Improved compression of DSD for Super Audio CD

Reiss, J. D., Reefman, D., Janssen, E., and Sandler, M. B.

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 bitrate compressed DSD, which is in contradiction with the lossless
The actual quantized 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.

\[
U(n) = 20I_1(n) + 6I_1(n) + I_1(n) \\
I_1(n+1) = I_1(n) + I_1(n) \\
I_1(n+1) = I_1(n) + I_1(n) \\
I_1(n+1) = I_1(n) + X(n) - Q(n)
\]

**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,... 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 \(0100101101101\) 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.](image-url)
The structure of PGM applied to a sigma delta shaping.

Noise Shaping PGM

![Diagram](image)

To overcome the problems associated with PGM applied as a 'post-processing' step, Magath 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.

\[
y(n) = \sum_{k=-\infty}^{\infty} v(n-k)
\]

Or equivalently, in the z-domain:

\[
Y(z) = \sum_{k=-\infty}^{\infty} V(z) z^{-k} = V(z) \left( \frac{1-z^{-N}}{1-z^{-1}} \right)
\]

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

Thus the summation is equivalent to a moving average filter of length N. Every Nth sample of the summation corresponds in amplitude to the group size of each PGM pulse. The operation of taking every Nth 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 aliasing and the PWM introduces harmonic distortion, carrier and sideband tones and intermodulation noise.

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.

![Diagram](image)

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

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Maximum Input</th>
<th>SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.83</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>0.79</td>
<td>68</td>
</tr>
<tr>
<td>3</td>
<td>0.67</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>0.52</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>Not Stable</td>
<td></td>
</tr>
</tbody>
</table>

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.

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.

Adaptive 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 converter (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.
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 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.

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Maximum Input SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.83</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>0.68</td>
</tr>
<tr>
<td>4</td>
<td>0.61</td>
</tr>
<tr>
<td>5</td>
<td>0.56</td>
</tr>
<tr>
<td>6</td>
<td>Not Stable</td>
</tr>
</tbody>
</table>

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) =** 
   \[ \frac{N}{-\log_2(N+1)} \]
   
   and the worst case compression ratio for nonuniform data is

2. **Compression Ratio (worst case) =** 
   \[ \frac{N}{-\log_2(N)} \]

where \( N \) is the length of the applied PGM window.

It should be noted 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 \( 1 \)s must be the same as it containing 4 \( 1 \)s. 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

\[ \binom{N}{i} = \frac{N!}{i!(N-i)!} \]

Combined, these outputs contribute

\[ \frac{1}{2^N} \binom{N}{i} \log_2 \left( \frac{1}{2} \right) \]

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

\[ \frac{1}{2^N} \binom{N}{N/2} \log_2 \left( \frac{1}{2} \right) \]

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 \( N/2 \) 1s followed by \( N/2 \) 1s occurring is

\[ \binom{N}{N/2} \left( \frac{1}{2} \right)^N \]

After PGM, there are \( 2^N \left( \frac{N}{N/2} \right) \) possible combinations. Thus, the best expected number of bits required to encode \( N \) symbols is

\[ H(P) = \frac{1}{2^N} \log_2 \left( \frac{1}{2} \right) - \frac{1}{2^N} \sum_{i=1}^{N/2} \frac{1}{2^N} \log_2 \left( \frac{1}{2} \right) \]

\[ = \frac{1}{8} \log_2 \left( \frac{N}{N/2} \right) = \frac{1}{2^N} \log_2 \left( \frac{N}{N/2} \right) \]

... Therefore, \( N \)-window PGM gives a best expected compression of

3. **Compression Ratio (random signal) =**

\[ \frac{N}{-\log_2 \left( \frac{N}{N/2} \right)} \]

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 \( 2^N \left( \frac{N}{N/2} \right) \) combinations of bit orderings occurs with equal probability. Then the expected number of bits required to encode \( N \) symbols is

\[ H(P) = \log_2 \left( 2^N \left( \frac{N}{N/2} \right) \right) \]

...which provides a lower bound on an optimal compression scheme.

4. **Compression Ratio (worst case) =**

\[ \frac{N}{-\log_2 \left( 2^N \left( \frac{N}{N/2} \right) \right)} \]

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

\[ \frac{N}{(N/2)!} \approx \frac{1}{\sqrt{2\pi e} \cdot N^{N+1}/e} \]

\[ \sqrt{2\pi e} \cdot N^{N+1}/e \]

\[ e^{-N} \cdot N^{N+1} \]

\[ e^{-N} \cdot N^{N+1} \]

\[ \frac{1}{\sqrt{2\pi e} \cdot N^{N+1}/e} \]

So \( 2^N \left( \frac{N}{N/2} \right) \) combine pairs \( 0 \rightarrow 1 \) \( 1 \rightarrow 0 \), and

\[ \frac{N}{(N/2)!} \]

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.
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.

### 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.

<table>
<thead>
<tr>
<th>Window Size</th>
<th>PGM Compression Ratio</th>
<th>Adaptive PGM Compression Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Signal</td>
<td>Worst Case</td>
<td>Random Signal</td>
</tr>
<tr>
<td>2</td>
<td>1.3333</td>
<td>1.2619</td>
</tr>
<tr>
<td>4</td>
<td>1.9698</td>
<td>1.7227</td>
</tr>
<tr>
<td>6</td>
<td>2.5714</td>
<td>2.1372</td>
</tr>
<tr>
<td>8</td>
<td>3.1444</td>
<td>2.5237</td>
</tr>
<tr>
<td>10</td>
<td>3.6949</td>
<td>2.8906</td>
</tr>
</tbody>
</table>

CONCLUSION

In this paper we have discussed techniques by which sigma delta bitstream 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


