One-Bit Audio: An Overview*

DERK REEFMAN, *AES Member* **AND ERWIN JANSSEN,** *AES Member*

Philips Research Laboratories, 5656 AA Eindhoven, The Netherlands

An overview of 1-bit audio processing is presented. Several characteristics of the sigma–delta modulator (SDM), currently the most often used device to generate 1-bit code, are discussed, as well as some simple design methodologies of SDMs. It is shown that 1-bit audio is capable of carrying very high-quality audio. The total audio production chain, from recording to replay, is displayed and its feasibility demonstrated. Finally, some recent developments in the field of 1-bit audio codecs are summarized, which show a further improvement over the already excellent audio characteristics of the SDM.

0 INTRODUCTION

In 1998, a 1-bit coding format was introduced as a successor to the Compact Disc (CD). Whereas the CD employs linear pulse-code modulation [(L)PCM] encoding, with 16-bit-wide words at a sample rate of 44.1 kilosamples per second to store the digital representation of the audio data, the new format stores a 1-bit representation of the audio at 2.8 megasamples per second, which is 64 times the CD data sampling rate. Obviously, this change to 1-bit audio has introduced the need for a change in signal processing.

While the application of 1-bit audio for audio storage and distribution is quite new, the underlying idea of employing a 1-bit coding scheme is not. Already in the early 1950s, the concept of 1-bit coding was proposed and implemented by de Jager [1]. The original idea by de Jager was that when transmitting a 1-bit code instead of a PCM code, the loss of 1 bit of information was not as detrimental as for the PCM code. In a 1-bit code, all bits carry equal weight and the loss of a single bit means a certain loss of accuracy. In PCM some bits are more significant than others, and the loss of the most significant bit (MSB) could lead to radically wrong results. While the application of a device, called delta modulator, invented by de Jager, first saw its applications in error reduction in communication applications, it soon appeared that the 1-bit code made the realization of a high-quality digital-to-analog (DAC)—(and, thus ADC)—relatively easy [2]. As a result of the appearance of the CD in the 1980s, 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 ADCs and DACs 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 1-bit format and the 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 bit-stream technology increased the performance of standard CD players dramatically in the 1990s. The typical sequence of digital audio generation would be the generation of 1-bit audio through a high quality ADC, followed by downconversion to 44.1~kHz/16 bit for storage on the CD, again followed by upconversion to 64 or $128 f_s$/1-bit in the CD player, after which it would be fed to a high-quality DAC. Nowadays, the general merits of 1-bit ADCs and DACs are widespread and many applications for frequencies much higher than typical audio bandwidths exist [3].

In the search of ultimate audio quality it seemed logical to introduce a format that would store this 1-bit output directly, instead of the “intermediate” CD format. In this way, all filtering and signal processing needed to convert to and from the 1-bit format is eliminated which, by definition, can only increase the sound quality. After the first experiments with 1-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 which 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

*Manuscript received 2003 November 4.

with the introduction of 1-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 multibit converter, had appeared and were feasible to implement for a limited number of bits (2 to 6). Because 1-bit converters are more sensitive to clock jitter, the "few-bit" converters took their place in the high-end audio market. To obtain a 1-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 1-bit audio. The fact that the datum is 1 bit instead of a few bits, 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 1-bit coding in relation to 1-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), an introduction to sigma delta modulation will be presented in Section 1. In Section 2, approximate modeling of SDMs will be used to present practical methods of SDM design, which also reveals some signal characteristics of SDMs. As linear modeling is far from perfect, however, a more extensive discussion about signal characteristics of SDMs follows in Section 3. In Section 4, the creation of 1-bit audio contents is discussed in the limited but important context of signal processing 1-bit audio. With the renewal of interest for 1-bit coding, several new developments have been published, and some interesting developments are discussed in Section 5. Finally, a summary and conclusions will be presented in Section 6.

1 Introduction to Sigma–Delta Modulation

Sigma-delta (SD) modulation has become a widespread name for a general class of devices that 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 that do have 1-bit output, with an occasional remark about other outputs. Also, while many of the remarks to be made hold equally well for ADCs, we will restrict the discussion to digital-to-digital converters. We will regard digital inputs and digital outputs, where the differences between input 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 a complete list of references detailing all progress that has been made since the conception of 1-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.

1.1 History and Fundamental Principle of SDM

The basic property of all 1-bit coders is that they try to obtain a sequence of -1's and $+1$'s such that, over a specified bandwidth, the output is an accurate representation of the input. This is depicted schematically in Fig. 1.

In Fig. 1 a 1-bit code is generated by a 1-bit coder. The input, which can have any number of bits n but runs at the same rate as the output, is subtracted, after an appropriate delay, from the output of the device. Subsequently an error measure ε is defined over a limited bandwidth. In the case under discussion, this is usually the audio band. This error measure can be the instantaneous error $\varepsilon(t)$, but it can also be an integral ε_{int}, integrated over a certain period of time. Obviously all 1-bit coders are designed such as to minimize some of these error measures in some way, which is basically the definition of a 1-bit coder. Historically, the so-called delta modulator [1] is the first device that 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 1-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 cycle $i + 1$, after a single clock cycle delay, indicated by the block labeled 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 $\varepsilon(t)$ as measured by a first-order low-pass filter. Note the term "instantaneous," as any future errors that will be produced are not taken into account.

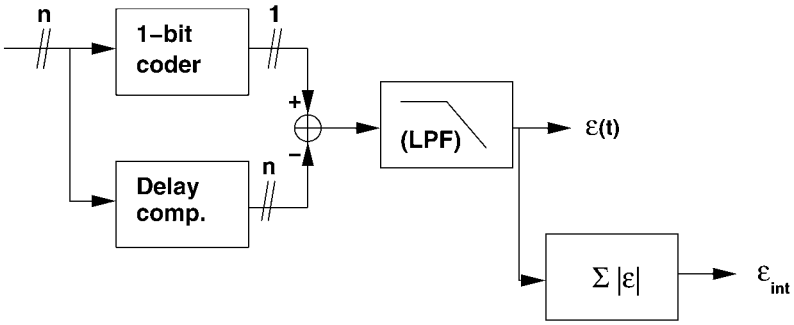


Fig. 1. Definition of error quantity ε which all coders try to minimize.

After the initial introduction of the delta modulator, various variations and improvements of this structure have been proposed in the period from 1960 to 1990 [3].

1.2 Basic Topologies of SDMs

Of all known topologies of 1-bit coders, some occur more frequently than others. In this section the most commonly occurring topologies will be discussed. These topologies are often also quite useful in connection with few-bit coders, and for that reason the quantizer devices are labeled by a Q (instead of a step function). However, the focus will be on 1-bit coders. The first topology is a generalization of the error feedback loop and is depicted in Fig. 3. This particular design is called a noise shaper. While in the original error feedback loop the quantizer error is only low-pass-filtered through a first-order filter, in a general noise shaper 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 (FFSDM). Its structure is depicted in Fig. 4. 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 $F(z) = H(z)/[H(z) + 1]$, and the input of the noise shaper is also pre-multiplied with $H(z)/[H(z) + 1]$, 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 up of four integrator sections, each consisting 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. Fig. 6 depicted the distributed feedback SDM (FBSDM). With the coefficients c_i and c'_i chosen properly the FFSDM and the 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 depends

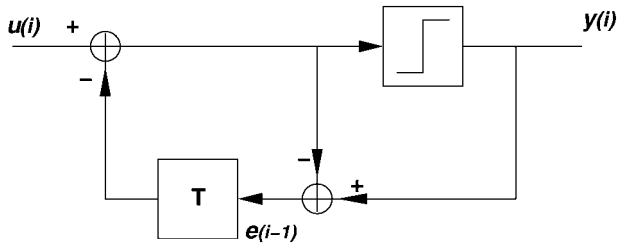


Fig. 2. Schematic of the error feedback loop [6].

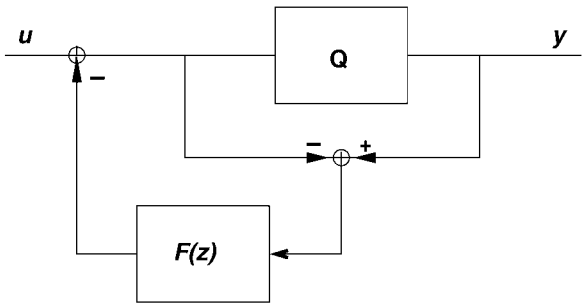


Fig. 3. Schematic of noise shaper.

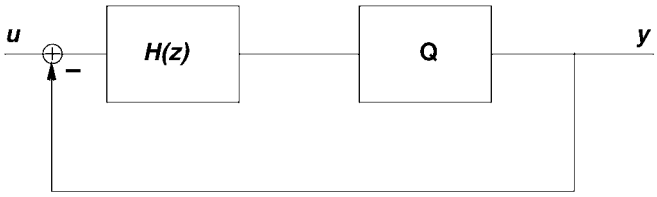


Fig. 4. Schematic of sigma–delta modulator.

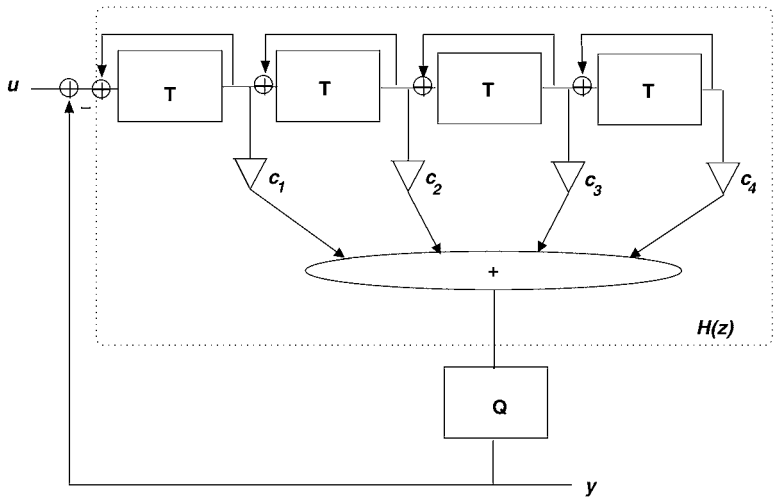


Fig. 5. Schematic of fourth-order feedforward sigma–delta modulator.

on the design of the complete system, and aspects such as system architecture and cost dictate what the optimum topology will be. From an analysis point of view, the 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.

2 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 1-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].

2.1 A Linear Model of the SDM

For applications in 1-bit audio, the quantizer Q in an SDM is a 1-bit quantizer, which outputs only values of $+1$ and -1. This is a highly nonlinear element, which renders the full analysis of an SDM difficult, if not impossible. Up to this moment, no complete mathematical theory exists that describes in full detail the behavior of an SDM. To gain some initial insight into the characteristics of the SDM, however, we will resort to a simple linear model and replace the highly nonlinear 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, related linearly to this SDM topology (see Section 1.2), we will restrict the discussion to SDMs only.

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

$$\mathrm{STF}(z) = \frac{cH(z)}{1 + cH(z)}, \qquad \mathrm{NTF}(z) = \frac{1}{1 + cH(z)}. \quad (1)$$

While models of various degrees of sophistication exist [9]–[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 an 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]–[11] are better suited, though still not flawless. Eqs. (1) show 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 1-bit audio applications, where the loop filter $H(z)$ typically is chosen as a low-pass filter with large low-frequency 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 nonlinearity included, differ significantly from those obtained by the use of the linearized model. Also other characteristics, discussed in Section 3, are not properly or not at all explained by the linearized model. It does give us some insight, though, in the way the quanti-

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

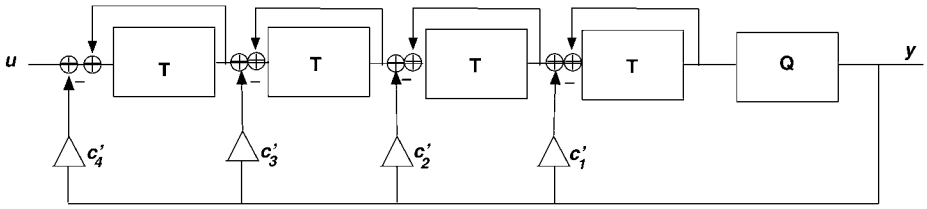


Fig. 6. Schematic of fourth-order distributed feedback sigma-delta modulator.

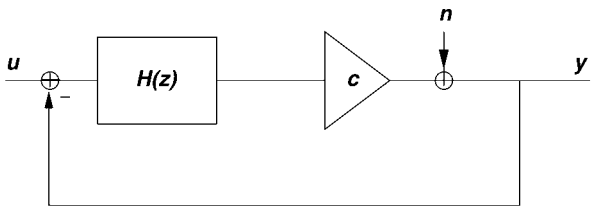


Fig. 7. Linearization of sigma–delta structure. Quantizer is replaced by a (signal-independent) gain and an additive noise source. Signal transfer function STF and noise transfer function NTF are defined by $Y = \mathrm{STF}U + \mathrm{NTF}N$, where Y is the Fourier transform of output y, U is the discrete Fourier transforms of input u and additive noise n.

zation noise is spectrally shaped and what filtering is applied to the input signal u.

2.2 Loop-Filter Design

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

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

and making use of the relation $\mathrm{NTF}(z) = 1/[1 + H(z)]$ we arrive at

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

which is to be recognized as a filter of the appearance $\mathrm{NTF}(z) = (1 - z^{-1})^n/P_n(z^{-1})$. 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 nth-order polynomial $P_n(z)$. Likewise, the signal transfer function can be computed as $\mathrm{STF}(z) = 1 - \mathrm{NTF}(z)$, resulting in

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

The approach that can now be followed is to design a high-pass filter for $\mathrm{NTF}(z)$, according to Butterworth or Chebyshev type 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 $1/\mathrm{NTF}(z)$ 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 dependence 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 $\mathrm{NTF}(z)$ takes the same shape as the noise transfer function of the feedforward structure discussed in the preceding. However, the signal transfer function is given by

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

which for low frequencies equals about 1 if the coefficients c_i are scaled as $c'_i = c_1/c_4$, $c'_2 = c_2/c_4$, $c'_3 = c_3/c_4$, $c'_4 = 1$. For higher frequencies the signal transfer function displays an almost third-order rolloff. This is in contrast to the feedforward topology, where the signal transfer function rolls off only very slightly (first order) for high frequencies. In Appendix A an example design of an SDM will be presented. In Fig. 8, the different signal transfer functions for a feedforward and a distributed feedback structure, with identical noise transfer function, have been calculated. The noise transfer functions are designed as fourth order Butterworth high-pass characteristics, with a cutoff frequency of 150 kHz. Clearly, the strong rolloff characteristic of the feedback structure can be observed. Interestingly the feedforward topology displays a strong peak in its transfer characteristic at the crossover 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 peak height in this case is about 6 dB.

This loop-filter design gives rise to an SDM with a maximum (peak) input of about -5 dB, that is, 0.57 with regard to the feedback signal from the quantizer. Above this input level the SDM turns unstable (see also Section 3.1). At an input of a sine with a peak amplitude of 0.5, the (unweighted) SNR in the band of 0–20 kHz is about 97 dB. In high-end audio applications often a SNR of better than 100 dB is desirable. However, 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

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

SNR is due to the fact that the noise floor 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 noise transfer function, and to use a Chebyshev type II high-pass filter for the noise transfer function design instead of a Butterworth characteristic. Another option is to create notches in the noise transfer function, which can easily be created in SDMs by the construction of resonator sections, as displayed in Fig. 9.

The construction in Fig. 9 is, in principle, applicable to a feedforward topology. For a feedback topology a similar arrangement with a feedback loop over two integrator sections is possible. 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 line 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

$$R_1(z) = \frac{h(z)}{1 = zfh(z)^2}, \qquad R_2(z) = h(z)R_1(z). \quad (6)$$

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

$$z_p = 1 - \frac{f}{2} \pm \frac{i}{2}\sqrt{4f - f^2}. \quad (7)$$

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:

$$f_{pole} = \arccos\left(1 - \frac{f}{2}\right) \quad (7)$$

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 fourth 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 Appendix A. Choosing the poles to move from

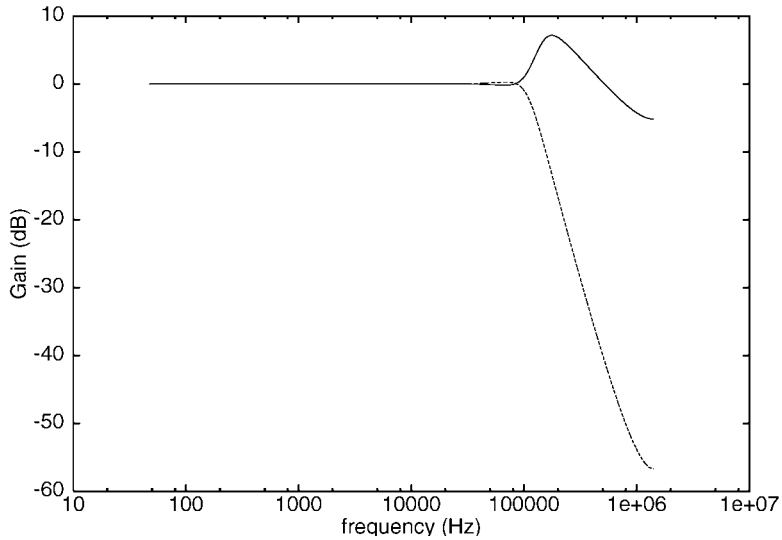


Fig. 8. Signal transfer functions for feedforward topology (——) and distributed feedback topology (– – –) with identical noise transfer functions.

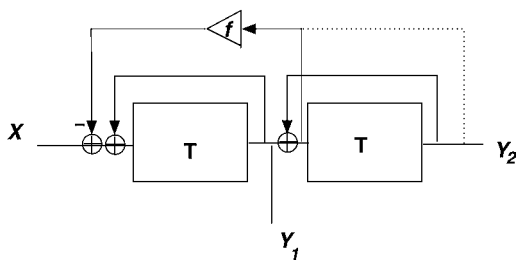


Fig. 9. Cascade of two integrator sections in an SDM, with a feedback loop between the integrators. The two different ways of incorporating the feedback loop result in slightly different pole characteristics. Indicated are two different outputs, characterized by transfer function $R_1(z)$ and $R_2(z)$,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

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 an SNR of 107 dB (97 dB without resonators). Indeed, the addition of the poles, turning the Butterworth characteristic in to a Chebyshev type II characteristic, gives a significantly better SNR. The dc suppression of the loop-filter is still better than 120 dB, which is sufficient. Compared to the A-weighted SNR figures, the improvement is less, because the poles serve primarily 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 and 19 kHz. In that case the SDM is stable to inputs up to 0.58, with an SNR of 120 dB. Note that in this case there is still one integrator with a pole at dc, and thus there cannot be any dc drift. To clarify the operation of such an SDM, pseudo code of the SDM is provided in Appendix B.

Fig. 10 illustrates the effect of the resonator sections. We see that with the resonators, the quantization noise is substantially suppressed in the area of 10–20 kHz, whereas below 10 kHz the SDM without resonators has the better performance.

3 SIGNAL CHARACTERISTICS OF 1-BIT SDMs

The fact that a 1-bit SDM (and likewise any other 1-bit coder topology) contains a strong nonlinearity, namely, a 1-bit quantizer, has its ramifications on the behavior of the device, which often cannot be predicted by a linearized model. In the next sections a number of the effects that 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 1-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 1-bit audio, which take half the level of the feedback path as 0 dB (50\% modulation depth). SNRs are determined as the SNR at the maximum signal level the SDM can accommodate without overload.

3.1 Stability

For every SDM design there is a tradeoff between the stability of the modulator and the SNR in the baseband. As an example, consider the results in Table 1 for different fifth-order SDMs, 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 shows a schematic representation of a clipped inte-

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).

Cut Off (kHz)	SNR (dB)	Maximum Input Level
80	95	0.77
90	100	0.71
100	104	0.66
110	106	0.60

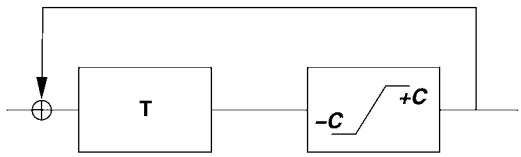


Fig. 11. Principle of clipped integrator. Absolute value of integrator output cannot exceed a value of *C*.

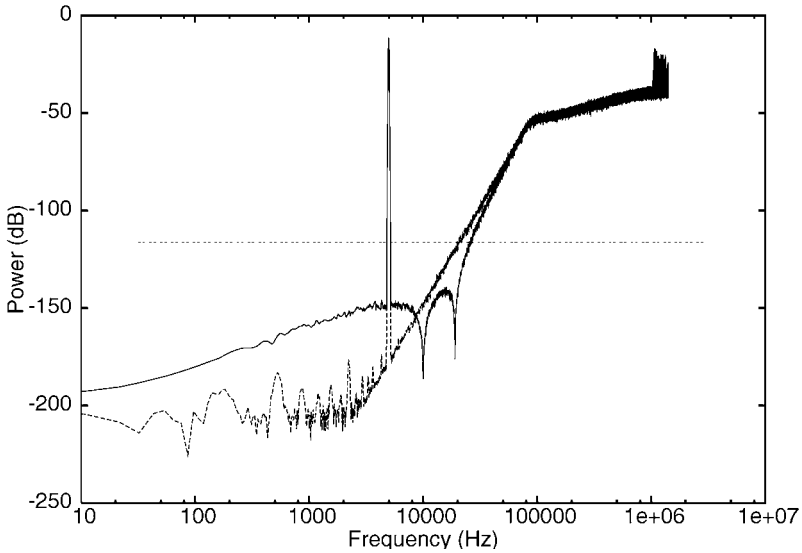


Fig. 10. Spectrum of fifth-order SDM with (——) and without (– – –) resonator sections. SDMs in this example are undithered. Horizontal line represents 16-bit resolution level (97 dB SNR over 20 kHz).

grator. 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.

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 oscillating), while still allowing integrator values that 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 two 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 the SNR with respect to the 120 dB without clippers. However, whereas the original SDM turned unstable at inputs of 0.59, its clipped version 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 1-bit audio the maximum input level is defined such that clipping occurs only rarely.

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.

Table 2. Typical example of the influence of clippers on normal SDM operation.

Input Level	Clipper Levels*					SNR (dB)
	C_1	C_2	C_3	C_4	C_5	
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

*Number of times a clipper was activated in a run of 300,000 samples.

3.2 Spectral Properties

Due to the inherent nonlinearity of the 1-bit quantizer, the spectrum of a 1-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 nonlinearity due to the quantization effect [13], this has been the first method resorted to in the literature to linearize SDMs too. In this section, we will discuss the appearance of nonlinearity in an SDM, even though in practical audio applications the influence of this nonlinearity is so benign as to be absent, as can be seen from inspection of Fig. 10, where no nonlinearity can be observed above the quantization noise floor at −150 dB.

3.2.1 Undithered SDMs

An appearance of the inherent nonlinearity due to the 1-bit quantizer can be observed in the spectrum of an SDM. Whereas for high-order SDMs, which are typically used for high-end audio applications, the effects of nonlinearity are hardly visible, they are for low-order SDMs, and they 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 characterized by the following noise transfer function:

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

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 nonlinearity of the SDM, the noise transfer functions that are expected from linear modeling are included in Fig. 12. The dotted curve is what would be expected on the basis of a 64 times coherent average [15] [thus reducing any uncorrelated component by $3\log_2(64) = 18$ dB]. The dashed curve is what would be expected without any coherent averaging at all. These curves show that the noise transfer function that is obtained does not resemble the theoretically expected noise transfer function. 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 one-half of the total power, which exceeds the signal power (−18 dB) by far. As a result of these high-powered high frequency 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 1-bit code is constant and equals 1.

This is in sharp contrast to any other quantized code. Therefore power that is spent 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 1-bit noise shaping does not follow the Gerzon–Craven theorem [16], in contrast to multibit noise shapers[3].

3.2.2 Dithered SDMs

While for PCM there exists mathematically optimal dither [13], namely, TPDF dither (dither with triangularly shaped pdf) spanning 2 least significant bits (LSBs), we cannot expect TPDF dither to linearize the 1-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 Section 3.3, 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,

[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 1-bit SDM $S + N = 1$ always, the Gerzon–Craven theorem does not hold in its published form.

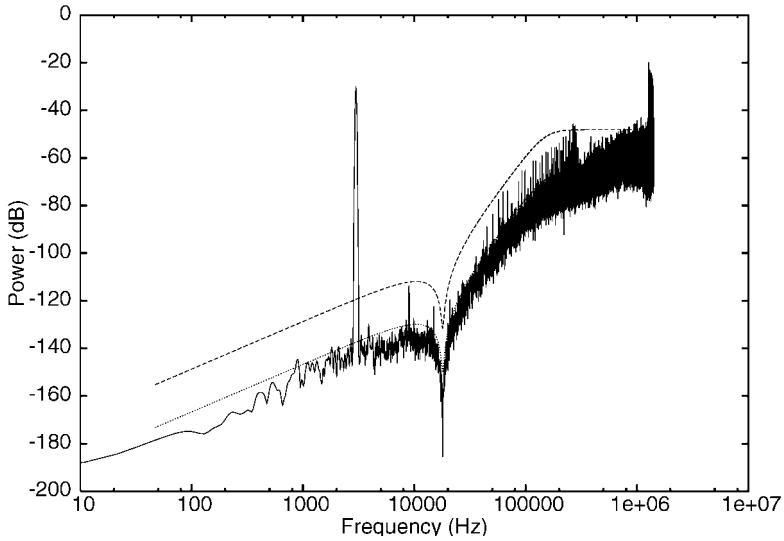


Fig. 12. Spectrum of third-order noise shaper used in analysis of linearization techniques. Input signal is a 3-kHz sine wave, −18 dB. To obtain this spectrum, a series of 64 coherent averages and 10 power averages has been used. – – – theoretically expected NTF; ····· expected NTF after 64 coherent averages.

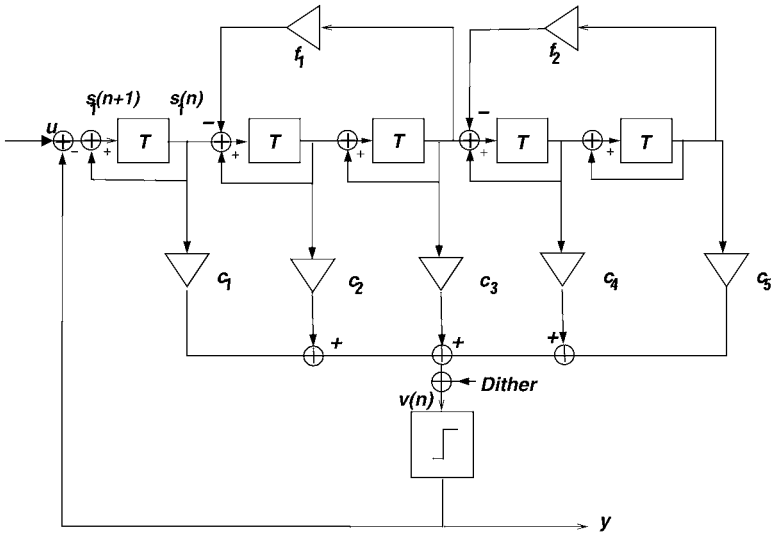


Fig. 13. Application of (amplitude domain) dither just before quantizer in a (here fifth-order) SDM. Also indicated are the first of the "states": $s_1(n + 1)$ before and $s_1(n)$ after the first delay element. States for subsequent delay elements can be assigned accordingly, which lead to state-space description of an SDM (see Section 3.3).

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 noise transfer function follows the theoretically predicted curve closely, and there is no obvious sign of distortion. Due to the fact that the high-powered high-frequency 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 to about 82 dB. Again, note that this is almost purely due to a redistribution of power, and not to the additive character of dither (as is the case in dithering a multilevel 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 of whether it would be necessary to linearize an SDM in the literal sense, since the only desire we have is that the lower 100 kHz be represented correctly, it might be unrealistic to require linearization to the extent that also the high-powered high-frequency 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 noise transfer function is slightly different from what is theoretically expected, but, more importantly, the high-frequency components have re-appeared.

As a result the SNR has increased from 82 to 88 dB, and the maximum dc input has increased from 0.53 to 0.69. Hence it is always advisable to judge the tradeoff between advantages and disadvantages of dithering. In the same

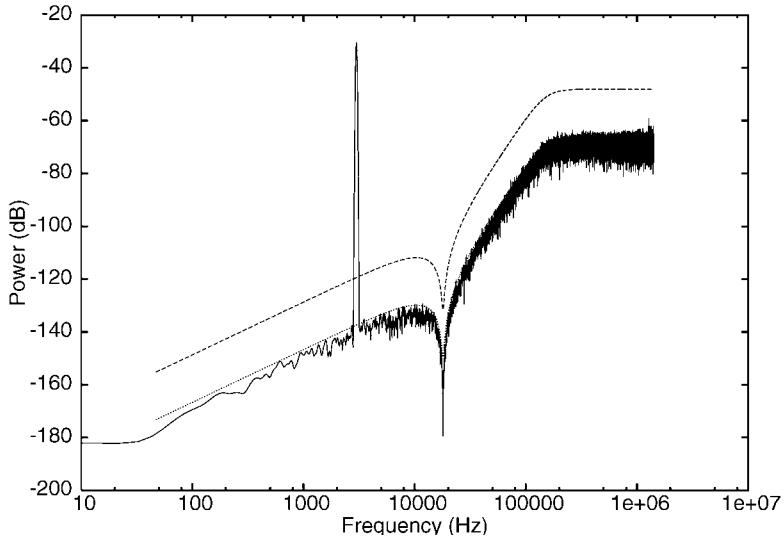


Fig. 14. Spectrum of third-order noise shaper with heavy dither. Input signal is a 3-kHz sine wave, −6 dB. To obtain this spectrum, a series of 64 coherent averages and 10 power averages has been used. The dashed curve is the theoretically expected NTF; the dotted curve the expected NTF after 64 coherent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

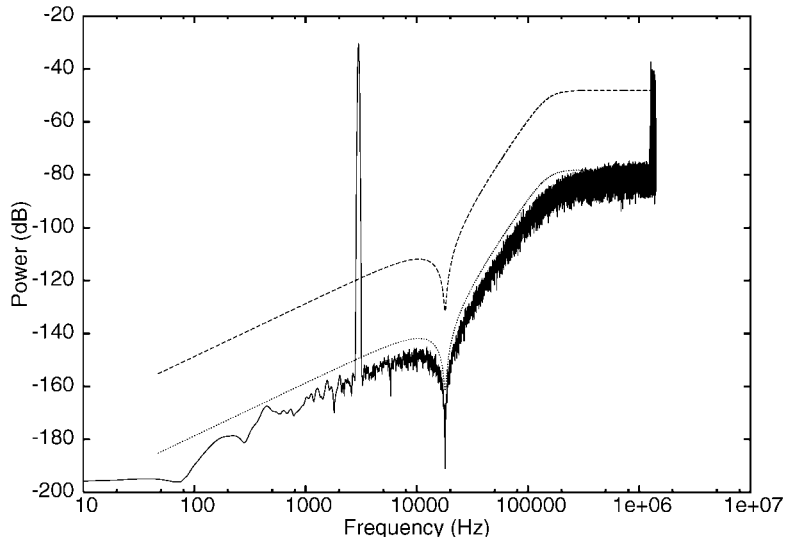


Fig. 15. Spectrum of third-order noise shaper with RPDF dither of half-width 0.5. Input signal is a 3-kHz sine wave, −6 dB. To obtain this spectrum, a series of 1024 coherent averages and 10 power averages has been used. The dashed curve is the theoretically expected NTF; the dotted curve the expected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

line of thinking, a precorrection 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], 1-bit SDM is compared to time-quantized 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 1-bit SDMs, even though optimum dither is yet to be defined, dithering schemes more effective than amplitude domain dither exist.

3.3 Limit Cycles

Limit cycles are a known phenomenon in any system with feedback and nonlinearity. As such they are also known from the design of digital IIR filters [21]. Not unexpectedly, therefore, limit cycles play also an important role in SDM design. We will use as a definition of a limit cycle a sequence of P output bits, that 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]. Even though limit cycles can exist for nonconstant input [24], we will restrict the discussion to dc inputs only, as these represent the situation occurring most often.

3.3.1 State-Space Description

The state-space description is a highly convenient way to describe the behavior of an SDM in the time domain. To illustrate the state-space description of an SDM, Fig. 13 will be examined. This figure displays the states s_i in a feedforward topology of an $N =$ fifth-order SDM with two resonator sections, as designed in Section 2.2. From Fig. 13 we can read that, in the absence of dither,

$$v(n) = \sum_{i=1}^{N} c_i s_i(n)$$
$$y(n) = \text{sign}\left[v(n)\right] \tag{10}$$

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], [26] that the evolution of states can be expressed concisely as

$$v(n) = \mathbf{c}^T \mathbf{s}(n)$$
$$\mathbf{s}(n+1) = \mathbf{A}\mathbf{s}(n) + \left[u(n) - y(n)\right]\mathbf{d} \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is called the "transition matrix" and $\mathbf{d} = (1, 0, 0, 0, 0)^{\mathrm{T}}$ 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 $\mathbf{s}(n)$,

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

From Eq. (12) we can infer that the initial integrator states are simply a kind of offset to the signal. The spectrum of the signal is determined completely by the second term on 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 an SDM is not determined by its initial integrator states.

With minor adaptations, the same state space formalism can be applied to other 1-bit coder topologies as well.

3.3.2 General Formulation of Limit Cycle Conditions

The compact representation Eq. (12) gives the means to view the consequences of a limit cycle directly. 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 [25] 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:

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

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 existence of a limit cycle with the assumed sequence of 1's. 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 limit cycles 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 an SDM with an aggressive noise transfer function (1) and for an SDM with a mild noise transfer function (2). The worst-case situation is depicted by + and ✻ signs. This represents the minimum amount of dither that is needed to certainly break up the most stable limit cycle for SDMs 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 SDM 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 1's followed by 11 −1's for SDM 1, and likewise 16 1's and 16 −1's for SDM 2. This corresponds to square waves of frequency 120 kHz and 80 kHz, which are exactly the corner frequencies of the noise transfer function design of SDMs 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 behavior 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 breakup 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, … , −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 breakup decreases. This is indicated in Fig. 16 by × and □ for SDMs 1 and 2, respectively. 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 amounts of dither are reduced to about 0.03 and 0.017 for SDMs 1 and 2, respectively, which is consistent with the intuition that longer limit cycles represent more boundary conditions to be fulfilled and are thus more easy to break up. While the amounts of dither discussed above are still significant, it should be remarked [25] 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.

4 THE CREATION OF 1-BIT CONTENT

In the previous sections the standard generation of 1-bit codes has been detailed. While obviously this is a crucial part in the high-sampling-rate concept of 1-bit audio, more complex signal processing is most often necessary in the production of a music release. Also, signal characteristics of 1-bit audio set some requirements on the replay of 1-bit audio.

In this section the signal chain leading to a 1-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 1-bit audio.

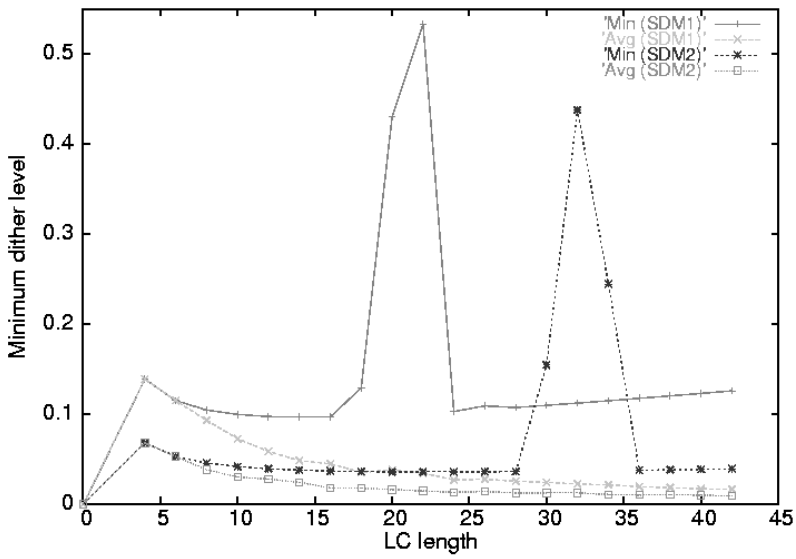


Fig. 16. Dither needed to break up a limit cycle corresponding to a dc input 0. +, ✻ worst-case situation: Level of dither necessary to certainly break up the most stable limit cycle, for aggressive and mild SDM, respectively. ×, □ average amount of dither needed to break up a limit cycle for aggressive and mild SDM.

4.1 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 ADC, this is not necessarily a native 1-bit converter. Often, high-end ADC, are 3–6-bit converters running at sample rates between 128 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 1-bit formats. The main reason is that many people in the recording industry want to save on necessary disk space while not giving in on the 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 1-bit format can be performed using an SDM, or any other 1-bit coder.

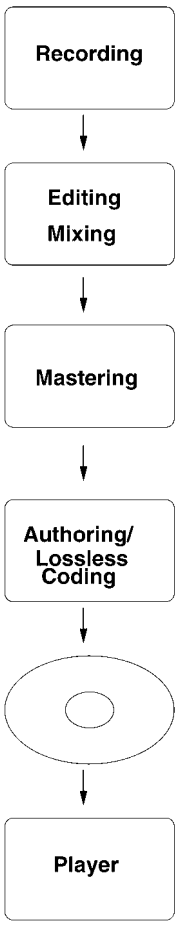


Fig. 17. Typical signal processing chain for 1-bit audio applications.

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 that is rather different from standard signal processing, and is detailed, for example, in [26]. 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 1-bit audio can be processed, will be shown in Section 4.3.

In the authoring phase, finally, changes to the signal content are no longer made. In most cases the data will be losslessly compressed in this phase. The compression that is employed for 1-bit audio is a scalable compression technique, and is detailed in the companion paper, "Lossless Compression of 1-Bit Audio," E. Knapen et al., this issue, pp. 190–199.

4.2 Playback Chain

An important aspect in replaying a 1-bit audio recording is the presence of a substantial amount of high-frequency (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 that 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 1-bit audio in a typical player. Such a configuration is shown in Fig. 18. A typical DAC chip (see for example, [27] or [28]) contains the first four 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 high-frequency signal present in the 1-bit audio signal can be removed without any compromise. As an example, consider a filter that is designed according to the following criteria: passband 0–100 kHz, flat within 0.01 dB; transition band 100–800 kHz; stopband 800 kHz–1.4 MHz, suppression −100 dB. This leads to a filter with only 22 taps, and thus does not pose any additional constraint in

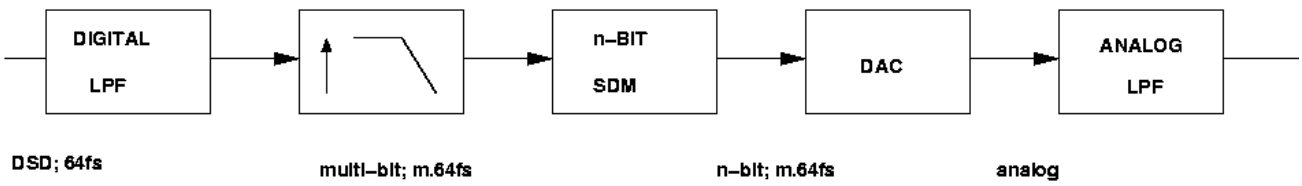


Fig. 18. Example of audio chain found in 1-bit capable player. The 1-bit audio is first low-pass-filtered in the digital domain, followed by upsampling to $m \cdot f_s$, typically 128 or $256f_s$. This high-rate signal is then fed to an *n*-bit SDM, where *n* typically varies between 1.5 and 5. Finally the analog output is passed through an analog low-pass filter.

terms of hardware. The filters that 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 low-pass filter does not influence the impulse response of 1-bit audio [29] 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 1-bit audio stream. Still, the signal contains substantial amounts of high-frequency, which is foremost white noise. The signal is then upsampled to a frequency that is used to perform the digital-to-analog conversion. As this upsampling also includes a digital low-pass operation, the first low-pass filter 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 [27] and 5 [28]. It is this signal that is converted to the analog domain. Due to the noise-shaping process, which is intrinsic in modern high-end DAC, 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 1-bit audio players, some additional filtering is provided to reduce the amount of high-frequency noise and signal even further to levels well below −30 dB. This filtering protects tweeters against full-scale high-frequency signals, which potentially could occur with the wide-band signal capabilities of 1-bit audio. It is important to remark that the high-frequency signal levels at which these additional filters need to operate are quite low due to the digital prefiltering (which removed a very substantial amount of high-frequency 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.

4.3 Signal Processing of 1-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 1-bit audio. For example, if a 1-bit audio signal is volume adjusted, with a gain $g = 0.123456$, the resulting output (the 1-bit signal multiplied by g) is a multibit word. Hence any signal processing for 1-bit audio principally always consists of a cascade of the actual processing step, followed by a requantization.

To obtain a realizable system, a low-pass filter before the requantizer is generally necessary. The reason for this is that the SDM that is used as a remodulator cannot cope with the high signal levels the 1-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 enough power such that the remodulator remains in stable operation. In this respect, the feedforward and feedback structures have quite different behavior. As shown in Section 2.2, the feedforward 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 noise transfer function filter characteristic. The feedback structure, on the contrary, has strong suppression of the input signal from the afore-mentioned corner frequency (see also Fig. 8). Hence a feedforward 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-tap) FIR filter that gives sufficient high-frequency suppression to allow subsequent requantization is shown in Fig. 19.

Because an FBSDM already contains input signal filtering, it is very tempting to contract some signal processing steps and the SDM remodulator. This approach has been investigated in [7], [30], [31]. An example where an IIR filter is contracted with an SDM is shown in Fig. 20. This device shows many practical advantages, such as the

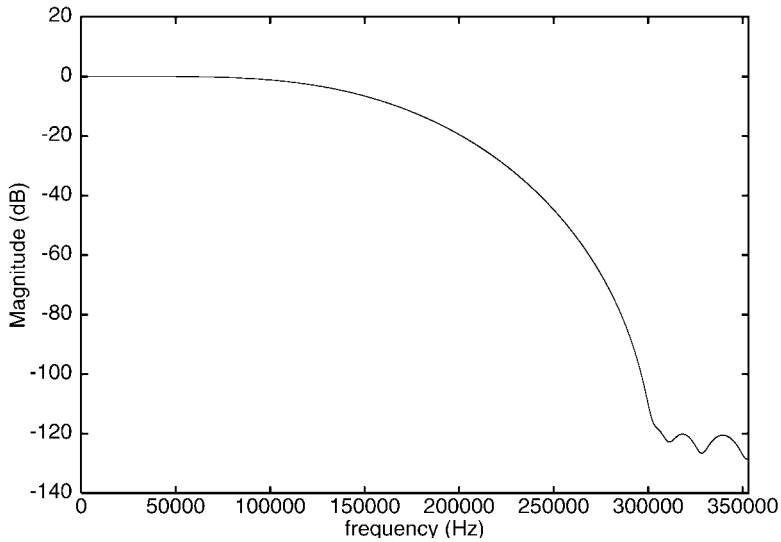


Fig. 19. Transfer function of filter which can be used to remove high-frequency of 1-bit audio signal such that it can be input to a subsequent SDM.

absence of multibit 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/remodulation, although the direct intermediate multibit path is absent.

While the work presented in [7], [30], [31] addresses several issues in 1-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. 20, the total signal path will contain multiple requantizations. Even though the low-pass filtering may have succeeded in the removal of 99% of all high-frequency energy, we certainly do not want to filter the signal in the region below 100 kHz, and still some energy resides in the band of 20–100 kHz if the 1-bit sample rate is 2.8 MHz. As a result of this, buildup 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 buildup of noise, typically the amount of noise that is left after filtering the 1-bit signal is much larger. This effect is illustrated in Fig. 21(a), where the effect of multiple requantizations is displayed schematically. This figure can be explained as follows. If we have a 1-bit signal, its noise starts to rise above 20–30 kHz, and reaches an almost flat level above 90 kHz. If, in a subsequent requantization, 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 and the quantization noise that has been introduced. If this cascade is repeated, it is easy to see why there will be a buildup of high-frequency 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 Fig. 21(b). As the number of requantizations increases, the signal quality drops slowly due to the increase in the noise floor. At the moment when the high-frequency noise is large enough to activate the clippers, the signal quality drops rapidly. This effect has been studied in more detail in [32], and when careful signal processing is performed, hundreds of requantizations can be performed before the signal degradation shown in Fig. 21 occurs.

Still, it is a most pragmatic approach to perform all signal processing in a multibit domain as illustrated in Fig. 22, such that buildup of noise is limited. The conversion to $64f_s$ 1-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 1-bit audio signal is significant in the higher part of the band of interest. Indeed, when the sample rate of the 1-bit audio system equals 2.8 MHz, as is the case with many concurrent systems, this is true. However, for rates of 5.6 MHz or higher, this reasoning is no longer true, and subsequent requantizations do not add substantial noise. Likewise, the analysis has focused 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

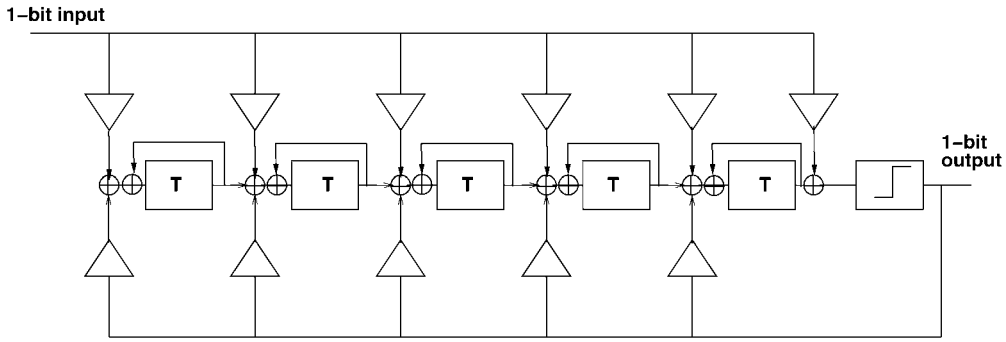
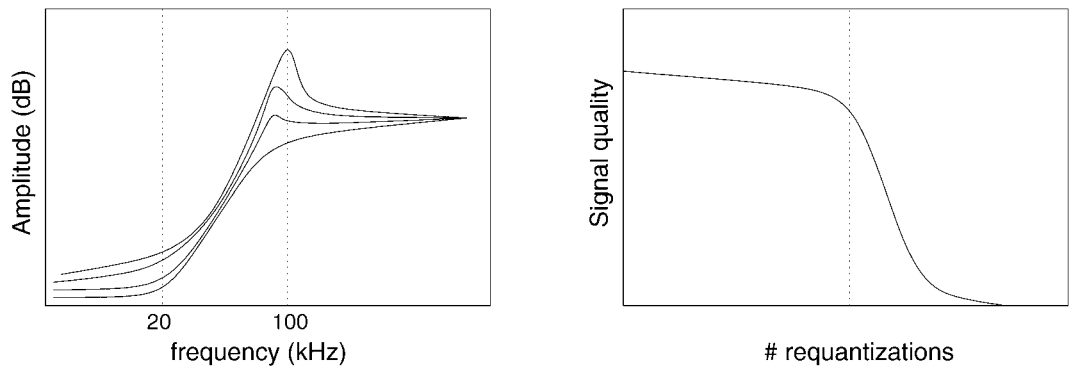


Fig. 20. Contraction of IIR filter characteristic and SDM, giving a structure with 1-bit input and 1-bit output.



(a) (b)

Fig. 21. Schematic presentation of effect of multiple quantizations for SDM running at 64 f_s. (a) due to multiple requantizations, a buildup of high-frequency noise occurs. Amplitude scale is arbitrary. (b) while for a limited number of requantizations the only adverse effect is a reduction in SNR, when buildup of high-frequency noise is too large, SDM will start to clip, leading to significant distortion.

of due to their increased stability (see Section 6). It is therefore yet undecided which type of processing will eventually be preferred.

5 RECENT DEVELOPMENTS IN 1-BIT AUDIO AND SIGNAL PROCESSING

While SD modulation has been around for a long time, it has seen a recent revival of interest due to the proposal to use 1-bit audio as a consumer delivery format. Recent research has focused on improvement of the characteristics of SDMs, most notably the stability of SDMs and the accuracy with which digital signals can be represented. In this section we will detail some of these new developments.

5.1 Controlling the Noise Shaping Characteristics

5.1.1 Pre-Correction SDMs

Sigma–delta precorrection (SDPC) has been introduced in [14]. The idea is based on the fact that, while distortion is a nonlinear phenomenon, it can be corrected for approximately when linear techniques employed. 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 Section 3.2.2) as an alternative linearization technique reduces the maximum stable input of SDM. SDPC has no stability penalty, while it is very successful in linearization.

Within SDPC (see Fig. 23) SDM is modeled as a nonlinear element ΣΔ, 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

$$s(u) = u + \alpha_2 u^2 + \alpha_3 u^3 + \dots . \tag{17}$$

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

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

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, $\alpha_i << 1$.)

To gain some insight in the performance of SDPC, it has been applied to the third-order SDM also used in Section 3.2. The spectrum of the resulting signal y is displayed in Fig. 24 in the range of 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 total harmonic distortion, which in this case is an increase in the noise floor by 3 dB. This is clear from an 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

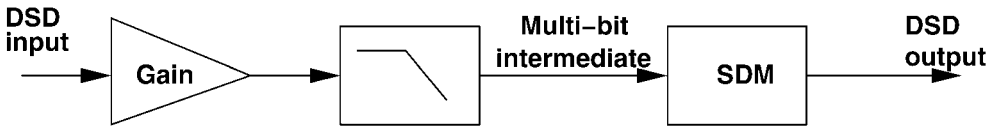


Fig. 22. Example of performing two sequential operations on 1-bit audio data. First a gain adjustment is applied, after which an IIR filter operation is applied without leaving intermediate high-rate multibit domain.

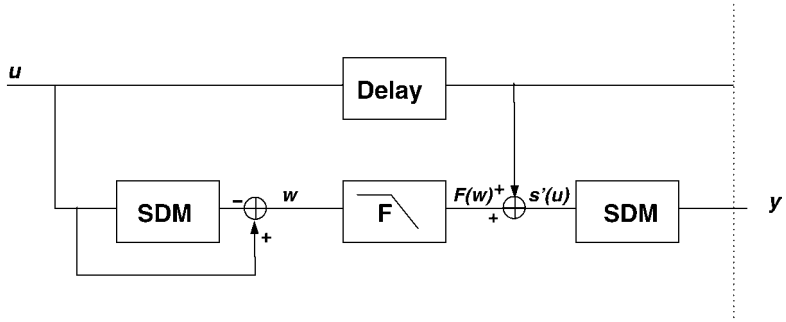


Fig. 23. Basic sigma–delta precorrection (SDPC) structure.

on the basis of the absence of correction components in the signal $s'(u)$.

It proves possible to apply this technique to slightly dithered and high-order SDMs as well. In this case any possible nonideality occurs at levels below −220 dB, which are not achievable in real life. It is a better result than probably could have been obtained by dithering the SDM (see Section 3.2.2). Hence due to the fact that SDPC poses no penalty with respect to stability, it is also more effective (at least in the band of 0–100 kHz) compared to dithering.

5.1.2 Parametrically Controlled Noise Shaping

Parametrically controlled noise shaping is introduced in [20], following the observation that in "classical" SDM design (see Section 2.2) the addition of more than seven integrators does not result in a noticeable 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 recognized, whereas the lower part implements a parametric equalizer. Such a system allows significant freedom in the choice of the final noise transfer function, which can be optimized. 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 noise transfer function, it is also extremely efficient in linearizing the SDM. This is illustrated in Fig. 26. In this figure a power spectrum of a para-

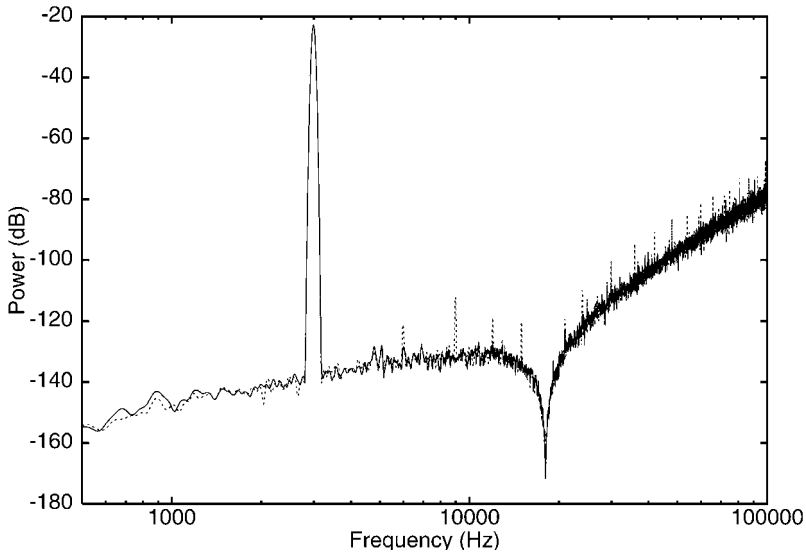


Fig. 24. Spectra of original SDM (– – –) and its implementation according to Fig. 23 (——). Spectrum of original SDM has been obtained using four coherent averages and 10 power averages; the other using eight coherent averages and 10 power averages. The fact that noise floors of spectra coincide precisely illustrates the 3-dB loss in SNR due to SDPC.

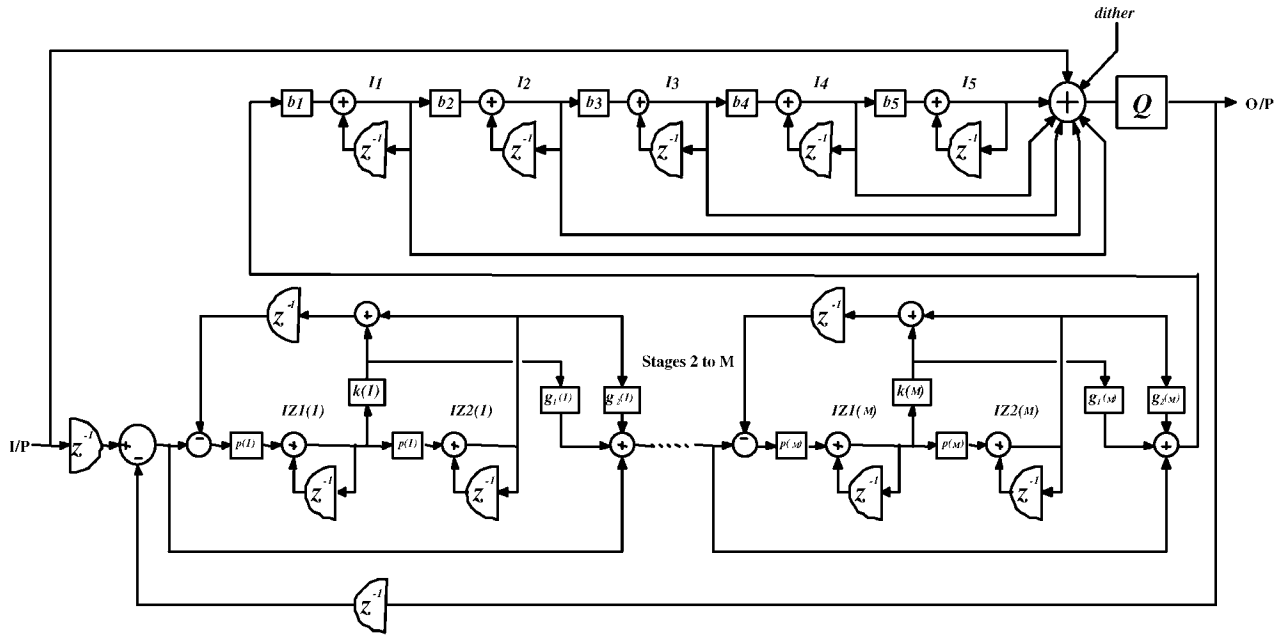


Fig. 25. Schematic representation of parametrically controlled noise shaper. (After [20].)

metric 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 nonlinearity, which would 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 of 0–20 kHz the parametric SDM displays a resolution equivalent to 24 bit.

As with classical SDMs, the final performance is limited by the stability of the SDM. While parametric SDM design addresses the question of how optimum noise shaping can be realized, it does not deal with the issue of stability. In the next two sections new developments are highlighted which do address this important question.

5.2 Controlling Stability

All designs that try to address the issue of stability improvement essentially return to the original question of 1-bit coders in Section 1.1: 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 longer any resemblance 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, 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 [33], [34]. Obviously the secret is in the algorithm hidden in the box labeled *VQ*, which decides the sign. It is, however, not obvious what this algorithm should look like. In [33] 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 optimum dependence on the individual state variables of the quantizer output.

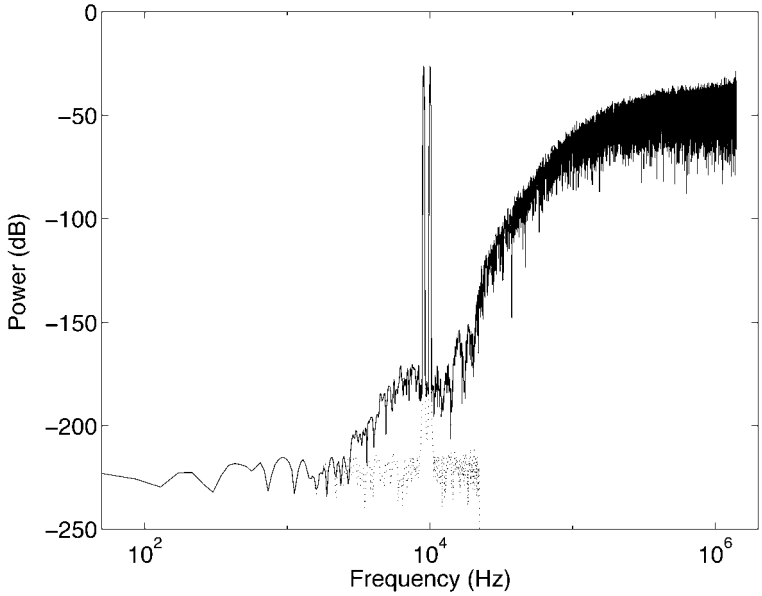


Fig. 26. Parametric SDM, fed with 9- and 10-kHz input sine waves. – – – 32-bit input to SDM. Note absence of 1-kHz intermodulation product. (From [20].)

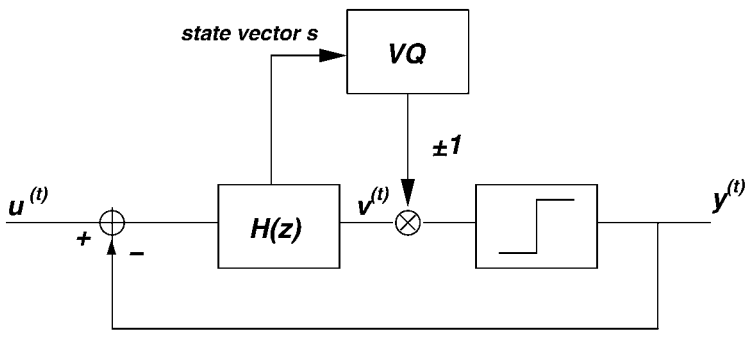


Fig. 27. The concept of vector quantizer *VQ*, embedded in SDM.

5.2.1 Step-Back SDMs

Historically step-back SDMs were the first to try 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 [33], [34]. 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 loop filter.) Certain preset 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 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 1-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 [35] and discussed more extensively in [36]. The basic idea is that two (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 [33] is the variable state step-back pseudo-trellis SDM presented in [37]. 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 successful in stabilizing SDMs. With the help of this algorithm a highly aggressive noise-shaping characteristic could be obtained which resulted in an SDM displaying the equivalent of 32-bit 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.

5.2.2 Trellis-Type SDMs

A completely new view on ways to minimize the time integral of the error ε was presented in 2002 by Kato in a seminal paper [38]. There connection was made between the trellis algorithm, known from error correction theory, and 1-bit noise shaping. The basic idea of the application of the trellis algorithm to 1-bit coding is to minimize the time integral of the loop filter output $v(t)$. This is in contrast with the approach sketched in, for example, [33], and [37], where the idea is to bound the loop filter output $v(t)$. Assume that up to the clock cycle $t = t_0$ the optimum 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 errors $v_{-1}(t_0 + 1)$ and $v_{+1}(t_0 + 1)$, respectively. One time instant later, again an output of either -1 or $+1$ is possible, resulting in four different possibilities (paths) for the two output bits. Every path has its own associated cost $C_{\omega_N}(t)$ (called pathmetric [38]), which can, for example, be defined as the sum of the squared frequency-weighted error values

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

with ω_N a sequence of N output bits.

Advancing time once more, the number of possibilities doubles again and becomes eight, 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 shows a trellis with order $N = 2$. The figure shows the four combinations of two 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 two possible sequences to two 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 one 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 [39], it turns out that if the system runs long enough, paths converge. This means that regardless of which path is examined, they all originate from the same "mother sequence" of bits. This mother sequence is then the sequence of bits that is the trellis approximation to the optimal sequence. Fig. 29 illustrates the convergence. 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 [38] and [40], application of the trellis con-

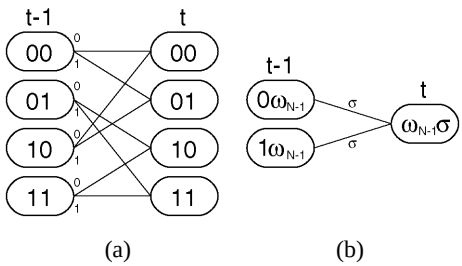


Fig. 28. Origination of new candidates (clock cycle t) from old candidates (clock cycle $t - 1$); complete state diagram. (a) $2^N = 4$ candidates. (b) General case. For clarity, signal level -1 is represented by 0 inside all figures.

verter increases the maximum stable input range and improves the SNR. Simulations have shown that for a significant gain in 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 sixth-order system contains $2^6 = 64$ SDMs which, together with bookkeeping overhead, results in a system that is about 100 times more expensive than a normal SDM.) In [40], and [41] 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.

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

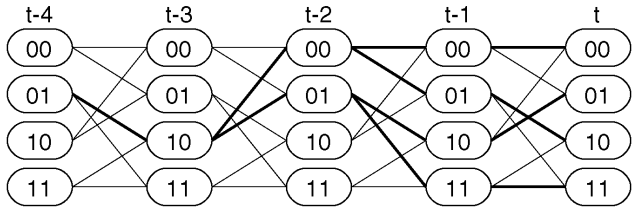


Fig. 29. Convergence of paths. Bold lines show origination of four candidates. Different candidates terminate with different output symbols, but in history ($t \rightarrow -\infty$) output sequences converge to a single solution.

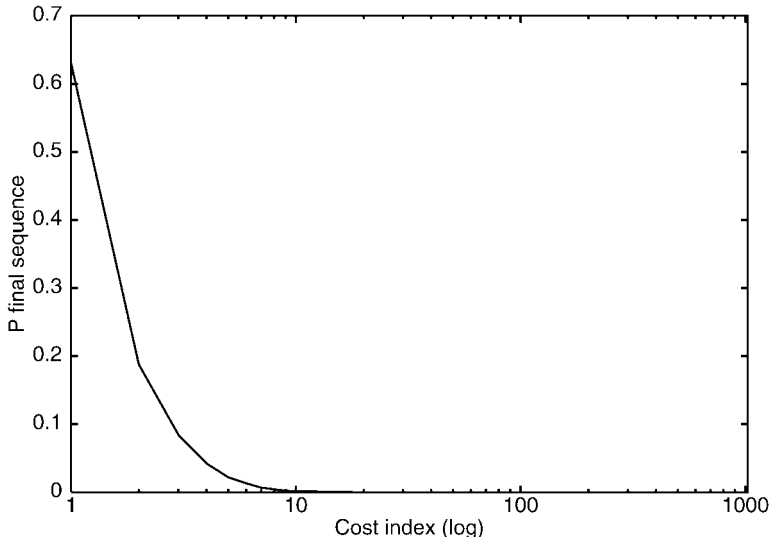


Fig. 30. Probability for candidate with a certain cost index to become optimum solution. Cost index ranks candidates on increasing cost function.

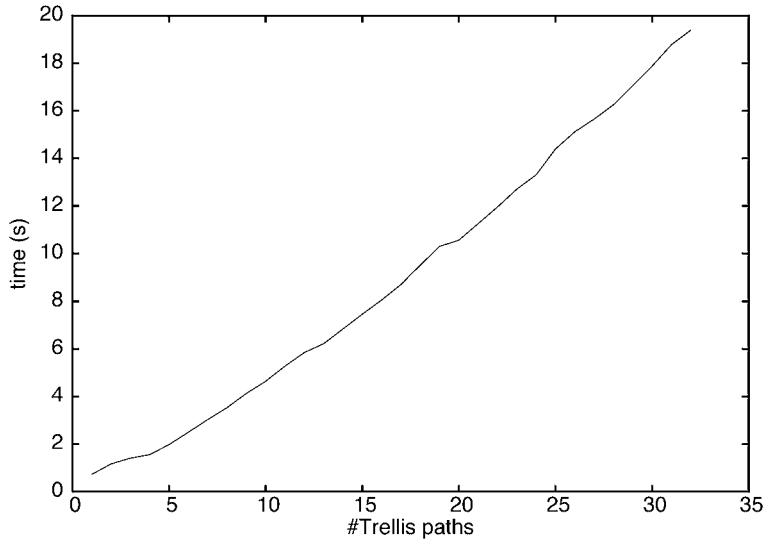


Fig. 31. Execution time for different data sets as a function of number of trellis paths.

CPU requirements increase only linearly with the trellis depth, instead of exponentially. In view of this a practical application of the efficient trellis algorithm has come within reach.

6 SUMMARY AND CONCLUSIONS

The concept of 1-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 ADCs and DACs converters. As a result of the goal to maintain an audio quality as high as possible in consumer audio delivery, 1-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 1-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 behavior of the SDM, or noise-shaping devices in general. While theoretically these devices are not perfectible in a mathematical sense as PCM is, it has been shown that in practice any nonlinearity is of a harmlessly low level. Also, linearization techniques have been reviewed which linearize a first-order SDM, seeding the thought that SDMs {\sl are} mathematically perfectible, only different from the way PCM is perfectible.

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 1-bit audio. Along the same line, replay of 1-bit audio is discussed. However, signal processing requires additional headroom (as well as PCM does) for which either an increase of the sample rate or an increase in the number of bits is required. Both of these are easily realizable.

We have shown that 1-bit audio offers a great deal of flexibility, as it does not rely on specific predefined coding paradigms. As a result the 1-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.

7 ACKNOWLEDGMENT

The authors would like to thank their colleques from Philips and Sony for many fruitful discussions and intense collaborations. Also, the many discussions with M. O. H. Hawksford, S. P. Lipshitz, J. D. Reiss, and J. Vanderkooy have greatly contributed to the increased understanding of 1-bit coding.

APPENDIX A
EXAMPLE DESIGN OF AN SDM

As an example, we will design a fourth-order SDM, with a noise transfer function according to a Butterworth high-pass filter design, cutoff frequency 150 kHz, as discussed in Section 2.2. Because the SDM needs to be realizable, the total loop needs to embody at least a single delay, that is, the term with z^0 in the signal transfer function 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 high-frequency gain that is larger than 1. With the foregoing in mind, we obtain for the noise transfer function

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

This results in the following coefficients in the feedforward 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 feedforward structure the signal transfer function is now given by

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

For the feedback structure the signal transfer function is given by

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

APPENDIX B
SDM CODE

In this appendix we provide the C-like pseudo code for the SDM discussed in Section 3.1. The code simulates 100 000

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);
       }
```

8 REFERENCES

[1] F. de Jager, "Delta Modulation—A Method of PCM Transmission Using the One Unit Code," *Philips Res. Rep.*, vol. 7, pp. 442–466 (1952).

[2] H. Inose, Y. Yasuda, and J. Murakami," A Telemetering System by Code Modulation—Δ–Σ Modulation," *IRE Trans. Space Electron. Telemetry*, vol. SET-8, pp. 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 Δ–Σ A/D Converter Incorporating a Novel Class of Dynamic Element Matching," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 35–51 (1992).

[5] R. T. Baird and T. S. Fiez, "Improved ΔΣ DAC Linearity Using Data Weighted Averaging," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1 (1995), pp. 13–16.

[6] H. Inose and Y. Yasuda, "A Unity Bit Coding Method by Negative Feedback," *Proc. IEEE*, vol. 51, pp. 1524–1535 (1963).

[7] J. A. S. Angus and N. M. Casey, "Filtering Sigma–Delta Audio Signals Directly," presented at the 102nd Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 45, p. 410 (1997 May), preprint 4445.

[8] R. W. Adams, P. F. Ferguson, Jr., A. Ganesan, S. Vincelette, A. Volpe, and R. J. Libert, "Theory and Practical Implementation of a Fifth-Order Sigma–Delta A/D Converter," *J. Audio Eng. Soc.*, vol. 39, pp. 515–528 (1991 July/Aug.).

[9] E. Stikvoort, "Higher Order One-Bit Coder for Audio Applications," presented at the 84th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 36, p. 382 (1988 May), preprint 2583.

[10] S. H. Ardalan and J. J. Paulos, "An Analysis of Nonlinear Behavior in Delta–Sigma Modulators," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 593–603 (1987).

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

[12] S. Wolfram, *The Mathematica Book*, 4th ed. (Wolfram Media/Cambridge University Press, Cambridge, MA, 1999).

[13] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and Dither: A Theoretical Survey," *J. Audio Eng. Soc.*, vol. 40, pp. 355–375 (1992, May).

[14] D. Reefman and E. Janssen, "Enhanced Sigma Delta Structures for Super Audio CD Applications," presented at the 112th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 50, p. 516 (2002 June), convention paper 5616.

[15] S. P. Lipshitz and J. Vanderkooy, "Towards a Better Understanding of 1-bit Sigma–Delta Modulators," presented at the 110th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 49, pp. 544–545 (2001 June), convention paper 5398.

[16] M. A. Gerzon and P. G. Craven, "Optimal Noise Shaping and Dither of Digital Signals," presented at the 87th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 37, p. 1072 (1989 Dec.), preprint 2822.

[17] S. R. Norsworthy, "Effective Dithering of Delta–Sigma Modulators," in *Proc. IEEE ISCAS'92*, vol. 3 (1992), pp. 1304–1307.

[18] S. R. Norsworthy and D. A. Rich, "Idle Channel Tones and Dithering in Delta–Sigma Modulators," presented at the 95th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 41, p. 1053 (1993 Dec.), preprint 3711.

[19] D. Reefman and E. Janssen, "DC Analysis of High Order Sigma–Delta Modulators," presented at the 113th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 50, p. 969 (2002 Nov.), convention paper 5693.

[20] M. Hawksford, "Time-Quantized Frequency Modulation with Time Dispersive Codes for the Generation of Sigma–Delta Modulation," presented at the 112th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 50, p. 516 (2002 June), convention paper 5618.

[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, 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 Syst. I*, vol. 49, pp. 646–656 (2002).

[24] J. D. Reiss and M. B. Sandler, "They Exist: Limit Cycles in High Order Sigma Delta Modulators," presented at the 114th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 51, p. 436 (2003 May), convention paper 5832.

[25] D. Reefman, J. Reiss, E. Janssen, and M. Sandler, "Stability Analysis of Limit Cycles in High Order Sigma Delta Modulators," presented at the 115th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 51, p. 1240 (2003 Dec.), convention paper 5936.

[26] D. Reefman and P. Nuijten, "Editing and Switching in 1-Bit Audio Streams," presented at the 110th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 49, p. 545 (2001 June), convention paper 5399.

[27] B. Adams, K. Nguyen, and K. Sweetland, "A 116-dB SNR Multi-Bit Noise-Shaping DAC with 192-kHz Sample Rate," presented at the 106th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 47, p. 533 (1999 June), preprint 4963.

[28] S. Nakao, H. Terasawa, F. Aoyagi, N. Terada, and T. Hamasaki, "A 117 dB D-Range Current-Mode Multi-Bit Audio DAC for PCM and DSD Audio Playback," presented at the 109th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 48, p. 1099 (2000 Nov.), preprint 5190.

[29] D. Reefman and P. Nuijten, "Why Direct Stream Digital Is the Best Choice as a Digital Audio Format," presented at the 110th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 49, p. 545 (2001 June), convention paper 5396.

[30] N. M. Casey and J. A. S. Angus, "One-Bit Digital Processing of Audio Signals," presented at the 110th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 41, p. 1053 (1993 Dec.), preprint 3717.

[31] J. A. S. Angus and S. Draper, "An Improved Method for Directly Filtering Sigma–Delta Audio Signals," presented at the 104th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 46, p. 569 (1998 June), preprint 4737.

[32] P. C. Eastty, C. Sleight, and P. D. Thorpe, "Research on Cascadable Filtering, Equalization, Gain Control, and Mixing of 1-Bit Signals for Professional Audio Applications," presented at the 102nd Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 45, p. 410 (1997 May), preprint 4444.

[33] L. Risbo, "Improved Stability and Performance from ΣΔ Modulators Using One-Bit Vector Quantization," *Proc. IEEE ISCAS'93* (1993), pp. 1361–1364.

[34] L. Risbo, *Sigma–Delta Modulators: Stability Analysis and Optimisation*, (Technical University of Denmark, 1994).

[35] A. J. Magrath, and M. B. Sandler, "A Use of Sigma–Delta Modulation in Power Digital-to-Analogue Conversion," *Int. J. Circuit Theory Appl*., vol. 25, pp. 439–455 (1997).

[36] A. J. Magrath, *Algorithms and Architectures for High Resolution Sigma–Delta Converters* (University of London, UK, 1996).

[37] M. Hawksford, "Parametrically Controlled Noise Shaping in Variable State-Step-Back Pseudo-Trellis SDM," presented at the 115th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 51, p. 1222 (2003 Dec.), convention paper 5877.

[38] H. Kato, "Trellis Noise-Shaping Converters and 1-Bit Digital Audio," presented at the 110th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 50, p. 516 (2002 June), convention paper 5615.

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

[40] P. Harpe, D. Reefman, and E. Janssen, "Efficient Trellis-Type Sigma Delta Modulator," presented at the 114th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 51, p. 439 (2003 May), convention paper 5845.

[41] E. Janssen and D. Reefman, "Advances in Trellis-Based SDM Structures," presented at the 115th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 51, p. 1257 (2003 Dec.), convention paper 5993.

THE AUTHORS

D. Reefman

E. Janssen

Derk Reefman was born 1967 in The Netherlands. In 1989 he received an M.Sc. degree (cum laude) in chemistry (on the subject of metallo-organic chemistry) and was awarded the 1990 Unilever Research prize for his work. He received a Ph.D. degree in physics (on the subject of theory and experiment in high temperature superconductivity) in 1993 and was awarded the biannual prize for the best Leiden University Ph.D. thesis in the fields of physics, chemistry, and biology. Later he joined Philips Research, The Netherlands, where he studied x-ray diffraction from 1993 until 1997. Having been an audiophile for quite a while, he decided to join the Philips Super Audio CD team, where he works on several digital and analog signal processing aspects of Super Audio CD, including professional audio as well as DA converter and power amplifier designs for consumer systems.

●

Erwin Janssen was born in 1976 in The Netherlands. He received a Master of Science degree (cum laude) in electrical engineering, along with an additional degree in computer science from the University of Twente, The Netherlands, in 2001. He joined Philips Research as a research scientist in the Mixed-Signal Circuits and Systems group in 2001. Since then he has been working on various aspects of signal processing for Super Audio CD. His research interests include audio signal processing and sigma–delta modulation.