a higher prediction order. Its curve shows a relative raise of 1, whereas the other predictors only improve by about 0.5.

4.3. Listening Test

Since the evaluation should respect subjective criteria as well, a simple MUSHRA [18] listening test was set up. Five items of the SQAM test set with different kinds of sound sources and characteristics were chosen: a cello, wind ensemble, male speech, vibraphone and a full mix. These test tracks were shortened to 10 seconds and processed like the tracks for the PEAQ measurements. The simulation parameters were set to a packet loss rate of 0.02, block length of 128, prediction order of 32, cross-fade with a cosine window over half a block. The small block length was chosen corresponding to simulate an AoIP system with a desirably small blocking delay, whereas the error rate was selected high to create a problematic network scenario, where error concealment is becoming essential.

The listening test was performed with mushraJS [19], a browser-based MUSHRA test tool, to easily reach many listeners. The 23 participants for this test were manifold. Researchers, audio professionals, musicians as well as unexperienced listeners took part. Five listeners produced irrelevant results, containing poorly rated hidden references for instance, that were excluded from the result set. The results per track, averaged over the listeners, are shown in Fig. 8. To begin with, the measured trend of quality is confirmed, revealing that the objective test with PEAQ was meaningful. There are no significant peculiarities for a single test track but the signal-dependency of the perceived error in the unprocessed material as well as the extrapolation quality is again observable.

For instance, the Burg predictor was rated $\mu = 72.7$ of 100 with a small standard deviation of $\sigma = 11.6$ for the vibraphone example, indicating that the listeners perceived the quality of the extrapolated track as acceptable without much uncertainty. In contrast, the cello track was only rated $\mu = 51.2$ with a clearly larger standard deviation of $\sigma = 17.01$. Hence, a larger disagreement of the listeners concerning the signal quality exists.

The small mean and standard deviation values of the error reference for the cello and the speech track of $\mu = 11.0, 14.0$ and $\sigma = 8.88, 10.72$ respectively, demonstrates that the majority of the listeners perceived the artifacts in this narrow-band content as very annoying. The corresponding results for the vibraphone and the Abba track of $\mu = 22.4, 25.2$ and $\sigma = 15.4, 17.6$ are clearly elevated. This again proves the reduced sensitivity to audible artifacts in wide-band audio material, as already pointed out in the previous section.

The most outstanding discrepancy between the objective measurements and the subjective listening test results is the superiority of the Burg method. A closer look on Fig. 6 and Fig. 8 reveals that the Burg method achieved a clearly larger relative advantage over ACM in the averaged listening test result than in the averaged PEAQ measurements.

5. INCLUSION IN AOIP FRAMEWORK

To test the real-time capability of the extrapolation with the most promising prediction scheme - the Burg method - the corresponding MATLAB code was ported to C/C++ and integrated in an already existent, proprietary AoiP system. The client of this AoIP software allows sending to and receiving from multiple other clients. It consists of several classes to handle the internal functionalities like audio, network, receiver buffers, and the front-end allowing easy control.

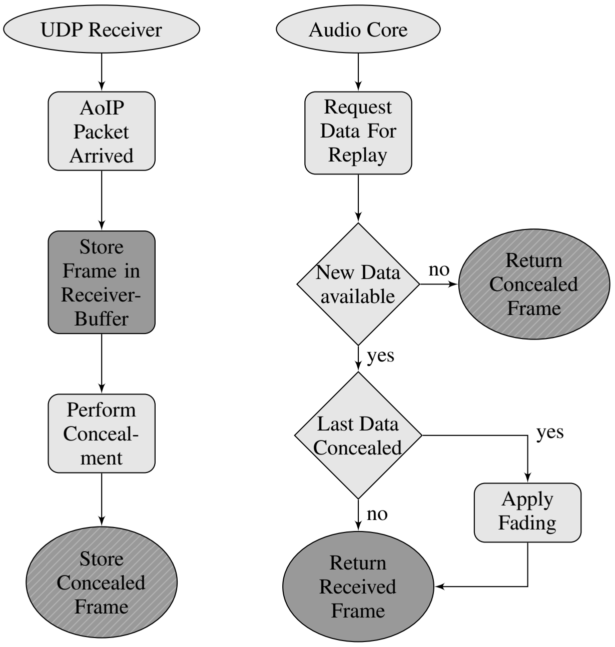


Figure 9: *Flow chart of concealment mechanism in AoIP software*

The actual concealment process in the software is illustrated in Fig. 9. Whenever an intact AoIP packet arrives at one of the UDP sockets, which is bound to a specific sender of another client, it is written to a free slot in the corresponding receiver buffer (marked in dark gray). The entire or just a fraction of the receiver buffer can then be used to perform the actual concealment using the techniques described above. The result is stored in a separate buffer (shaded dark gray). Note, that the extrapolated audio frame is longer than the actual audio frames of size N. For this implementation, the overlap was chosen to be half the block length $N_{OL} = \frac{1}{2}N$.

Every time the audio core gets activated, and the audio processing callback is triggered, a stereo mix of the current audio frames of all receivers is written to the output buffer of the sound card. Therefore, the audio core requests data from all receiver buffers. If a receiver buffer does not contain new data, it will return the last concealed audio frame truncated to length of N. When new data is available but the last returned block of data was concealed, it is necessary to perform a cross-fade between the current received frame and the extended fraction of the last concealed frame for N_{OL} samples. This is required to allow smooth transitions between regular and concealed frames, resulting in glitch-free audio. In the case of new available data and a correctly received last data block, the current audio frame can be returned without further processing.

Computing the prediction coefficients for several input audio data streams is computationally expensive. A simple test setup on a modern notebook with an Intel i3-3120M DualCore processor and AoIP settings of 8 active receivers, using 3 concealment input data blocks of length 256 samples, and a prediction order of 32 reveals that the AoIP tool requires about 70 % CPU load using the proposed concealment implementation. If the concealment is disabled only 12 % are necessary. Although the computational effort is large, the extrapolation is done in real-time allowing massive sound quality improvement for erroneous network conditions.

6. CONCLUSIONS AND PROSPECTS

The problem of faulty audio material due to non-optimal network conditions in an AoiP scenario is discussed in this paper. Kauppinen's approach of audio extrapolation seems to be a legit way to perform error concealment in the case of packet loss or critical packet delay. The basic idea of the extrapolation technique and its integration in a possible AoIP framework is presented. Four different prediction approaches to obtain a signal model for the extrapolation process are investigated. The performance of the predictors is evaluated in detail following objective and subjective criteria. The objective assessment is done with a large-scale automated measurement with PEAQ over the SQAM data set. To judge the subjective impact of the prediction technique a listening test with 23 participants was applied. The conclusions of both evaluation strategies are consistent. The block-based methods (Burg and ACM) appear to be very promising for the described purpose, whereas the sample-based methods can't be recommended, due to their adverse performance (PNLMS) or their algorithmic complexity (GAL). The results demonstrate the typical trade-off between performance and computational complexity. ACM is clearly the cheaper block-based prediction strategy but objectively and subjectively worse rated than the Burg method, which was outstandingly judged best by the listeners for all test items.

A possible implementation within a real-time AoIP software is demonstrated. The high-quality concealment for multiple receivers is achievable in real-time but the computations are extensive and therefore, consume a majority of the CPU resources. Optimizing the implementation and analyzing simplifications should be done for inclusion in other projects. Typically, AoIP systems include audio codecs to reduce the transmission data rate. Embedding the proposed method within a prediction based audio codec might lead to systems of higher efficiency, due to shared usage of prediction filter coefficients. Perceptual audio codecs operate on frequency-domain representations of the source signal. This additional information could also be utilized for enhanced concealment schemes.

7. REFERENCES

[1] D. Konstantas and Y. Orlarey, "The distributed musical rehearsal environment," *Multimedia, IEEE*, pp. 54–64, 1999.

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

[4] A. Renaud, A. Carôt, and P. Rebelo, "Networked music performance: State of the art," in *Proceedings of the AES 30th International Conference*, Saariselkä, Finland, 2007, number 4, pp. 1–7.

[5] A. Carôt, C. Werner, and T. Fischinger, "Towards a Comprehensive Cognitive Analyisis of Delay-Influenced Rhytmical Interaction," in *Proceedings of the International Computer Music Conference (ICMC 2009)*, Montreal, Canada, 2009.

[6] A. Carôt and C. Werner, "External latency-optimized soundcard synchronization for applications in wide-area networks," in *AES 14th Regional Convention, Tokio, Japan*, Tokyo, Japan, 2009.

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

[8] 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.

[9] R. McAulay and T.F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 4, pp. 744–754, 1986.

[10] A. Stenger, K. B. Younes, R. Reng, and B. Girod, "A new error concealment technique for audio transmission with packet loss," in *in Proc. EUSIPCO*, 1993, pp. 1965–1968.

[11] 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, number 1, pp. 105–110.

[12] F. Keiler, D. Arfib, and U. Zölzer, "Efficient linear prediction for digital audio effects," in *Proc. of the 15 th Int. Conference on Digital Audio Effects (DAFx-12)*, Verona, Italy, 2000, pp. 1–6.

[13] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, Second edition, 1991.

[14] G. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least squares adaptive algorithms for fir transversal filtering," *Signal Processing Magazine, IEEE*, vol. 16, no. 4, pp. 13–41, 1999.

[15] J. Durbin, "The fitting of time-series models," *Revue de l'Institut International de Statistique*, pp. 233–244, 1960.

[16] 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 18th 2013.

[17] European Broadcast Union, "EBU SQAM CD," Website, Available online at `tech.ebu.ch/publications/sqamcd` visited on March 18th 2013.

[18] International Telecommunication Union, "Bs.1534: Method for the subjective assessment of intermediate quality levels of coding systems," Website, Available online at `http://www.itu.int/rec/R-REC-BS.1534`; visited on March 18th 2013.

[19] S. Kraft, "mushrajs," Website, Available online at `https://github.com/seebk/mushraJS` visited on March 18th 2013.

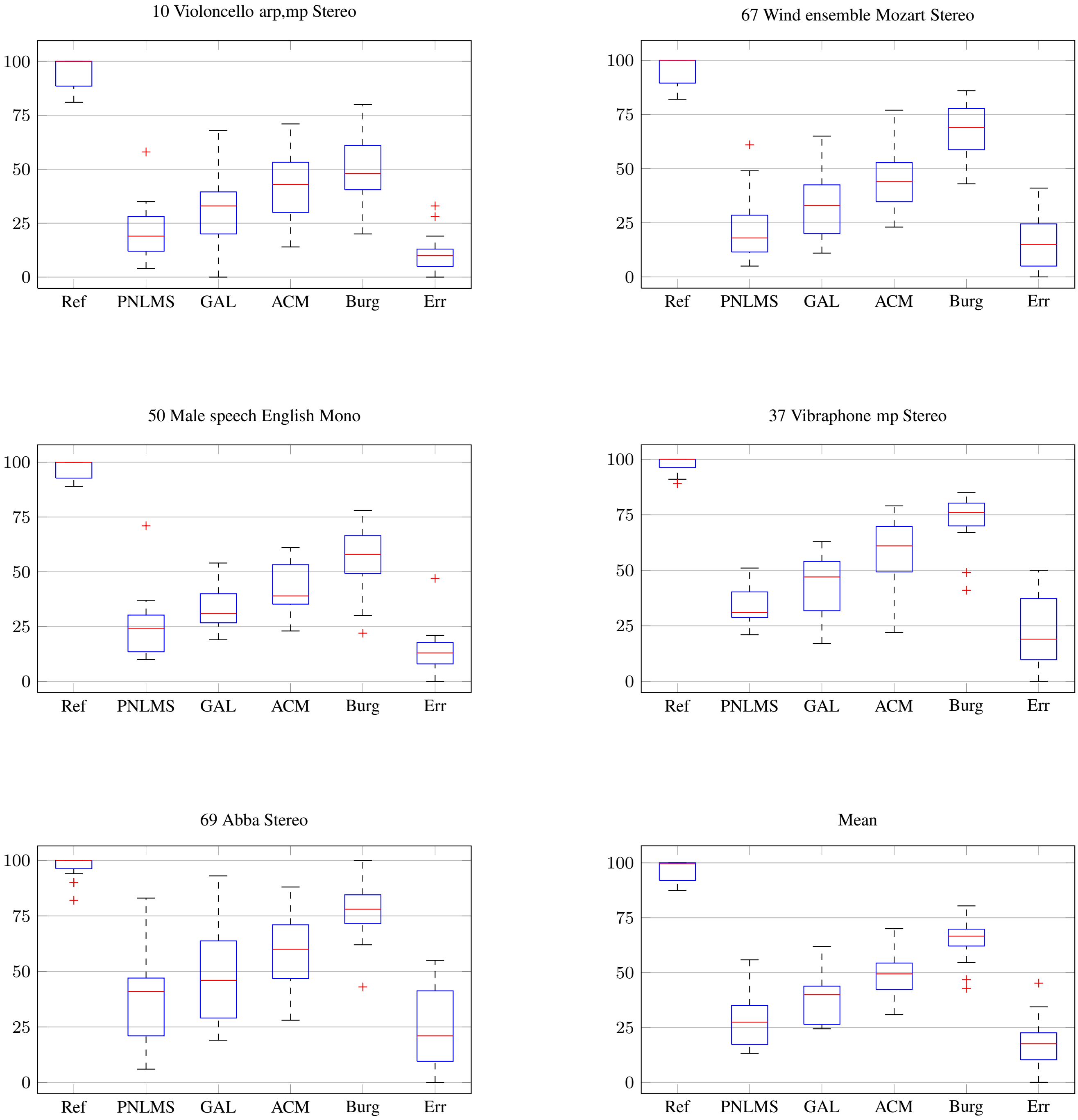


Figure 8: *Results of MUSHRA listening test*