DELAY-FREE AUDIO CODING BASED ON ADPCM AND ERROR FEEDBACK

Martin Holters, Christian R. Helmrich, Udo Zölzer

Dept. of Signal Processing and Communications,
Helmut Schmidt University — University of the Federal Armed Forces
Hamburg, Germany
martin.holters|udo.zoelzer@hsuhh.de, c.helmrich@ecodis.de

ABSTRACT

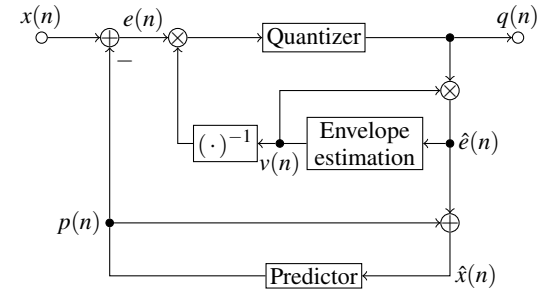
Real-time bidirectional audio applications, like microphones and monitor speakers in live performances, typically require communication systems with minimum latency. When digital transmission with limited bit rate is desired, this poses tight constraints on the algorithmic delay of the audio coding scheme. We present a delay-free approach employing adaptive differential pulse code modulation (ADPCM) and adaptive spectral shaping of the coding noise. To achieve zero-delay operation, both prediction and quantization logic of the ADPCM structure are realized in a backward-adaptive fashion. Noise shaping is accomplished via two feedback loops around the quantizer for efficient exploitation of the auditory selectivity and masking phenomena, respectively. Due to automatic optimization of the involved parameters, the performance of the proposed system is on par with that of prior low-delay approaches.

1. INTRODUCTION

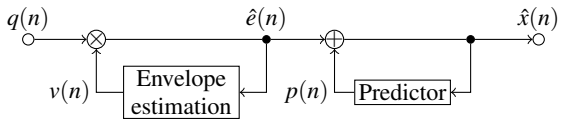
Contemporary perceptual audio codecs such as MPEG-1 Layer 3 (MP3) and MPEG-4 (AAC), due to block-wise processing of the input waveform, introduce an algorithmic latency of at least 100 ms between the original signal and the decoded output [1]. In off-line applications such as the preparation of digital recordings for efficient storage, this delay generally does not represent an issue. In real-time bidirectional communication, however, delays above roughly 10 ms are likely to compromise performance [2]. Härmä and Laine [3] proposed a total codec latency of 2–5 ms for "live" situations. Given that the above codecs are not suitable for real-time use, different approaches are required. Recent developments include MPEG-4 AAC-ELD [4] and Fraunhofer's ULD [5] with delays of 15 and 6 ms, respectively. It is also possible to achieve delay-free operation by use of linear predictive coding (LPC), a technique commonly used for speech coding since the 1970s [6]. Because of their inability to obtain detailed spectral information about the input, however, LPC codecs typically fall behind their transform-based counterparts in terms of attainable audio quality for a given bit rate.

2. THE PROPOSED SYSTEM

The coding system proposed in this paper consists of a delay-free backward-adaptive ADPCM codec and feedback noise shaping. The ADPCM codec exploits the signal's statistics to reduce the distortion when requantizing to a lower bit rate. The resulting coding error is nearly white noise. By feeding the error back to the input through suitably chosen filters, the coding noise may be spectrally shaped to reduce its audibility.



(a) Encoder.



(b) Decoder.

Figure 1: *Structure of the ADPCM codec.*

2.1. The ADPCM codec

A fairly standard ADPCM encoder and decoder similar to the ones from [7], [8] and [9] has been employed in this work, see figure 1. From the input $x(n)$ to the ADPCM encoder, a prediction $p(n)$ is subtracted. The resulting prediction error $e(n) = x(n) - p(n)$ is then divided by an estimate $v(n)$ of its expected absolute value. This normalized prediction error is finally quantized to the reduced bit rate.

The value $q(n)$ associated with the quantization index is then multiplied with $v(n)$ in both decoder and encoder to obtain the reconstructed prediction error $\hat{e}(n) = v(n) \cdot q(n)$. From this $\hat{e}(n)$, the level estimate $v(n+1)$ to be used for the next sample is determined. By further adding the predicted value back to the prediction error, we obtain the reconstructed signal $\hat{x}(n) = \hat{e}(n) + p(n)$, again in both decoder and encoder. The predictor uses $\hat{x}(n)$ to calculate the prediction $p(n+1)$ for the next input sample. As the same values of $\hat{e}(n)$ and $\hat{x}(n)$ are used in encoder and decoder to update $v(n)$ and $p(n)$, respectively, it is not necessary to transmit any side information, provided that the same initial states are used. The overall error due to coding, as can easily be verified, equals the quantization error scaled by $v(n)$ and is approximately white [10].

The level or envelope estimation is performed by low-pass filtering $\hat{e}^2(n)$ according to

$$v^2(n+1) = \max\left(v_{min}^2,\, \lambda \hat{e}^2(n) + (1-\lambda) v^2(n)\right) \qquad (1)$$

and taking the square root of the result. The coefficient λ controls the cut-off frequency of the low-pass and hence determines how fast the estimated envelope follows the signal, while v_{min} sets a lower bound on the envelope to avoid using excessive gains in the codec during periods of silence. To minimize quantizer overload when the prediction error power suddenly increases, the low-pass switches between two coefficients λ_{AT} and λ_{RT} corresponding to two cut-off frequencies: when $\hat{e}^2(n) > v^2(n)$, a higher cut-off frequency is used to facilitate fast adaptation, while otherwise, a lower cut-off frequency is used to achieve a smoother $v(n)$ for stationary segments.

The predictor is realized as an FIR lattice filter

$$f_m(n) = f_{m-1}(n) - \kappa_m(n) b_{m-1}(n-1) \tag{2}$$

$$b_m(n) = b_{m-1}(n-1) - \kappa_m(n) f_{m-1}(n) \tag{3}$$

$$f_0(n) = b_0(n) = \hat{x}(n), \tag{4}$$

where

$$p(n) = \hat{x}(n) - f_M(n) = \sum_{m=1}^{M} \kappa_m(n) b_{m-1}(n-1) \tag{5}$$

yields the desired prediction. The lattice filter structure has the important advantage that it can easily be ensured to be minimum-phase by enforcing $|\kappa_m(n)| < 1$, a prerequisite for the stability of the feedback structure employed.

The coefficients are adapted using the gradient adaptive lattice (GAL) algorithm [11]

$$\kappa_m(n+1) = \kappa_m(n) + \mu_m(n) \cdot \left(f_m(n) b_{m-1}(n-1) + f_{m-1}(n) b_m(n) \right), \tag{6}$$

which is very attractive because of its low computational complexity. The step size $\mu_m(n)$ is derived from a base step size $\tilde{\mu}$ by stage-wise power normalization [12]

$$\mu_m(n) = \frac{\tilde{\mu}}{l_m(n) + l_{min}}, \tag{7}$$

$$l_m(n) = (1-\tilde{\mu}) \cdot l_m(n-1) + \tilde{\mu} \cdot \left(f_{m-1}^2(n) + b_{m-1}^2(n-1) \right), \tag{8}$$

where l_{min} is a small constant to avoid division by zero. In practice, after updating κ_m according to equation (6), $|\kappa_m| \leq 1 - \varepsilon$ is enforced to ensure filter stability in the feedback structure, with ε being a small constant.

The quantizer is non-uniform symmetric. Each input sample is quantized to the nearest codebook entry q_i. No dithering is used, as the signal to be quantized is approximately white, so that no severe harmonic distortion is to be expected.

2.2. The feedback noise shaper

The ADPCM system takes advantage of any redundancy present in the audio signal to minimize the power of the coding noise induced by the requantization. However, the coding noise is white, and its audibility can be further reduced by appropriate spectral shaping, even at the cost of increasing its power. While in [8] and [9], adaptive pre- and post-filtering were used to achieve the spectral shaping, in this paper, noise feedback shall be employed.

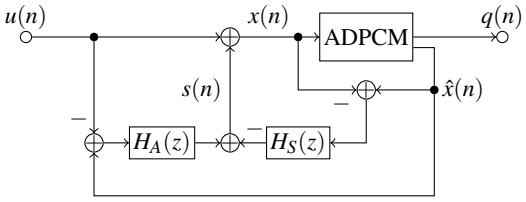


Figure 2: *Structure of the noise shaper around the ADPCM codec.*

As depicted in figure 2, the noise shaper used consists of two filters, $H_S(z)$ and $H_A(z)$. If $e(n) = \hat{x}(n) - x(n)$ denotes the error introduced by the ADPCM codec, then

$$\hat{X}(e^{j\Omega}) = U(e^{j\Omega}) + \frac{1 - H_S(e^{j\Omega})}{1 - H_A(e^{j\Omega})} E(e^{j\Omega}), \tag{9}$$

that is, the error is spectrally shaped by

$$H(z) = \frac{1 - H_S(z)}{1 - H_A(z)}. \tag{10}$$

To obtain a realizable system without delay-free loops, both filters $H_S(z)$ and $H_A(z)$ will be free of a direct path.

The denominator filter $H_A(z)$ is adapted so as to exploit the simultaneous masking of the original signal and shape the noise spectrum accordingly. To reduce computational complexity, the masking threshold is approximated by the simple heuristic of using a smoothened version of the signal spectrum itself. Conveniently, the spectral information implicitly present in the prediction coefficients $\kappa_m(n)$ may be reused by applying another lattice filter

$$\bar{f}_m(n) = \bar{f}_{m-1}(n) - \kappa_m(n) \bar{b}_{m-1}(n-1) \tag{11}$$

$$\bar{b}_m(n) = \beta \cdot \left(\bar{b}_{m-1}(n-1) - \kappa_m(n) \bar{f}_{m-1}(n) \right) \tag{12}$$

$$\bar{f}_0(n) = \hat{x}(n) - u(n) \qquad \bar{b}_0(n) = \beta \left(\hat{x}(n) - u(n) \right) \tag{13}$$

giving the first component of the shaping signal

$$s_A(n) = \sum_{m=1}^{M} \kappa_m(n) \bar{b}_{m-1}(n-1). \tag{14}$$

The additional factor $\beta \in [0,1]$ scales the resulting poles compared to the all-pole model of the linear predictor, thereby smoothing the spectrum and avoiding excessive noise shaping which might result in undesirable peaks in the noise spectrum.

The noise shaping achieved by this simple adaptation scheme has two particular disadvantages. First, for reasonable choices of the filter order M, the resolution at the low end of the spectrum is, psychoacoustically, insufficient. As a consequence, partial unmasking of the coding error, and thus audible low-frequency noise, is likely to occur, as reported in [13, 14, 15]. Second, the considerably reduced hearing sensitivity at very high frequencies is not exploited. A certain amount of noise energy can be shifted to the upper end of the spectrum before the audibility threshold is reached and before the overall codec performance is negatively affected. Fortunately, both issues can be remedied by applying a second, static filter $H_S(z)$ as

$$s_S(n) = \sum_{l=1}^{L} h_S(l) \cdot \left(\hat{x}(n-l) - x(n-l) \right) \tag{15}$$

with coefficients such that $1 - H_S(z)$ attenuates low frequencies at the cost of amplifying high frequencies.

Combining both components then gives the shaping signal $s(n) = s_A(n) - s_S(n)$ which is added to the original signal $u(n)$ to yield the input signal $x(n)$ of the ADPCM codec. Note that the decoder is not modified when introducing error-feedback noise shaping, since $\hat{u}(n) = \hat{x}(n)$ already is the correct reconstruction.

3. PARAMETER OPTIMIZATION

The proposed coding system has a wide range of parameters that can be adjusted. All of these can have a significant impact on the audio quality delivered by the system for a given bit rate. In fact, the choice of parameters determines whether the coding system is usable at all. Unfortunately, the optimal parameter values in terms of achieved audio quality cannot be derived analytically. Instead, a search in the parameter space is required. As the parameter space has too many dimensions for manual trial-and-error style optimization, automating this search is desirable.

The first prerequisite for automated optimization is a suitable cost function. For the case at hand, this requires choosing audio test material and a method to evaluate the achieved audio quality after coding and decoding. As the test material, we have chosen the EBU sound quality assessment material (SQAM) [16], featuring a total of 70 test items including artificial signals, single instrument, ensemble and voice recordings as well as pop music excerpts. The resulting audio quality was determined for each item by itself using the PEAQ method [17, 18], giving an objective difference grade (ODG) between 0 (no audible distortion) and -4 (very annoying distortion). To get a single value per parameter vector, the fourth powers of the ODGs are averaged. Taking the fourth power strongly emphasizes items of low quality. The motivation is that an audio codec with good performance for any signal is more universally useful than one with excellent performance for most, but failing miserably for some.

The optimization consists of a global search followed by a local search. The global search is based on simulated annealing, the local search is conducted according to the Rosenbrock method.

3.1. Simulated annealing

Originally, simulated annealing is a technique for discrete optimization, however, it can easily be modified for continuous parameter spaces [19]. Let $\boldsymbol{r}$ denote the current parameter vector of the search and $C(\boldsymbol{r})$ the associated cost function as described above. Then, a new tentative parameter vector is constructed as

$$\boldsymbol{r}^* = \boldsymbol{r} + \Delta\boldsymbol{r}, \tag{16}$$

where $\Delta\boldsymbol{r}$ is a random vector with zero mean and covariance $\boldsymbol{S}$ to be discussed shortly. The tentative vector $\boldsymbol{r}^*$ is accepted as the new parameter vector $\boldsymbol{r}$ with probability

$$P = \min\left(1, \exp\left(-\frac{C(\boldsymbol{r}^*) - C(\boldsymbol{r})}{T}\right)\right). \tag{17}$$

That is, an $\boldsymbol{r}^*$ resulting in better quality is always accepted, while decreases in quality are only accepted with decreasing probability. The parameter T, the temperature, is lowered in the course of the optimization, so that accepting worse parameter vectors becomes increasingly improbable. The covariance $\boldsymbol{S}$ is determined from past accepted solutions such that the steps $\Delta\boldsymbol{r}$ taken depend on the local topology of the cost function in a reasonable way.

For suitable choices of initial temperature T and the cooling scheme with which T is decreased, convergence to the optimal solution is guaranteed. However, the cost function employed would take too long to evaluate for such a cooling scheme. We therefore reduce T faster, which no longer delivers the optimal solution, but usually reaches a sufficiently good solution in an acceptable amount of time.

3.2. The Rosenbrock method

As the simulated annealing does not necessarily converge to the optimal solution, a local search starting at the best parameter vector evaluated during the simulated annealing can further refine the result. We apply the Rosenbrock method [20], as it does not require the cost function's gradient.

Again starting from a parameter vector $\boldsymbol{r}$, a tentative parameter vector

$$\boldsymbol{r}^* = \boldsymbol{r} + \Delta\boldsymbol{r}_i \tag{18}$$

is formed and accepted as new $\boldsymbol{r}$ only if it results in a quality improvement. The $\Delta\boldsymbol{r}_i$ is chosen out of a set of orthogonal vectors. If $\boldsymbol{r}^*$ is accepted, the length of the respective $\Delta\boldsymbol{r}_i$ is increased, otherwise it is decreased. This is repeated iteratively for all $\Delta\boldsymbol{r}_i$ until in each direction, at least one $\boldsymbol{r}^*$ was accepted and one was rejected. After such a round, the complete step taken in the whole process is used as $\Delta\boldsymbol{r}_1$ and the remaining $\Delta\boldsymbol{r}_i$ are determined by Gram-Schmidt orthogonalization. The process in started over with the new set of $\Delta\boldsymbol{r}_i$, where the length of $\Delta\boldsymbol{r}_1$ may be used as termination criterion.

4. EVALUATION

Parameter optimization and evaluation were performed using mono downmixes of the SQAM material. The sampling rate of the material was kept at 44.1 kHz. For the bit rate, both 3 bit and 4 bit per sample were examined, resulting in 132.3 kbit/s and 176.4 kbit/s, respectively, for a single channel.

To evaluate not only the quality of the complete coding system, but also the impact of the noise shaping, in a first optimization run, the noise shaper was disabled to give parameters for the ADPCM codec only. We constrained $\varepsilon = 10^{-8}$ and $v_{min} = l_{min}$ to reduce search space dimensionality without sacrificing too much flexibility. The ADPCM parameters were then kept fixed for the optimization of the noise shaper, where the order of the static noise shaper was set to $L = 8$.

The resulting parameters of this two-stage optimization are listed in table 1. Not surprisingly, the quantizer codebook is denser for lower values. The time constants of the envelope estimation are relatively high, which allows the coding systems to handle transient sounds without excessive quantizer overload. The predictor order $M = 79$ for the 4 bit per sample case is somewhat high, but thanks to the GAL algorithm, it should not stand in the way of an efficient implementation. As expected, the magnitude transfer function of the static noise shaper, depicted in figure 3, attenuates low-frequency components at the cost of amplifying higher frequencies. The scaling factors $\beta = 0.28$ or $\beta = 0.49$ used in the adaptive noise shaper result in a significant smoothing of the spectrum.

The ODGs for the same items for which listening tests were conducted in [7] are shown in figure 4. It should be noted that

Table 1: *Codec parameters used for evaluation (negative entries q_i of the symmetric codebook omitted for brevity).*

Parameter	for 3 bit / sample	for 4 bit / sample
q_1	1.46159×10^{-01}	1.08515×10^{-01}
q_2	5.49513×10^{-01}	3.00403×10^{-01}
q_3	1.04466	5.21187×10^{-01}
q_4	1.80110	7.87807×10^{-01}
q_5		1.00004
q_6		1.31301
q_7		1.73766
q_8		2.69500
λ_{AT}	8.35779×10^{-01}	8.65607×10^{-01}
λ_{RT}	9.87227×10^{-02}	1.05238×10^{-01}
M	49	79
$\tilde{\mu}$	2.08112×10^{-03}	2.07903×10^{-03}
$v_{min} = l_{min}$	1.46494×10^{-05}	1.44400×10^{-05}
ε	1.00000×10^{-08}	1.00000×10^{-08}
$h_S(1)$	4.95363×10^{-01}	4.02220×10^{-01}
$h_S(2)$	4.06237×10^{-02}	1.34312×10^{-01}
$h_S(3)$	2.45061×10^{-02}	-1.92533×10^{-02}
$h_S(4)$	6.51422×10^{-04}	2.40213×10^{-03}
$h_S(5)$	2.73930×10^{-01}	1.22069×10^{-01}
$h_S(6)$	-1.04735×10^{-01}	6.56308×10^{-04}
$h_S(7)$	-2.52869×10^{-03}	-2.32595×10^{-02}
$h_S(8)$	2.17942×10^{-01}	3.22201×10^{-01}
β	4.91936×10^{-01}	2.84474×10^{-01}

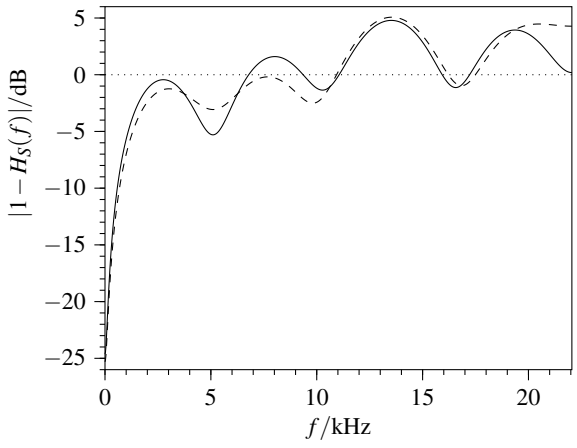


Figure 3: *Magnitude transfer function of the static noise shaping filter $1 - H_S(z)$ for 3 bit per sample (- -) and 4 bit per sample (—).*

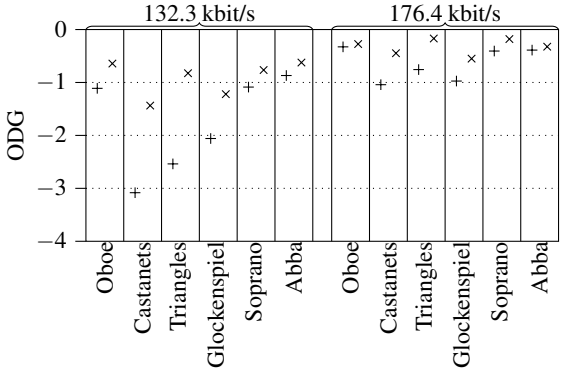


Figure 4: *Objective evaluation results for the coding system without (+) and with (×) noise shaping.*

these items include the castanets, triangles and glockenspiel signals which are particularly problematic for the ADPCM codec. The results for 132.3 kHz/s are almost on par with the approach proposed in [7], where admittedly the codec was operated at only 128 kbit/s, but on the other hand, was not delay-free.

Adding the noise shaping shows a significant improvement in audio quality for the lower bit rate across all signal types. For the higher bit rate, adding the noise shaping still shows an improvement, alas a smaller one. The reason for this rather small improvement is that the ADPCM codec shows worst performance for transients, while for stationary segments, at the higher bit rate the noise is low enough to be almost inaudible. But for transients, the adaptive noise shaper does not help much as it is not adapted fast enough, leaving only the static noise shaper.

Of course, the ODGs were the cost function of optimization, so the proposed codec is tuned for this objective evaluation instead of real human perception. However, informal listening tests confirm that the results of figure 4 match with subjective evaluation.

5. CONCLUSIONS

We have presented a delay-free audio coding approach based on backward-adaptive ADPCM and noise-shaping feedback. An efficient realization of the noise shaping filter was achieved through separation into a signal-adaptive structure, derived from the continuously adapted prediction filter, and a time-invariant structure with a high-pass characteristic. Thanks to an automatic tuning of the numerous parameters of the codec, the achieved audio quality is comparable to that of prior low-delay coding systems while the computational complexity remains very moderate. Only for very transient signals, our delay-free codec is at a disadvantage due to its inability to "look ahead in time". Nonetheless, in the trade-off between bit rate, audio quality and coding latency, the proposed approach represents a viable option when minimum coding delay is of high priority.

6. REFERENCES

[1] M. Lutzky, G. Schuller, M. Gayer, U. Krämer, and S. Wabnik, "A guideline to audio codec delay," in *Proc. 116th AES Convention*, Berlin, May 2004, AES.

[2] G. Schuller and A. Härmä, "Low delay audio compression using predictive coding," in *Proc. ICASSP '02*, Orlando, May 2002, IEEE, vol. 2, pp. 1853–1856.

[3] A. Härmä and U.K. Laine, "Warped low-delay CELP for wideband audio coding," in *Proc. 17th AES Int. Conf. on High Quality Audio Coding*, Florence, Sept. 1999, AES.

[4] M. Schnell, R. Geiger, M. Schmidt, M. Jander, M. Multrus, G. Schuller, and J. Herre, "Enhanced MPEG-4 low delay AAC — low bitrate high quality communication," in *Proc. 122nd AES Convention*, Vienna, May 2007, AES.

[5] S. Wabnik, G. Schuller, J. Hirschfeld, and U. Krämer, "Reduced bit rate ultra low delay audio coding," in *Proc. 120th AES Convention*, Paris, May 2006, AES.

[6] J.L. Flanagan, M.R. Schroeder, B.S. Atal, R.E. Crochiere, N.S. Jayant, and J.M. Tribolet, "Speech coding," *IEEE Trans. Commun.*, vol. 27, no. 4, pp. 710–737, Apr. 1979.

[7] F. Keiler, "Real-time subband-ADPCM low-delay audio coding approach," in *Proc. 120th AES Convention*, Paris, May 2006, AES, Paper 6748.

[8] M. Holters, O. Pabst, and U. Zölzer, "ADPCM with adaptive pre- and post-filtering for delay-free audio coding," in *Proc. ICASSP '07*, Honolulu, Apr. 2007, IEEE.

[9] M. Holters and U. Zölzer, "Delay-free lossy audio coding using shelving pre- and post-filters," in *Proc. ICASSP '08*, Las Vegas, Apr. 2008, IEEE, pp. 209–212.

[10] J.D. Gibson, S.K. Jones, and J.L. Melsa, "Sequentially adaptive prediction and coding of speech signals," *IEEE Trans. Commun.*, vol. 22, no. 11, pp. 1789–1797, Nov. 1974.

[11] L.J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. ICASSP '77*, Boulder, May 1977, IEEE, vol. 2, pp. 683–686.

[12] C.J. Gibson and S. Haykin, "Learning characteristics of adaptive lattice filtering algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 6, pp. 681–691, Dec. 1980.

[13] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 1, pp. 63–73, Feb. 1979.

[14] B.S. Atal and M.R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 3, pp. 247–254, June 1979.

[15] R. Bastian, "Subjective improvements in DPCM-AQ performance based on adaptive noise shaping," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 5, pp. 1067–1071, Oct. 1981.

[16] EBU Tech. 3253-E, "Sound quality assessment material. Recordings for subjective tests," Apr. 1988.

[17] ITU Recommendation ITU-R BS.1387-1, "Method for objective measurements of perceived audio quality," Nov. 2001.

[18] P. Kabal, "An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality," Tech. Rep., Department of Electrical & Computer Engineering, McGill University, Montreal, Dec. 2003.

[19] D. Vanderbilt and S.G. Louie, "A monte carlo simulated annealing approach to optimization over continuous variables," *J. Comput. Phys.*, vol. 36, pp. 259–271, 1984.

[20] H.H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Comp. J.*, vol. 3, no. 3, pp. 175–184, 1960.