One-bit Audio: An Overview

Derk Reefman, Erwin Janssen
Philips Research Laboratories,
Prof. Holstlaan 4, 5656 AA Eindhoven,
the Netherlands

October 31, 2003

Abstract

This paper presents an overview of one-bit audio processing. Several characteristics of a Sigma Delta Modulator (SDM), which currently is the most often used device to generate one-bit code, are discussed, as well as some simple design methodologies of SDMs. It is shown that one-bit audio is capable of carrying very high quality audio. The total audio production chain, from recording to replay, is displayed and its feasibility is demonstrated. Finally, some recent developments in the field of one-bit audio codecs are summarized, which show a further improvement over the already excellent audio characteristics of a Sigma Delta Modulator.

List of abbreviations

FBSDM	(distributed) Feedback Sigma Delta Modulator, sometimes also called 'error feedback SDM'.
FFSDM	Feedforward Sigma Delta Modulator, sometimes also called 'interpolative SDM' or 'predictive SDM'.
LPF	Low Pass Filter
kS/s, MS/s	unit of sample rate: kilo samples or mega samples per second
NL	Non-Linearity
NS	Noise Shaper
NTF	Noise Transfer Function
SDM	Sigma Delta Modulator
SDPC	Sigma Delta Precorrection
SNR	Signal to Noise Ratio
STF	Signal Transfer Function

1 Introduction

In 1998, a one-bit coding format was introduced as a successor to the Compact Disc (CD). Whereas CD employs (L)PCM encoding, with 16 bit wide words at a sample rate of 44.1 kS/s to store the digital representation of the audio data, the new format stores a one-bit representation of the audio at 2.8 MS/s, which is 64 times the CD data sampling rate. Obviously, this change to one-bit audio has introduced the need for a change in signal processing.

While the application of one-bit audio for audio storage and distribution is quite new, the underlying idea of employing a one-bit coding scheme is not. Already in the early fifties, the concept of one-bit coding was proposed and implemented by de Jager [1]. The original idea by de Jager was that when transmitting a one-bit code instead of a PCM code, loss of 1 bit of information was not as detrimental as for PCM code. In a one-bit code, all bits carry equal weight and loss of a single bit meant a certain loss of accuracy. In PCM, some bits are more significant than others and loss of the MSB, could lead to radically wrong results. While the application of the device, coined 'Delta Modulator', invented by de Jager first saw its applications in error reduction in communication applications, it soon appeared that the one-bit code made the realisation of a high quality DAC (and, thus ADC) relatively easy [2]. As a result of the appearance of the CD in the eighties, demands for reduced distortion levels in audio reproduction were becoming more stringent. It proved virtually impossible, and at least economically unfeasible, to create low distortion DAC devices with many (16) bits. Contrary to that, it was much easier to create low-distortion AD and DA converters using a digital format of 1 bit, which were running at very high sample rates such as 64 or 128 times 44.1 kHz. Conversions between this high speed, one-bit format and 44.1 kHz/16 bit CD format can easily be accomplished in the digital domain using filtering and signal processing. This technique has been highly successful, and the so-called 'oversampling' and/or 'bitstream' technology dramatically increased the performance of standard CD-players in the nineties. The typical sequence of digital audio generation would be the generation of one-bit audio through a high quality ADC, followed by downconversion to 44.1 kHz/16 bit for storage on CD, again followed by upconversion to 64 or 128 f_s/one-bit in the CD player, after which it would be fed to a high quality DAC. The general merits of one-bit ADCs and DACs are wide spread nowadays, and many applications for frequencies much higher than typical audio bandwidths exist [3].

In search of ultimate audio quality, it seemed logical to introduce a format that would store this one-bit output directly, instead of the 'intermediate' CD format: in this way, all filtering and signal processing needed to convert to and from the one-bit format is eliminated which, by definition, can only increase the sound quality. After the first experiments with one-bit audio, it appeared indeed that the perceived sound quality was significantly better compared to the 44.1 kHz/16 bit format. Also, at the same time, new ADCs and DACs were appearing on the market, that were still using high sample rates (64 or 128 times 44.1 kHz), but exploited a few bits (1.5 to 5) instead of 1. As with the introduction of one-bit audio ADCs and DACS, this had purely technical fundamentals: ingenious techniques such as dynamic element matching [4, 5] to reduce the distortion problems of a multi-bit converter had appeared, and were feasible to implement for a limited number (2-6) of bits. Because one-bit converters are more sensitive to clock-jitter, the 'few-bit' converters took their place in the high-end audio market. To obtain a one-bit audio representation from the few-bit representation, the need for some mild digital signal processing was introduced. Interestingly, this did not lead to any observable change in sound quality in any test performed by studio engineers. Therefore, it is now believed, that the very high sample rate is the key factor in the extremely good sound quality of one-bit audio. The fact that the data is 1 bit instead of few bit, however, has retained its value because it reduces the storage requirements of the audio.

The purpose of this paper is to present an overview of the field of 'one-bit coding' in relation to one-bit audio, in a mix of practical and theoretical aspects. Many of the results presented in this review have been published elsewhere already, and will be discussed in a concise way. References for further reading are provided. However, some results are less well-known and will be presented in a more self-contained manner. As a core technology is formed by Sigma Delta Modulators (SDMs), in Sec. 2, an introduction to Sigma Delta Modulation will be presented. In Sec. 3, approximate modelling of SDMs will be used to present practical methods of SDM design, which also reveals some signal characteristics of SDMs. As linear modelling is far from perfect, however, in Sec. 4 a more extensive discussion about signal characteristics of SDMs follows. In Sec. 5, the creation of one-bit audio contents is discussed in the limited but important context of signal processing one-bit audio. With the renewal of interest for one-bit coding, several new developments have been published, and some interesting developments are discussed in Sec. 6. In Sec. 7, finally, a summary and conclusions will be presented.

2 Introduction to Sigma Delta Modulation

Sigma Delta Modulation (SD modulation) has become a wide spread name for a general class of devices, which characterizes itself by the phenomenon of *noise shaping* (and, in virtually all applications, *oversampling*); in principle, it bears little relation to the number of bits that the device outputs. In this section, however, we will focus mostly on devices which do have one-bit output, with an occasional remark about other outputs. Also, while many of the remarks to be made equally well hold for AD converters, we will restrict the discussion to 'digital-to-digital' converters: we will regard digital inputs and digital outputs, where the differences between in- and output can be word length and/or sample rate.
Obviously, it is not feasible to present a complete overview of the history of (Sigma) Delta Modulation, nor is it possible to provide the complete list of references detailing all progress that has been made since the conception of one-bit coding. Throughout the paper, therefore, reference will frequently be made to the compendium [3] which presents a detailed description of work related to SD modulation.

2.1 History and fundamental principle of SDM

The basic property of all one-bit coders, is that they try to obtain a sequence of -1s and +1s such, that over a specified bandwidth, the output is an accurate representation of the input. This is schematically depicted in Fig. 1.
In Fig. 1, a one-bit code is generated by a device 'one-bit coder'. The input, which can have any number of bits n but runs at the same rate as the output, is subtracted after appropriate delay, from the output of the device. Subsequently, an error measure ϵ is defined over a limited bandwidth; in the currently discussed case, this is usually the audio band. This error measure can be the instantaneous error $\epsilon(t)$, but can also be an integral ϵ_{int}, integrated over a certain period of time. Obviously, all one-bit coders are designed such as to minimize some of these error measures in some way, which is basically the definition of a one-bit coder. Historically, the so-called 'Delta Modulator' [1] is the first device which purposely tried to achieve this.
A decade after its introduction, the 'error feedback loop' [6] was introduced, which is very familiar to the current SDM designs. A schematic of an error feedback loop is depicted in Fig. 2. Central is the one-bit quantizer, indicated by the block with the step-function, which, in clock cycle i, produces an output bit $y(i)$ and introduces an error $e(i)$. In the next clock

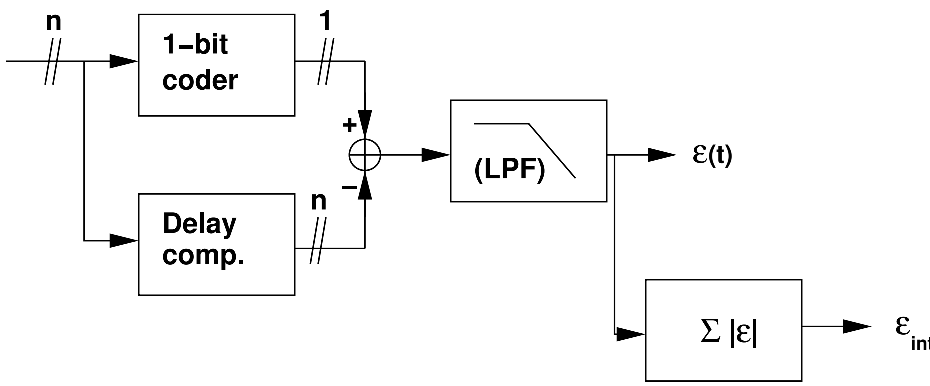

Figure 1: Definition of the error quantity ϵ that all coders try to minimize.

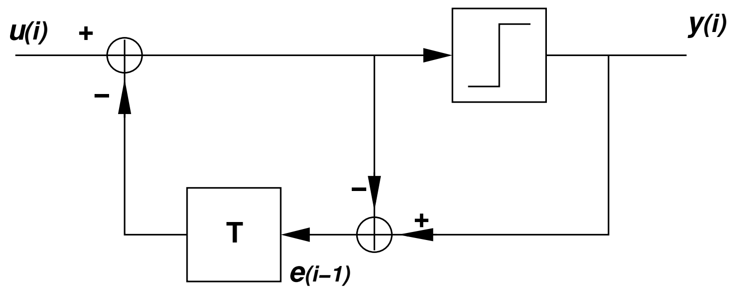

Figure 2: Schematic of the 'error feedback loop' [6].

cycle $i+1$, after a single clock cycle delay, indicated by the block labelled 'T', an attempt is made to correct for this error by subtracting it from the input $u(i+1)$. Hence, in accordance with Fig. 1, the device tries to minimize the *instantaneous* error $\epsilon(t)$ as measured by a first-order low pass filter. Note the wording 'instanteneous', as any future errors that will be produced, are not taken into account.
After the initial introduction of the delta modulator, various variations and improvements over this structure have been proposed in the period 1960-1990 [3].

2.2 Basic topologies of SDMs

Of all known topologies of one-bit coders, some occur more frequently than others. In this section, the most commonly occurring topologies will be discussed. These topologies are often quite useful too in connection to few-bit coders, for which reason the quantizer devices are labelled with a 'Q' (instead of a step-function). However, the focus will be on one-bit coders. The first topology is a generalization of the error feedback loop, and is depicted in Fig. 3. This particular design is called Noise-Shaper (NS). While in the original error feedback loop the quantizer error is only low pass filtered through a first order filter, in a general NS the error is filtered by a filter $F(z)$, which can, in principle, be any design. Clearly, a typical design will choose $F(z)$ such as to minimize the error ϵ (with ϵ defined by the designer). Another frequently employed structure is the 'feedforward Sigma Delta Modulator' (or FFSDM). Its structure is depicted in Fig. 4. Clearly, it bears great resemblance to the noise shaper, and, in fact, the two structures can be made identical. When the filter $F(z)$ is chosen as $F(z) = \frac{H(z)}{H(z)+1}$, and the input of the NS is pre-multiplied with $\frac{H(z)}{H(z)+1}$, too, the two topologies are identical. To introduce the distributed feedback type SDM, we first present a more detailed implementation of a fourth order FFSDM in Fig. 5. We see that the (loop)filter $H(z)$ is made

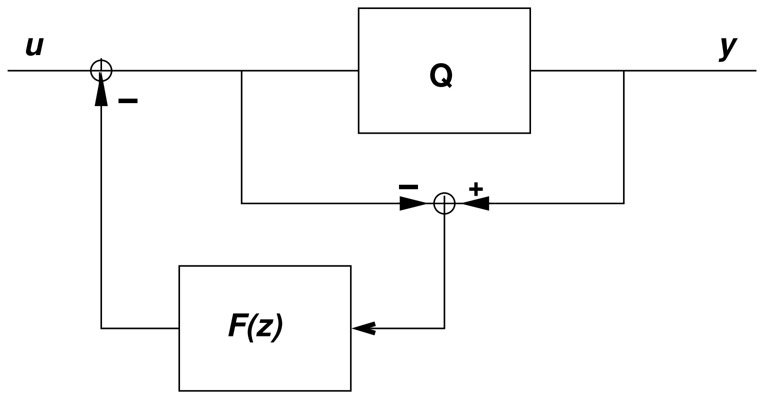

Figure 3: Schematic of a 'Noise Shaper'.

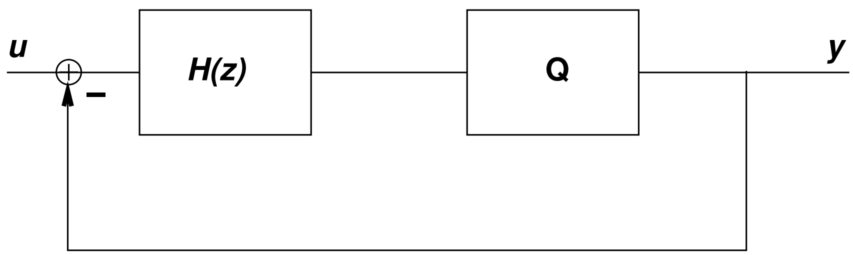

Figure 4: Schematic of a Sigma Delta Modulator.

up of 4 integrator sections, each of which consists of a delay and a summing element. The outputs of all integrators are weighted by coefficients c_i, and the weighted contributions of all integrators are summed and fed to the quantizer Q. In Fig. 6, the 'distributed feedback SDM' (FBSDM) is depicted. With the coefficients c_i and c_i' properly chosen, the FFSDM and FBSDM can be made almost identical; with a slightly more generic representation of the FBSDM, they can be made completely identical with respect to their noise shaping characteristics. [7].

With all these different noise shaper and SDM designs, it is clear that the choice of which topology to use is dependent on the design of the complete system, and aspects like system architecture and cost dictate what the optimal topology will be. From an analysis point of view, study of a single SDM design will provide, after simple linear manipulations, the results for all topologies.

In the next section, we will study a simplified model of noise shapers and SDMs, which allows us to gain some initial insight into the design of SDMs and their noise shaping properties.

3 Sigma Delta design

The most important characteristic of an SDM is its (quantization) noise shaping function. While the precise description of the noise shaping characteristic of a one-bit SDM is very difficult, a useful pragmatic approach has been developed, based on linear system theory, which allows engineers to create a realistic SDM design [8]. In this section, the most important assumptions will be outlined, and an SDM design method will be described closely following [8].

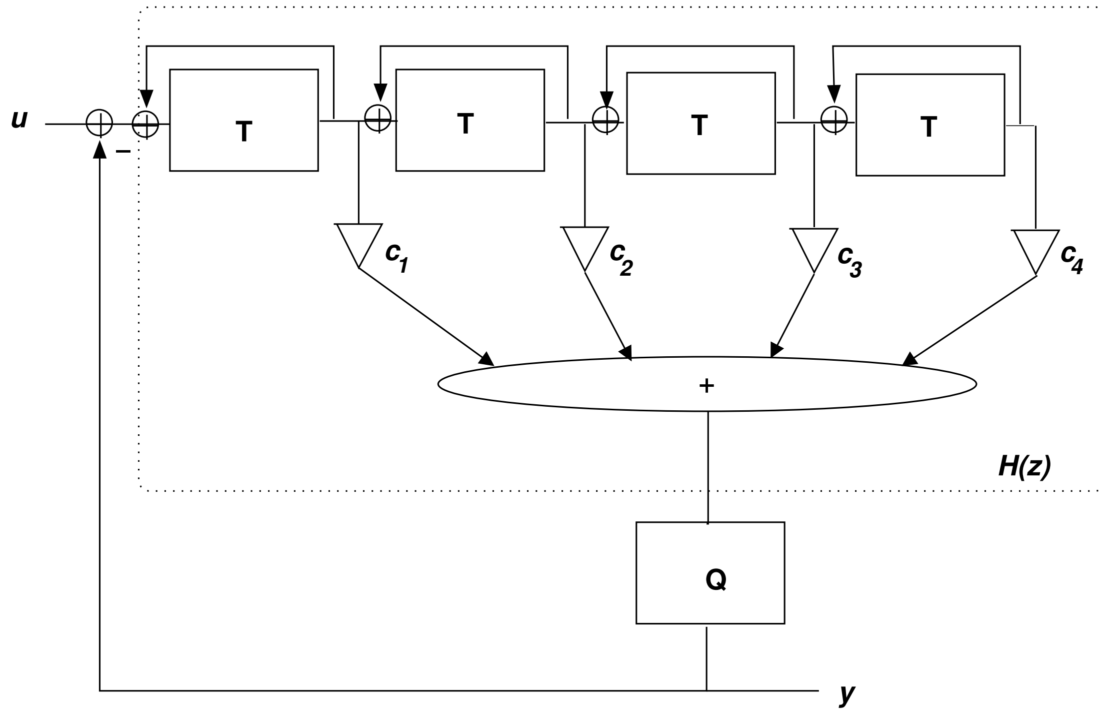

Figure 5: Schematic of a fourth-order, feedforward Sigma Delta Modulator.

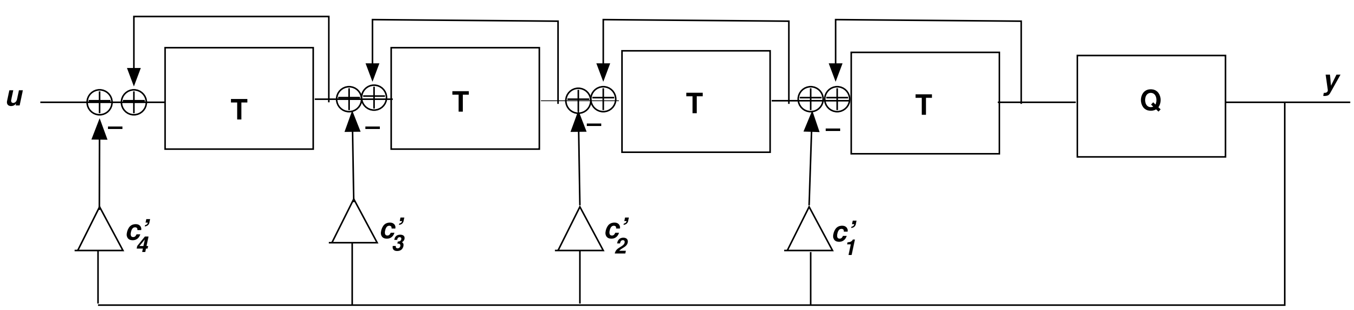

Figure 6: Schematic of a fourth-order, distributed feedback Sigma Delta Modulator.

3.1 A linear model of the SDM

For applications in one-bit audio, the quantizer Q in a SDM is a one-bit quantizer, which outputs only values of $+1$ and -1. This is a highly non-linear element, which renders the full analysis of a SDM difficult, if not impossible. Up to this moment, no complete mathematical theory exists which describes in full detail the behaviour of a SDM. To gain some initial insight in the characteristics of the SDM, however, we will resort to a simple linear model and replace the highly non-linear quantizer by a (linear) gain c and an additive noise source n, which models the quantization error, as indicated for the SDM topology in Fig. 7. Because the other topologies are, within the same approximation, linearly related to this SDM topology (see Sec. 2.2), we will restrict the discussion to SDMs only.

While this linear model is a reasonable assumption for multi-bit quantizers, it is hardly justifiable for a one-bit quantizer. Still, it is the only approximation which results in tractable mathematics[1]. Doing this, we can write for the signal transfer function (STF) and the noise

[1] Corroborating the proverb 'If the only tool you have is a hammer, you will see every problem as a nail'.

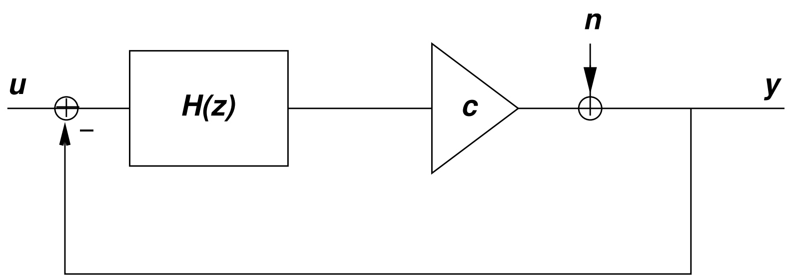

Figure 7: Linearization of Sigma Delta structure. The quantizer is replaced by a (signal independent) gain, and an additive noise source. The signal transfer function STF and noise transfer function NTF are defined by $Y = STF.U + NTF.N$, where Y is the fourier transform of the output y, U is the discrete fourier transform (DFT) of the input u and N the DFT of the additive noise n.

transfer function (NTF) the following expressions:

$$STF(z) = \frac{cH(z)}{1 + cH(z)}$$

$$NTF(z) = \frac{1}{1 + cH(z)} \tag{1}$$

While models of various degrees of sophistication exist [9, 10, 11] to obtain the (signal dependent) values of the quantizer gain c and its possible phase shift, we will for simplicity assume that the gain $c \approx 1$ because it allows us to obtain some information on the most basic aspect of a SDM: its general noise shaping characteristic. It should be stressed, that this assumption does not allow any detailed prediction with respect to its signal characteristics; for these analyses, the methods of [9, 10, 11] are better suited, though still not flawless. Eq. 1 shows how, in a situation where the loop-gain $H(z)$ is very large, the signal transfer function approximates 1. The noise transfer function, on the contrary, is negligible for large $H(z)$. This shows that in one-bit audio applications, where the loop-filter $H(z)$ typically is chosen as a low pass filter with large LF gains, the quantization noise in the audio band is strongly suppressed.
It is of crucial importance, however, to realize that the replacement of the quantizer by a gain element c and an additive noise source, is a very crude approximation, the more so if $c = 1$ is taken. Typically, the Signal-to-Noise Ratios (SNRs) as calculated from simulations on the actual SDM with the non-linearity included, differ significantly from those obtained by the use of the linearized model. Also other characteristics, discussed in Sec. 4, are not properly, or not at all, explained by the linearized model. It does give us some insight, though, in the way the quantization noise is spectrally shaped and what filtering is applied to the input signal u.

3.2 Loop-filter design

A very convenient way to start the design of a SDM modulator [8] is the linear model of Fig. 7, where we take the gain $c = 1$. We take the feed-forward structure from Fig. 5, and write down the NTF that is associated with it. We can write for the loop-filter $H(z)$:

$$H(z) = c_1 \frac{z^{-1}}{1 - z^{-1}} + c_2 (\frac{z^{-1}}{1 - z^{-1}})^2 + c_3 (\frac{z^{-1}}{1 - z^{-1}})^3 + c_4 (\frac{z^{-1}}{1 - z^{-1}})^4 \tag{2}$$

and making use of the relation $NTF(z) = 1/(1 + H(z))$ we arrive at:

$$NTF(z) = \frac{(1 - z^{-1})^4}{(1 - z^{-1})^4 + c_1 z^{-1}(1 - z^{-1})^3 + c_2 z^{-2}(1 - z^{-1})^2 + c_3 z^{-3}(1 - z^{-1}) + c_4 z^{-4}} \quad (3)$$

which is to be recognized as a filter of the appearance $NTF(z) = (1 - z^{-1})^n / P_n(z^{-1})$. This is the form of a Butterworth or a Chebyshev type II filter[2]; the choice of either of those realizations dictates the final appearance of the n'th order polynomial $P_n(z)$. Likewise, the STF can be computed as $STF(z) = 1 - NTF(z)$, resulting in:

$$STF(z) = \frac{c_1 z^{-1}(1 - z^{-1})^3 + c_2 z^{-2}(1 - z^{-1})^2 + c_3 z^{-3}(1 - z^{-1}) + c_4 z^{-4}}{(1 - z^{-1})^4 + c_1 z^{-1}(1 - z^{-1})^3 + c_2 z^{-2}(1 - z^{-1})^2 + c_3 z^{-3}(1 - z^{-1}) + c_4 z^{-4}} \quad (4)$$

The approach that can now be followed is to design a high-pass filter for $NTF(z)$, according to Butterworth or Chebyshev-II (or any other) rules, and reorganize terms such that it is in the shape of Eq. (3). One way of approaching this is to use a symbolic manipulation package such as Mathematica [12], or to collect terms in powers of z and equate identical powers. From an engineering point of view, a very easy way of obtaining the coefficients c_i is by recognizing that $1/NTF(z)$ is linear in the coefficients c_i. It is then possible to set up a linear system for (at least as many as the order of the system) different values of z. These values must have no simple relation to each other to avoid linear dependency in the system, but need not be complex. In this way, it is also irrelevant whether the Butterworth filter is provided as a cascade of biquads, or as a direct realization.
When we inspect the feedback structure (Fig. 6), we see that the transfer characteristic for the $NTF(z)$ takes the same shape as the NTF of the feed-forward structure discussed above. However, the STF is given by

$$STF(z) = \frac{z^{-4}}{(1 - z^{-1})^4 + c'_1 z^{-1}(1 - z^{-1})^3 + c'_2 z^{-2}(1 - z^{-1})^2 + c'_3 z^{-3}(1 - z^{-1}) + c'_4 z^{-4}} \quad (5)$$

which, for low frequencies equals about 1 if the coefficients c_i are scaled as $c'_1 = \frac{c_1}{c_4}$; $c'_2 = \frac{c_2}{c_4}$; $c'_3 = \frac{c_3}{c_4}$; $c'_4 = 1$. For higher frequencies, the STF displays an almost third-order roll-off. This is in contrast to the feed-forward topology, where the STF rolls off only very slightly (first order) for high frequencies. In App. A, an example design of a SDM will be presented. In Fig. 8, the different STF's for a feed-forward and a distributed feedback structure, with an identical NTF, have been calculated. The NTF's are designed as 4'th order Butterworth high pass characteristics, with a cut-off frequency of 150 kHz. Clearly, the strong roll-off characteristic of the feedback structure can be observed. Interestingly, the feed-forward topology displays a strong peak in its transfer characteristic at the cross-over frequency. Because this feature is due to the complex nature of $H(z)$, it is not obvious from Eq. (1) if only the magnitude response $|H|$ is used. The maximum peak height is in this case about 6 dB.
This loop-filter design gives rise to an SDM with a maximum (peak) input of about -5 dB (*i.e.*, 0.57 w.r.t. the feedback signal from the quantizer). Above this input level, the SDM turns unstable (see also Sec. 4.1). At an input of a sine with a peak amplitude of 0.5, the (unweighted) Signal to Noise Ratio (SNR) in the band 0-20 kHz is about 97 dB. In high-end audio applications, often a signal-to-noise ratio of better than 100 dB is desirable. However,

[2] albeit scaled such that the first term $c_0 z^0$ of $H(z)$ equals zero. If this term were non-zero, the resulting SDM would not contain a delay in the closed loop and hence would not be realizable.

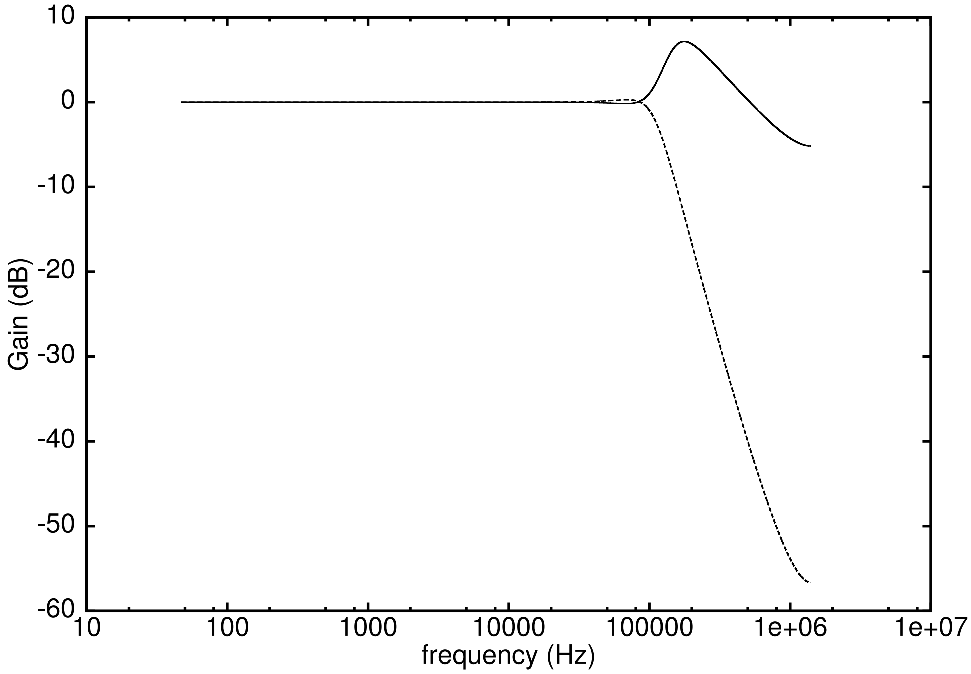

Figure 8: Signal transfer functions for a feed-forward topology (solid) and a distributed feedback topology (dashed) with identical NTF's.

one might argue that the A-weighted SNR is much better, because the noise floor is large only for frequencies close to 20 kHz. Indeed, for this example, the A-weighted SNR amounts to about 105 dB, where the large apparent improvement in SNR is due to the fact that the noisefloor increases with frequency. Still, this is judged to be insufficient for hifi applications. One way of increasing the SNR in the audio band, while hardly reducing the maximum input level, is to use higher order filters for the NTF, and to use a Chebyshev type II -like high pass filter for the NTF design instead of a Butterworth characteristic. Another option is to create notches in the NTF, which can easily be created in SDM's by the construction of resonator sections, as displayed in Fig. 9.

The construction in Fig. 9 is, in principle, applicable to a feed-forward topology; for a feedback topology, a similar arrangement with a feedback loop over two integrator sections is possible.

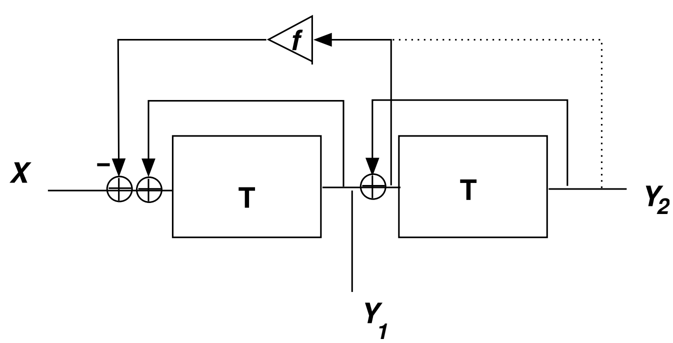

Figure 9: A cascade of two integrator sections in a SDM, with a feedback loop between the integrators. The two different ways of incorporating the feedback loop result in slightly different pole characteristics. Indicated are the two different outputs, which are characterized by a transfer function $R_1(z)$ and $R_2(z)$, respectively.

In Fig. 9, two outputs of the resonator section are indicated as Y_1 and Y_2; the relation between these is that $Y_2(z) = h(z)Y_1(z)$, designating the transfer characteristic of the integrator section as $h(z) = z^{-1}/(1 - z^{-1})$.
Also, two different realizations of the feedback path (with coefficient f) are possible. The solid curve in Fig. 9 does not incorporate the delay that the dotted realization does. The general effect of a resonator can be obtained by studying the structure corresponding to the solid drawn topology. The resonator transfer functions $R_1(z), R_2(z)$, defined by $Y_1(z) = R_1(z)X(z)$ and $Y_2(z) = R_2(z)X(z)$, are given by:

$$R_1(z) = \frac{h(z)}{1 + zfh(z)^2}; \quad R_2(z) = h(z)R_1(z) \tag{6}$$

The poles of $R_1(z)$ and $R_2(z)$ are given by

$$z_p = 1 - \frac{f}{2} \pm \frac{i}{2}\sqrt{4f - f^2} \tag{7}$$

These poles are exactly on the unit circle; this differs from the dotted structure where the poles are outside the unit circle. The pole frequencies are given by:

$$f_{pole} = \mathrm{acos}(1 - \frac{f}{2}) \tag{8}$$

which, for small values of f, virtually coincides with the pole frequencies for the dotted structure. As such a feedback loop over two integrator sections transforms the two poles at DC ($z^{-1} = 1$) into two complex conjugate poles away from DC, care should be taken that there is enough DC gain in the loop-filter to avoid DC drift. As an example, consider the 4th order SDM with a Butterworth design, corner frequency 150 kHz, and made up by a cascade of two resonator sections as in Fig. 9; the design details are provided in App. A. Choosing the poles to move from DC to ± 10 and ± 19 kHz, the corresponding numerical values of the feedback coefficients are 0.000496 and 0.001789. The SDM obtained has a maximum input of 0.57 (0.57 without resonators) and a SNR of 107 dB (97 dB without resonators). Indeed, the addition of the poles, turning the Butterworth characteristic in to a Chebyshev II - like characteristic, gives significantly better SNR; the DC suppression of the loopfilter is still better than 120 dB, which is sufficient. Compared to the A-weighted SNR figures, the improvement is less, because the poles primarily serve to suppress the noise between 10 and 20 kHz.
A further improvement can be obtained when using a fifth order SDM, with a Butterworth NTF design (corner frequency 110 kHz) plus the poles at 10,19 kHz: in that case the SDM is stable to inputs up to 0.58, with a SNR of 120 dB. Note, that in this case, there is still 1 integrator with a pole at DC, and thus there cannot be any DC drift. To clarify the operation of such a SDM, pseudo-code of the SDM is provided in App. B.
In Fig. 10, the effect of the resonator sections is illustrated. We see, that with the resonators, the quantisation noise is substantially suppressed in the area 10-20 kHz, whereas below 10 kHz the SDM without resonators has the better performance.

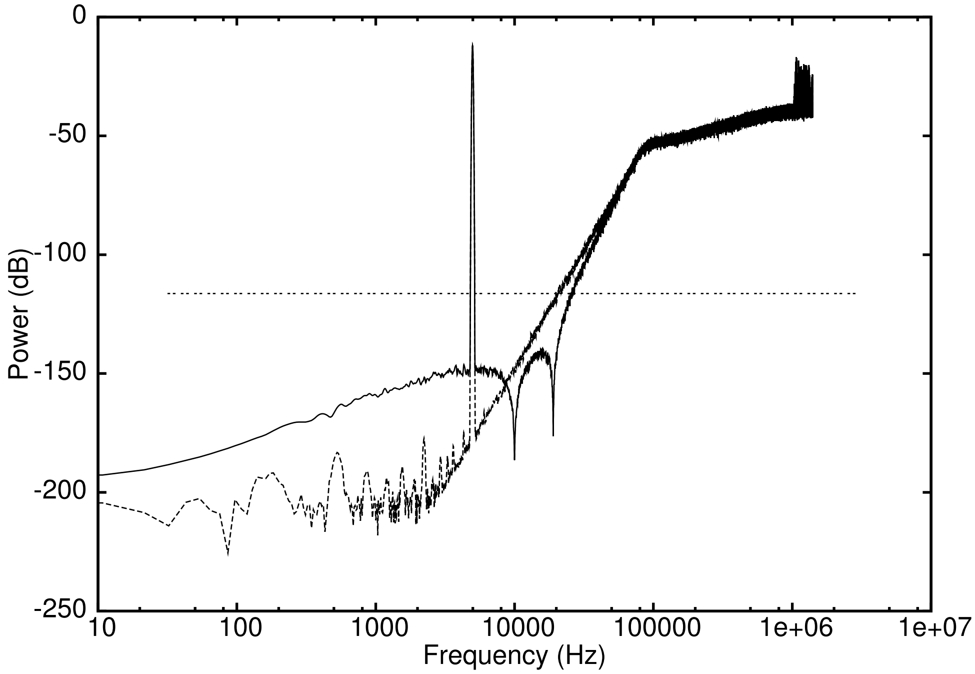

Figure 10: Spectrum of a fifth-order SDM, with (solid) and without (dashed) resonator sections. The SDMs in this example are undithered. The horizontal line represents the 16 bit resolution level (97 dB SNR over 20 kHz).

4 Signal Characteristics of one-bit SDMs

The fact that a one-bit SDM (and likewise any other one-bit coder topology) contains a strong non-linearity, namely a one-bit quantizer, has its ramifications on the behaviour of the device, which often cannot be predicted by a linearized model. In the next sections, a number of the effects which cannot be described by (currently known) linear approximations will be described heuristically. Whenever not mentioned specifically, it is assumed that the sample rate equals 64 times 44.1 kHz, and that the SDM is used as the one-bit coder topology.
Further, we will use as a reference level (0 dB) the level of the feedback path. This differs from the often used definitions in one-bit audio, which take half the level of the feedback path as 0 dB (50% modulation depth). Signal-to-Noise Ratios (SNRs) are determined as the SNR at the maximum signal level the SDM can accomodate without overload.

4.1 Stability

For every SDM design, there is a trade-off between stability of the modulator and the SNR in the base-band. As an example, consider the results in Table 1 for different 5'th order SDM's, which have all been created using Butterworth high pass filters as design NTF.
So far, we have not bothered about what happens if the SDM input exceeds its maximum: the SDM gets into wild oscillations, with constantly increasing amplitude in the integrator states and decreasing frequency. Even worse, when the input is removed from the system, the SDM does not return to its original state. To avoid such a situation, it is customary to use *clippers* in each integrator stage. In Fig. 11, a schematic representation of a clipped integrator is given. The idea is that the output of the integrator can never exceed its *clip* value, C. In other words, the integrator section simply stops integrating when the clipping level C has been reached

cut-off (kHz)	SNR (dB)	max. input level
80	95	0.77
90	100	0.71
100	104	0.66
110	106	0.60

Table 1: Trade-off of the maximum input range and the SNR in the base-band for a series of 5'th order modulators (Butterworth NTF design).

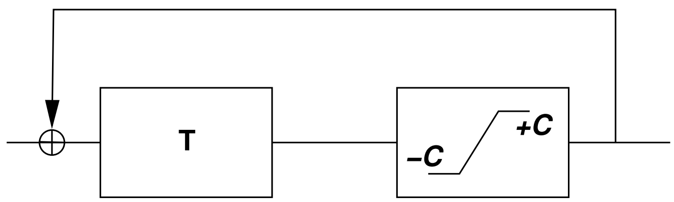

Figure 11: Principle of a clipped integrator. The absolute value of the output of the integrator cannot exceed a value of C.

The purpose of these clippers is to avoid a situation where the values in the integrator stages get too high (and cause the SDM to start to oscillate), while still allowing integrator values which occur during normal operation. Whereas the main purpose of the clippers is to let the SDM return to normal operation *after* overload, it is also desirable to avoid serious distortion in the signal if clipping occurs.

A heuristic way of obtaining reasonable numerical values for the clipper levels is to monitor the integrator levels during very large sine wave inputs and square wave inputs, close to overload of the SDM. The clipper levels C_1 and C_2 of the first 2 integrator stages can be set according to these values. If the higher integrator stages are assigned values according to this recipe as well, the situation occurs that the SDM returns to normal operation after overload, but can have all clippers activated simultaneously. This will cause serious clicks and pops (especially if the first integrators run into their clippers). Hence, the higher order clippers should be designed such that the high order clippers are activated first, before the low order clippers are activated.

From table 2, we can obtain some idea about the influence of the clippers on the SDM operation. The clippers are sometimes activated during continuous operation at 0.5 input level, which causes a small reduction in SNR with respect to the 120 dB without clippers. However, whereas the original SDM turned unstable at inputs of 0.59, its clipped version

Input level	C_1	C_2	C_3	C_4	C_5	SNR (dB)
0.5	0	0	0	0	1836	118
0.55	0	0	0	0	6595	117
0.59	0	0	12	57	16285	107
0.60	0	5	48	175	18829	104
0.65	0	512	2283	3258	38155	67

Table 2: Typical example of the influence of clippers on normal SDM operation. The columns with clippers C_i indicate the number of times a clipper was activated in a run of 300,000 samples.

shows continuous stable operation. Even at inputs of 0.65, the first integrator is not clipped, indicating that the signal distortion is still limited, and highly audible clicks are absent. In fact, only at input levels exceeding 0.75, will the initial integrator clip, causing a clearly audible effect. At the level of 0.75, the SNR has dropped to about 60 dB. Typically, in one-bit audio the maximum input level is defined such, that clipping only rarely occurs.
As an alternative, or in addition to, clipping in the SDM, clipping *before* the SDM might be considered. However, in this case dynamic range must be sacrificed, although the resulting system is unconditionally stable for large inputs.

4.2 Spectral properties

Due to the inherent non-linearity of the one-bit quantizer, the spectrum of a one-bit coder potentially exhibits signs of distortion or other spurious signals. This is a well-known issue, and various ways of reducing or removing the effects caused by such a non-linearity have been proposed [3]. Since dithering the quantizer has proven in (L)PCM to remove any non-linearity due to the quantization effect [13], this has been the first method resorted to in literature to linearize SDMs, too. In this section, we will discuss the appearance of non-linearity (NL) in an SDM, even though in practical audio applications the influence of this NL is so benign as to be absent, as can be seen from inspection of Fig. 10, where no NL can be observed above the quantization noise floor at -150 dB.

4.2.1 Undithered SDM's

An appearance of the inherent non-linearity due to the one-bit quantizer can be observed in the spectrum of a SDM. Whereas for high order SDMs, which are typically used for high-end audio applications, the effects of non-linearity are hardly visible, they are for low-order SDMs, and are also well-documented [3]. For that reason, we will restrict our analysis in this section to a third-order SDM, as used in [14], which is notorious for its bad signal properties. The spectrum of the third order SDM that will be used in the remainder of this paper devoted to linearization techniques, is shown in Fig. 12.
The SDM is of the feedforward type, and is characterised by the following NTF:

$$NTF(z) = \frac{1 - 3.00z^{-1} + 3.00.z^{-2} - z^{-3}}{1 - 2.34z^{-1} + 1.87z^{-2} - 0.51z^{-3}} \tag{9}$$

While this third-order SDM displays a dynamic range of about 91 dB, its third harmonic is at a level of -110 dB. While this is still a rather respectable number, it is an unacceptable number for high-end audio applications. Also, the higher order harmonic distortion products are significant, too. It should be remarked, that this type of SDM is not recommended for practical use. To further illustrate the non-linearity of the SDM, the NTFs which are expected from linear modelling are included in Fig. 12. The dotted curve is what would be expected on basis of a 64 time coherent average [15] (thus reducing any uncorrelated component by $3\log_2(64) = 18dB$), the dashed curve is what would be expected without any coherent averaging at all. These curves show that the NTF that is obtained does not resemble the theoretically expected NTF. In particular, in the frequency regime above 700 kHz a significant number of highly correlated components is visible. The total amount of coherent power of the SDM shown in Fig. 12 amounts to almost half of the total power, which exceeds the signal power (-18 dB) by far. As a result of these high powered HF components, the noise floor in other parts of the spectrum is actually *lower* than expected. The explanation for this phenomenon is that the total output power of a one-bit code is *constant* and equals 1; this is in sharp contrast with any other quantized code. Therefore, power which is spent

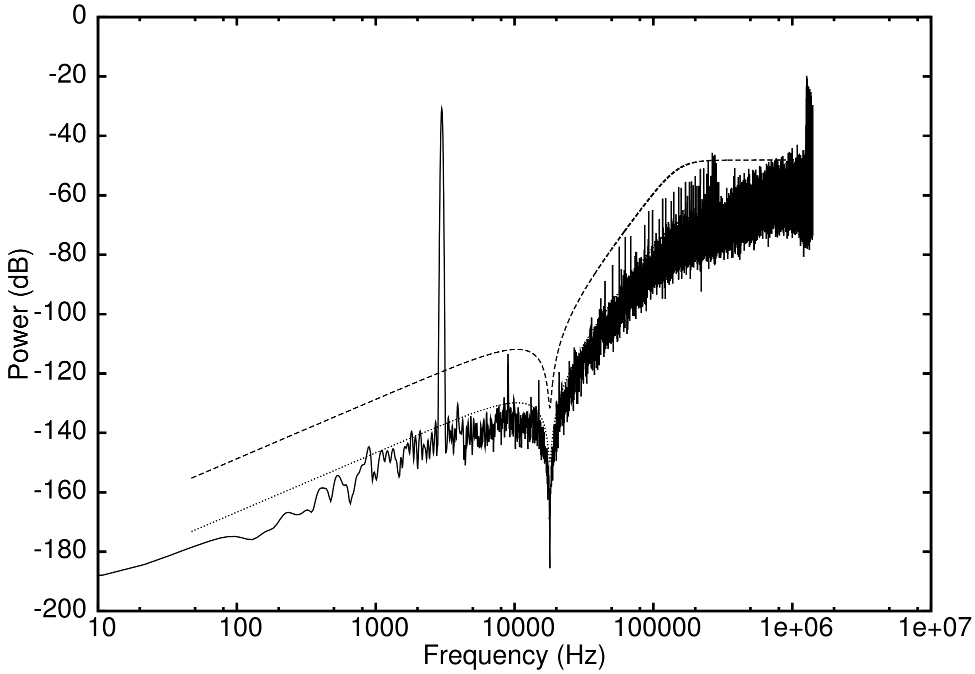

Figure 12: Spectrum of the third order noise-shaper used in the analysis of linearization techniques. The input signal is a 3 kHz sine wave, -18 dB. To obtain this spectrum, a series of 64 coherent averages and 10 power averages has been used. The dashed curve is the theoretically expected NTF; the dotted curve the expected NTF after 64 coherent averages.

in a particular spectral region, will be removed from another region and vice versa. This is quite a special situation, as due to this phenomenon one-bit noise shaping does *not* follow the Gerzon-Craven theorem [16] in contrast to multi-bit noise shapers[3].

4.2.2 Dithered SDM's

While for PCM there exists mathematically optimal dither [13], namely TPDF dither (dither with triangularly shaped pdf) spanning 2 LSBs, we cannot expect TPDF dither to linearize the one-bit quantizer simply because it spans only 1 bit. Many dithering schemes have been proposed for SDMs, some being more effective than others [17, 18]. Based on PCM knowledge, it has been tempting to apply the dither just before the quantizer as shown in Fig. 13. In the sequel, we will refer to this dither as amplitude domain dither. A clear advantage of such an approach is that the dither will be noise-shaped, and thus has little influence on the SNR in the signal band [17]. Nevertheless, it is not clear whether dithering just before the quantizer is most effective; as we may see in Sec. 4.3, and as conjectured in [19] it is most probably not. The dither that we found optimal for most applications is RPDF (dither with rectangularly shaped pdf) with a width that strongly depends on the SDM used. In Fig. 14, the spectrum of our third order SDM is depicted when it is dithered by RPDF dither of a half width of 0.8 (with quantizer levels -1,+1, this means the dither has a peak-to-peak value of 1.6). Obviously, the dither does a very good job in linearizing the system. The observed NTF

[3]This can be most easily inferred from the basic assumption in [16] that the Shannon theorem can be employed in the noise shaping case, which assumes a signal-independent SNR. As for a one-bit SDM $S+N=1$ always, the Gerzon-Craven theorem does not hold in its published form.

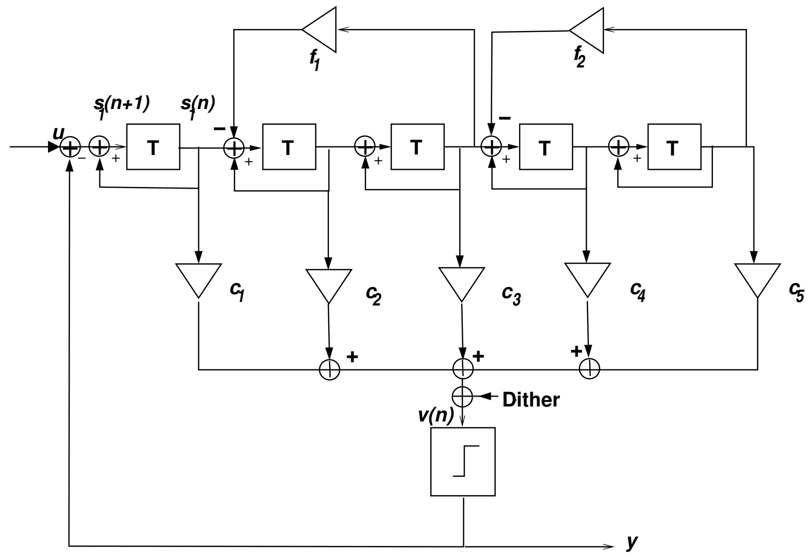

Figure 13: Application of (amplitude domain) dither just before the quantizer in a (in this case fifth order) SDM. Also indicated is the first of the 'states': $s_1(n+1)$ before, and $s_1(n)$ after the first delay element. The states for the subsequent delay-elements can be assigned accordingly, which lead to the state-space description of a SDM - see Sec. 4.3

closely follows the theoretically predicted curve, and there is no obvious sign of distortion. Due to the fact that the high powered HF components are absent now, the noise floor in the low frequency part has increased. As a result, the SNR of the SDM has dropped from 94 dB to about 82 dB. Again, note that this is almost purely due to re-distribution of power, and not to the additive character of dither (as is the case in dithering a multi-level quantizer!). Hence, this is a penalty to pay for linearizing the SDM. A further penalty to pay is the decrease in stability. In this example, the original SDM is stable for DC inputs up to 0.84; the dithered SDM up to only 0.55.

These observations have seeded the thought whether it would be necessary to linearize a SDM in the literal sense; as the only desire we have is that the lower 100 kHz is represented correctly, it might be unrealistic to require linearization to the extent that also the high powered HF components disappear. When, for example, the SDM is dithered using RPDF dither of half width 0.5, the linearization is not complete, as can be inferred from Fig. 15. The NTF is slightly different from what is theoretically expected, but, more importantly, the HF components have re-appeared.

As a result, the SNR has increased from 82 dB to 88 dB, and the maximum DC input has increased from 0.53 to 0.69. Hence, it is always advisable to judge the trade-off between advantages and disadvantages of dithering. In the same line of thinking, a pre-correction scheme for SDMs has been proposed [14], which aims at correcting any errors made by a first SDM with a second SDM. This technique has proven to be much more powerful compared to dithering a SDM; without dithering, distortion components in the audio band can be reduced to levels below -130 dB; when a tiny dither level of 0.01 is applied, distortion lowers to levels beneath -140 dB.

Another line of investigation has been initiated due to the fact that the standard way of dithering ('amplitude dithering') appears to be quite inefficient. In fact, there is strong evidence that points in the direction that dither applied in the amplitude domain is not optimal at all [20, 19]. In [20], one-bit sigma delta modulation is compared to time-quantized

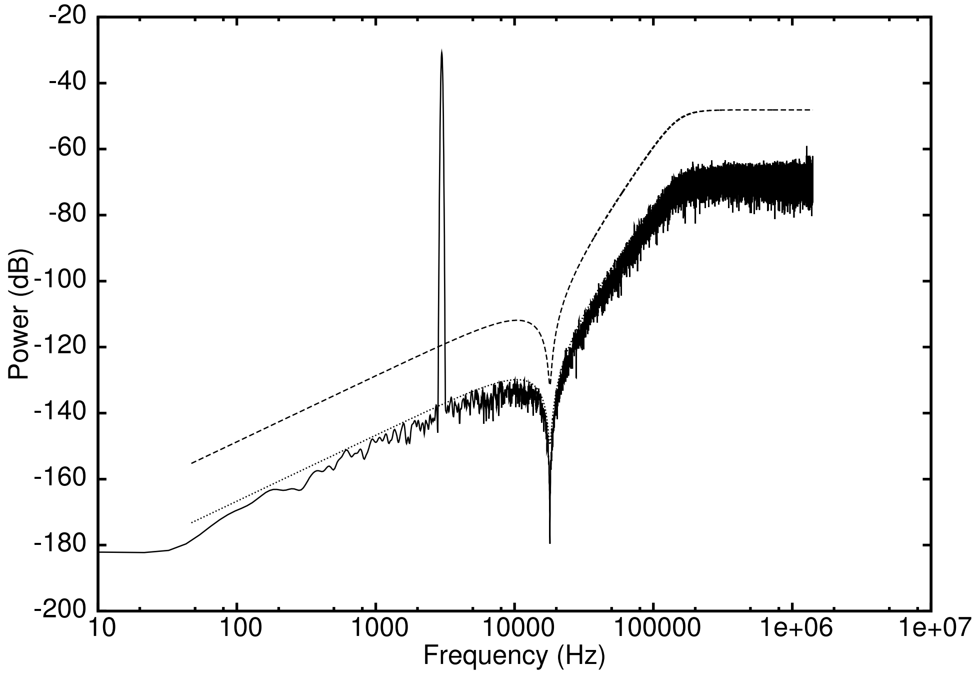

Figure 14: Spectrum of the third order noise-shaper with heavy dither. The input signal is a 3 kHz sine wave, -6 dB. To obtain this spectrum, a series of 64 coherent averages and 10 power averages has been used. The dashed curve is the theoretically expected NTF; the dotted curve the expected NTF after 64 coherent averages.

frequency modulation. To apply dithering, a technique called 'time-dispersion' is introduced, which is fundamentally equivalent to dither in the time domain. This technique effectively linearizes even a first-order SDM. This observation emphasizes the thought that for one-bit SDMs, even though optimal dither is yet to be defined, dithering schemes more effective than amplitude domain dither exist.

4.3 Limit cycles

Limit cycles are a known phenomenon in any system with feedback and non-linearity; as such, they are also known from the design of digital IIR filters [21]. Not unexpectedly, therefore, also in SDM design, limit cycles play an important role. We will use as a definition of a limit cycle a sequence of P output bits, which repeats itself indefinitely. Based on a state space description, several qualitative results can be obtained [22, 23], but it also proves possible to present an exact description of limit cycles in SDMs [24, 25, 26]. Even though limit cycles can exist for non-constant input [24], we will restrict the discussion to DC inputs only as these represent the most often occurring situation.

4.3.1 State Space description

The state space description is a highly convenient way to describe the behaviour of an SDM in the time domain. To illustrate the state space description of a SDM, Fig. 13 will be examined. This figure displays the states s_i in a feedforward topology of an $N = 5$'th order SDM with 2 resonator sections, as designed in Sec. 3.2. From Fig. 13 we can read that, in the absence of dither,

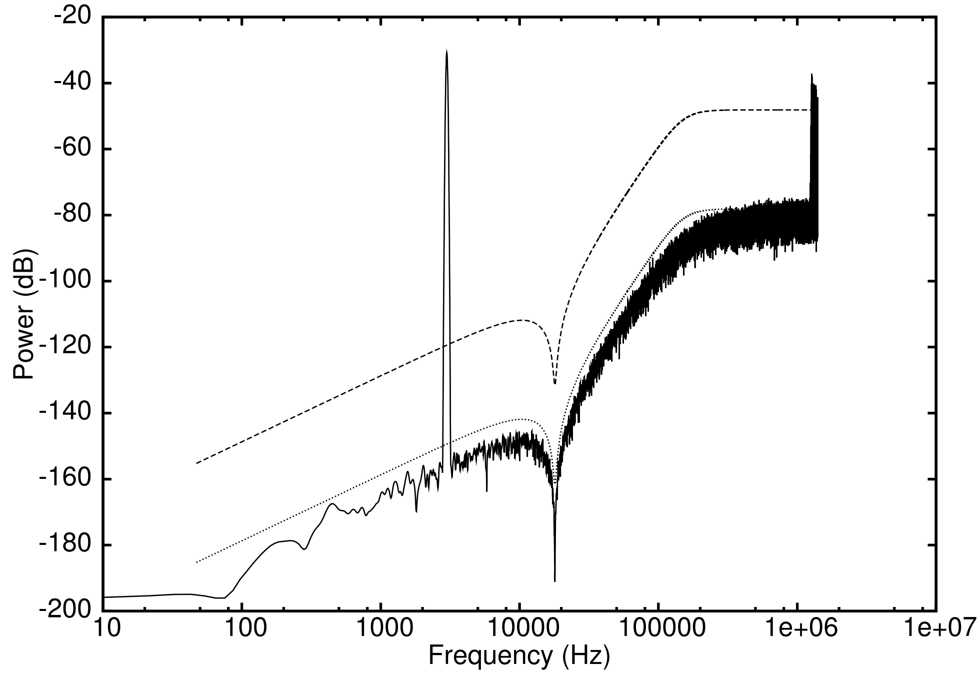

Figure 15: Spectrum of the third order noise-shaper with RPDF dither of half width 0.5. The input signal is a 3 kHz sine wave, -6 dB. To obtain this spectrum, a series of 1024 coherent averages and 10 power averages has been used. The dashed curve is the theoretically expected NTF; the dotted curve the expected NTF after 64 coherent averages.

$$\begin{aligned} v(n) &= \sum_{i=1}^{N} c_i s_i(n) \\ y(n) &= \text{sign}(v(n)) \end{aligned} \tag{10}$$

where $y(n)$ is the output bit at clock cycle n, $v(n)$ is the quantizer input signal, and $s(n)_i$ are the integrator outputs, called state variables. The c_i are the feedforward coefficients.
It can be shown [22, 27] that the evolution of states can be expressed concisely as

$$\begin{aligned} v(n) &= \mathbf{c}^T \mathbf{s}(n) \\ \mathbf{s}(n+1) &= \mathbf{A}\mathbf{s}(n) + (u(n) - y(n))\mathbf{d} \end{aligned} \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is called the *transition matrix* and $\mathbf{d} = (1,0,0,0,0)^T$ describes how the input and feedback are distributed. The power of the state space description is that it allows us to create a very compact description of the propagation of the SDM from time $t = 0$ to time $t = n$, as repeated application of Eq. (11) to $\mathbf{s}(0)$ leads to $\mathbf{s}(n)$:

$$\mathbf{s}(n) = \mathbf{A}^n \mathbf{s}(0) + [\sum_{i=0}^{n-1} (u(i) - y(i)) \mathbf{A}^{n-i-1}]\mathbf{d} \tag{12}$$

From the above equation, we can infer that the initial integrator states are simply a kind of an offset to the signal. The spectrum of the signal is determined completely by the second term in the right-hand side of Eq. (12); the first term carries no signal information. Hence,

this confirms the known fact that the signal content of a SDM modulator is not determined by its initial integrator states.
With minor adaptations, the same state space formalism can be applied to other one-bit coder topologies as well.

4.3.2 General formulation of limit cycle conditions

The compact representation Eq. (12) gives the means to directly view the consequences of a limit cycle. In dynamical systems theory, a limit cycle of period P can exist only if, for initial conditions $s(0)$,

$$\mathbf{s}(P+n) = \mathbf{s}(n) \tag{13}$$

for all n greater than or equal to zero [22]. However, from a practical point of view, we are interested in periodic behavior in the output y. It can be proven [26] that periodic y guarantees that a limit cycle exists. Thus we can use the limit cycle definitions and as a consequence, we have a strict set of necessary (but not sufficient!) equalities that need to hold for the initial states if periodic output is sustained:

$$(\mathbf{I} - \mathbf{A}^P)\mathbf{s}(0) = [\sum_{i=0}^{P-1}(u(i) - y(i))\mathbf{A}^{P-i-1}]\mathbf{d} \tag{14}$$

For most SDMs, the solution of Eq. (14) fixes all integrator states, except for the last integrator state [25], in order to have a valid limit cycle. Further requirements for a valid limit cycle are posed by the fact that, if the limit cycle is defined as a sequence $\{y(i)\}(i = 1, \ldots, P-1)$, we have for each $y(i)$:

$$y(i)v(i) = y(i)\mathbf{c}^T\mathbf{s}(i) > 0 \tag{15}$$

which either reduces the solution to Eq. 14 from a line to a line*piece* of limited length, or excludes the existance of a limit cycle with the assumed sequence of 1s. Because the line piece of solutions allows a limited variation of the last integrator, this is equivalent to the statement that a certain amount of dither can be added to the quantizer before the limit cycle is broken up. An important consequence is that since all other integrators need to have specified values, dithering any of these is extremely efficient in breaking up a limit cycle, and preferred over the classical way of dithering the quantizer.
The state space approach allows us to obtain important quantitative results on limitcycles in SDMs. For example, the minimum level of amplitude dither (when added in the classical way just before the quantizer, as illustrated in Fig. 13) that is necessary to remove any limit cycles, can been obtained [25]. While this may not be the most effective way to remove limit cycles, as dithering any but the last quantizer is more efficient, this situation occurs often in practice, justifying special attention. In Fig. 16, the minimum dither level that is needed to break up a limit cycle for DC input is depicted for a SDM with an aggressive NTF (1) and for a SDM with a mild NTF (2). The worst case situation is depicted with plus and star signs. This represents the minimum amount of dither that is needed to certainly break up the most stable limit cycle for SDM 1 and 2, respectively. While slightly more stable limit cycles can sometimes be found for non-DC inputs, these situations do not represent a practical situation. The first interesting observation is that the limit cycles for the aggressive SDM 1 are *more stable* with respect to dither than those of the less aggressive SMD 2. Also, we can see that there is a very stable limit cycle occurring around limit cycle length 22 for SDM 1, and for limit cycle length 32 for SDM 2. Upon investigation of these limit cycles, it appears that they consist of a series of 11 1s followed by 11 -1s for SDM 1, and likewise 16 1s

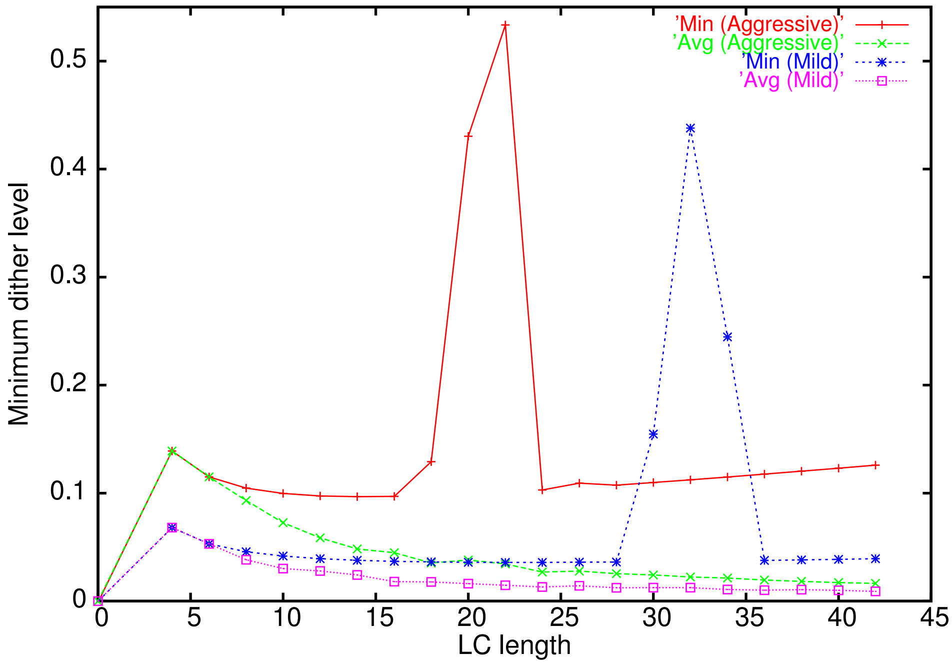

Figure 16: Dither needed to break up a limit cycle corresponding to a DC input 0. Plus-signs and stars represent the worst case situation: the level of dither necessary to certainly break up the most stable limit cycle, for an aggressive and mild SDM, respectively. Crosses and squares represent the average amount of dither needed to break up a limit cycle for the aggressive and mild SDM.

and 16 -1s for SDM 2. This corresponds to a square wave of frequency 120 kHz and 80 kHz, which are exactly the corner frequencies of the NTF design of the SDM 1 and 2 respectively. In practice, however, these limit cycles could never occur; upon the slightest disturbance of the integrators, the SDM runs unstable.

This is to be contrasted with the limit cycle behaviour for other limit cycle lengths. The shortest limit cycle, the sequence $\{1, -1\}$, appears to be most stable (disregarding the previously discussed limit cycles) for both SDMs. For longer limit cycles, the amount of dither needed for break-up decreases to a minimum value close to the peak, after which the limit cycle becomes more stable. All these limit cycles consist of the sequence $\{-1, 1, -1, 1, \ldots, -1, 1, -1, -1, 1, 1\}$, which represents the minimally possible deviation for the simple $\{-1, 1\}$ sequence. While these most stable limit cycles slightly increase in stability for longer limit cycles, on average the amount of dither necessary for break-up decreases. This is indicated with crosses and squares for SDM 1 and 2, respectively, in Fig. 16. The average amount of dither is defined as the average of the minimum dither levels that are needed to break up the individual limit cycles. Again, we see that SDM 1 presents limit cycles that are in general more stable than those of SDM 2. At limit cycle lengths of 42, the average amount of dither is reduced to about 0.03 and 0.017 for SDM 1 and 2, respectively, which is consistant with the intuition that longer limit cycles represent more boundary conditions to be fullfilled and are thus more easy to break up. While the amounts of dither discussed above are still significant, it should be remarked [25, 28] that dithering any but the last integrator is far more effective in disrupting the limit cycle than classical dither, and therefore presents a preferred alternative over dithering the quantizer.

5 The creation of one-bit content

In the previous sections, the standard generation of one-bit codes has been detailed. While this is, obviously, a crucial part in the high sampling rate concept of one-bit audio, more complex signal processing is most often necessary in the production of a music release. Also, signal characteristics of one-bit audio set some requirements on the replay of one-bit audio.
In this section, the signal chain leading to a one-bit audio delivery medium, and the signal chain for home audio delivery will be outlined. The section ends with an overview of signal processing techniques for one-bit audio.

5.1 The recording chain

In Fig. 17, several steps are envisaged which occur typically in the recording chain leading to the creation of a disc. Most of these steps involve analog or digital signal processing in one way or another. Starting with the AD converter, this is not necessarily a native one-bit converter. Often, high-end AD converters are 3-6 bit converters running at sample rates between $128f_s$ and $512f_s$, where f_s is symbolic for a sample rate of 44.1 kHz. While not necessary in principle, in practice these signal formats are often converted to one-bit formats. The main reason is that many people in the recording industry want to save on necessary disk space, whilst not giving in on sample rate; obviously, the only way to achieve this is to reduce the number of bits. If a change in sample rate is necessary, this can be done using standard upsampling or downsampling techniques [21]; most often, sample rate changes by more than a factor of 2 are not necessary. The change of few-bit to one-bit can be performed using a SDM, or any other one-bit coder.
In the editing phase, mild signal processing such as volume adjustments need to be done, and often switching between bit streams is necessary. Switching of bit streams is a technique which is rather different from standard signal processing, and is detailed in, for example, [27]. In the mixing phase, and to a lesser extent also in the mastering phase, heavy signal processing is often involved, ranging from relatively simple equalization to sophisticated reverberation techniques. Some examples of how one-bit audio can be processed, will be shown in Sec. 5.3.
In the authoring phase, finally, no changes to signal content are made anymore. In most cases, the data will be losslessly compressed in this phase. The compression that is employed for one-bit audio is a scalable compression technique, and is detailed in a companion paper [28].

5.2 The playback chain

An important aspect in replaying a one-bit audio recording is the presence of a substantial amount of HF (quantization) noise. This represents a large signal, and when the analog components in the delivery chain are not of exceptional quality, this could easily lead to distortion and other problematic effects. It appears that for most realistic SDM designs about 90% of the total amount of (the substantially coherent) quantization noise power, is above 800 kHz, as can also be judged from Fig. 10. The exact value of the frequency above which most of the correlated signal is found, is dependent on the signal which is input to the SDM; it will, however, never be very much lower than the quoted 800 kHz.
To judge whether these quantization noise components are harmful, we need to look at the full audio chain which is used to replay one-bit audio in a typical player. Such a configuration is shown in Fig. 18. A typical DAC-chip (see *e.g.* [29] or [30]) contains the first 4 blocks displayed in Fig. 18. The digital filter in the path leading to the n-bit SDM is a crucial part, where most of the HF signal present in the one-bit audio signal can be removed without any

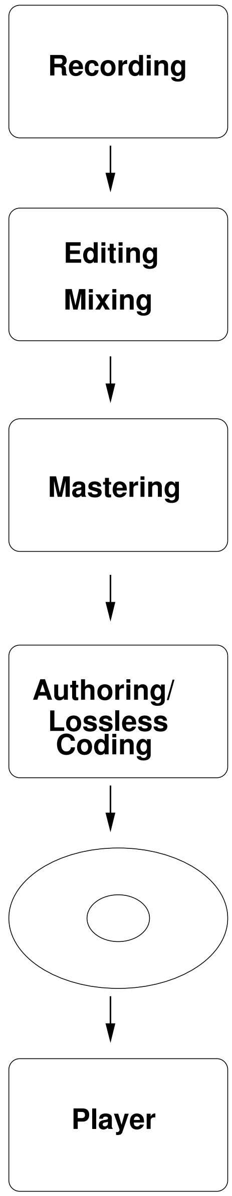

Figure 17: Typical signal processing chain for one-bit audio applications.

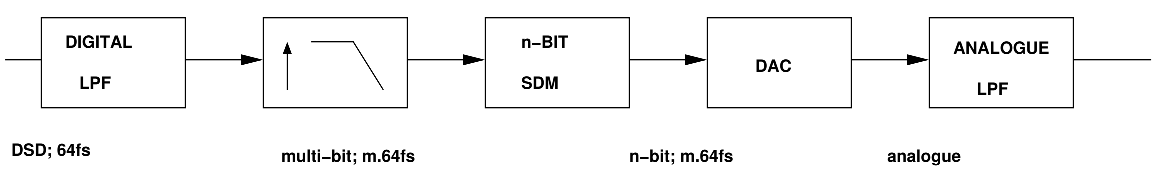

Figure 18: Example of an audio chain found in an one-bit capable player. The one-bit audio is first low pass filtered in the digital domain, followed by upsampling to m $\cdot$ f_s, typically, 128 or 256 f_s. This high-rate signal is then fed to an n-bit SDM, where n typically varies between 1.5 and 5. Finally, the analog output is passed through an analog low pass filter.

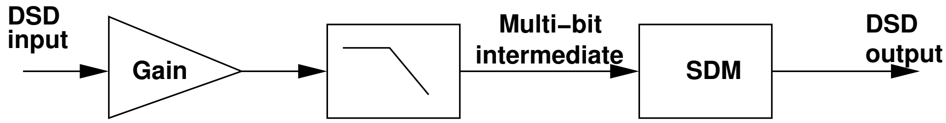

Figure 19: Example of performing two sequential operations on one-bit audio data. First, a gain adjustment is applied, after which an IIR filter operation is applied *without* leaving the intermediate high rate, multi-bit domain.

compromise. As an example, consider a filter that is designed according to the following criteria: pass-band: 0-100 kHz, flat within 0.01 dB; transition band 100 kHz - 800 kHz; stop band: 800 kHz - 1.4MHz, suppression -100 dB. This leads to a filter with only 22 taps, and thus does not pose any additional constraint in terms of hardware; the filters which are necessary to do proper upsampling from a low sample rate format to the required $m \cdot 64f_s$, are much more demanding. Also, the digital LPF does not influence the impulse response of one-bit audio [31], as the transition width is extremely large. It is clear, that the application of this filtering will lead to significant suppression of the high frequency components present in the original one-bit audio stream. Still, the signal contains substantial amounts of HF, which is foremost white noise. The signal is then upsampled to a frequency that is used to perform the digital-to-analog conversion on. As this upsampling also includes a digital lowpass operation, the first LPF in Fig. 18 could be combined with the upsampling section. The SDM will noise-shape this signal into an n-bit signal, where n typically varies between 3 [29] and 5 [30]. It is this signal, which is converted to the analog domain. Due to the noise shaping process, which is intrinsic in modern, high-end DA convertors, and is the sole basis for their very high performance, some additional high frequency noise extending to frequency regimes well above 1 MHz is introduced. This noise is usually removed by an analog low pass filter of first or second order. This filtering is most often passive, and can thus be performed with exceptionally low distortion and intermodulation.

In most one-bit audio players, some additional filtering is provided, to further reduce the amount of HF noise and signal even further to levels well below -30 dB. This filtering protects tweeters against full scale HF signals, which potentially could occur with the wide-band signal capabilities of one-bit audio. It is important to remark, that the HF signal levels at which these additional filters need to operate are quite low due to the digital pre-filtering (which removed a very substantial amount of HF signal causing the total signal power to be substantially less than 1); hence, the linearity of the filters can be quite high and the filtering operation is performed without additional intermodulation products.

5.3 Signal processing of one-bit audio

A crucial point in any audio chain is signal processing, ranging from simple volume adjustments to complex equalizations. It is immediately apparent, that a direct translation of the 'PCM-way' of signal processing does not exist in one-bit audio. For example, if a one-bit audio signal is volume-adjusted, with a gain $g = 0.123456$, the resulting output (the one-bit signal multiplied with g) is a multi-bit word. Hence, any signal processing for one-bit audio principally always consists of a cascade of the actual processing step, followed by a re-quantization.

To obtain a realizable system, a low pass filter before the requantizer is generally necessary. The reason for this is that the SDM which is used as a re-modulator, cannot cope with the high signal levels the one-bit audio presents. As virtually all of the power of these signals is above 100 kHz, a low pass filter reducing signals above this frequency is sufficient to remove

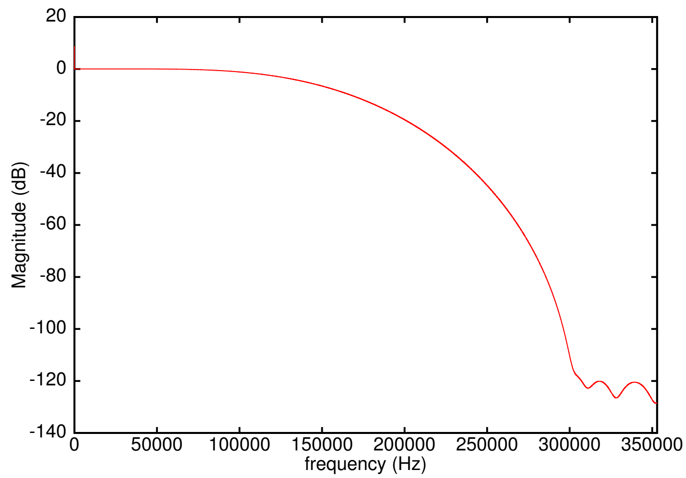

Figure 20: Transfer function of a filter which can be used to remove the HF of a one-bit audio signal, such that it can be input to a subsequent SDM.

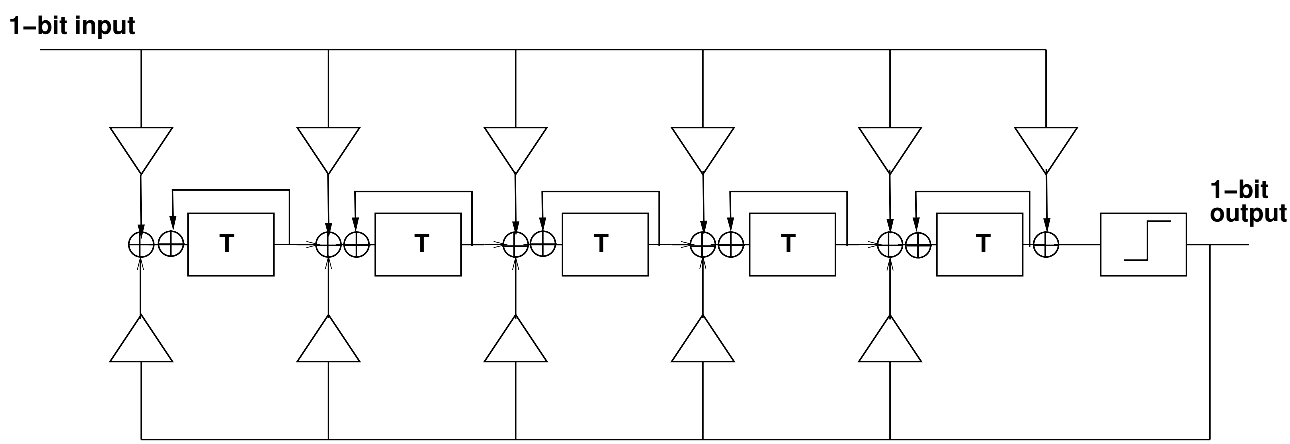

Figure 21: Contraction of IIR filter characteristic and SDM, giving a structure with one-bit input *and* one-bit output.

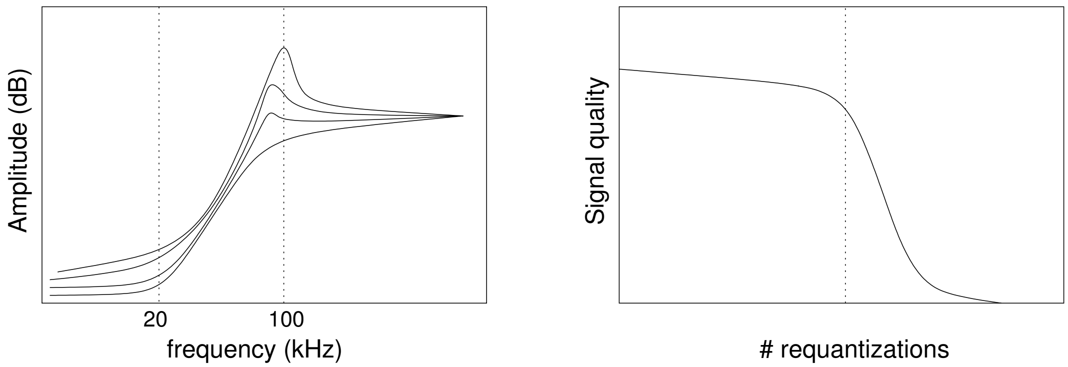

Figure 22: Schematic presentation of the effect of multiple quantizations for a SDM running at 64 f_s. Left: due to multiple requantisations, a build-up of HF noise occurs. The amplitude scale is arbitrary. Right: while for a limited number of requantizations the only adverse effect is a reduction in SNR, when the build-up of HF noise is too large, the SDM will start to clip leading to significant distortion.

enough power such that the re-modulator remains in stable operation. In this respect, the feed-forward and feedback structures have quite different behaviour. As shown in Sec. 3.2, the feed-forward structure has little suppression of the input signal over the whole band (up to Nyquist), and sometimes even a gain just at the corner frequency of the NTF filter characteristic. The feedback structure, on the contrary, has strong suppression of the input signal from the fore mentioned corner frequency (see also Fig. 8). Hence, a 'feed-forward' SDM will need more severe filtering of its input signal compared to a 'feedback' SDM in order to maintain stability. The response of a (64 taps) FIR filter which gives sufficient HF suppression to allow subsequent re-quantization, is shown in Fig. 20.

Because a FBSDM already contains input signal filtering, it is very tempting to contract some signal processing steps and the SDM re-modulator. This approach has been investigated in [32, 7, 33]. An example, where an IIR filter is contracted with a SDM, is shown in Fig. 21. This device shows many practical advantages, such as the absence of multi-bit multipliers, which at high sample rates is a major benefit. It is important to note, however, that such a device is *not different* from the cascade of signal processing/re-modulation, although the direct intermediate multi-bit path is absent.

While the work presented in [32, 7, 33] addresses several of the issues in one-bit signal processing, there is one further issue. Suppose, that a sequence of signal processing steps is necessary. If each of these steps is built according to Fig. 21, the total signal path will contain multiple requantizations. Even though the low pass filtering may have succeeded in removal of 99% of all HF energy, we certainly do not want to filter the signal in the region below 100 kHz, and still some energy resides in the band 20-100 kHz if the one-bit sample rate is 2.8 MHz. As a result of this, build-up of noise will occur. While this is not principally different from LPCM, where at each processing step dithering has to be applied, thus also resulting in a build-up of noise, typically the amount of noise that is left after filtering the one-bit signal is much larger. This effect is illustrated in the left of Fig. 22, where schematically the effect of multiple requantizations is displayed. This figure can be explained as follows. If we have a one-bit signal, its noise starts to rise above 20-30 kHz, and reaches an almost flat level at above 90 kHz. If, in a subsequent re-quantization, 80-100 kHz bandwidth is maintained, which is the goal in high end applications, the signal is low pass-filtered at a frequency of about the same value. If this signal is fed to a next SDM, its output signal will contain both its own quantization noise, as well as the quantization noise that has been introduced. If this cascade is repeated, it is easy to see why there will be a build-up of HF noise in the area of about 80-90 kHz. Eventually, this signal will be large enough to drive the SDM into its clippers, thus reducing the signal quality. This effect is shown in the right of Fig. 22; as the number of requantizations increases, the signal quality drops slowly due to the increase in noise floor. At the moment that the HF noise is large enough to activate the clippers, the signal quality drops rapidly. This effect has been studied in more detail in [34], and when careful signal processing is performed, hundreds of requantizations can be performed before the signal degradation shown in Fig. 22 occurs.

Still, it is a most pragmatic approach to perform all signal processing in a multi-bit domain, such that build-up of noise is limited. The conversion to $64f_s$ one-bit signals should be made only after the final signal processing step. However, this reasoning hinges on the fact that the quantization noise floor in the one-bit audio signal is significant in the higher part of the band of interest. Indeed, when the sample rate of the one-bit audio system equals 2.8 MHz, as is the case with many concurrent systems, this is true. However, for rates which are 5.6 MHz or higher, this reasoning is not true anymore, and subsequent requantizations do *not* add substantial noise. Likewise, the analysis has focussed completely on 'classical' SDMs, but now a much wider range of SDM techniques has become available which are able to push the quantization noise to much higher frequencies than classical SDMs are capable of due to their increased stability (see Sec. 6). It is, therefore, undecided yet which type of processing is eventually to be preferred.

6 Recent developments in one-bit audio and signal processing

While Sigma Delta Modulation has been around for a long time, it has seen a recent revival of interest due to the proposal to use one-bit audio as a consumer delivery format. Recent research has focussed on improvement of the characteristics of SDMs, most notably the stability of SDMs and the accuracy with which digital signals can be represented. In the next section we will detail some of these new developments.

6.1 Controlling the noise shaping characteristics

6.1.1 Pre-Correction SDMs

Sigma Delta Pre-Correction (SDPC) has been introduced in [14]. The idea is based on the fact that, while distortion is a non-linear phenomenon, if can be approximately corrected for when employing linear techniques. While standard (even undithered!) SDM distortion ratios are typically below -140 dB, the method actually serves an aesthetic purpose only, because any distortion introduced by always present (analog) equipment will be much more severe. The method does not address increased stability of the SDM, and only improves the accuracy with which signals can be represented. However, dithering the SDM (see Sec. 4.2.2) as an alternative linearization technique reduces the maximum stable input of a SDM. SDPC has no stability penalty, while it is very succesful in linearization.

Within SDPC (see Fig. 23), a SDM is modelled as a non-linear element $\Sigma\Delta$, with a transfer characteristic written as:

$$\Sigma\Delta(u) = u + \alpha_2 u^2 + \alpha_3 u^3 + \dots \tag{16}$$

Then, an approximation $s'(u)$ to a signal $s(u)$ is created, where $s(u)$ is defined according to:

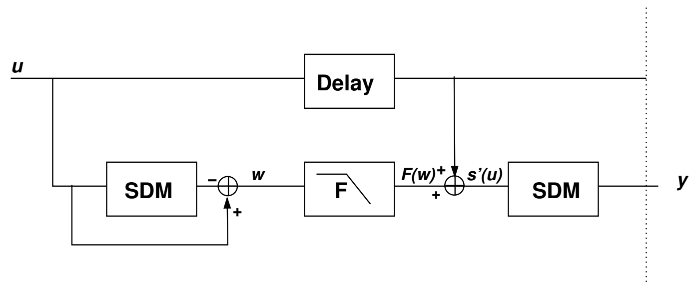

Figure 23: Basic Sigma Delta Pre-Correction (SDPC) structure.

$$s(u) = u - \alpha_2 u^2 - \alpha_3 u^3 - \ldots \tag{17}$$

Thus, $s'(u)$ differs from $s(u)$ in that it contains some residual quantization noise. If signal $s(u)$ is fed to an identical SDM, the resulting output signal $\Sigma\Delta(s(u))$ is be given by:

$$\Sigma\Delta(s(u)) = u - 2\alpha_2^2 u^3 + \mathcal{O}(u^4) \tag{18}$$

The use of $s'(u)$ in Eq. (18) will give the same result, but also adds some quantization noise. In other words, the second harmonic distortion component has been completely removed, and the third harmonic component has been substantially reduced (note, that for the low distortions we are dealing with, $\alpha_i \ll 1$).

To gain some insight in the performance of SDPC, it has been applied to the third order SDM also employed in Sec. 4.2. The spectrum of the resulting signal y is displayed in Fig. 24 in the range 0-100 kHz. The huge suppression of the distortion components is clearly visible. Typically, the distortion has been reduced by about 20 dB. As always, there is a price to pay for this improvement in THD, which in this case is an increase in the noise floor by 3 dB. This is clear from inspection of Fig. 24, when one realizes that the corrected spectrum has been obtained using twice as many coherent averages which lowers the noise floor by 3 dB, and that the noise floor is identical to the noise floor of the uncorrected spectrum. This also corroborates the fact that this is white noise indeed; if it were correlated, it would have resulted in a more than 3 dB increase. The origin of the increase of the noise floor is the fact that the signal $s'(u)$ still contains the quantization noise present in the low frequency range; the second SDM in the cascade adds its own quantization noise to it. Though not visible in Fig. 24, the high frequency signals above 1 MHz are completely unchanged using the new topology, which is as expected on basis of the absence of correction components in the signal $s'(u)$.

It proves possible to apply this technique also to slightly dithered and high order SDMs; in this case, any possible non-ideality occurs at levels below -220 dB, which are unachievable in real life. As well, it is a better result than probably could have been obtained by dithering the SDM (see Sec. 4.2.2). Hence, also due to the fact that SDPC poses no penalty with respect to stability, it also is more effective (at least in the band 0-100 kHz) compared to dithering.

6.1.2 Parametrically controlled noise shaping

Parametrically controlled noise shaping is introduced in [20], following the observation that in 'classical' SDM design (see Sec. 3.2) the addition of more than 7 integrators does not result in a noticable increase in performance (when the sample rate is approximately 2.8 MHz).

To remedy this situation, the structure in Fig. 25 has been introduced. In the upper part of Fig. 25, a standard fifth order SDM can be recognised, whereas the lower part implements a

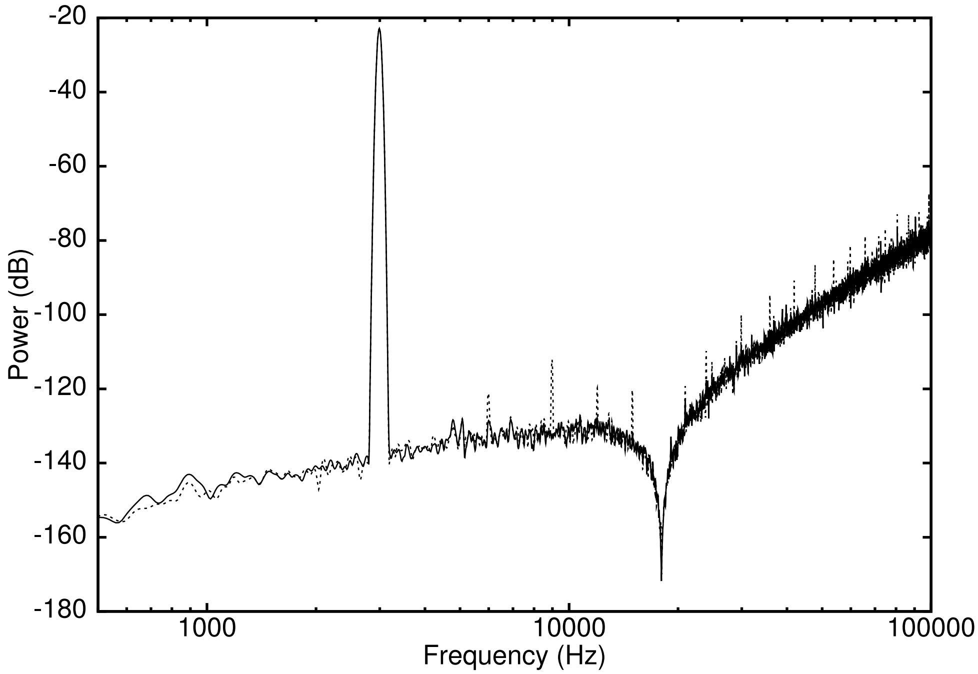

Figure 24: Spectra of the original SDM (dashed), and its implementation according to Fig. 23 (solid line). The spectrum of the original SDM has been obtained using 4 coherent averages and 10 power averages; the other using 8 coherent averages and 10 power averages. The fact that the noise floors of the spectra coincide precisely illustrates the 3 dB loss in SNR due to SDPC.

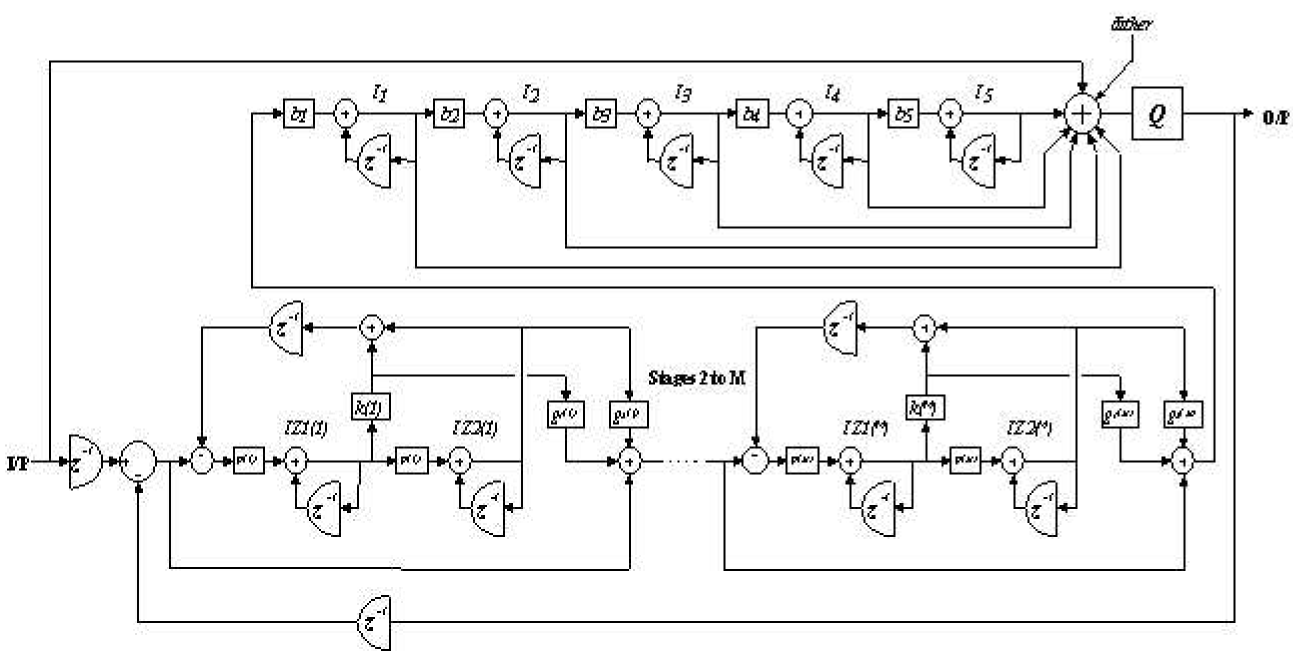

Figure 25: Schematic representation of a parametrically controlled noiseshaper (after [20]).

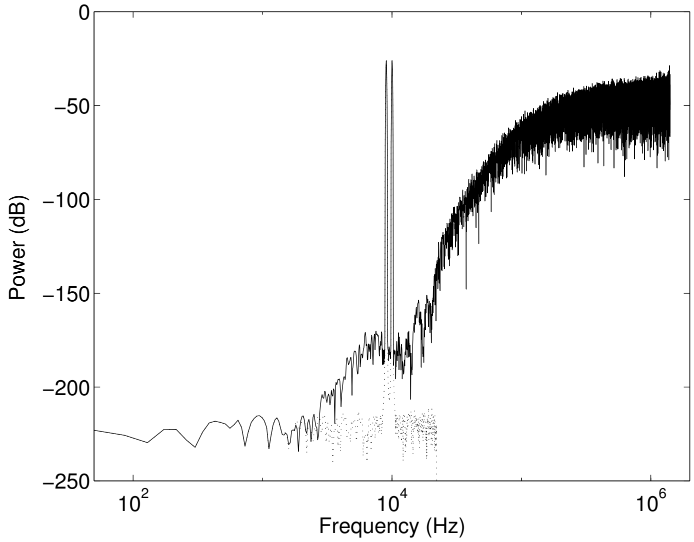

Figure 26: Example taken from [20]; a parametric SDM, with fed with a 9 and 10 kHz input sine wave. The lower curve (dashes) represents the 32 bit input to the SDM. Note the absence of a 1 kHz intermodulation product.

parametric equalizer. Such a system allows significant freedom in the choice of the final NTF, which can be optimised. For example, specific attention can be paid to the suppression of low frequency quantization noise, which is exemplified in [20]. While this structure is highly flexible with respect to the shape of the NTF, it is also extremely efficient in linearizing the SDM. This is illustrated in Fig. 26. In this figure, a power spectrum of a parametric SDM is displayed (taken from [20]), which is fed with a 9 and 10 kHz input sine wave. Also shown is the noise level that corresponds to 32-bit, TPDF (of width 2 LSB) dithered LPCM. Any sign of non-linearity, which whould expose itself as intermodulation products around 1 kHz, is absent. Further, with the parametric SDM a resolution is achieved that exceeds its 32 bit PCM equivalent by far below 2 kHz, which is believed to be the area where the ear is most sensitive. Because the trace in Fig. 26 has been generated with a 32 bit input to the SDM, this feature is masked by the resolution of the input and the final trace shows a resolution not better than its 32 bit equivalent. Over the band 0-20 kHz, the parametric SDM displays a resolution equivalent to 24 bits.
As with classical SDMs, the final performance is limited by the stability of the SDM; while parametric SDM design addresses the question of how optimal, spuriae free, noise shaping can be realised, it does not deal with the issue of stability. In the next two sections, new developments are highlighted which do address this important question.

6.2 Controlling stability

All designs that try to address the issue of improvement of stability, essentially return to the original question of one-bit coders in Sec. 2.1: how to minimize the error ϵ in Fig. 1. Classical SDMs try to minimize the instantaneous error. That this can be far from optimal, is illustrated by the instability phenomenon itself: even though the SDM continues to minimize the instantaneous error, the output signal has no resemblance any more to the input signal. Intuitively, it is clear that solutions with a better (integrated) error metric must exist. This points in the direction, that improvements should be sought in minimizing a metric of the error, which has a finite extent over time. This is what stability improved SDM designs do,

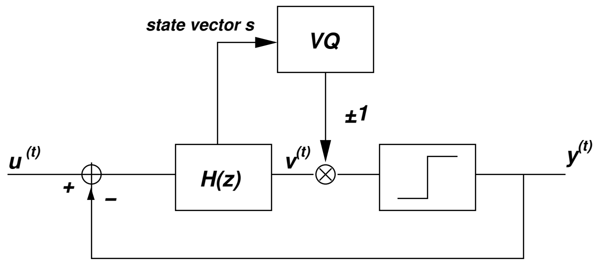

Figure 27: The concept of a vector quantizer VQ, embedded in a SDM.

and the difference between the different designs lies solely in the fact *how* this error is defined, and how it is attempted to be minimized.
A theoretically appealing concept is depicted in Fig. 27, and is based on a vector quantizer that employs knowledge about all state variables in the filter instead of only the filter output value.
The idea has been introduced in 1993 by Risbo [35, 36]. Obviously, the secret is in the algorithm hidden in the box labelled VQ, that decides upon the sign. It is, however, not obvious what this algorithm should look like; in [35] a neural network algorithm is proposed. The vector quantizer concept has not yet been exploited to a great extent, most probably because of the difficult task of revealing the optimal dependence on the individual state variables of the quantizer output.

6.2.1 Step-back SDMs

Historically, step-back SDMs were the first that tried to address the problem of stability. Several algorithms exist, and often borrow from experience that has been obtained in the design of digital Class-D amplifiers. As early as 1993, a concept called 'step-back' has been introduced, in conjunction with the earlier mentioned vector quantizer concept [35, 36]. In this idea, the absolute value $|v|$ of the filter output v is monitored, which functions as an estimate for the error ϵ (in fact, it *equals* the error ϵ when the low pass filter in Fig. 1 is chosen to equal the loopfilter). Certain pre-set bounds are given for $|v|$, and whenever $|v|$ exceeds such a value, the step-back algorithm is activated. The purpose of the step-back algorithm is to find a sign inversion of an output bit (or 'bit-flip') such that the total error including this bit flip, is again within its bounds. This algorithm has proven to be quite efficient in increasing the SDM stability, and, at the same time, improving the linearity in the signal band. Its main drawback is in the apparent arbitrary choice of the boundaries for $|v|$. When these boundaries are chosen too tight, chances are that no one-bit code exists that represents the input. On the other hand, if the boundaries are chosen too unrestrictive, the stability improvement may prove to be only marginal.
Another useful and easily implementable concept has been coined 'bit-flipping with look-ahead', introduced in [37] and more extensively discussed in [38]. The basic idea is that 2 (identical) SDMs run in parallel, the second fed with the same input as the first but delayed with one sample. In this case, the first SDM functions as a 'look-ahead' for the second; when signs of instability occur, the output bit of the second SDM can be changed in the hope that this will remove the instability.
A method that is akin to the method presented in [35] is the 'variable state-step-back pseudo-Trellis SDM', presented in [39]. In this approach, a set of heuristic rules are defined for the decision to step back in history and create another decision. This approach proves to be extremely succesful in stabilizing SDMs. With the help of this algorithm a highly aggres-

sive noise shaping characteristic could be obtained which resulted in a SDM displaying the equivalent of 32 bits resolution over 30 kHz bandwidth.
While from a conceptual point of view these step-back SDMs are quite appealing, a possible drawback might be that it is very difficult to implement these designs in real-time hardware. This problem is alleviated by the designs in the next section.

6.2.2 Trellis type SDMs

A completely new view upon ways to minimize the time-integral of the error ϵ was presented in 2002 by Kato in a seminal paper [40]. In the paper the connection was made between the Trellis algorithm, known from error correction theory, and one-bit noise shaping. The basic idea of the application of the Trellis algorithm to one-bit coding is to minimize the time integral of the loop filter output $v(t)$. This is in contrast with the approach sketched in, *e.g.* [35] and [39] where the idea is to bound the loop filter output $v(t)$. Assume that up to clock cycle t=t_0 the optimal output sequence of bits is known. The output $y(t_0 + 1)$ can be either -1 or +1, which will result in the instantaneous frequency weighted error $v_{-1}(t_0 + 1)$ and $v_{+1}(t_0 + 1)$, respectively. One time instant later, again an output of either -1 or +1 is possible, resulting in 4 different possibilities (*paths*) for the 2 output bits. Every path has its own associated cost $C_{\omega_N}(t)$ (called pathmetric [40]), which can, for example be defined as the sum of the squared frequency weighted error values:

$$C_{\omega_N}(t) = \sum_{\tau=0}^{t} [v(\tau)]^2 \tag{19}$$

with ω_N a sequence of N output bits.
Advancing time once more, the number of possibilities doubles again and becomes 8, and so on. The full Trellis algorithm limits the number of paths by selecting, and continuing with, only half of the newly generated paths. In a full Trellis system of order N, 2^N possible solutions are investigated at every moment in time. Advancing time by 1 results in 2^{N+1} candidates, of which 2^N are selected. The 2^N solutions under investigation are forced to be all different in the newest N bits, in order to maintain the trellis structure.

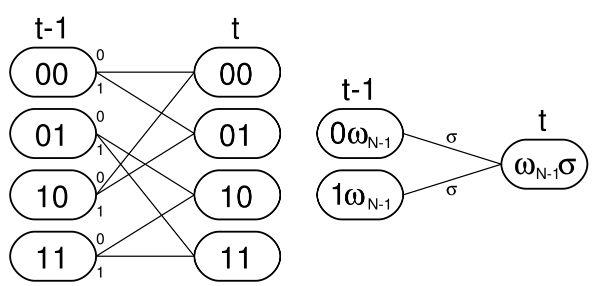

Figure 28: Origination of new candidates (clock cycle t) from old candidates (clock cycle $t-1$). Complete state diagram for $2^N = 4$ candidates (left), and the general case (right). For clarity, the signal level -1 is represented by the symbol “0” inside all figures.

Figure 28 shows a Trellis with order N=2. The figure shows the 4 combinations of 2 bits that are possible for clock cycle *t-1*. If a ’0’ is concatenated to the sequence ’00’, we obtain ’000’. Adding a ’1’ results in ’001’. Reducing the length of the 2 possible sequences to 2 again, results in ’00’ and ’01’, respectively. It is clear that starting with ’10’ would also result in ’00’ and ’01’, therefore a choice has to be made and 1 path has to be selected. The selection

criterion is the total cost of the path; it is assumed that the path with the lower cost will turn out to be the best solution of the two.
From Trellis and Viterbi theory [41], it turns out that if the system runs long enough, paths converge. This means, that independently of which path is examined, they all originate from the same 'mother sequence' of bits. This mother sequence is then the sequence of bits which is the Trellis approximation to the optimal sequence. Figure 29 illustrates the convergence.

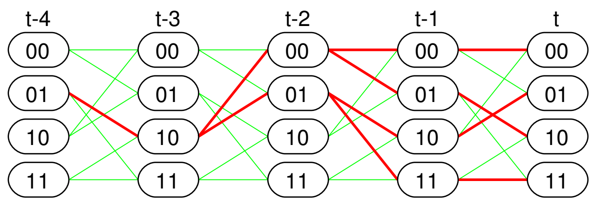

Figure 29: Convergence of paths: the bold lines show the origination of the four candidates. The different candidates terminate with different output symbols, but in history ($t \to -\infty$) the output sequences converge to a single solution.

In practice, an output latency up to several thousand bits, depending on the Trellis order, is enough to find the convergence point.
As shown in [40, 42], application of the Trellis converter increases the maximum stable input range and improves SNR. Simulations have shown, that in order to significantly gain performance, the Trellis order needs to be large (say, > 8). Since the workload doubles for every increment of the Trellis order, orders higher than 5 or 6 can hardly be used (a 6th order system contains $2^6 = 64$ SDMs, together with bookkeeping overhead, results in an about 100 times more expensive system as a normal SDM). In [42, 43] an efficient Trellis SDM was introduced, which makes it possible to reach the performance of a high order full Trellis converter at only a fraction of the cost. The idea was conceived after the observation that in a full Trellis algorithm, only a fraction of all paths that are calculated return in the final solution of the algorithm, which is illustrated in Fig. 30.

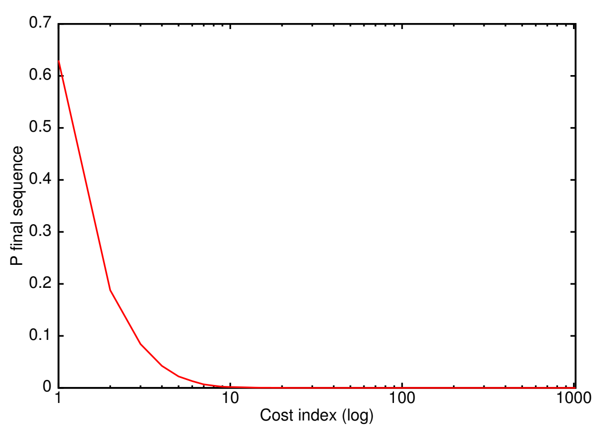

Figure 30: The probability for a candidate with a certain cost index to become the optimum solution. The cost index ranks the candidates on increasing cost function.

The 'efficient Trellis' algorithm thus only tracks those paths which have a high probability of proving to be optimal, thus allowing for a dramatic increase in computational efficiency. Figure 31 shows the relation between the number of Trellis paths and required CPU time.

Clearly, the CPU requirements increase only linear with the Trellis depth, instead of exponentially. In view of this, practical application of the efficient Trellis algorithm has come within reach.

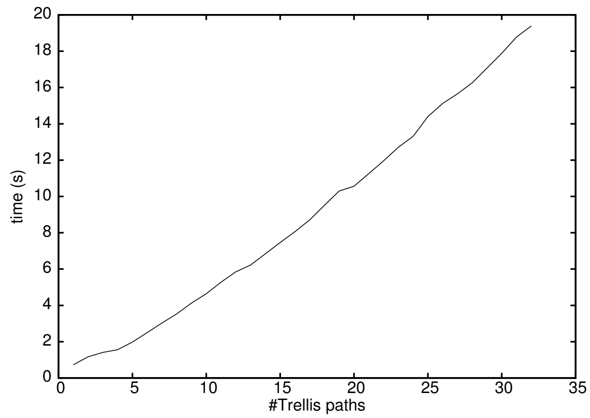

Figure 31: Execution time for different data sets as function of the number of Trellis paths.

7 Summary and Conclusions

The concept of one-bit audio has been described. It is characterized by noise shaping and large oversampling ratios, which is in line with the general trend observed in high-end audio analogue-to-digital and digital-to-analogue converters. As a result of the goal to maintain an audio quality as high a possible in consumer audio delivery, one-bit audio has been conceived as a means to realize this goal. A historical element in this development is the noise shaping device, or, more specifically, the Sigma Delta Modulator (SDM) which has provided a means to generate a high quality one-bit audio representation.

We have reviewed some simple SDM design techniques which show that the design of a functional SDM has become a standard engineering practice. We have further analyzed the behaviour of the SDM, or noise shaping devices in general. While, theoretically, these devices are not perfectable in a mathematical sense as PCM is, it has been shown that in practice any non-linearity is of a harmlessly low level. Also, linearization techniques have been reviewed which linearize a first-order SDM, seeding the thought that SDMs *are* mathematically perfectable, only different from the way PCM is perfectable.

Several signal processing aspects have been discussed, which showed that all signal processing required for disc production is highly feasible, while maintaining the high audio quality offered by one-bit audio. Along the same line, replay of one-bit audio is discussed. However, signal processing requires additional headroom (as well as PCM does) for which either an increase of sample rate, or an increase in the number of bits is required. Both of these are easily realizable.

We have shown that one-bit audio offers a great deal of flexibility, as it does not rely on specific predefined coding paradigms. As a result, the one-bit code can be tailored to any specific demand, offering ample freedom to satisfy any user's need. Several examples of newly developed coding schemes are given. In particular, the so-called Trellis-based coding techniques appear to be both highly flexible and of very high quality. We expect that these developments are only the first to inspire the audio community, and hope that many new and exciting developments will follow, with the ultimate goal of reconstructing a perfect sound field to the benefit of every home environment.

Acknowledgements

The authors would like to thank their colleques from Philips and Sony for many fruitful discussions and intense collaborations. Also, the many discussions with Professors M.O.H. Hawksford, S.P. Lipshitz, J. D. Reiss and J. Vanderkooy have greatly contributed to the increased understanding of one-bit coding.

References

[1] F de Jager. Delta modulation - a method of PCM transmission using the one unit code. *Philips Res. Rep.*, 7:442–466, 1952.

[2] H. Inose, Y. Yasuda, and J Murakami. A telemetering system by code modulation - $\Delta - \Sigma$ modulation. *IRE Trans. Space Electron. Telemetry*, SET-8:204–209, 1962.

[3] S.R. Norsworthy, R. Schreier, and G.C. Temes. *Delta-Sigma Converters, Theory, Design and Simulation*. IEEE Press, New York, 1997.

[4] B.H. Leung and S Sutarja. 'multi-bit $\Delta - \Sigma$ A/D converter incorporating a novel class of dynamic element matching. *IEEE Trans. Circuits Syst. II*, 39:35–51, 1992.

[5] R.T. Baird and T.S. Fiez. Improved $\Delta\Sigma$ DAC linearity using data weighted averaging. *Proc. IEEE Int. Symp. Circuits Syst.*, 1:13–16, 1995.

[6] H. Inose and Y Yasuda. A unity bit coding method by negative feedback. *Proc. IEEE*, 51:1524–1535, 1963.

[7] J. A. S. Angus and N. M Casey. Filtering $\Sigma\Delta$ audio signals directly. In *Proceedings of the AES 102'nd convention*, 1998. Preprint 4445, 1998 march 22-25 Munich.

[8] R.W. Adams, P.F. Ferguson, A. Ganesan, S. Vincelette, A. Volpe, and A. Libert. Theory and practical implementation of a fifth order sigma-delta A/D converter. *J. Audio Eng. Soc.*, 39:515–528, 1991.

[9] E. Stikvoort. Higher order 1-bit coder for audio applications. In *Proceedings of the AES 84th convention*, 1988. preprint 2583, March 1988, Paris.

[10] S.H. Ardalan and J.J. Paulos. An analysis of nonlinear behavior in delta-sigma modulators. *IEEE Transactions on Circuits and Systems*, CAS-34:593–603, 1987.

[11] J. van Engelen and van de Plassche. *Bandpass Sigma Delta Modulators : Stability Analysis, Performance and Design Aspects*. Kluwer Academic Publishers, New York, 1999.

[12] S. Wolfram. *The Mathematica Book*. Wolfram Media/Cambridge University Press, Cambridge, 4 edition, 1999.

[13] S.P. Lipshitz, R.A. Wannamaker, and J. Vanderkooy. Quantization and dither: a theoretical survey. *J. Audio Eng. Soc.*, 40:355–375, 1992.

[14] D. Reefman and E. Janssen. Enhanced sigma delta structures for Super Audio CD application. In *Proceedings of the AES 112th convention*, 2002. preprint 5616, 2002 May 10-13 Munich.

[15] S.P. Lipshitz and J. Vanderkooy. Towards a better understanding of 1-bit sigma-delta modulators. In *Proceedings of the AES 110th convention*, 2001. preprint 5398, 2001 Amsterdam.

[16] M.A. Gerzon and P.G. Craven. Optimal noise shaping and dither of digital signals. In *Proceedings of the AES 87th convention*, 1989. preprint 2822, 1989.

[17] S.R. Norsworthy. Effective dithering of delta-sigma modulators. In *IEEE Proc. ISCAS'92*, 1992. Vol.3, pp 1304–1307.

[18] S.R. Norsworthy and D.A. Rich. Idle channel tones and dithering in delta-sigma modulators. In *Proceedings of the AES 95th convention*, 1993. preprint 3711, 1993 October New York.

[19] D. Reefman and E. Janssen. DC analysis of high order sigma delta modulators. In *Proceedings of the AES 113th convention*, 2002. preprint 5693 2002 October 5-8 Los Angeles.

[20] M.O.J. Hawksford. Time-quantized frequency modulation with time dispersive codes for the generation of sigma-delta modulation. In *Proceedings of the AES 112th convention*, 2002. Preprint 5618, 2002 May 10-13 Munich.

[21] A.V. Oppenheim and A.W. Shafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[22] S. Hein and A. Zakhor. *Sigma Delta Modulators: nonlinear decoding algorithms and stability analysis*. Kluwer Academic Publishers, New York, 1993.

[23] D. Hyun and G. Fischer. Limit cycles and pattern noise in single-stage single-bit delta-sigma modulators. *IEEE Trans. Circuits and Systems I*, 49:646–656, 2002.

[24] J.D. Reiss and M.B. Sandler. They exist: Limit cycles in high order sigma delta modulators. In *Proceedings of the AES 114th convention*, 2003. preprint 5832, March 22-25 2003, Amsterdam.

[25] D. Reefman, J.D. Reiss, E. Janssen, and M.B. Sandler. Stability analysis of limit cycles in high order sigma delta modulators. In *Proceedings of the AES 115th convention*, 2003. preprint 5936 2003 October 10-13 New York.

[26] D. Reefman, J.D. Reiss, E. Janssen, and M.B. Sandler. Stability analysis of limit cycles in high order sigma delta modulators. *to be published*, ??:??, 2004.

[27] D. Reefman and P.A.C.M. Nuijten. Editing and switching in 1-bit audio streams. In *Proceedings of the AES 110th convention*, 2001. preprint 5399, 2001 May 12-15 Amsterdam.

[28] D. Reefman and E. Janssen. One-bit Audio: An Overview. *J. Audio Eng. Soc.*, ??:??, 2004.

[29] B. Adams, K. Nguyen, and K. Sweetland. A 116 db SNR multi-bit noise shaping DAC with 192 kHz sample rate. In *Proceedings of the 106th AES convention*, 1999. preprint 4963, Munich (1999).

[30] S. Nakao, H. Terasaw, F. Aoyagi, N. Terada, and T. Hamasaki. A 117db D-range current-mode multi-bit audio DAC for PCM and DSD audio playback. In *Proceedings of the 109th AES convention*, 2000. preprint 5190, Los Angeles (2000).

[31] D. Reefman and P.A.C.M. Nuijten. Why direct stream digital is the best choice as a digital format. In *Proceedings of the AES 110th convention*, 2001. preprint 5396, 2001 May 12-15 Amsterdam.

[32] N. M. Casey and J. A. S. Angus. One bit digital processing of audio signals. In *Proceedings of the AES 95'th convention*, 1998. Preprint 3717, 1993 october 7-10 New York.

[33] J. A. S. Angus and S Draper. An improved method for directly filtering $\Sigma\Delta$ audio signals. In *Proceedings of the AES 104'th convention*, 1998. Preprint 4737, 1998 may 16-19 Amsterdam.

[34] P. Eastty, C. Sleight, and P. Thorpe. Research on cascadable filtering, equalisation, gain control and mixing of 1-bit signals for professional audio applications. In *Proceedings of the AES 102'nd convention*, 1998. Preprint 4444, 1998 march 22-25 Munich.

[35] L. Risbo. Improved stability and performance from $\Sigma\Delta$ modulators using one-bit vector quantization. *IEEE Proc. ISCAS'93*, pages 1361–1364, 1993.

[36] L. Risbo. *Sigma-delta modulators: stability analysis and optimisation*. Technical university of Denmark, 1994.

[37] A. J. Magrath and M.B. Sandler. A use of sigma-delta modulation in power digital-to-analogue conversion. *Int. J. of Circuit theory and applications*, 25:439–455, 1997.

[38] A. J. Magrath. *Algorithms and Architectures for High resolution Sigma-Delta converters*. University of London, 1996.

[39] M.O.J. Hawksford. Parametrically controlled noise shaping in variable state-step-back pseudo-trellis SDM. In *Proceedings of the AES 115th convention*, 2003. Preprint 5877, 2003 October 10-13 New York.

[40] H. Kato. Trellis noise-shaping convertors and 1-bit digital audio. In *Proceedings of the AES 112th convention*, 2002. Preprint 5615, 2002 May 10-13 Munich.

[41] A.J. Viterbi and J.K. Omura. *Principles of Digital Communications and Coding*. McGraw-Hill, New York, 1979.

[42] P. Harpe, D. Reefman, and E. Janssen. Efficient trellis-type sigma delta modulator. In *Proceedings of the AES 114th convention*, 2003. Preprint 5845, 2003 March 22-25 Amsterdam.

[43] E. Janssen and D. Reefman. Advances in trellis based SDM structures. In *Proceedings of the AES 115th convention*, 2003. Preprint 5993, 2003 October 10-13 New York.

A Example Design of a SDM

As an example, we will design a fourth order SDM, with a NTF according to a Butterworth high-pass filter design, cut-off frequency 150 kHz, as discussed in Sec. 3.2. Because the SDM needs to be realizable, the total loop needs to embody at least a single delay, *i.e.*, the term with z^0 in the STF needs to be zero. This corresponds with the requirement that the high pass filter should have 1 as its first value of the impulse response. This can be accomplished by multiplying the high pass filter with a certain coefficient (larger than 0), resulting in a HF gain which is larger than 1. With the above in mind, we obtain for the NTF:

$$NTF(z) = \frac{+1.00z^{-0} - 4.00z^{-1} + 6.00z^{-2} - 4.00z^{-3} + 1.00z^{-4}}{+1.00z^{-0} - 3.13z^{-1} + 3.75z^{-2} - 2.03z^{-3} + 0.42z^{-4}} \tag{20}$$

This results in the following coefficients in the feed-forward structure:

$$\begin{aligned} c_1 &= 0.8707115357 \\ c_2 &= 0.3594322506 \\ c_3 &= 0.0811807847 \\ c_4 &= 0.0083240406 \end{aligned} \tag{21}$$

For the feed-forward structure, the STF is now given by:

$$STF(z) = \frac{+0.00z^{-0} + 0.87z^{-1} - 2.25z^{-2} + 1.97z^{-3} - 0.58z^{-4}}{+1.00z^{-0} - 3.13z^{-1} + 3.75z^{-2} - 2.03z^{-3} + 0.42z^{-4}} \tag{22}$$

For the feedback structure, the STF is given by:

$$STF(z) = \frac{z^{-4}}{+1.00z^{-0} - 3.13z^{-1} + 3.75z^{-2} - 2.03z^{-3} + 0.42z^{-4}} \tag{23}$$

B SDM-code

In this appendix, we provide the C-like pseudo code for the SDM discussed in Sec. 4.1. The code simulates 100000 clock cycles of the SDM, with a DC input of 0.1 .

```
/* Coefficients: */
c = {
    0.791882,
    0.304545,
    0.069930,
    0.009496,
    0.000607
    };
f = {
    0.000496,
    0.001789
    };

/* Initialization */
    s0 = s1 = s2 = s3 = s4 = 0;
    y = 1;

    N = 100000;

/* Main loop */
    for (i = 0; i < N; i++) {
        sum = c[0]*s0 + c[1]*s1 + c[2]*s2 + c[3]*s3 + c[4]*s4;
        if (sum >= 0)
            y = 1;
        else
            y = -1;

        x = 0.1;

        s4 = s4 + s3;
        s3 = s3 + s2 - f[1]*s4;
        s2 = s2 + s1;
        s1 = s1 + s0 - f[0]*s2;
        s0 = s0 + (x-y);
    }
```