Audio Engineering Society
Convention Paper
Presented at the 112th Convention
2002 May 10–13 Munich, Germany

Improved compression of DSD for Super Audio CD

Reiss, J. D.[1], Reefman, D.[2], Janssen, E.[2] and Sandler, M. B.[1]

[1] *Philips Research Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands*
[2] *Queen Mary, University of London, Mile End Road, London E14NS United Kingdom*
Correspondence should be addressed to derk.reefman@philips.com

ABSTRACT

A method is presented for improving current coding efficiency in DSD signals. The goal of this work is to explore new compression techniques which are tailored to the DSD format and which are meant to complement the current lossless DST compression practice used for SACD. The new technique builds on principles illustrated in previous papers. The method makes use of the highly oversampled character of DSD. Example implementations and results have been obtained. Losses to stability and signal-to-noise ratio have been measured and their audio effects have been minimised and quantified. Lower bounds are established on the compression ratio of these methods. This is viewed as a first step for a potentially constant bitrate compression scheme.

INTRODUCTION

Recently, Philips and Sony have devised and implemented a new audio storage format known as Super Audio Compact Disc, or SACD. At the core of this format is its primary enabling technology, Direct Stream Digital (DSD). DSD is a new recording format that employs 1-bit oversampling sigma-delta modulation. Whereas traditional compact discs use 16 bit PCM encoding at 44.1kHz, DSD uses 1-bit sampling of audio at 64x44.1kHz. Thus, DSD requires four times the data to record the same amount of time. To support this additional data requirement, the Super Audio CD has a 4.7 Gigabyte layer that holds both a 2-channel stereo, and a 6 channel multichannel recording. For a typical 74 minutes recording, this would require about 12 Gigabytes of data storage. This is accomplished using a lossless coding scheme referred to as Direct Stream Transfer (DST), which involves data framing, prediction and entropy encoding. The coding gains that are achieved in practical situations allow this amount of DSD to be stored on a single disk [1]. Compact Discs, on the other hand, fit 74 minutes of 2 channel audio into approximately 780 Megabytes. Thus, despite vastly improved storage technology, there has been no advantage in the total playback time of the audio. All benefits have been to audio quality and additional functionality.

Furthermore, DSD and related sigma delta modulation based systems may see use in areas such as internet audio streaming. Should DSD become popular as a consumer distribution format, one would expect there to be a demand for internet streaming of a DSD encoded signal, rather than conversion to 16 bit PCM with all the losses that that might entail. This would call for constant bit rate compressed DSD, which is in contradiction with the lossless

DST compression scheme that is currently applied to DSD[2] which is inherently not constant bit rate.
Then there is the issue of storing a DSD encoded signal on other media, such as hard disk drives. For 6 channel DSD, this would require 12 times as much storage space as that of a stereo 16 bit Wave file.
For all these reasons, compression becomes a primary concern. The goal of the work presented here is to explore some new compression techniques, which are tailored to the DSD format and which are meant to complement the current DST compression practice in that they are not lossless. The main aim of this paper is to establish lower bounds on the compression ratio of these methods. This is to be viewed as a first step for a potentially constant bitrate compression scheme.

BACKGROUND

Sigma-delta (or delta-sigma) modulation is a popular method for high-resolution A/D and D/A converters. Sigma-delta modulators operate using a trade-off between oversampling and low-resolution quantization. That is, a signal is sampled at much higher than the Nyquist frequency, typically with one bit quantization, so that the signal may be effectively quantized with a resolution on the order of 14-20 bits[3]. Recent work has concentrated on tone suppression[4], multibit modulation[5] and chaotic modulation[6, 7].
The simplest, first order sigma-delta modulator consists of a 1-bit quantizer embedded in a negative feedback loop that also contains a discrete-time integrator. The analogue output is sampled at a frequency higher than the Nyquist frequency and is converted into a binary output. The system may be represented by the map[8]

$$U(n) = U(n-1) + X(n-1) - Q(n-1)$$

where X represents the analogue input signal and Q is the quantizer

$$Q(n) = \begin{cases} 1 & \text{if } U(n) \ge 0 \\ -1 & \text{if } U(n) < 0 \end{cases}$$

In this representation, $Q(n)$ represents the quantization of input $X(n-1)$. The actual quantised output is converted into binary data, 1 and 0 (as opposed to 1 and –1, respectively). This system works by quantizing the difference between the input and the accumulated error. Thus when the error grows sufficiently large, the quantizer will flip in order to reduce the error. On average, the quantization output will be approximately equal to the input. Higher order modulators are typically used in commercial applications since they often yield improved signal-to-noise ratios. However, the essential structure: oversampling, quantization, and noise shaping, remains the same.
In this paper, we only consider oversampling and quantization that is relevant to Direct Stream Digital. Thus, a 1 bit quantizer is used, and, unless otherwise noted, all simulations were performed with 64 times oversampling of signals with Nyquist frequency no greater than 44.1kHz. However, the sigma delta modulator used in DSD implements a complicated 5th order sigma delta modulator and incorporates other sophisticated technologies. In order to concentrate on the essential properties of our compression scheme, we chose to investigate a simpler sigma delta modulator. First and second order modulators are inappropriate models because they do not exhibit the instability problems commonly found in higher order modulators. Thus our analysis concentrated on a 3rd order sigma delta modulator, which may be implemented as depicted in Figure 1.
This results in the following difference equations.

$$\begin{aligned} U(n) &= 20I_1(n) + 6I_2(n) + I_3(n) \\ I_3(n+1) &= I_3(n) + I_2(n) \\ I_2(n+1) &= I_2(n) + I_1(n) \\ I_1(n+1) &= I_1(n) + X(n) - Q(n) \end{aligned}$$

PULSE GROUP MODULATION

The average pulse repetition frequency (PRF) is defined as the reciprocal of the average time between consecutive rising edges of the pulse stream. The PRF of the output of a sigma-delta modulator depends on the oversampling ratio L, the sampling frequency f_s and the composition of the limit cycles in the output. The maximum possible pulse repetition frequency of a SDM is $L \cdot f_s / 2$, which occurs for the repeating limit cycle 1,-1,1,-1,1... A straightforward method of reducing the Pulse Repetition Frequency of the SDM bitstream and forcing it to be constant is to group together samples with the same sign, so that after the sample-and-hold the transitions are reduced. This technique is sometimes referred to as pulse group modulation (PGM) [9]. The output is divided into frames of length N and the samples in each frame are reordered so that all the 1s occur in a single group at the end. For instance, if N=4, then the sequence 0100101101101101 would become 0001,0111,0011,0111.
PGM may be applied selectively, and a number of different implementations of PGM are described in [9]. Typically, the errors introduced by PGM are shaped by the use of an additional feedback loop. However, PGM may also be applied as a post-processing procedure. In which case there is no PGM feedback and the pulse grouping may be applied directly on the output bitstream.

Post-Processing PGM

To analyze the effect of post-processing PGM, consider a sequence $v(n)$ of 1-bit data from the SDM. At each sample instant, the sum of the present sample and the previous N-1 samples is taken.

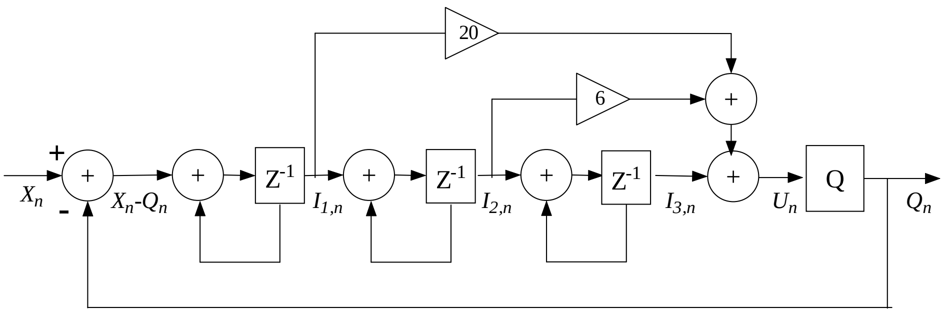

Figure 1. A block diagram of the third order sigma delta modulator used in simulations.

$$y(n)=\sum_{k=0}^{N-1}v(n-k)$$

Or equivalently, in the *z*-domain:

$$Y(z)=\sum_{k=0}^{N-1}V(z)z^{-k}=V(z)\left(\frac{1-z^{-N}}{1-z^{-1}}\right)$$

$$M(z)=Y(z)/V(z)=\left(\frac{1-z^{-N}}{1-z^{-1}}\right)$$

Thus the summation is equivalent to a moving average filter of length *N*. Every N^{th} sample of the summation corresponds in amplitude to the group size of each PGM pulse. The operation of taking every N^{th} sample and discarding the remaining samples is that of decimation, and the conversion to a pulse group is uniformly sampled pulse width modulation (PWM), which involves a sample rate increase by a factor *N*. The decimation produces aliassing and the PWM introduces harmonic distortion, carrier and sideband tones and intermodulation noise.

Noise Shaping PGM

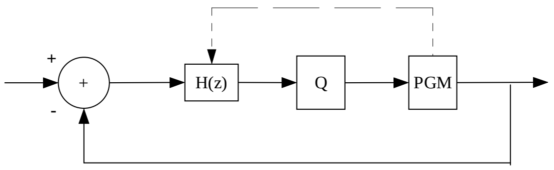

Figure 2. The structure of PGM applied to a sigma delta modulator with noise shaping.

To overcome the problems associated with PGM applied as a `post-processing' step, Magrath and Sandler proposed a feedback loop after the PGM module[9]. This structure is depicted in Figure 2. This generic structure applies to sigma delta modulators of any order. A specific implementation can be summarized as follows:

1. At each clock cycle, store the integrator states;
2. After *N* cycles, replace the *N* bits with a fixed pattern of -1 and +1 (PGM), and re-calculate the integrator states as if the SDM gave this output.
3. Continue with these new integrator states.

The recalculation of the integrator states serves as noise shaping and error correction of the effects of PGM, although it is also a potential cause of instability.

For the case of the third order modulator described previously, the filter H is made up by the cascade of integrators whose output is summed with weights 20, 6 and 1 to form the quantizer (Q) input.

Window Size	Maximum Input	SNR (dB)
1	0.83	75
2	0.69	65
3	0.56	62
4	0.47	58
5	Not Stable	

Table 1. Maximum input and signal-to-noise ratio for full noise-shaping PGM.

Table 1 depicts how stability and signal quality are affected by PGM. Both decrease until a window size of 5 is used. At which point the design becomes unstable. Modifications of the modulator coefficients can permit both higher maximum input and improved SNR. These modifications are justified because the extra delays in the feedback loop due to PGM change the nature of the noise shaping.

Several ways exist to apply such feedback in a control loop; and other examples of such systems can be found [10]. A system in the spirit of the architecture advocated by the latter authors can be built as shown in Figure 3.

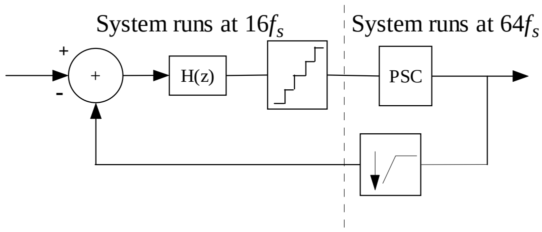

Figure 3. Block diagram depicting an equivalent system to noise shaping PGM.

The basic principle of operation is that a sigma delta modulator running at a low oversampling ratio of, say, 16, is accommodated with a 5-level quantizer. The code that is produced by this quantizer is subsequently passed through a parallel to series convertor (PSC), which also functions as a PGM block. The PSC translates the 5 level code to a series of 4 equally weighted bits, which hence run at a rate of 64 fs.

After the PSC, feedback is applied to the input of the system. In this feedback path, down sampling must be applied as the input of the system runs at 16 fs only. This downsampling function must be such that aliasing components in the baseband are small. On the other hand, if the delays in the feedback path are too large, instability of the system occurs quickly and renders it useless.

Window Size	Maximum Input	SNR (dB)
1	0.83	75
2	0.79	68
4	0.67	46
8	0.52	32
16	Not Stable	

Table 2. Maximum input and signal-to-noise ratio for downsampling PGM.

In Table 2, results are presented depicting how stability and SNR are affected as a function of PGM window size for this system, Figure 3, Clearly, stability is increased due to the modified structure. This is likely due to the fact that errors take longer to accumulate due to the downsampling. Although the maximum input is improved, the signal-to-noise ratio remains in roughly the same range, and in fact deteriorates at a slightly quicker rate.

Adaptive PGM

Noise shaping PGM can lead to increased instability. PGM also introduces complications such as SNR degradation. Thus another grouping procedure has been proposed. The concept behind this procedure, is to minimise the application of PGM while still shaping its effects to compensate for aliasing and distortion. Therefore it is only applied when it would have the largest impact on the compression rate. In this situation, only the bit combination with an equal number of 1s and –1s in the output is reordered. All other bit combinations have no grouping applied. This is equivalent to saying that PGM is applied only on windows where the sum of the bitstream is zero. The procedure is as follows.

1. At each clock cycle, store the integrator states;
2. After N cycles, calculate the integrated signal;
3. If it is zero, replace the N bits with a fixed pattern of -1 and +1, and re-calculate the integrator states as if the SDM gave this output.
4. Continue with these new integrator states, or with the old ones if the previous set of N was not totaling zero.

Table 2 depicts how stability and signal quality are affected by adaptive PGM. This should be contrasted with Table 1. Stability and signal-to-noise ratio both compare favorably with those achieved under noise-shaping PGM. This is further indication that there is no one method of pulse group modulation that is preferred. The choice of PGM that is applied should be determined by the constraints of the system, and the desired compression ratio.

Window Size	Maximum Input	SNR
1	0.83	75
2	0.75	71
3	0.68	68
4	0.61	64
5	0.56	59
6	Not Stable	

Table 2. Maximum input and signal-to-noise ratio for adaptive PGM.

COMPRESSION USING PULSE GROUP MODULATION

In previous work [11], it was shown that the expected compression ratio of random data using post-processing PGM or noise shaping PGM is

1. Compression Ratio (random signal)= $\dfrac{N}{-\sum_{i=0}^{N}\frac{\binom{N}{i}}{2^N}\log_2\frac{\binom{N}{i}}{2^N}}$,

and the worst case compression ratio for nonuniform data is

2. Compression Ratio (worst case)= $\dfrac{N}{\log_2(N+1)}$.

where N is the length of the applied PGM window.

It should be noted here that the worst case compression ratio is an extreme example, since it would require all possible output bit combinations to occur with equal probability. For instance, if N=8, then it would require that the probability of an N bit sequence containing no 1s must be the same as it containing 4 1s. Sequences with such a property have to be carefully constructed, and sliding the window over by just one bit would, in almost all cases, destroy this unusual nature.

The case of adaptive PGM needs to be treated differently. This is because PGM is applied in only certain situations. That is, unordered bit combinations may exist in the output that could not occur in post-processing or full noise shaping PGM. The choice of which combinations to reorder is dictated by the effects of reordering on the compression ratio.

Consider a window of length N in the bitstream without PGM where we assume a uniform and uncorrelated distribution. The number of possible outputs with exactly i 1s in the window is N choose i, or $\binom{N}{i}=\frac{N!}{(N-i)!i!}$. Combined, these outputs contribute $\frac{\binom{N}{i}}{2^N}\log_2\frac{\binom{N}{i}}{2^N}$ to the entropy of the bitstream. Thus, the combination that contributes to the most to the entropy is the one with an equal number of 1s and –1s in the output, which contributes exactly $\frac{\binom{N}{N/2}}{2^N}\log_2\frac{\binom{N}{N/2}}{2^N}$ to the entropy. If PGM is to be applied selectively, this is a logical choice for where it would have the most significant impact in increasing the Compression Ratio. Also, it is expected that the impacts on SDM stability are minimal in this case because combinations with an equal number of 1s and –1s typically occur at relatively small input values. As was accomplished in the case of full PGM, bounds on the compression ratio can be derived.

We can use the entropy formula to estimate the typical compression ratio achieved. The probability of any of the combinations with N/2 0s followed by N/2 1s occuring is $\binom{N}{N/2}\Big/2^N$. After PGM, there are $2^N-\binom{N}{N/2}+1$ possible combinations. Thus, the best expected number of bits required to encode N symbols is

$$H(P)=-\frac{\binom{N}{N/2}}{2^N}\log_2\frac{\binom{N}{N/2}}{2^N}-\sum_{i=1}^{2^N-\binom{N}{N/2}}\frac{1}{2^N}\log_2\frac{1}{2^N}$$

$$=-\frac{\binom{N}{N/2}}{2^N}\log_2\frac{\binom{N}{N/2}}{2^N}+N-\frac{N}{2^N}\binom{N}{N/2}=N-\frac{\binom{N}{N/2}}{2^N}\log_2\binom{N}{N/2}$$

…

Therefore, N-window PGM gives a best expected compression of

3. Compression Ratio (random signal)= $\dfrac{N}{N-\frac{\binom{N}{N/2}}{2^N}\log_2\binom{N}{N/2}}$

Of course, output prior to PGM is not expected to be completely random and uncorrelated. Thus we can consider the input which would result in random and uncorrelated output *after* PGM.

For a nonrandom sequence, the worst case for the compression ratio occurs when each of the $2^N-\binom{N}{N/2}+1$ combinations of bit orderings occurs with equal probability. Then the expected number of bits required to encode N symbols is $H(P)=\log_2(2^N-\binom{N}{N/2}+1)$, which provides a lower bound on an optimal compression scheme

4. Compression Ratio (worst case)= $\dfrac{N}{\log_2(2^N-\binom{N}{N/2}+1)}$

Furthermore, Stirling's formula, $N!\approx\sqrt{2\pi}e^{-N}N^{N+\frac{1}{2}}$, can be used to show that this partial compression is very limited in its effectiveness as the window size is increased.

$$\binom{N}{N/2}=\frac{N!}{(N/2!)(N/2!)}\approx$$

$$\frac{\sqrt{2\pi}e^{-N}N^{N+\frac{1}{2}}}{\sqrt{2\pi}e^{-N/2}(N/2)^{N/2+\frac{1}{2}}\sqrt{2\pi}e^{-N/2}(N/2)^{n/2+\frac{1}{2}}}$$

$$=\frac{e^{-N}N^{N+\frac{1}{2}}}{\sqrt{2\pi}(N/2)^{N+1}}=\frac{e^{-N}2^{N+1}}{\sqrt{2\pi N}}$$

So $2^N-\binom{N}{N/2}+1\approx 2^N(1-\frac{2e^{-N}}{\sqrt{2\pi N}})\to 2^N$ as $N\to\infty$.

Therefore, the worst case Compression Ratio approaches 1, which is the same as if there was no PGM applied. In fact, we can show that the expected compression also approaches 1 for signals which are random when PGM is not applied.

$$\frac{N}{N-\frac{\binom{N}{N/2}}{2^N}\log_2\binom{N}{N/2}} \approx \frac{N}{N-\frac{e^{-N}}{\sqrt{\pi N/2}}\log_2\frac{2^N e^{-N}}{\sqrt{\pi N/2}}} \to 1$$

These limiting cases serve to indicate a trend, but they are not applicable to any practical implementation. Any PGM scheme where the window size is larger than the oversampling ratio would have far too much noise introduced to be of use.

Window Size	PGM Compression Ratio		Adaptive PGM Compression Ratio	
	Random Signal	Worst Case	Random Signal	Worst Case
2	1.3333	1.2619	1.3333	1.2619
4	1.9698	1.7227	1.3199	1.1563
6	2.5714	2.1372	1.2905	1.0925
8	3.1444	2.5237	1.2591	1.0590
10	3.6949	2.8906	1.2443	1.0423

Table 3. Theoretical Compression Ratios for full and adaptive PGM.

Actual bounds on the compression ratio for each of these cases are given in Table 3 The expected compression ratio for a random signal, and the worst case compression ratio for a nonrandom signal are calculated for full PGM and adaptive PGM. These values are fairly low, but they indicate worst case scenarios and are still a significant improvement over lossless schemes (which would give a lower bound of 1 for the compression ratio of a random, uncorrelated signal.

Again its important to note that these are the upper bounds on the compression that can be achieved under such a distribution. Thus, Equations 1 and 3 referred to the best compression that PGM (post-processing or adaptive respectively) can achieve given that the bitstream would have a uniform, random and uncorrelated distribution if PGM had not been applied. Similarly, Equations 2 and 4 referred to the best compression that PGM can achieve given the worst case scenario for the distribution of the bitstream after PGM has been applied. Here, we have ignored the effects of noise shaping (which can improve compression) and of the use of inferior compression algorithms (which would lower the compression ratio).

RESULTS

In [11] it was confirmed that adaptive arithmetic coding algorithms can give compression ratios very close to those predicted from Table 3 for post-processing PGM of a random signal. Here, we consider how practical compression methods perform on more realistic data. In Table 4, the resulting compression ratios are given for the application of noise shaping PGM for two inputs each under a different lossless compression scheme.

First, a 4kHz sine wave of amplitude 0.25 was used as input to a third order modulator of the form given in Figure 3. The resulting bitstream was compressed using the Direct Stream Transfer algorithm that is implemented in the SuperAudioCD. The compression ratios are all quite high. With the exception of a window size *N*=2, the compression ratio increases with PGM window size. Then random input between 0.8 and –0.8 with rectangular pdf was used as input to the same modulator. The resulting bitstream was compressed using gzip, a popular compression utility based on the Lempel-Ziv algorithm [12]. The compression here still increased with window size, but was considerably lower than that of the sine wave.

	Compression Ratio		
Window Size	Sine Wave	noise-shaped random signal	Worst Case
1	3.3	1.48	1.00
2	3.1	1.84	1.26
4	4.2	1.79	1.72
8	5.4	2.86	2.52

Table 4. Compression ratio as a function of window size for sine wave input with the DST encoding algorithm and for a bandlimited noise shaped random signal with the gzip compression algorithm applied. The worst case compression ratios are also depicted for comparison.

Clearly, the compression ratios that were achieved were better than the worst case values. Sinusoidal input has far more structure than random input and thus gave very high compression ratios. However, in many cases the expected compression of a random bitstream with post-processing PGM applied actually outperformed the actual compression achieved with a random input signal. This implies that noise-shaping in some cases made compression more difficult, or that there were imperfections in the encoding algorithm. The latter is known to be correct at least in part, because the Lempel-Ziv algorithm typically performs slightly worse than compression algorithms such as arithmetic encoding [13]. The compression ratio is also dependent on the choice of sigma delta modulator, the specific implementation of the compression algorithm and any unknown structure in the input.

CONCLUSION

In this paper we have discussed techniques by which sigma delta bitstreams can be highly compressed. PGM based compression schemes are a potentially useful, lossy method of compressing the DSD bitstream. The losses in stability and in the signal-to-noise ratio can be compensated for by selectively applying PGM only when it is most beneficial or has the least impact.

In many cases, the signal-to-noise ratio drops to low values. This proves unacceptable for use with high quality formats as DSD. The attempted solution, adaptive PGM as proposed here, may, as yet, not be a sufficient remedy because it results in only minimal compression ratio gains. Still, the methods presented allow strong lower bounds to be derived, and therefore such schemes may be particularly useful in the development of constant bit-rate streaming of DSD audio. It is thus clear that further investigation is both necessary and warranted.

References

[1] D. Reefman and P. Nuijten, “Why Direct Stream Digital is the best choice as a digital audio format,” presented at 110th Audio Engineering Society Convention, Amsterdam, the Netherlands, 2001.

[2] F. Bruekers, W. Oomen, R. van der Vleuten, and L. van de Kerkhof, “Improved Lossless Coding of 1-Bit Audio Signals,” presented at 103rd Convention of the Audio Engineering Society, New York, 1997.

[3] W. Chou and R. M. Gray, “Dithering and its effects on sigma delta and multi-stage sigma delta modulation,” *IEEE Transactions on Information Theory*, vol. 37, pp. 500-513, 1991.

[4] A. J. Magrath and M. B. Sandler, “Efficient Dithering of Sigma-Delta Modulators with Adaptive Bit Flipping,” *Electronics Letters*, vol. 31, 1995.

[5] S. J. Park, R. M. Gray, and W. Chou, “Analysis of a sigma delta modulator with a multi-level quantizer and a single-bit

feedback," in *Proceedings of the ICASSP 91*. Toronto, Canada, 1991, pp. 1957-1960.

[6] O. Feely, "Nonlinear dynamics of discrete-time electronic systems," *IEEE Circuits and Systems Society Newsletter*, vol. 11, pp. 1-12, 2000.

[7] J. Reiss and M. B. Sandler, "The Benefits of Multibit Chaotic Sigma Delta Modulation," *CHAOS*, vol. 11, pp. 377-383, 2001.

[8] R. M. Gray, "Oversampled Sigma-Delta Modulation," *IEEE Transactions on Communications*, vol. COM-35, pp. 481-487, 1987.

[9] A. J. Magrath and M. B. Sandler, "Hybrid Pulse width Modulation / Sigma-Delta Modulation Power Digital-to-Analogue Converter.," *IEE Transactions on Circuits, Devices and Systems*, vol. 143, pp. 149-156, 1996.

[10] D. Birru, "Optimized reduced sample rate sigma-delta modulation," *IEEE Trans. Circuits Syst.*, vol. CAS-44, pp. 896-906, 1997.

[11] J. D. Reiss and M. B. Sandler, "Efficient compression of oversampled 1-bit audio signals," presented at Audio Engineering Society 111th Convention, New York, NY, USA, 2001.

[12] Ziv J., Lempel A., ``A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, Vol. 23, No. 3, pp. 337-343.

[13] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic Coding Revisited," *ACM Transactions on Information Systems*, vol. 16, pp. 256-294, 1998.