SDM versus LPCM: the debate continues

M.O.J. Hawksford
Centre for Audio Research and Engineering
University of Essex
UK CO4 3SQ

mjh@essex.ac.uk
http://esewww.essex.ac.uk/research/audio

Abstract: Significant misrepresentation of both 1-bit SDM and multi-bit LPCM coding paradigms persist within both professional and commercial arenas that impacts directly upon the perception of DVD-A and SACD formats. A balanced appraisal of these schemes is presented in order to expose the core differences in the technology both in the theoretical and instrumentation domains. Some observations are made about the fallacy of performance comparisons and the consequence of misinformation that subsequently is derived.

Abbreviations

ADC	analogue-to-digital converter	LPCM	linear pulse-code modulation
DAC	digital-to-analogue converter	PDF	probability density function
DSD	direct stream digital	PWM	pulse-width modulation
SDM	sigma-delta modulator	PZC	positive zero crossing
DVD_A	digital versatile disc (audio)	NPWM	naturally sampled pulse-width modulation
DVD_V	digital versatile disc (video)	SACD	super-audio compact disc
LFM	linear frequency modulation	UHRA	ultra-high resolution audio

1 Introduction

There has been considerable debate about the new high-resolution digital formats that embraces both theoretical signal processing aspects as well as the instrumentation of filter and conversion technology. This paper attempts to take a broad perspective of sigma-delta modulation (SDM) and linear pulse-code modulation (LPCM) with a focus on algorithmic design assuming transcoding from an ultra high-resolution audio format. A significant part of the theoretical consideration of SDM is based on an alternative method of encoding and this is investigated specifically with respect to linearity and decorrelation.

Our main thesis is that the fundamental differences between both the converters, coders and the signal formats that surround the technologies are not as different as some would believe especially when a realistic appraisal is made of how channel processing is performed. For example, it is not uncommon for the basis of a strong held opposition to the LPCM format to be focused on the supposition that LPCM is fundamentally flawed and "*creates stress reaction in people*" [1]. Also, it is conjectured that SDM bypasses these perceived problems because it operates with a 1-bit signal format and "*is the only digital recording system that does not to have these effects*" [1]. This paper emphasises that in the practical world such distinctions are not clear cut and that caution should be exercised when offering arguments based upon ideological concepts. It is probable that the perceived differences are more a function of system instrumentation than the underlying fundamental principles.

A major conclusion drawn is that given proper design, systems differences are relatively small and may become insignificant when real-world conversion technology and signal processing are introduced into the signal path, especially as many converters are now multi-bit structures.

2 Core philosophy of direct-stream digital and generalisations

The founding core philosophy of direct-stream digital (DSD) is that the signal from a uniformly sampled and streamed, 1-bit SDM analogue-to-digital converter (ADC) is retained entirely in the 1-bit domain and applied subsequently to the digital-to-analogue converter (DAC). In this scheme the output bit rate forms a uniformly sampled data structure and each bit assigned equal weight. Historically there was a period in time when state-of-the-art ADCs were implemented using SDM technology. Under these conditions it was expedient to retain the bitstream in its purest form especially where no intermediate signal processing was performed. In this restricted mode of operation the digital signal presented to the DAC is then identical to that generated at the output of the ADC and therefore can retain maximum fidelity, commensurate with ADC performance.

There is merit in storing directly the bitstream generated by the ADC especially if the converter is a state-of-the-art device, where for archival purposes maximum fidelity is then assured. It makes even greater sense when the ADC and DAC operates symmetrically and converts the bitstream directly to analogue without requiring format conversion or intermediate signal processing. SDM offers the unique property that the signal spectrum is contained directly within the bit pattern so that after jitter reduction, only a low-pass filter is required to recover the analogue signal. Hence, at the heart of this process is an intrinsic simplicity. However, similar arguments can be applied to other forms of converter technology especially those using a LPCM kernel with noise shaping, oversampling and decorrelation. The role of decorrelation (quantization level scrambling) in association with a thermometer style DAC is particularly important and offers a bridge between conventional LPCM with dither and 1-bit structures when considering real-world system imperfections. In practice, this particular LPCM structure forms a hierarchy of ADC where a basic ADC architecture is shown in Figure 2-1.

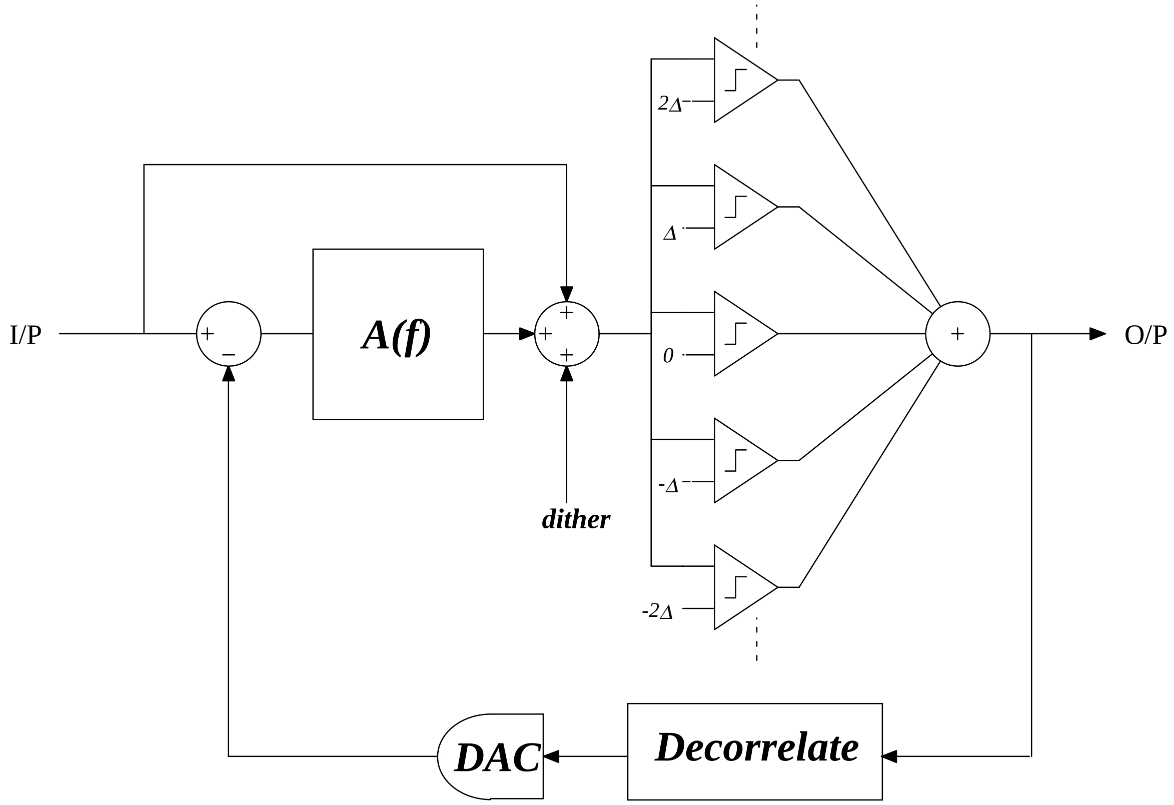


Figure 2-1 Generic UHRA noise shaped ADC with oversampling and decorrelation.

In this system the ADC generates multi-bit output code to represent the multi-level signals formed at the output of the uniform quantizer. So to maximise system transparency and by applying a similar philosophy to that of DSD, then the channel should convey the multi-bit samples from the ADC and not just a 1-bit transcoded representation. For archival applications this then guarantees maximum fidelity commensurate with the performance of the source ADC. The symmetry argument used with DSD can still apply to the multi-bit ADC and DAC where samples are converted directly to their analogue representation to mirror the ADC quantizer output without loss of information through intermediate signal processing either in the channel or the converters themselves.

Proposal

It is proposed that the transparent channel, 1-bit philosophy first mooted by Sony should be generalised to encapsulate multi-bit converter technology that uses oversampling, noise shaping and decorrelation processing.

In this paper a digital audio format that employs multi-level samples combined with optimum dither, noise shaping, a high oversampling ratio (e.g. 64-times Nyquist) and appropriate use of decorrelation techniques within the DACs is termed ultra-high resolution audio (UHRA). It is also strongly advised that the output code from the ADC should be scrambled with a random function to reduce the likelihood of jitter correlation resulting from practical system imperfections, this should be mandatory in a UHRA format. These features distinguish the approach from the limiting case of 1-bit SDM.

However, UHRA has a significant cost function in terms of channel capacity. DSD has a sampling rate f_{sdm} that is identical to the channel bit rate and has been chosen to be 2.8224 Mbit/s (i.e. 64 *44.1 kHz), where if the LPCM word length is increased from 1 to N bit, the channel bit rate becomes $N\ f_{sdm}$. Also, if optimum dither is used just prior to linear quantization then the principal criticism [2] of DSD is circumvented and the ADC can in theory be linearized. This latter observation is considered paramount and has been cited as the major problem in adopting DSD as a release format. However, add to this the importance of using decorrelation processing just prior to low resolution but high accuracy DAC then the advantages of DSD are incorporated but elevated to a higher performance level.

For a given bit rate there can be an exchange between bandwidth and LPCM word length, where the 1-bit case yields the maximum bandwidth albeit with the highest amplitude high frequency quantization noise. Consequently, if the bandwidth of the release format is to be retained then the sampling rate in the intermediate channel must match that of the DSD bit rate implying that a higher channel bit rate is mandatory for UHRA.

In practice it has been shown [2] that if the in-audio band signal-to-noise ratio is to be maintained (or bettered), then the multi-bit oversampled format can out perform DSD at a given bit rate although there is a reduction in signal bandwidth. The advantage of a properly dithered uniform quantizer has also been demonstrated where the system exhibits no correlated distortion artefacts unlike SDM. It is this latter area where strong criticism has been raised to DSD, where in the conventional implementation there are fundamental distortion mechanisms. This area is explored further in this paper.

3 The UHRA domain

It is important in establishing the differences between DSD and LPCM to appreciate that it may well become common practice to use a UHRA format in the intermediate recording channel located between the converters and release format transcoding. UHRA is well matched to many converters that now operate using oversampled and multi-bit architectures and also because signal processing can be performed with extreme accuracy while in the UHRA domain. The concept of performing signal processing in the 1-bitstream domain is problematic, that results in additional distortion products while actually requiring the generation of internal multi-bit processing, thus corrupting the philosophy on which the system is based.

In this sense although the core philosophy of DSD is now a historic relic, there has been a most welcome outcome in that signal-processing technology is currently being developed within the UHRA domain. In this domain the performance of the channel is considerably in advance of the much lower sampling rate LPCM processes. It is surmised that if it had not been for DSD then this development would probably not have occurred. This single fact may lead ultimately to much higher recording quality that is ideally aligned both to music archives as well as optimising the performance of a wide range of release formats.

There is a fundamental high-resolution audio concept to consider in the context of UHRA. A transparent system with very wide bandwidth and no internal correlated distortion processes will yield no audible signal coloration. Even though, converters may introduce signal coloration, this in theory can be engineered out; it is not fundamental to the concept of UHRA. However, all release formats other than UHRA data, must discard a fraction of the UHRA code and by the finite limitations of the transcoding mathematics introduce a degree of signal impairment.

It is in this context that DVD-A and DSD should be viewed. DVD-A requires sub-sampling of the UHRA data so the filters used may introduce some minor impairment, although their accuracy is under the control of the system designer. There may also be some minor psycho-acoustic implications by reducing signal rise times, although this is a controversial and as yet unproven conjecture. DSD in reducing the data from N-bit to 1-bit also encounters some fundamental problems associated with correlated distortion, so can not be a totally transparent system. However, the relaxation of filter specifications may be beneficial, although the substantial increase in high frequency noise and the need to filter this at the DAC output somewhat offsets the bandwidth advantage. Expedient use of pre-emphasis in the transcoder can in theory reduce this problem. Consequently, there are inevitable subtle differences between the release formats of DVD-A and DSD where some minor performance difference should be anticipated as each algorithm introduces mild coder-specific imperfections.

UHRA source code may also provide an input signal for all other forms of perceptual coder that in turn add their individual signature to the sound quality. Nevertheless, it is often a feature of coders that although they may add coloration to the sound, the quality of the source material can be detected. A possible scenario for multiple release formats is shown in Figure 3-1. However, it is an error in judgement to assume that because an audio signal is to be released in a data reduced format, that the quality of the audio processing up stream of the transcoder should be compromised.

Observing future trends, there is a need to develop loss-less encryption of UHRA code. A computationally efficient technique previously reported [3] used cascaded stages of differentiation together with a partial clipping correction technique to account for occasional transient overload. Efficient compression of the UHRA format not only has implications on archival storage but could also find application with higher capacity DVD such as blue-laser technology, where the high capacity should enable UHRA code to form the most transparent release format. As such the core philosophy of 1-bit DSD code is extended to a higher performance multi-bit format, where as well as offering an extreme resolution capability, eliminates the problem of correlated distortion.

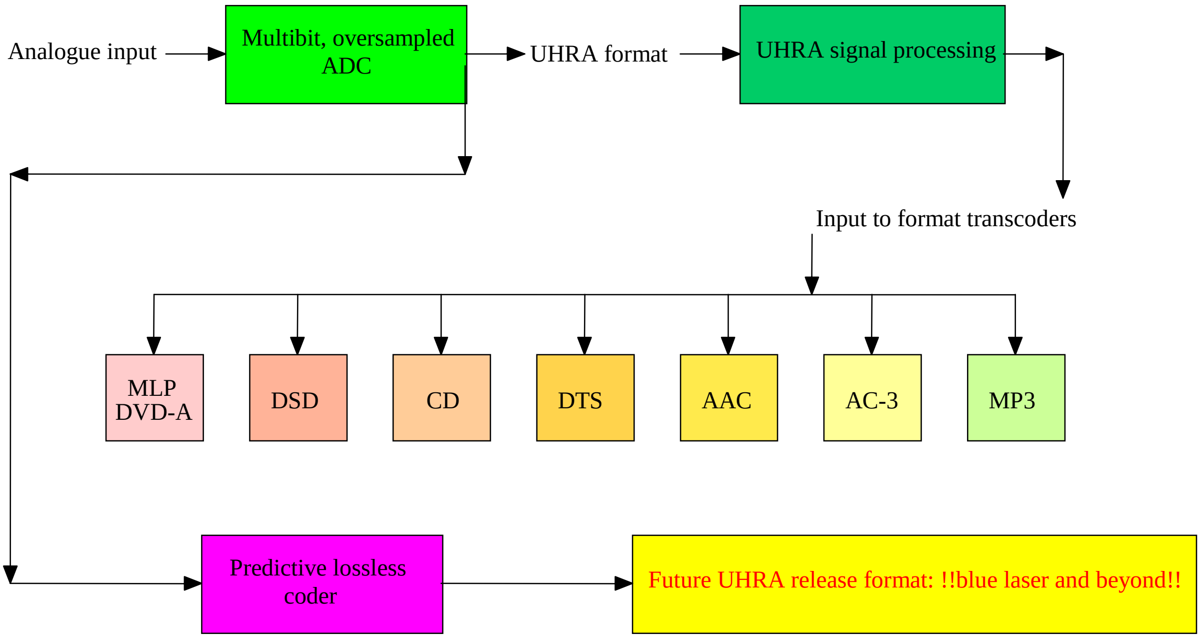

Analogue input
Multibit, oversampled ADC
UHRA format
UHRA signal processing
Input to format transcoders
MLP DVD-A
DSD
CD
DTS
AAC
AC-3
MP3
Predictive lossless coder
Future UHRA release format: !!blue laser and beyond!!

Figure 3-1 UHRA format and transcoding to multiple release formats.

4 1-bit encryption: Is there a fundamental problem of distortion correlation?

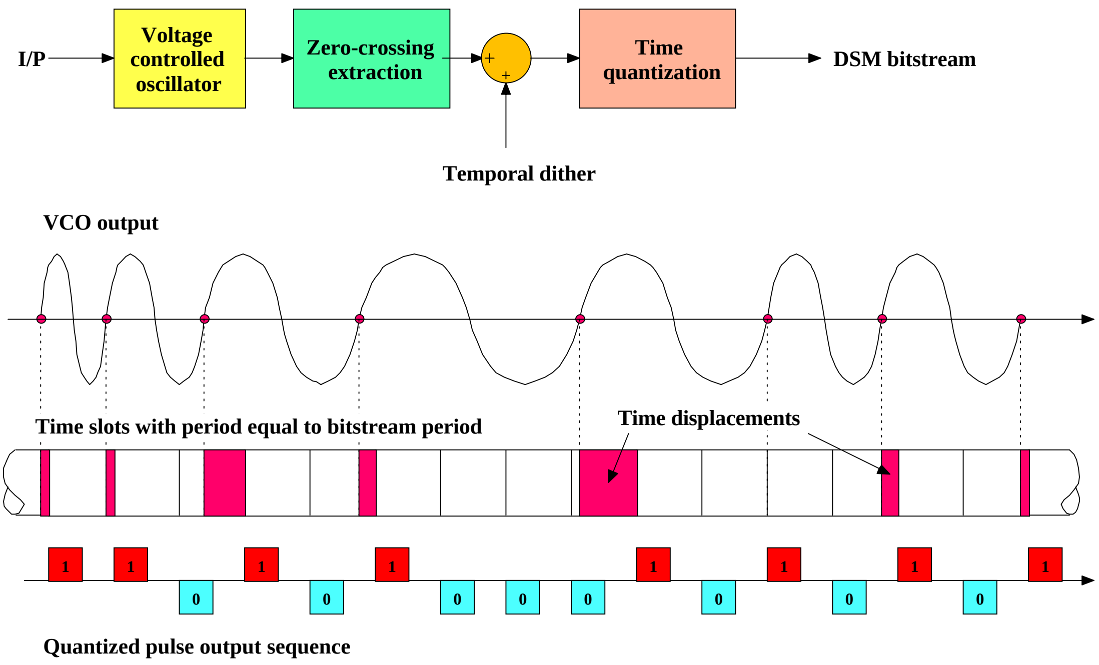
It is well established that conventional SDM encryption exhibits problems of correlated distortion. This has been reported earlier [4,5,6,7] although a more rigorous treatise has since been published by *Lipshitz and Vanderkooy* [2]. The principal observation of this work is that when a uniform quantizer is preceded by the addition of a correctly designed dither signal, that within the constraints of a 2-level quantizer, the quantizer is fully loaded even before any signal is applied. Consequently, when the signal is applied to the SDM and the 2-level quantizer is observed as a multi-level quantizer, it is always operating in a condition of overload and therefore the dither statistics are corrupted.

The specific question raised in this Section is whether saturation in a 2-level quantizer is fundamental or is there a relaxation of the operating conditions, which can avoid saturation. It is not intended here that discussion will lead necessarily to a higher performance encoder, only whether a coder without saturation can be conceived. A number of examples are presented to develop this thesis.

4-1 LFM model equivalence of SDM

Earlier publications [8,9] have established the equivalence of single integrator delta-modulation and time-quantized phase modulation and also of first-order SDM and time-quantized frequency modulation. Hence, by including amplitude modulation to describe sampling, these two models link analogue modulation processes with quantization and noise shaping. The SDM equivalent model incorporates linear frequency modulator (LFM) where the centre frequency is set normally to one half the bitstream pulse rate. The input signal then modulates the LFM where frequency is proportional to input amplitude. Reference points are taken on the oscillator output waveform, such as the positive zero crossings (PZC), where subsequently these locations are quantized along the time axis using a grid of equally spaced time slots. Where a time-quantized PZC occurs a 1 pulse is introduced, otherwise a –1 pulse is inserted (logic 0). The process of time quantization is therefore analogous to amplitude quantization.

Extending the analogy between time quantization and amplitude quantization, it is proposed to introduce a time dither sequence into the system but where the time dither is added to the time co-ordinates of the zero crossings, thus de-correlating the temporal quantization process. A further extension to this process is to introduce temporal noise shaping at the point of quantization to enhance to low frequency resolution of the coder. A first-order model of oscillator, time dither and time quantization is illustrated in Figure 4-1.



Process of time quantization

Figure 4-1 Time-quantized frequency modulation model of SDM.

In such a scheme time dither is a temporal perturbation function and requires identical characteristics to that of conventional dither with a triangular probability distribution function (PDF), but scaled to span –T to T, where T is the bitstream-sampling period. The time dither PDF is shown in Figure 4-2.

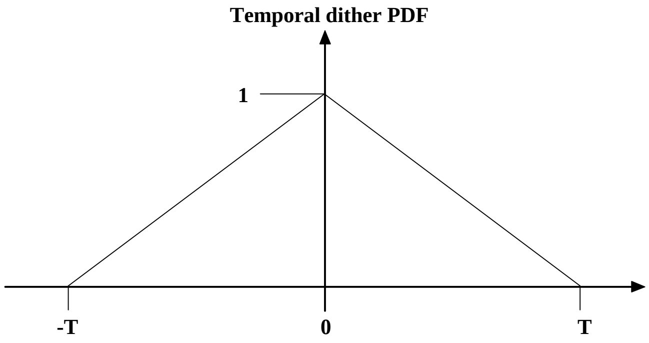


Figure 4-2 Time dither sequence with triangular PDF.

It is evident that with the addition of time dither, the PZC time locations can be perturbed such that two can occur within a single time slot, requiring a time-quantized pulse of double amplitude to be generated. The bitstream code is then no longer binary. To circumvent this problem the time resolution can be increased, while maintaining the centre frequency of the LFM. Fundamentally, this breaks the linkage with the sampling rate of the source data and the sampling rate of the bitstream converter, where it is here that the problem of correlated quantization distortion may find a solution, even if other area of performance are compromised.

Postulate

For a bitstream coder to eliminate correlated distortion resulting from quantization, a necessary condition is for the bitstream pulse rate f_{sdm} to be greater than the sampling rate f_s of the of the source information. This ratio is defined as the H-factor.

where, $$\mathrm{H} = \frac{f_{sdm}}{f_s}$$

The use of LFM in defining instants in time that describe the source data without distortion has a parallel with naturally sample pulse-width modulation (NPWM) and is considered in Section 4-4. Also, it maybe conjectured that when a noise-shaping scheme is employed to increase the low-frequency resolution of a bitstream converter that this ratio must increase significantly above 2.

4-2 Validation and demonstration of time domain quantization model

The process of deriving a SDM coded signal through the use of LFM and subsequent time-domain quantization is investigated here in greater detail. The objective is first to confirm the method of encryption delivers acceptable results and second to explore the linearity of a system based upon a 1-bit serial code. However, it is not the intention to produce a definitive coding strategy, but only to demonstrate a condition whereby linearity can be achieved using a 1-bit sequential code.

4-2-1 Frequency modulation

The process of LFM is defined in terms of a frequency-modulated carrier with a centre frequency $f_{sdm}/(2H)$ and amplitude A as,

$$s_{lfm} = A\cos\left(\int_{t=0}^{t} \omega\, dt\right)$$

where $$\omega = \pi \frac{f_{sdm}}{\mathrm{H}}\left(1 + \frac{y(t)}{Y_{\max}}\right)$$

giving, $$s_{lfm} = A\cos\left(\int_{t=0}^{t}\left(\pi\frac{f_{sdm}}{\mathrm{H}}\left(1 + \frac{y(t)}{Y_{\max}}\right)\right)dt\right) = A\cos\left(\pi\frac{f_{sdm}}{\mathrm{H}}\left(t + \int_{t=0}^{t}\frac{y(t)}{Y_{\max}}dt\right)\right)$$

where s_{lfm} is the LFM signal, y(t) is the input signal, Y_{max} the maximum peak value of $|y(t)|$ and f_{sdm} the bit rate of the equivalent first-order SDM encoder. The H-factor was introduced in Section 4-1 where initially let $\mathrm{H}=1$, although this is explored later in more detail. Assume the reference point used for digitisation is the PZC of the LFM waveform s_{fm}. These time instants $\{t_r\}$ are located when the phase of s_{fm} is multiples of 2π such that,

$$\frac{f_{sdm}}{\mathrm{H}}\left(t + \int_{t=0}^{t}\frac{y(t)}{Y_{\max}}dt\right)_r = 2r$$

Consequently, r represents the rth PZC in the LFM output. For a given input function y(t), natural sampling solutions in terms of $\{t_r\}$ have to be sought to match this integer relationship. In discrete signal processing terms it is a difficult task to seek precise solutions, however from a verification and performance stance, the functional form of y(t) is known enabling an initial coarse search followed by a finer estimate using interpolation. This problem can be visualised with reference to Figure 4-3, where it is revealed that each time location can be associated with a unique integer r providing the slope of the phase ϕ versus time t graph is greater than zero.

That is,

if $$\phi = \pi \frac{f_{sdm}}{\mathrm{H}} \left(t + \int_{t=0}^{t} \frac{y(t)}{Y_{\max}} dt \right)$$

then by differentiating $\partial\phi / \partial t > 0$, the bound on y(t) is $1 + y(t)/Y_{\max} > 0$. A condition that sits comfortably within the defined operating constraints placed on the system.

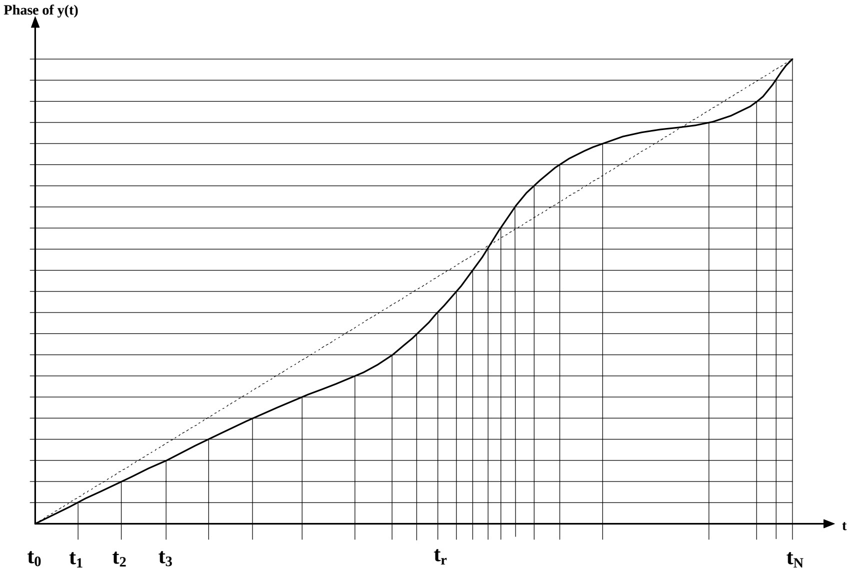


Figure 4-3 Seeking natural sampling solutions for $\{t_r\}$.

4-2-2 Time quantization and positive zero crossing estimation

The final stage in the transcoding process is time-quantization. Initially it may appear sufficient to seek solutions for $\{t_r\}$ with a time-resolution that is just less than a bit period in the output code. However, if time dither is to be applied as well as temporal noise shaping to the time quantization process, then the estimates must be accurate to well within the bit period, otherwise the approximations in $\{t_r\}$ appear as an additional layer of quantization distortion. Fortunately, the output bit rate is significantly higher than the signal bandwidth. Consequently, by sampling at, say 8-times the bit rate, a search can be performed to localise approximately each integer boundary followed by linear interpolation performed on points either side a transition to estimate more precisely the time location of the integer boundary.

The search was implemented using the following signal processing techniques:

- The FM signal is computed with an oversampling factor 'of' over the bit rate f_{sdm}.
- The signal is converted to a square wave using a squaring (*or sign*) function.
- A difference signal is computed between adjacent samples, that is non-zero only at the zero crossing transitions.
- By interrogating the sign of the inter-sample difference, the PZCs can be identified and by using a *sort* function applied to this difference sequence, a vector zr(1:L) computed that contains only the sequenced sample numbers of samples just prior to an actual PZC.
- Hence, knowing the time location of a sample that precedes a PZC, a sample of the FM signal s_{lfm} each side of a PZC is calculated and a more accurate time location t_r estimated using linear interpolation as,

$$t_r = zr(r) - \frac{s_{lfm}\left(zr(r)\right)}{s_{lfm}\left(zr(r+1)\right) + s_{lfm}\left(zr(r)\right)}$$

The MATLAB subroutine presented below was used to perform a PZC search for $\{t_r\}$:

```
% search for PZC: 1 pulse comes just before PZC
sd=.5*(1+sign([sign(s(2:L*of))-sign(s(1:L*of-1)) 0]-.1));
% sort approximate PZC locations to determine their time co-ordinates
[p q]=sort(sd.*(1:L*of));
[mx my]=min(q(L*of/2:L*of));
% zr is a vector that defines the sample number of the sample just preceding a PZC
L1=my-1+L*of/2; L2=L*of;
zr=q(L1:L2);
```

Knowing the time locations $\{t_r\}$, quantization can be performed using a combination of time dither and temporal noise shaping, where the process takes as input the time sequence $\{t_r\}_{r=0}^{N}$. The process with only time dither and time quantization is illustrated in Figure 4-4, while Figure 4-5 includes temporal noise shaping. It is significant, that within the temporal noise-shaping loop no amplitude limits are imposed on the quantizer

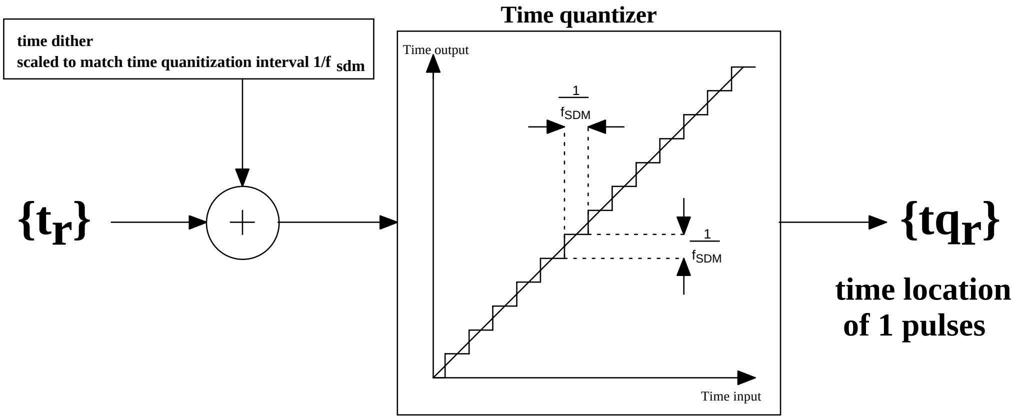


Figure 4-4 Time quantization with time dither.

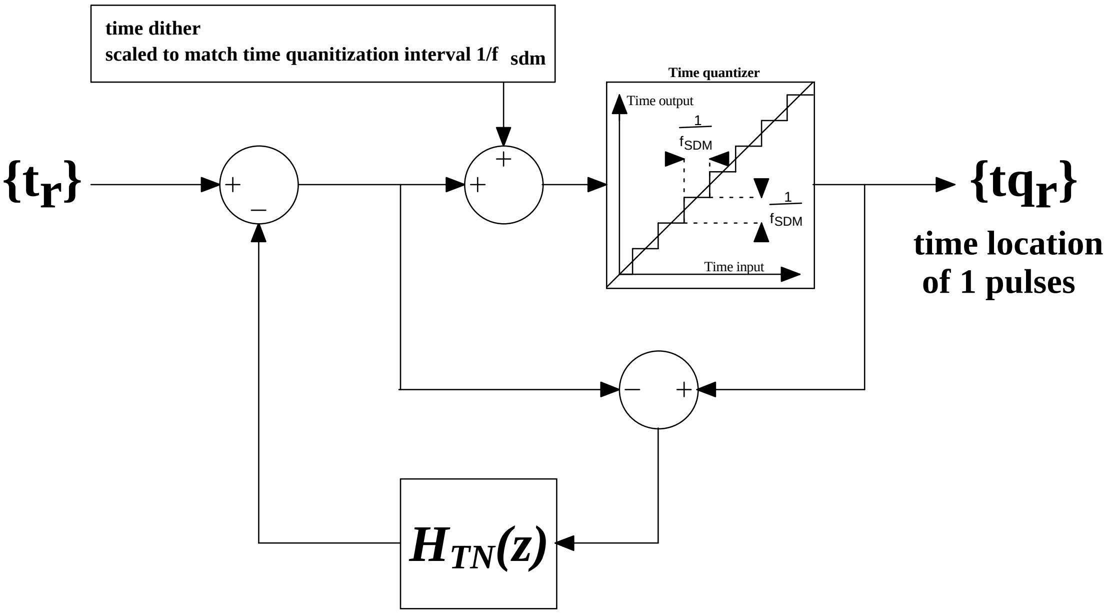


Figure 4-5 Time quantization with temporal noise shaping and time dither.

4-2-3 Temporal error correction

After generating the SDM code, the precise PZC locations were defined by the function {t_r}. However, following quantization and especially when temporal noise shaping is employed, then the noise-shaped output sequence {tq_r} may include coded sample times that are no longer sequenced in an arithmetic progression. This can be observed in the differential of the time-quantized pulse co-ordinates {tq_{r+1} - tq_r} where zero or negative values are returned when the sequence is in error. For example, assume that a segment of {tq_r} is as follows:

$$tqr \;=\; [\ldots\ 50\ 55\ 60\ 65\ \underline{75\ 70}\ 85\ 90\ \ldots]$$

This sequence defines the sample numbers where 1-pulses are to be located. It can be seen that the quantization process has returned an invalid sequence [… 75 70 85 ….] However, this problem is simply corrected by the expedient of re-ordering the bit pattern as,

$$tqr \;=\; [\ldots\ 50\ 55\ 60\ 65\ \underline{70\ 75}\ 85\ 90\ \ldots]$$

This process is valid since each pulse carries equal weight and normally has duration equal to the SDM bit rate; the only requirement is a vacant 0-pulse slot. Consequently, once the sequence {tq_r} is determined, a sequential *sort* function can re-order the pulses into a format suitable for the formation of SDM code. This is an unusual process in the formation of SDM code. The *sort* allows pulses to jump time-slot boundaries and therefore, to a degree, to circumvent the 2-level amplitude quantization limit normally imposed within a conventional SDM noise-shaping loop, where the output quantizer forges the binary amplitude and not the time location of the pulse.

However, where the basic *sort* fails is when the quantizer returns non-unique values in {tq_r}, as this implies two or more coincident pulses. The probability of this occurrence can be relatively low although it does depend upon the H-factor and the order of noise shaping selected. A simple expedient is to time displace the multi-valued pulses into the nearest unoccupied time slots. This maintains the correct overall area under the SDM code, although pulse time displacement introduces a small non-linear error with an error spectrum that rises with frequency. An improved method is to modify adjacent pulse locations in a more symmetrical manner.

A novel *sort* procedure has been identified that guarantees the required positive arithmetic progression, seeks out both dual and multiple coincident pulses and reconfigures coincident samples into a near-symmetric bi-directional distribution. A critical requirement of this process is that the number of unit pulses remains invariant, thus there is no loss of pulse area. Consider a vector $[X_r]_{r=1}^{N}$ of length N that contains the sample locations of 1-pulses. The sorted vector $[Y_r]_{r=1}^{N}$ with non-coincident pulses is computed as follows,

$$[Y_r]_{r=1}^{N} \;=\; sort\Big([X_r]_{r=1}^{N} \;-\; [1:N]\Big) \;+\; [1:N]$$

where $[1:N]$ implies a vector [1 2 3 … r … N]. To demonstrate the functionality and performance of this algorithm, two examples are consider with two and three coincident samples respectively. The vector length is arbitrarily selected as N = 10.

Example 1 Let, $[X_r]_{r=1}^{10} \;=\; [2\ \ 5\ \ 8\ \ 11\ \ 9\ \ 9\ \ 13\ \ 15\ \ 16\ \ 19]$

whereby after sorting,

$$[Y_r]_{r=1}^{10} \;=\; [2\ \ 5\ \ 6\ \ 8\ \ 10\ \ 12\ \ 14\ \ 15\ \ 16\ \ 19]$$

Sum of sample co-ordinates = 107 in both cases and the error between input and output vectors is,

$$error \;=\; [X_r]_{r=1}^{10} \;-\; [Y_r]_{r=1}^{10} \;=\; [0\ \ 0\ \ 2\ \ 3\ \ -1\ -3\ \ -1\ \ 0\ \ 0\ \ 0]$$

Example 2 Let, $[X_r]_{r=1}^{10} \;=\; [2\ \ 5\ \ 8\ \ 9\ \ 13\ \ 13\ \ 13\ \ 16\ \ 18\ \ 19]$

whereby after sorting,

$$[Y_r]_{r=1}^{10} \;=\; [2\;\;5\;\;8\;\;9\;\;11\;\;13\;\;15\;\;16\;\;18\;\;19]$$

Sum of sample co-ordinates = 116 in both cases and the error between input and output vectors is,

$$error \;=\; [X_r]_{r=1}^{10} \;-\; [Y_r]_{r=1}^{10} \;=\; [0\;\;0\;\;0\;\;0\;\;2\;\;0\;\;-2\;\;0\;\;0\;\;0]$$

Observe how the sorted values are unique and arranged in an arithmetic progression. Also, the area under the input and output sequences are invariant. This latter observation suggests that the pulse redistribution can be considered a form of time re-quantization where effective noise shaping preserves the area of the integrated function.

4-2-4 Bitstream coding using time-quantized LFM model *without temporal noise shaping*

The LFM model described in Section 4-2-2 was simulated to obtain the spectrum of the output bit pattern using the MATLAB program in Appendix 1. A bitstream-sampling rate of 128 time 44.1 kHz was selected with an H-factor of unity. The input signal consisted of two equal amplitude sinusoids of frequency 100 Hz and 5000 Hz and no dither was used. Figure 4-6(a) shows the output spectrum when each sine wave has an amplitude 0.3, while in Figure 4-6(b), this is repeated but with an amplitude of 0.0001. The reference level shown at about –125 dB corresponds to the spectral density of a 16 bit @ 44.1 kHz system. The result in Figure 4-6(a) reveals a spectral envelope slope of about 6-dB/octave although there is evidence of signal-to-quantization distortion correlation. At the lower input signal level, Figure 4-6(b) gross distortion is revealed.

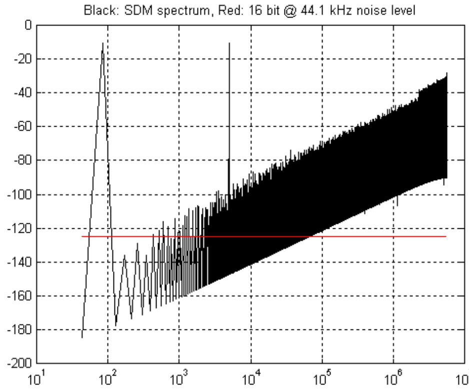


Figure 4-6(a) SDM, no dither H = 1.

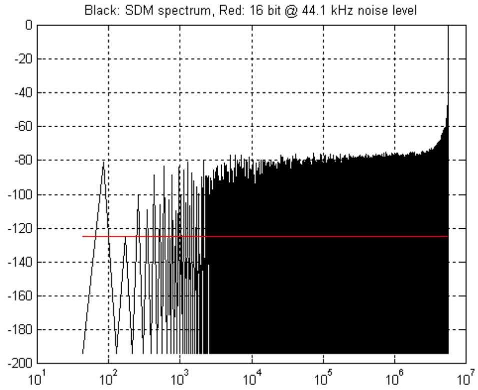


Figure 4-6b SDM, no dither H = 1.

A second pair of spectra were recalculated but using a triangular PDF dither sequence prior to quantization as shown in Figure 4-2, where the time span T corresponds to the bitstream-sampling period. However, the program returned numerous incorrectly sequenced and non-unique estimates of $\{tq_r\}$. In other words, when dither is added to $\{t_r\}$ prior to time quantization, it is possible to transgress a sample period and obtain a value identical to that from an adjacent quantized PRZ, say t_{r-1} or t_{r+1}. This overlap mode is a direct result of the Lipshitz-Vanderkooy bound [2] being breached in that dither + signal exceeds the linear coding capability of a 1-bit system. Although quantization is performed here in the time domain, it is equivalent to an amplitude quantization process when mapped against a multi-level quantizer.

The results were then repeated for an H-factor, H = 2 and are shown in Figure 4-7(a,b). In this case proper bitstream coding is confirmed without pulse ambiguity, where a significant conclusion is that there is now minimal correlation between signal and quantization error. However, a consequence of H = 2 is that the centre frequency of LFM is halved, so that there is an apparent loss of coding resolution and the dc level of the code has been lowered, assuming each pulse width remains equal to a bit interval. The dc error can be corrected by stretching each output pulse to H clock periods duration; although following convolution occasional multi-level outputs can then be formed. However, this is not a breach of the SDM code, as this data is still binary and time quantized to the f_{sdm} clock. Nevertheless, a fundamental result is that within the constraint of H = 2, the bitstream code is substantially linearized and can be compare favourably in this respect with dithered LPCM, confirming our earlier postulate introduced in Section 4-1. This implies that for correctly dithered first-order

SDM, the SDM sampling rate is at least of 4-times the bandwidth of input signal + dither, where the bandwidth is determined by the selection of LFM centre frequency with respect to the SDM bit rate. The output spectral results formed by including dither are statistically identical with no evidence of correlated distortion. Two further spectra were then computed with dither but now error correction was included and the H-factor reset to unity. These spectra are shown in Figure 4-7(c,d).

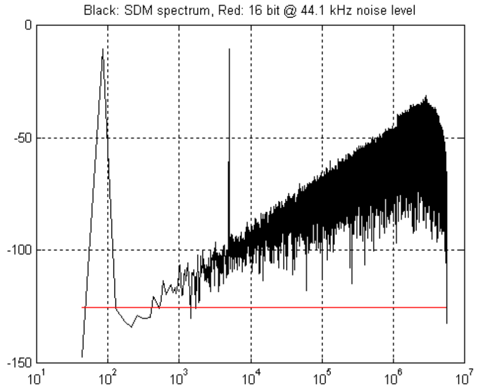


Figure 4-7(a) SDM, with dither H = 2.
(no error correction)

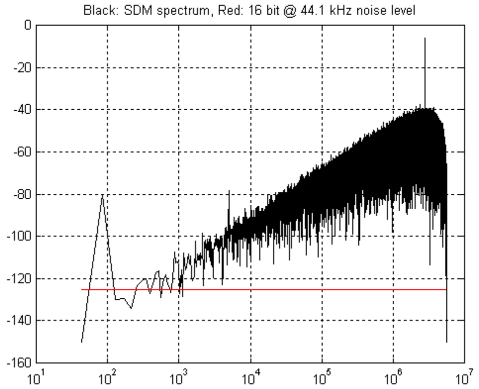


Figure 4-7(b) SDM, with dither H = 2.
(no error correction)

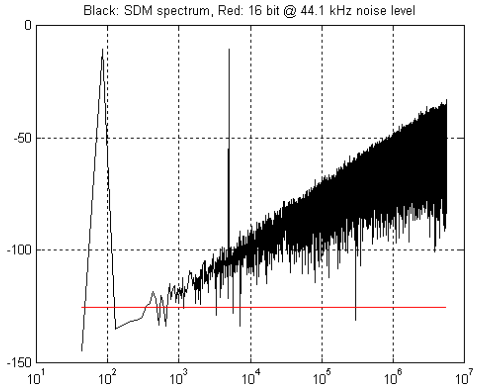


Figure 4-7(c) SDM, with dither H=1.
(with error correction)

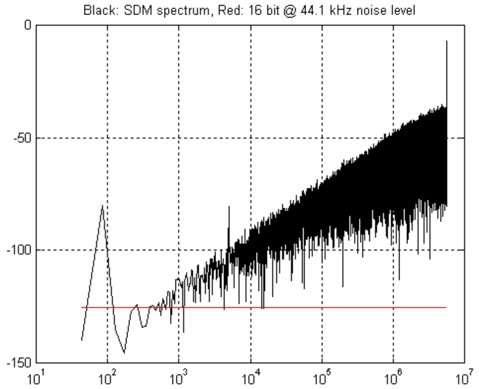


Figure 4-7(d) SDM, with dither H = 1.
(with error correction)

Because the system is based on frequency modulation, there is evidence of sidebands either side the LFM carrier. In Figure 4-6(a-b) because H= 1, the carrier frequency is on the extreme right of the spectrum being at $f_{sdm}/2$, while in Figure 4-7(a-b) where H = 2, the carrier at $f_{sdm}/4$, is shifted to the left. The region where the sidebands are significant results from the LFM carrier deviation and therefore relates to the modulation depth. This spectral spread is quite broad although the decay is rapid and sinks into the substantial quantization noise. This mechanism actually sets a bound on the maximum modulation depth, where if the modulation index is high, significant sidebands can extend down into the audio band. An extreme example of sideband interference is where the instantaneous carrier frequency f_c actually enters the audio band. For example, it follows from earlier analysis that,

$$f_c \;=\; \frac{f_{sdm}}{2\mathrm{H}}\left(1 \;+\; \frac{y(t)}{Y_{\max}}\right) \;=\; \Box f_{\,audio}$$

where $\Box f_{\,audio}$ is the upper frequency of the audio band. Hence, by observing the minimum value of f_c the maximum percentage modulation depth $m_{max\%}$ is

$$m\big|_{\max\%} \;=\; 1 \;-\; 2H\,\frac{\Delta f_{audio}}{f_{sdm}}$$

For example, if H = 4, $\Delta f_{audio} = 20\,kHz$ and $f_{sdm} \approx 2.8224\,MHz$, then $m\big|_{\max\%} = 94\%$. However, in practice this is an extreme case where because of the non-linear spectral distribution of LFM sidebands, a value of about 40 % should not be exceeded.

As an illustration of intermediate processes within the time-quantized LPCM model, Figure 4-8 shows a segment of the LFM output together with the signal z_r that identifies a sample just preceding a PZC. Also, in Figure 4-9 a graph is shown of the quantized time differential (tq_{r+1} - tq_r) after error correction, see Figure 4-4 and 4-5, taken over an analysis frame and plotted against the PZC number r. The critical factor is that the differential must remain positive if pulse ambiguity is to be avoided, where in practice a minimum value of 1 is revealed. Monitoring this signal (without error correction) allows the minimum H-factor to be determined.

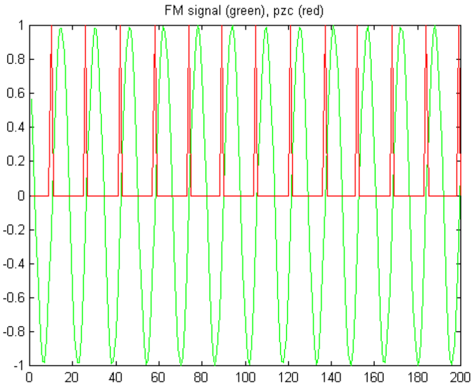


Figure 4-8 LFM segment with z_r.

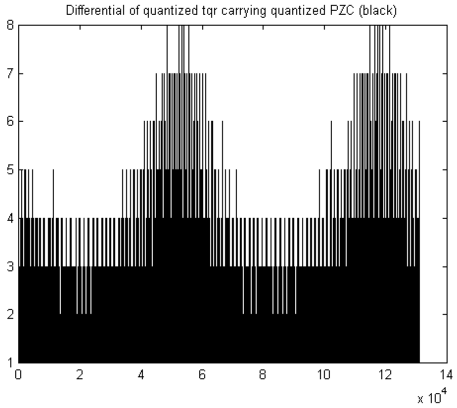


Figure 4-9 Map of {tq_{r+1} - tq_r}.

4-2-5 Bitstream coding using time-quantized LFM model *with temporal noise shaping*

To demonstrate the feasibility of using temporal noise shaping to enhance resolution both 2nd and 3rd-order noise shapers were included in the process of quantizing the {t_r} sequence. The two noise shaping filters $H_{T1}(z)$, $H_{T2}(z)$ selected were a single and dual integrator topology respectively as shown in Figure 4-5 is [10], where

$$H_{T1}(z) \;=\; z^{-1} \qquad\qquad H_{T2}(z) \;=\; 2z^{-1} \;-\; z^{-2}$$

For the single integrator case it was necessary for reasons previously discussed to select H = 4, while for the dual integrator case H = 6. Dither was used in both simulations together with an identical input signal to that used in Section 4-2-4 with amplitudes set to 0.3. The results are shown in Figures 4-10 and 4-11, where the blue curve also indicates the shape of the noise-shaping transfer function. It is important to observe that the process of non-noise shaped time-quantized LFM is equivalent to a first order SDM loop, therefore a 1storder noise shaper is similar to a 2nd-order SDM, while a 2nd-order noise shaper is similar to a 3rd-order loop. Hence, a 3rd-order system achieves stable operation that is not normally possible in the SDM without significant loop modification. The effect of selecting H = 4 and H = 6 respectively is evident in the high frequency region of the output spectrum where spectrum replications about $f_{sdm}/4$ and $f_{sdm}/6$ can be observed.

The simulations were then repeated but using error correction as described in Section 4-2-3. The nature of the error correction forces an SDM code pattern without ambiguity where it was found that stable operation could now be achieved both with one and two integrator systems and with H = 1. The corresponding spectra are shown in Figures 4-12 and 4-13. The results compare favourably with those in Figures 4-10 and 4-11, except mild intermodulation distortion is just evident either side of the 5 kHz signal. However, this is for a relatively high input with a normalised peak value 0.6, for lower levels the sidebands rapidly become insignificant. However, although there is mild intermodulation, the overall noise spectra are slightly lower in level as a unity H-factor is used. Also, the noise spectrum achieved with a single integrator temporal noise shaper is better than that of the second-order case, indicating some additional noise due to error correction that now has to cope with more extreme perturbations in $\{tq_r\}$.

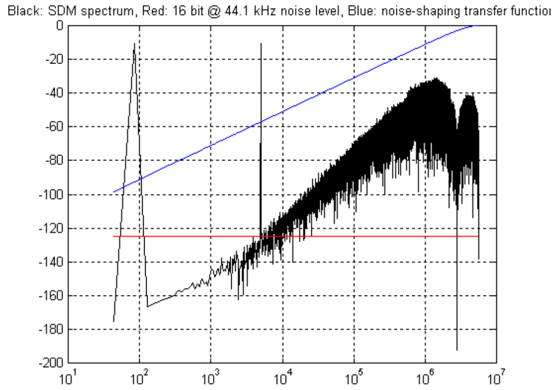


Figure 4-10 First order, H = 4,
(no error correction).

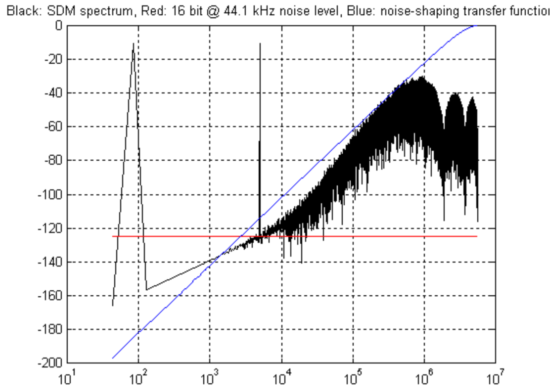


Figure 4-11 Second order, H = 6,
(no error correction).

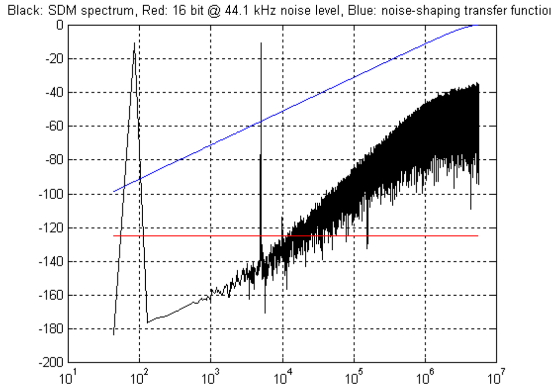


Figure 4-12 First order, H = 1,
(with error correction).

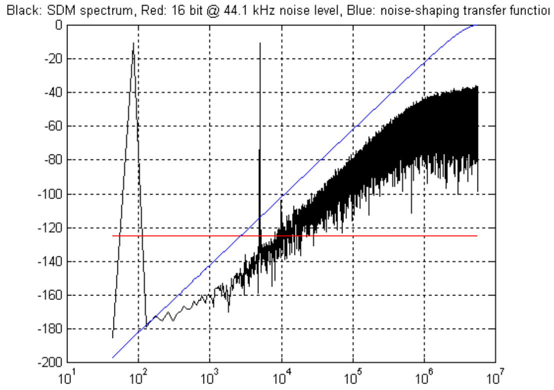


Figure 4-13 Second order, H = 1,
(with error correction).

4-2-6 Bitstream coding using time-quantized LFM model *with generalised temporal noise shaping*

From the noise shaping topology of Figure 4-5, the z-domain noise shaping transfer function $NST(z)$ is,

$$NST(z) = 1 - H_{TN}(z)$$

where $H_{TN}(z)$ may be expressed as an all-zero polynomial, $H_{TN}(z) = \sum_{r=1}^{N} a_r z^{-r}$. A re-configuration of the noise shaping topology is shown in Figure 4-14, where observing the lower topology it is evident that the loop transfer function $LPT(z)$ can be expressed,

$$LPT(z) = \frac{-H_{TN}(z)}{1 - H_{TN}(z)} = \frac{NST(z) - 1}{NST(z)}$$

For a stable system $NST(z)$ must be a minimum phase function. The program in Appendix 1 includes an option for entering an arbitrary specification for the magnitude response of the noise-shaping transfer function. From this function, after generating a symmetrical response about f_{sdm}, the minimum-phase impulse response is computed from which $H_{TN}(z)$ is derived. This response can then be used in the noise-shaping loop.

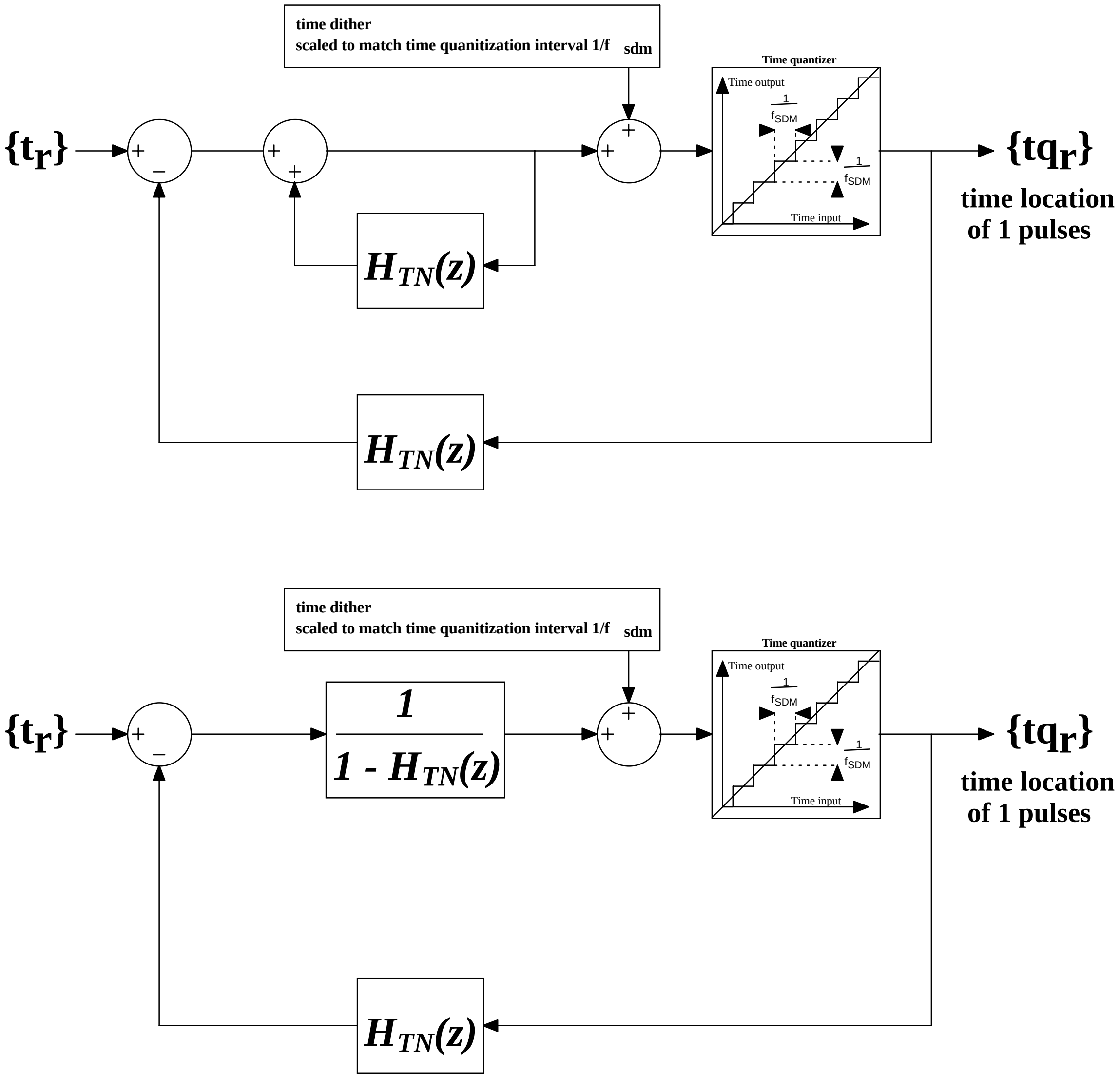


Figure 4-14 Re-configuration to canonic noise shaping topology

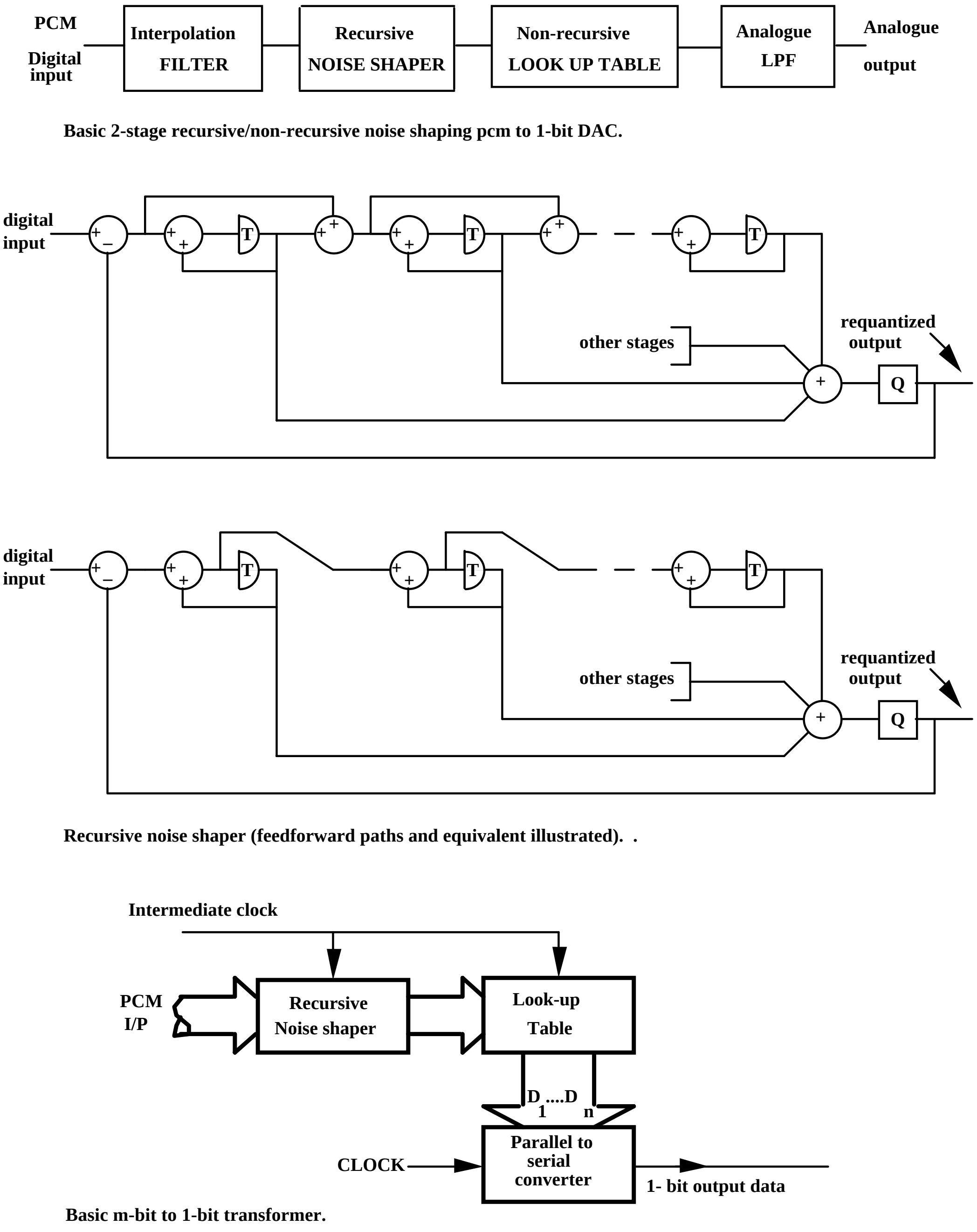
5 Multi-level noise shaping with multi-bit to binary conversion

An alternative bitstream coder that matches the linearity criteria of LPCM has been previously described [11,12], so the principles are reviewed in brief. This technique uses three cascaded stages as shown in Figure 5-1. In this example, the first stage uses a fourth-order noise-shaping loop with a continuous quantizer to relax the stability criteria, where the multi-level output data is truncated open loop to a range of +/- 8 quanta. The error correction procedure shown in Figure 5-2 can enhance truncation and is similar in concept to that described earlier [3].

The final stage uses an open-loop code converter based upon a look-up table that transforms the multi-level data from the noise shaper output into a serial bitstream, where code conversion is shown in Table 1. The generation of the code table follows an earlier proposal [11,12] where sequences are selected to minimise gain-error modulation at around 20 kHz of the normalised Fourier transform. This technique reduces base-band non-linearity in the code transformation and consequently achieves a lower distortion compared with a pulse-width modulation code typical of some MASH systems. The significance of this technique is that it shows another example of the generation a serial bit stream code that allows proper noise shaping and dither to be employed in the conversion. However, it also reveals that the input signal bandwidth must be sampled at a rate substantially below the sampling rate of conventional SDM.

level	15 bit code	level	15 bit code
-7.5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	7.5	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
-6.5	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	6.5	1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
-5.5	0 0 0 1 0 0 0 0 0 0 0 1 0 0 0	5.5	1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
-4.5	0 0 1 0 0 0 0 1 0 0 0 0 1 0 0	4.5	1 1 0 1 1 1 1 0 1 1 1 1 0 1 1
-3.5	0 1 0 0 0 0 1 0 1 0 0 0 0 1 0	3.5	1 0 1 1 1 1 0 1 0 1 1 1 1 0 1
-2.5	0 1 0 0 1 0 0 1 0 0 1 0 0 1 0	2.5	1 0 1 1 0 1 1 0 1 1 0 1 1 0 1
-1.5	0 1 0 1 0 1 0 0 0 1 0 1 0 1 0	1.5	1 0 1 0 1 0 1 1 1 0 1 0 1 0 1
-0.5	0 1 1 0 0 1 0 1 0 1 0 0 1 1 0	0.5	1 0 0 1 1 0 1 0 1 0 1 1 0 0 1

Table 1 - Optimised codes for a 16 level to 15-bit transformer.



Basic 2-stage recursive/non-recursive noise shaping pcm to 1-bit DAC.

Recursive noise shaper (feedforward paths and equivalent illustrated). .

Basic m-bit to 1-bit transformer.

Figure 5-1 Open loop look-up table bitstream converter.

This family of bitstream encoding can been see to include other forms of binary conversion including pulse-width modulation (PWM). In fact pulse width modulation can be seen as an extension of SDM but where the output ones and zeros are grouped into blocks of 1 and 0 sequences. In this sense there is a family resemblance to the error correction scheme described in Section 4-2-3. In practice where open-loop pulse grouping schemes are used, methods of linearization are required to reduce the effects of code specific spectral modulation. One method is reviewed above while other linearization methods for PWM have been reported [13,14,15].

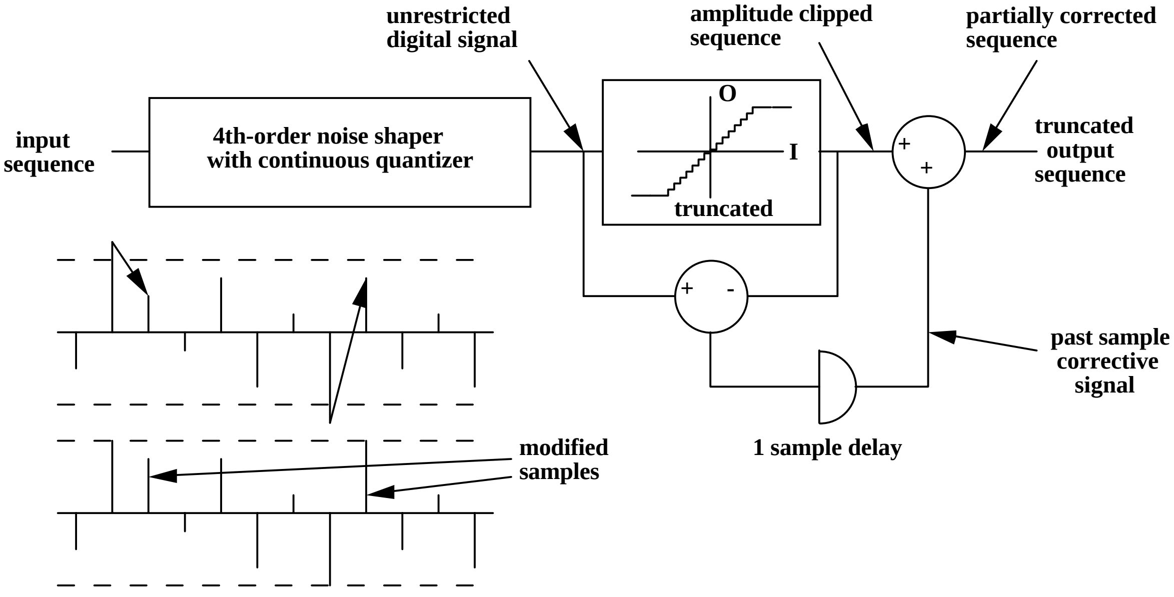

unrestricted digital signal
amplitude clipped sequence
partially corrected sequence
input sequence
4th-order noise shaper with continuous quantizer
O
I
truncated
+
+
truncated output sequence
+
-
past sample corrective signal
modified samples
1 sample delay

Figure 5-2 Sample truncation with error feedforward correction.

6 Hardware and system comparisons

In making comparisons between LPCM and SDM it is evident that there are significant difference in the coding algorithm. The question therefore arises whether there is a corresponding significant difference in sound quality. This is a seemingly simple question on first encounter yet it is fraught with problems. Section 3 exposed the desire to employ a UHRA format within the recording chain that transcends the quality of either release format. Also, it should be noted that it is inappropriate to use different electronics including ADCs etc in the two signal paths as these inevitably introduce a sonic signature and therefore invalidate any comparison. Consequently, fundamental to any meaningful comparison is the need to use extremely high quality source material, preferably in a format that is better than either release format and that has a common origin. As such, the fundamental format difference is then related only to the quality and idiosyncrasies of the conversion algorithms used to generate the release code. Because this stage of the process is performed in the digital domain, it is not modified in any way by sample jitter or analogue related problems. However, it is dependent on the internal word length within each algorithm and also the correct application of dither in any truncation process. Following the observations of *Lipshitz and Vanderkooy* [2] then there are potential implications on linearity in SDM where some degree of correlation with signal and quantization should be anticipated when using conventional techniques with H = 1. Also, it has been demonstrated that for a given bit rate, LPCM can yield better in-band SNR than SDM, although theoretically with greater filtering in the signal path to prevent aliasing at the 96 kHz sampling rate.

The comment on signal bandwidth requires special attention. In practice the signal transfer function of SDM can have a frequency independent signal transfer function that is constant to $f_{sdm}/2$, a frequency substantially higher than LPCM. However, the noise shaping transfer function rises rapidly with frequency and produces high levels of quantization noise in the ultrasonic band. This noise requires substantial filtering if gross overloading of electronics downstream of the DAC is to be avoided. When this requirement for bandlimiting is introduced, then the difference between SDM and LPCM in this domain is marginal, making this a minor issue. There is already debate as to the value of ultrasonic signals in high-resolution audio that can be become audible through intermodulation because of system non-linearity. The presence of high frequency noise may well exacerbate this problem and contribute a further level of uncertainty.

Originally when DSD was proposed it used a 1-bit system that would employ 1-bit ADC and DAC technology. It is evident that with UHRA the ADC is more likely to employ multi-bit architectures with noise shaping. However, it is also probable that DAC electronics will employ multi-bit converters in association with noise shaping and de-correlation processing as these in general perform better than 1-bit converters. Consequently, there is a requirement for code conversion within the replay electronics that may in principle be combined with additional signal processing in multi-channel applications. If a universal system employs such DACs then there is also a need to transcode from either LPCM or SDM code to the data format of the DAC. Such a system can then form the basis of a realistic comparison, as the core converter technology is the same and the only areas where errors occur are within the transcoder, a process that mirrors to some degree the encryption process via UHRA. Nevertheless, if the transcoding is performed to a high degree of mathematical accuracy, then any additional error should be below those introduced at source coding.

Dichotomy

If the release formats are compared using non-identical electronics, these are rendered invalid because there are too many variables to allow a meaningful conclusion, even though one system may be preferred over another. In such a circumstance subjective differences would be anticipated, as there are performance differences between audio electronic equipment. This creates a dichotomy as in the purest sense the inherent simplicity of SACD with its electronics operating with uniformly streamed binary data is a critical factor, where it could be argued that the hardware configuration is as influential as the means of encoding. Such performance differences, if they exist, can be attributable to one or more of the following factors:

- Differences in (the various) filter responses in terms of bandwidth, hence rise-time, also in-band ripple together with truncation distortion within practical digital processors.
- Differences in jitter sensitivity. DSD code is more sensitive to jitter even when a switched capacitor DAC is used. However, offsetting this is the lack of word structure in DSD where, in an ideal sense, data is streamed continuously. Word patterns that result in LPCM increases the likelihood of word dependent jitter. This can effect the operation of a phase lock loop, or be introduced by mutual coupling between circuits and problems of ground rail and power supply interaction in practical circuitry.
- Differences introduced by correlated distortion as a function of a particular coding technique, including the internal resolution and correct use of dither in process implementation.
- Accumulated distortion in intermediate processes, thus supporting the need for a UHRA format.
- Imperfect decorrelation of converter distortion products from the signal.
- Imperfections in associated analogue electronics that respond to the spectral content of the audio signals. In particular, intermodulation distortion related to high frequency noise resulting from noise shaping.

7 Conclusions

This paper has considered factors that differentiate LPCM from SDM based systems. In a broad sense and within the context of a practical recording chain, the performance differences between these release formats are not so great as might first be perceived. Issues of overall bandwidth relate to the need for system filters that either prevent aliasing distortion or suppress noise generated through noise shaping in SDM. Viewed in this way, these specific differences are rather marginal even though the function of each filter differs to some extent, the final result is similar. This is particularly true when the signals are transcoded to match the data formats of identical DACs using oversampling, noise shaping and decorrelation techniques. Other factors that relate to post processing that are required both in digital and active loudspeaker systems and, for example, where signals must be matched to a replay environment for surround sound reproduction are not considered in this discussion. However, such processes do nevertheless have far reaching ramifications on the debate.

Clearly within any practical system the underlying theory of LPCM applies at some stages to both release-formats especially where common source data is employed. The view that somehow SDM is a radically different paradigm from LPCM that is a more natural process for encoding music is misguided, misrepresents the physics and fails to observe the practicalities of signal processing, even where a simple process such as a gain change is required. Such views are mischievous, divisive and serve to mislead. In practice it is a rare situation for a recording chain to consist of only back-to-back 1-bit converters with no intermediate processing.

Considerable attention has been directed to the question of whether a 1-bit coding system can be linear in the same way as LPCM with dither. In the way that conventional SDM is implemented, where the effective sampling rate is f_{sdm}, the answer is no. However, by relaxing the encoder design by employing an H-factor >1, linear encoding become feasible as there is now room in the signal space to accommodate the dither and to use a multi-level quantizer. However, in forwarding this observation a somewhat different view of SDM must be taken based upon a model of LFM. This technique enables quantization with noise shaping to be performed in the time domain. It was observed that when the quantizer output produces time locations that were multi-level and therefore not properly sequenced, that a simple *sort* function could re-order the pulses without distortion, as all pulses carry equal weight. The only occasions where distortion occurs in this process is because of coincident output levels. However, a modified *sort* function was shown to translate this occurrence by dispersing the pulses either side of the coincidence event. Simulations confirmed that levels of correlated distortion introduced by this method were extremely low and only evident on maximum level signals.

It was acknowledged that within the context of conventional generation of SDM code, there is a fundamental limitation due to quantizer saturation. The proposal that a higher resolution code be considered is therefore prudent where it is critical that the quantizer be multi-level, even if this is limited to 1 or 2 bit. It has been shown that by relaxing the bandwidth of SDM, linearity is achievable, although this probably does not represent a complete solution. It is suggested that the key advantage of using 1-bit uniformly streamed code, where other than memory effect caused by interaction of adjacent bit patterns during reconstruction, the 2-level DAC if not the coding, exhibits inherent linearity. Hence, in extending the quantizer to multi-level, decorrelation procedures are mandatory if this advantage is to be retained. This is why in proposing a UHRA format; decorrelation is cited as a fundamental element in the process. Also, for practical system reasons, scrambling the LPCM data is also supported in order to reduce equipment susceptibility to jitter correlation.

Simulations of SDM derived from the LFM model cannot match the noise performance of LPCM, although they do demonstrate linear encoding is feasible with SDM under certain conditions. It is conceivable therefore, that using this model with more advanced noise shaping and error correction, further improvements can be made on lowering in-band noise levels, thus opening the way to alternative methods of generating 1-bit code that bypasses the severe conditional loop stability intrinsic to conventional SDM. However, when all factors are considered and the philosophy of not using back-to-back 1-bit converters without intermediate processing, such as gain control and filtering, then the current performance status favours the LPCM format. LPCM with loss-less compression [16] is the currently the most efficient and transparent release format for conveying data between UHRA domain and a state-of-the-art multi-bit converter. However, forging UHRA code and making it the release format in future high capacity DVD applications raises an interesting prospect.

8 References

1 THE INTERVIEW, Kessler, K., *Hi-Fi News and Record Review*, vol. 45, no. 10, pp 64-67, October 2000

2 WHY PROFESSIONAL 1-BIT SIGMA-DELTA CONVERSION IS A BAD IDEA, Lipshitz, S.P. and Vanderkooy, J., *109th AES Convention*, Los Angeles, September 2000, preprint 5188

3 TRANSPARENT DIFFERENTIAL CODING FOR HIGH-RESOLUTION DIGITAL AUDIO, Hawksford, M.O.J., *107th AES Convention*, New York, September 1999, preprint 5037 (L-1)

4 MULTI-CHANNEL HIGH-DEFINITION DIGITAL AUDIO SYSTEMS FOR HIGH-DENSITY COMPACT DISK, Hawksford, M.O.J., *101st AES Convention*, Los Angeles, November 1996, preprint 4362 (J-2)

5 HIGH-DEFINITION DIGITAL AUDIO IN 3-DIMENSIONAL SOUND REPRODUCTION, Hawksford, M.O.J., *103rd AES Convention*, New York, September 1997, preprint 4560

6 DVD-AUDIO: HIGH-RESOLUTION, MULTI-CHANNEL DIGITAL FORMATS, Hawksford, M.O.J., *PALA'98 - AES Singapore Section Seminar*, World Trade Centre, Singapore, 17th July 1998

7 BITSTREAM VERSUS PCM DEBATE FOR HIGH QUALITY COMPACT DISC, Hawksford, M.O.J., *private publication*, ARA web: http://www.meridian-audio.com/ara/

8 EXACT MODEL FOR DELTAMODULATION PROCESSES, Flood, J.E. and Hawksford, M.J., *Proc. IEE*, vol.118, pp.1155-1161, 1971

9 UNIFIED THEORY OF DIGITAL MODULATION, Hawksford, M.J. *Proc. IEE*, vol.121, no.2, pp.109-115, February 1974

10 RELATIONSHIP BETWEEN NOISE SHAPING AND NESTED DIFFERENTIATING FEEDBACK LOOPS, Hawksford, M.O.J. and Vanderkooy, J., *JAES*, vol. 47, no. 12, pp 1054-1060, December 1999

11 MULTI-LEVEL TO 1 BIT TRANSFORMATIONS FOR APPLICATIONS IN DIGITAL-TO-ANALOGUE CONVERTERS USING OVERSAMPLING AND NOISE SHAPING, Hawksford, M.O.J., *Proc. Institute of Acoustics*, vol.10, Part 7, pp. 129-143, 1988

12 A COMPARISON OF TWO-STAGE 4TH-ORDER AND SINGLE-STAGE 2ND-ORDER DELTA-SIGMA MODULATION IN DIGITAL-TO-ANALOGUE CONVERSION, Hawksford, M.O.J., *IEE Conference on Analogue to Digital and Digital to Analogue Conversion*, Conference publication 343, pp

148-152, Swansea, September 1991

13 DYNAMIC MODEL-BASED LINEARIZATION OF QUANTIZED PULSE-WIDTH MODULATION FOR APPLICATIONS IN DIGITAL-TO-ANALOG CONVERSION AND DIGITAL POWER AMPLIFIER SYSTEMS, Hawksford, M.O.J., *JAES*, vol. 40, no. 4, pp 235-252, April 1992

14 TOWARDS THE 24-BIT DAC: NOVEL NOISE SHAPING TOPOLOGIES INCORPORATING CORRECTION FOR THE NONLINEARITY IN A PWM OUTPUT STAGE, Craven, P., *JAES*, vol. 41, no. 5, pp 291-313, May 1993

15 LINEARIZATION OF MULTI-LEVEL, MULTI-WIDTH DIGITAL PWM WITH APPLICATIONS IN DIGITAL-TO-ANALOGUE CONVERSION, Hawksford, M.O.J., *JAES*, vol. 43, no. 10, pp 787-798, October 1995

16 LOSSLESS CODING FOR AUDIO DISCS, Craven, P.G. and Gerzon, M.A., *JAES*, vol. 44, no. 9, pp 706-720. September 1996

Appendix 1: MATLAB program to model LFM SDM converter including first-order and second-order temporal noise shaping filter, dither and error correction

```
% SDM investigation using FM model with time quantization and temporal noise shaping
% special program with all noise-shaping modes integrated into a single routine
% 03.03.01
close; clear; home; colordef white;

% Nyquist sampling rate, fsam
fsam=44100;

% SDM sampling rate,fsdm
fsdm=256*fsam;

% oversampling factor to enhance fm resolution, of
of=8;

% H-factor: ratio of SDM sampling rate for LFM centre frequency
H=1;

% input parameters (2 sine waves)
a1=.3; a2=.3; f1=100; f2=5000;

% rx=0 no dither, rx=1 dither
rx=1;

% select error correction to eliminate pulse co-incidence and improper sample ordering
% err=0 no correction in tqr sequence
% err=1 prevent negative differentials in tqr sequence
% err=2 prevent negative and zero differentials in tqr sequence
err=2;

% temporal noise shaping: G=0 none, G=1 first order, G=2 second order, G=3 general noise shaping
G=3;

% general noise-shaping filter length, HL
HL=2^floor(log2(fsdm/1000));

% save Figures if sav = 1
sav=0;

% if wav = 1 generate and save filtered time domain output in .wav fsam @ 16-bit format
wav=0;

% system parameters:
```

```
Lx=2^18; preamble=2^5;
L=Lx+preamble;

% FM centre frequency
fm=fsdm/2;

% computational time interval
dt=1/(of*fsdm);


%******************************************************************************
% initialse general noise-shaping filter HNS vector
if G==0
HL=2;
fprintf('No temporal noise shaper')
NSTmin=[1 0];
elseif G==1

% first-order noise-shaping transfer function
fprintf('Initialising general noise-shaping filter for one integrator')
HL=2; erh=zeros(size(1:HL));
NSTmin=[1 -1];
HNS=[-NSTmin(2:HL) 0];
elseif G==2

% second-order noise-shaping transfer function
fprintf('Initialising general noise-shaping filter for two integrators')
HL=3; erh=zeros(size(1:HL));
NSTmin=[1 -2 1];
HNS=[-NSTmin(2:HL) 0];
elseif G==3

% general minimum-phase noise-shaping transfer function
fprintf('Initialising general noise-shaping filter: length HL')
HL
erh=zeros(size(1:HL));
% define single-sided noise-shaping amplitude spectrum
M=40; W=20;
NST=[.0001*(1:M)/M .0001+.5*(1-cos(pi*(0:W-1)/W)) .0001+ones(size((M+W+1):HL/2))];
NST=[NST(1) NST(1:HL/2) NST(HL/2-1:-1:1)];
% calculate minimum-phase impulse response of noise-shaping transfer function
NST=exp(conj(hilbert(log(NST)))); NST(1)=0;
NSTmin=real(ifft(NST));
NSTmin=NSTmin/NSTmin(1);
NSTmin=conv(NSTmin,[1 -2 1]);
% calculating noise-shaping transfer function HNS vector (1 - NSTmin(z)), length HL
HNS=[-NSTmin(2:HL) 0];

else
fprintf('Terminating, invalid G value\\')
return
end
%******************************************************************************

% SDM equivalent bit rate
fsdm

% signal frequency
f0=fsdm/Lx;
```

```
h1=round(f1/f0); h2=round(f2/f0);
if h1==0
h1=1;
end
if h2==0
h2=1;
end
fsig1=h1*f0
fsig2=h2*f0
%*******************************************************************************************************

% generate input signal
w=2*pi*f0;
asi=-(a1/(h1*w))*cos(h1*w*(1:L*of)*dt)-(a2/(h2*w))*cos(h2*w*(1:L*of)*dt);

% generate '1' pulses
%*******************************************************************************************************
% FM process:
g=2*pi*fm/H;
s=cos(g*((1:L*of)*dt+asi));

% search for PZC: 1 pulse comes just before PZC
sd=.5*(1+sign([sign(s(2:L*of))-sign(s(1:L*of-1)) 0]-.1));
% sort approximate PZC locations to determine their time co-ordinates
[p q]=sort(sd.*(1:L*of));
[mx my]=min(q(L*of/2:L*of));
% zr is a vector that defines the sample number of the sample just preceding a PZC
L1=my-1+L*of/2; L2=L*of;
zr=q(L1:L2);

% perform linear interpolation to give better time estimate of actual PZC
tr(1:L2-L1+1)=zr(1:L2-L1+1)-s(zr(1:L2-L1+1))./(s(zr(1:L2-L1+1)+1)-s(zr(1:L2-L1+1)));

% the system has been oversampled above the sdm rate by a factor 'of'
% perform noise shaped, time quantization with dither on the sequence tr
k=size(tr);
rd=rx*(rand(1,k(2))+rand(1,k(2))-1);
if G==0
tqr=round(H*of)+round(tr/of+rd)*of;
else
erh=zeros(size(1:HL));
tqr=zeros(size(1:k(2)));
for n=2:k(2)
pe=tr(n)-sum(erh(1:HL).*HNS(1:HL));
tqr(n)=round(pe/of+rd(n))*of;
erh(2:HL)=erh(1:HL-1);
erh(1)=tqr(n)-pe;
end; end
tqr0=tqr;

% ERROR CORRECTION 1: sort to prevent negative differentials only
if err==1
tqr=sort(tqr);
end

% ERROR CORRECTION 2: sort to prevent zero and negative differentials
if err==2
tqr=sort(tqr-of*(1:k(2)))+of*(1:k(2));
end
```

```
error=(tqr0-tqr)/of;
error=error-round(mean(error));

% offset tqr by constant to prevent negative or zero vector index
tqr=tqr-min(tqr-of);

% construct a sequence of 1, -1 pulses located at time quantised locations tqr
sdm(1:L2)=zeros(size(1:L2));
sdm(tqr(1:k(2)))=1;

% decimate sequence by a factor 'of' to form vector length L
[p q]=max(abs(sdm));
sdm=[zeros(size(1:q/of-1)) sdm(q:of:L2)];

% stretch pulse to length 'H', noting in reconstruction this can cause multi-level pulses
sdm=2*(conv(sdm,ones(size(1:round(H))))-.5);
[b c]=size(sdm);
sdm=sdm(c+1-Lx:c);
ss=sum(sdm);
sdm(1)=sdm(1)-sum(sdm);
%*************************************************************************************


% Fourier transform of sdm (sdmc complex, sdmf log magnitude)
sdmc=2*fft(sdm)/Lx;
sdmf=20*log10(2^-32+abs(sdmc));
if max(sdmf)>0
sdmf=sdmf-max(sdmf);
end

% calculate 16-bit and reference level assuming a 44.1 kHz sampling rate
ref16=3+10*log10((2/2^16)^2/12)+10*log10(2*fsdm/(44100*Lx));

% generate .wav file in 44.1 kHz @ 16 bit format
% define low-pass filter bandwidth
R=round(fsdm/fsam);
upper=ceil(0.5*Lx/R);
lowpass=[ones(size(1:upper)) zeros(size(upper+1:Lx/2))];
filt=[0 lowpass(2:Lx/2) 0 lowpass(Lx/2:-1:2)];
out=real(ifft(filt.*sdmc))*Lx/2;
% decimate in time
out=out(1:R:Lx);
% quantize to 16 bit and save .wav
if wav==1
no=size(out);
out=round(out*2^15+rand(1,no(2))+rand(1,no(2))-1)/2^16;
cd E:\Temp
wavwrite([[out out out out out]',[out out out out out]'],fsam,16,'SDM')
cd D:\matlab\bin
end


% plot routines

% LFM time domain with PZC
plot(s(1:200),'k')
hold
plot(sd(1:200),'r')
title('LFM signal (black), PZC (red)')
```

```
pause
if sav==1
print -dbitmap E:\Temp\Figure_a
end
close

% SDM signal segment
plot(sdm(2000:2300),'k')
title('SDM code (black)')
pause; close
p=size(tqr);
plot((tqr(2:p(2))-tqr(1:p(2)-1))/of,'k')
title('Differential of quantized {tqr} carrying quantized PZC (black)')
if sav==1
print -dbitmap E:\Temp\Figure_b
end
pause; close

% Timing error resulting from error correction
plot(error,'k')
title('Timing error resulting from tqr error correction, (black))')
pause
if sav==1
print -dbitmap E:\Temp\Figure_c
end
close

% SDM spectrum
NSTF=fft([NSTmin(1:HL) zeros(size(1:Lx-HL+1))]);
semilogx(f0*(1:Lx/2-1),sdmf(2:Lx/2),'k')
hold
semilogx(f0*(1:Lx/2-1),ref16*ones(size(2:Lx/2)),'r')
if G>0
semilogx(f0*(1:Lx/2-1),20*log10(abs(NSTF(2:Lx/2)/max(abs(NSTF)))),'b')
title('Black: SDM spectrum, Red: 16 bit @ 44.1 kHz noise level, Blue: noise-shaping transfer function')
else
title('Black: SDM spectrum, Red: 16 bit @ 44.1 kHz noise level')
end
grid; pause
if sav==1
print -dbitmap E:\Temp\Figure_d
end
close

% Low-pass filtered and decimated time domain output
plot(out,'k')
title('Low-pass filtered and decimated time domain output, (black))')
pause
if sav==1
print -dbitmap E:\Temp\Figure_e
end
close


% end of program
return
% *********************************************************
```