ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA
Ciclo XX
Settore/i scientifico disciplinari di afferenza: ING-INF/01

Two-and-Three level representation of analog and digital signals by means of advanced sigma-delta modulation

Presentata da Ludovico Ausiello

Coordinatore Dottorato:
CLAUDIO FIEGNA

Relatore:
MASSIMO FERRI
RICCARDO ROVATTI
Corelatore:
GIANLUCA SETTI
MARC MOONEN

Esame Finale 2009

For(e)ward $\boxed{\gg}$

I guess everything started in 1997 when i first heard the song "Starship Trooper" by the Yes band in my living room: an old Sansui amplifier with two AR-6 loudspeakers.

Four seconds of pure miracle, with a shimmering bass line passed trough a *Tremolo*, litterally moving into my head and asking my soul to understand all of this. I needed to know how did the bassist Chris Squire make that incredible bass sound, I wanted to recreate it, use it on stage. After some months of research (at that time Google was not) I found out the most important part of the truth: It wasn't the player who invented that bass sound, but someone else. This someone was Eddie Offord, producer and sound engineer for the Yes band.

FAST FORWARD $\boxed{\gg}$

This work of thesis is a personal tribute to all young and old geniuses that during decades of *4-track* recording fought to preserve quality, invented new sounds, imagined new creative possibilities to be offered to the most unstable people living on this planet, namely musicians and, even worst, rock bands.

Furthermore, it is an homage to engineers and inventors that dedicated their lives and strength to unveil musical instruments instead of yet another racing car: Laurens Hammond, Rupert Neve, Leo Fender, Robert Moog, Adolphe Sax, Jim Marshall, Orville Gibson, Bartolomeo Cristofori are in my mind as high as Leonardo or Galilei. They left their intellectual work to people who are musicians first, people like me. I appreciate the intuitions behind their inventions much more than I blame their lack in business strategy. Their creations will still stand in time and will be played on and on while digital technology that tries to emulate them will be gone and born again a hundred times. The physical principles on which their instruments are based upon offer sensory and emotional perceptions. These are instruments that can be *felt*.

REWIND $\boxed{\ll}$

In all the recipes there are secret ingredients and hidden details: this is what I found on my path. Although in the last years the market has definitely embraced digital music and low quality *mp3*s as a standard, this research on audio authoring techniques is my little attempt to investigate in the opposite direction, far away from business, offering the highest quality for audio recordings.

In this thesis we present some innovative techniques of audio signals representation and compression. In the first we will introduce the typical production flow used in the music industry as a standard since the last thirty years: an important step of this chain is the digitalizing of the musical content by the mean of Analog-to-Digital converters. So we will briefly investigate some important aspects of A/D conversion and we will present a class of devices as our system choice.

Then in chapter 2 we will focus on the authoring step, meaning the phase of the global process in which the audio content is converted to fit the optical support where is eventually stored. There are several analogies between this phase and the initial A/D conversion, due to the presence of a last A/D or D/D conversion step. There are some algorithms presented in literature and used as a *de facto* standard, and from these already proposed techniques we will introduce an innovative software converter which embeds some new features. On one side the research for a better performance, on the other side a consideration on the system power efficiency. In chapter 3 a short introduction on the power losses causes in a digital amplification system will be presented and evaluated to underline the good results achievable. In the following chapters (4 and 5) two versions of the innovative software D/D conversion will be presented to underline the computational needs and possible savings of the proposed technique.

In chapter 6 we will introduce a broader innovation of A/D converter design, that is the frequency-warping applied to Sigma-Delta Modulators; this design technique could be implemented both in software and hardware (i.e. A/D or D/D converters) and it provides improved performances on both low-order and high-order modulators.

In the last chapter we will provide an extension of the state of the art in lossless compression algorithm necessary to fit the recorded musical material on the optical support (e.g. DVD) used as storage media. A dedicated three levels Linear Prediction Encoding lossless compressor algorithm will be presented together with a symbol-entropy encoder; we will then compare the obtained results with the state of the art.

Chapter 1

Analog to Digital Conversion

In this chapter we briefly present some characteristics of the fundamental process on which all digital audio recording systems are based: the analog to digital conversion. In the last decades the music industry has produced a *de facto* standard signal flow, from the recorded material to the consumer good. We present this flow in figure 1.1.

The musical content is first recorded, edited and mixed using analog and digital equipment (e.g. tape or optical compressors and digital reverberation systems) and then mastered. This is the last phase of the flow that is under the control of the artist and the producer and that needs both technical and artistic skills. After that in our music-business scheme we can transform the master in many possible audio formats that are now-a-days all in the digital domain (the master itself is in the largest percentage of cases already in a digital format). So the authoring step needs or A/D or D/D conversion, depending on the master format.

In general, thinking of the digital signal that we have to store on some kind of support we can represent the digital conversion as a step in which we transform a time-and-amplitude continuous signal into a quantized and discrete time one;

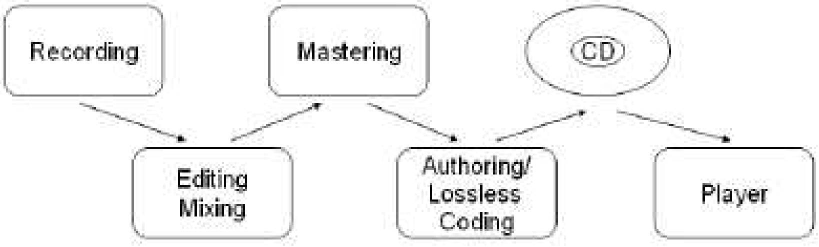


Figure 1.1: Typical music industry signal flow

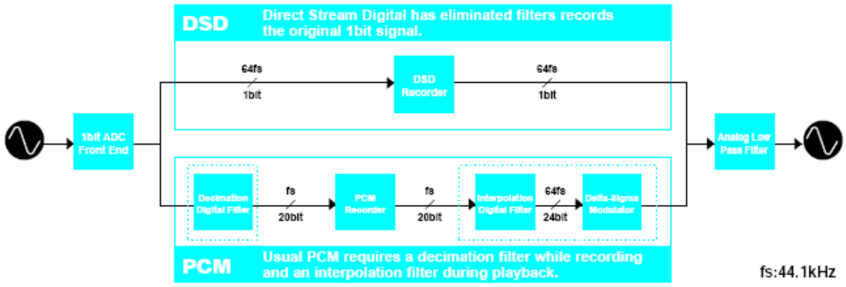

DSD
Direct Stream Digital has eliminated filters records the original 1bit signal.
64fs
1bit
DSD Recorder
64fs
1bit
1bit ADC Front End
Analog Low Pass Filter
Decimation Digital Filter
fs
20bit
PCM Recorder
fs
20bit
Interpolation Digital Filter
64fs
24bit
Delta-Sigma Modulator
PCM
Usual PCM requires a decimation filter while recording and an interpolation filter during playback.
fs:44.1kHz

Figure 1.2: Comparison between Sample-to-Sample and Sample-to-Stream conversion

there is a substantial loss of information due to the fact that we are restricting our signal to have a finite number of ”states” representing its own dynamics. On the other hand the sampling theorem by Shannon is not a demanding constraint for audio signals, because their band is limited from $20Hz$ to $20kHz$. We can indeed trade the sampling frequency with the resources used to represent the amplitude with some advantages in each case: the first case is the so called Nyquist conversion, or Sample-to-Sample conversion, in which we transform the master into a stream that has a large dynamics thanks to a large number of bits representing it (from 16 to 24). The latter one is the Oversampling conversion, or Sample-to-Stream conversion, in which we use a larger sampling frequency (F_s) multiplying the normal Nyquist F_s for an OverSampling Ratio (**OSR**) that can vary from 64 to 256; using this method we reduce drastically the dynamic range of the digital stream and we can even use a single bit representing the amplitude of the signal. These techniques can recall the parallel or serial representation of information content used in communication systems.

Looking at figure 1.2 we can compare the two possible way to store audio information once converted into the digital domain: the lower signal flow represents the standardized Linear Pulse Code Modulation (**LPCM**) used for audio Compact Disc (**CD**) and Digital Versatile Disc-Audio (**DVD**-Audio) [1], [2]. The upper signal flow is a different choice: it uses the Sample-to-Stream conversion that is implemented, for example, in the Super Audio CD systems (**SACD**)[3]. In reference there are exhaustive descriptions of the two encoding systems. Here is important to notice that the LPCM system needs several more processing passages (in the digital domain) and has been proposed in 1996 to be the standard audio representation for the DVD-Audio, supported by Pioneer. On the other side the 1-bit audio representation has been proposed as standard by Sony and Philips in 2001 under the name of Direct Stream Digital (**DSD)** and it is used in the proprietary Super Audio CD format.

In fact if we want to playback a multibit digital signal we have to convert it again in the analog domain and then pass it trough an analog amplifier; this is extremely useful if the intention is to create a system compatible with other (and older) musical media as tuners, turntables or cassete desks, but it still affects the system with the problems related to analog power amplification. On

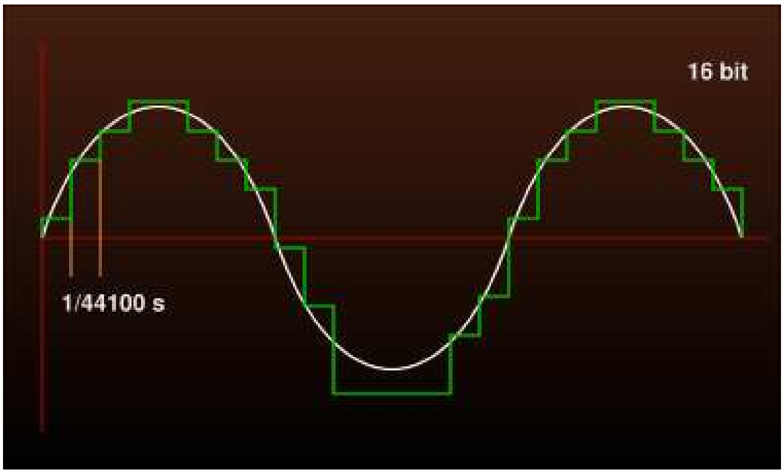


Figure 1.3: Quantization effect in a sample to sample conversion process

the other side the 1-bit audio system needs a totally redesigned power section. The truth is that in the last decade several manufacturers of power amplifiers have started realizing the so called ”digital” amplifiers [4] that are perfectly tailored to use the 1-bit encoding system. This is also one of the reasons why Sony and Philips has presented the DSD technology as the reasonable future evolution for high-resolution audio.

1.1 Sample to Sample converters

The typical signal representation after a Sample-to-Sample conversion is given in picture 1.3 and the device capable of this encoding is the Nyquist converter. We can think of the output as an approximation of the original analog input. The standard amplitude for audio content is, in the analog domain, 1.24 Volt (referred normally as 0**dB**) so if we desire to increase the accuracy of our system we can think of increasing the number of levels representing our input swing in the digital domain: this is what has happened in the music industry since 1982 when the audio Compact Disc format has been presented; the CD uses 16 bits (i.e. 2^{16} levels) and a sampling frequency F_s equal to $44.1kHz$, assuming an audio band of $20.05kHz$. The original standard of LPCM has evolved in the last decade and also the sampling frequency has been increased in this type of conversion, reaching the actual standard of 24 bits and $192kHz$ that is used, for example, in the DVD-Audio standard.

1.2 Sample to Stream converters

The typical signal representation after a Sample-to-Stream conversion is given in figure 1.4. The output can vary only on few levels, normally on two;

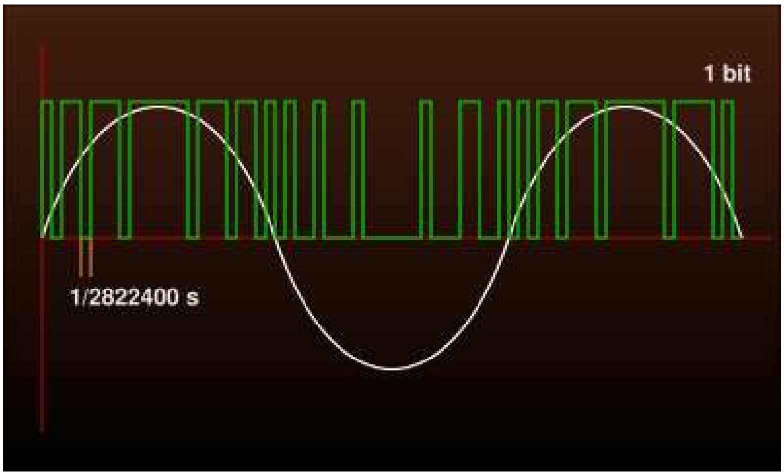


Figure 1.4: The amplitude of the input is translated into the density of $[-1,+1]$ symbols of the output

they could be $[-1,+1]$ or, already coded to fit an optical storage media, $[0,1]$. The higher sampling frequency is providing at least 64 samples in the same time slot where a Nyquist converter would have given as output a single one; we have consequently traded in time the accuracy that before has been reached in the amplitude domain. In a classical telecommunication view we could desire to transmit a larger word of information simply demultiplexing (or parallelizing) these short bit word into a **OSR**-long word. Indeed we could desire to preserve this digital signal in this primitive form and to storage it directly on a large capacity support; doing this we can hope to reduce the approximation due to the digital conversion only on the quantization error. In the real world a Nyquist converter is a Sample-to-Stream converter after which there is a **decimator** that shortens the length of the transmitted word with a mathematical transformation, the decimation.

In practice we can think of the sample-to-stream conversion as a modulation in which the amplitude of the analog input is converted into the density of $[-1,+1]$ symbols of the output. A Nyquist converter, on the other side, collapse this density of symbols in a single value that is chosen to represent the instantaneous amplitude of the signal. The single-bit type of conversion is of major interest in this thesis and it will be covered in detail in the following. Here is a list of figure of merit that apply to both systems:

- Storage capacity needed, or length of the recorded material
- Frequency response
- Dynamic range (Signal-to-Noise ratio)
- Number of available channels
- Lossless compression needed
- Content protection

In the next sections we will compare the two systems according to these features and we will further investigate on some aspects of sample-to-stream converters.

Parameter	DVD-Audio	SuperAudio CD	CD
Audio coding	16, 20 or 24-bit LPCM	1-b DSD	16-b LPCM
Sampling rate	44.1, 48, 88.2, 96, 176.4, or 192 kHz	2,822.4 kHz	44.1 kHz
Playback time	62-843 min*	70-80 min	74 min
Frequency response	DC-96 kHz	DC-100 kHz	DC-20 kHz
Dynamic range	Up to 144 dB	Over 120 dB	96 dB
Channels	1-6	2-6	2
Compression	Yes (MLP)	Yes (DST)	None
Content protection	Yes	Yes	No

Table 1.1: Comparison between high-resolution audio systems (* Playback time depends on resolution, sampling rate and number of channels)

1.3 Signal flow comparison

We eventually said that the Nyquist converter and a single bit converter are based on the same device, that produces a high frequency low bit output stream; then, the difference between the two is that the first one approximates with the decimation the density of $[+1, -1]$ symbols with a single value that is emitted every $1/F_s$ instant of time, while the second one leaves the original stream untouched and produces consequently a sample every $1/OSR \cdot F_s$ instant of time. The device used to produce the high frequency stream is normally a Sigma Delta Modulator (**SDM**) that is a hardware A/D converter realized in several different silicon technologies. This kind of ADCs have gained a major role in audio systems because of the high accuracy achievable with relative poor analog matching needs in the silicon technology; they are affordable and reliable. Now we want to analyze figure 1.2 and compare the two signal flows following the criteria expressed at the end of the last section. Starting with the storage capabilities we summarize some data in table 1.1:

As we can see the DVD-Audio is surely more versatile regarding the possible playback length and the number of channels even if the 1-bit DSD technology has been improved to handle till 6 channels. Both the new systems embed a form of content protection, meaning the capability to protect the stored audio to be copied without permission and both of them need a lossless compression algorithm to preserver the original quality of the signal but reducing the storage space needed. In each case a proprietary encoding system has been developed [5], [6] and it is transparent to the final user because the original musical content is reconstructed during the playback process in real time. We have to notice that the declared SNR is extremely high for both new proposed standard but in the real world there are no amplification systems capable of granting 144 dB of dynamic range to the final user.

Again we have to stress that the original signal has to pass through several more passages if we want to author a multibit audio file; the great advantage of the LPCM is, indeed, the presence of several recording-editing-and-mixing softwares (the so-called Digital Audio Workstation or **DAW**) that are now-a-days the *workhorse* in music production. These DAWs are including in a portable digital environment hundreds of possibilities: virtual instruments, dynamic filtering, equalization, reverberations and all the possible creative and

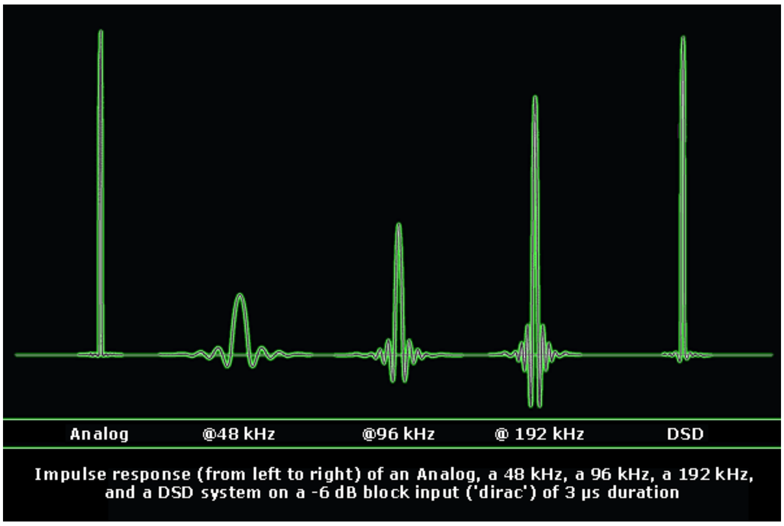


Figure 1.5: Comparison between different impulse responses for several LPCM systems and the Direct Stream Digital encoding

psychoacoustic filters that are almost always used by artists and producers. So the normal output of these software tools are multibit and Nyquist-Frequency sampled audio files. This to say that the so straightforward signal flow presented for the 1-bit audio encoding is, in fact, *following* the normal production flow.

Another great difference between the two systems is the need for the LPCM of a Digital-to-Analog converter (**DAC**) that is completely not necessary in the DSD system. In this second system the DAC role is played by a simple passive low-pass filter that, in many cases, could be included in the loudspeakers.

1.4 Impulse response comparison

For sure one advantage of the 1-bit oversampled encoding is the transient response clarity; in figure 1.5 we present a comparison between several LPCM systems and the DSD one. It's clear that the higher sampling frequency is reaching the same performance of the best analog tape recorders available. The ringing effect noticeable in the LPCM encoding is due to the presence of a steep low-pass filter that must be inserted before the decimator downsampler. In the 1-bit audio representation this step is absent and so there are softer constraints on the low-pass filter present in the whole system, that reduces to the first anti-aliasing filter before the SDM at the beginning of the converter chain.

1.5 Sigma Delta Modulation

In this section we investigate in some detail the Sigma Delta Modulator, already presented as fundamental device in both possible high-resolution audio systems introduced. Although the sigma-delta modulator was first introduced in 1962

[7], it did not gain importance until recent developments in digital Very Large Scale Integrated (**VLSI**) technologies which provide the practical means to implement the large digital signal processing circuitry. The increasing use of digital techniques in communication and audio application has also contributed to the recent interest in cost effective high precision A/D converters. A requirement of analog-to-digital (A/D) interfaces is compatibility with VLSI technology, in order to provide for monolithic integration of both the analog and digital sections on a single die. In fact Σ-Δ A/D converters are based on digital filtering techniques and almost 90% of the die is implemented in digital circuitry according to this constraint.

Conventional high-resolution A/D converters, such as successive approximation and flash type converters, operating at the Nyquist rate often do not make use of exceptionally high speeds achieved with a scaled VLSI technology. These Nyquist samplers require a complicated analog lowpass filter (anti-aliasing filter) to limit the maximum frequency input to the A/D, and sample-and-hold circuitry. On the other hand, Σ-Δ A/D converters use a low resolution A/D converter (1-bit quantizer), noise shaping, and a very high oversampling rate. The high resolution can be achieved by the decimation (sample-rate reduction) process or, as we previously said, collecting directly the 1-bit output stream. Moreover, since precise component matching or laser trimming is not needed for the high-resolution Σ-Δ A/D converters, they are very attractive for the implementation of complex monolithic systems that must incorporate both digital and analog functions. These features are somewhat opposite from the requirements of conventional converter architectures, which generally require a number of high precision devices. We will now introduce the working principle of the SDM and the concepts of quantization noise, noise shaping, oversampling, and decimation.

1.5.1 Quantization error

The process of converting an analog signal (which has infinite resolution by definition) into a finite resolution one introduces an error signal that depends on how the signal is being approximated. This quantization error is on the order of one least-significant-bit (**LSB**) in amplitude, and it is quite small compared to full-amplitude signals. However, as the input signal gets smaller, the quantization error becomes a larger portion of the total signal. When the input signal is sampled to obtain the sequence $x(n)$, each value is encoded using finite wordlengths of B-bits including the sign bit. Assuming the sequence is scaled such that $|x(n)| \leq 1$ for fractional number representation, the pertinent dynamic range is 2. Since the encoder employs B-bits, the number of levels available for quantization $x(n)$ is 2^B. The interval between successive levels, q, is therefore given by:

$$q = \frac{1}{2^{B-1}} \tag{1.1}$$

which is called the quantization step size. The sampled input value $x^*(t)$ is then rounded to the nearest integer present in the code. Then the A/D converter output is the sum of the actual sampled signal $x^*(t)$ and an error (quantization noise) component $e(n)$, that is:

$$x(n) = x^*(t) + e(n) \tag{1.2}$$

A common statement of the approximation is that the quantization error $e(n)$ has the following properties, which will be referred later on as the "input-independent additive white-noise approximation".

- $e(n)$ is statistically independent of the input signal $x(k)$ for all *n,k* (strong version) or $e(n)$ is uncorrelated with the input signal $x(n)$ (weak version);
- $e(n)$ is uniformly distributed in $[-\Delta/2, \Delta/2]$, where Δ is the quantization step;
- $e(n)$ is an independent identically distributed (i.i.d.) sequence (strong version) or $e(n)$ has a flat power spectral density (it is "white")(weak version);

when an input signal which is large compared to an LSB step, and the error term $e(n)$ is as defined before, then the noise power (variance), σ_e^2, can be found as [11]:

$$\sigma_e^2 = \mathbf{E}[e^2] = \frac{1}{q}\int_{\frac{-q}{2}}^{\frac{q}{2}} e^2 de = \frac{q^2}{12} = \frac{2^{-2B}}{3} \tag{1.3}$$

where **E** denotes the statistical expectation operator. Figure 1.6 shows the spectrum of the quantization noise. Since the noise power is spread over the entire frequency range equally, the level of the noise power spectral density can be expressed as:

$$N(f) = \frac{q^2}{12 \cdot \mathbf{F_s}} = \frac{2^{-2B}}{3 \cdot \mathbf{F_s}} \tag{1.4}$$

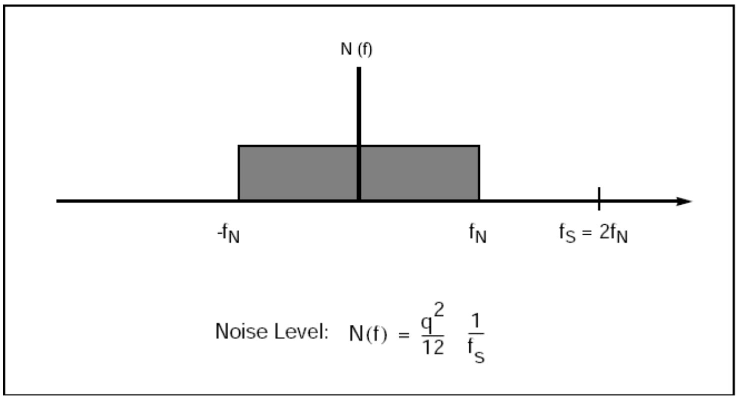


Figure 1.6: *Noise spectrum for a Nyquist converter*

These concepts apply in general for A/D converters. Indeed the quantization process in a Nyquist-rate A/D converter is generally different from that in an oversampling converter. While a Nyquist-rate A/D converter performs the quantization in a single sampling interval to the full precision of the converter, an oversampling converter generally uses a sequence of coarsely quantized data at the input oversampling rate of $F_s = OSR{\cdot}F_{sNyquist}$ followed by a digital-domain decimation process (or left untouched as seen in the DSD case) to compute a more precise estimate for the analog input at the lower output sampling rate, $F_{sNyquist}$, that is the same as used by an equivalent Nyquist converter.

1.5.2 Oversampling and Decimation basics

Regardless of the quantization process oversampling has immediate benefits for the anti-aliasing filter. To illustrate this point, consider a typical digital audio application using a Nyquist sampler and then using a two times oversampling approach: The data samples from Nyquist-rate converters are taken at a rate of at least twice the highest signal frequency of interest. For example, a $48kHz$ sampling rate allows signals up to $24kHz$ to pass without aliasing, but because of practical circuit limitation, the highest frequency that passes is actually about $22kHz$. Also, the anti-aliasing filter in Nyquist A/D converters requires a flat response with no phase distortion over the frequency band of interest (e.g., $20kHz$ in digital audio applications). To prevent signal distortion due to aliasing, all signals above $24kHz$ for a $48kHz$ sampling rate must be attenuated by at least $96dB$ for 16 bits of dynamic resolution (thinking for example of a CD format application).

These requirements are tough to meet with an analog low-pass filter. Figure 1.7(a) shows the required analog anti-aliasing filter response, while figure 1.7(b) shows the digital domain frequency spectrum of the signal being sampled at $48kHz$. Now consider the same audio signal sampled at $2 \cdot F_s$, $96kHz$. The anti-aliasing filter only needs to eliminate signals above $74kHz$, while the filter has flat response up to $22kHz$. This is a much easier filter to build because the transition band can be $52kHz(22kto74kHz)$ to reach the -96 dB point. The smaller ringing effect in the time domain due to the gradual slope of the transition band has been illustrated in the section presenting and comparing the impulse responses. However, since the final sampling rate is $48kHz$, a sample rate reduction filter, commonly called a decimation filter, is required but it is implemented in the digital domain, as opposed to anti-aliasing filters which are implemented with analog circuitry. Figure 1.7 illustrate in the (d) and (e) part the analog anti-aliasing filter requirement and the digital-domain frequency response, respectively. The spectrum of a required digital decimation filter is shown in figure 1.7(f).

This two-times oversampling structure can be extended to OSR times oversampling converters. Figure 1.8(a) shows the frequency response of a general anti-aliasing filter for OSR times oversamplers, while the spectra of overall quantization noise level and baseband noise level after the digital decimation filter is illustrated in figure 1.8(b).

Since a full precision quantizer was assumed, the total noise power for oversampling converters and for the Nyquist sampler is the same. However, the

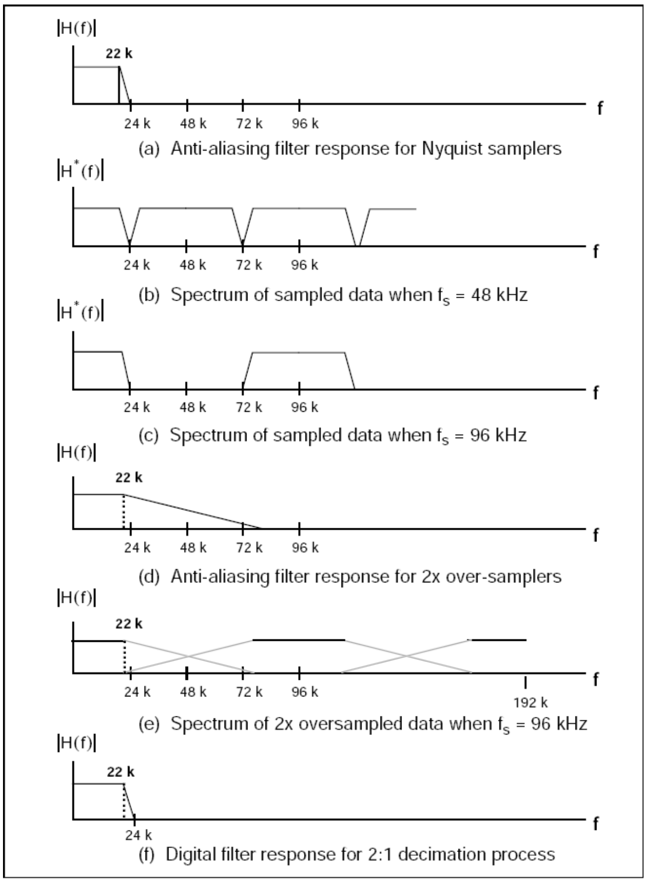

|H(f)|
22 k
24 k
48 k
72 k
96 k
f
(a) Anti-aliasing filter response for Nyquist samplers
|H*(f)|
(b) Spectrum of sampled data when f_S = 48 kHz
(c) Spectrum of sampled data when f_S = 96 kHz
(d) Anti-aliasing filter response for 2x over-samplers
192 k
(e) Spectrum of 2x oversampled data when f_S = 96 kHz
(f) Digital filter response for 2:1 decimation process

Figure 1.7: *Comparison between Nyquist and* $2X$ *Oversampling converter spectra*

percentage of this noise that is in the bandwidth of interest, the baseband noise power N_B is:

$$N_B = \int_{-f_B}^{f_B} f\, df = \frac{2f_B}{\mathbf{F_s}} \frac{q^2}{12} \tag{1.5}$$

where f_B is band of interest for the signal spectrum, which is much smaller (especially when F_s is much larger than f_B) than the noise power of Nyquist samplers described in equation 1.4.

The transition band of the anti-aliasing filter of an oversampled A/D converter is much wider than its passband, because anti-aliasing protection is required only for frequency bands between $OSR \cdot F_s - f_B$ and $OSR \cdot F_s + f_B$, when $N = 1, 2, \cdots$, as shown in figure 1.7(b). Since the complexity of the filter is a strong function of the ratio of the width of the transition band to the width of the passband, oversampled converters require considerably simpler anti-aliasing filters than Nyquist rate converters with similar performance. For example, with $N = 64$, a simple RC lowpass filter at the converter analog input is often sufficient as illustrated in figure 1.7(a).

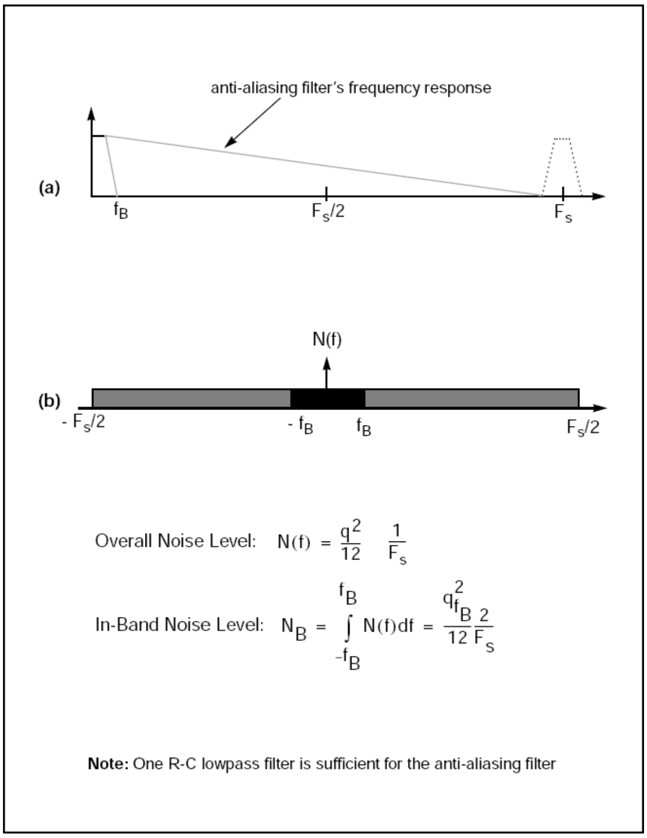

anti-aliasing filter's frequency response
(a)
f_B
$F_s/2$
F_s
N(f)
(b)
$- F_s/2$
$- f_B$
f_B
$F_s/2$
Overall Noise Level: $N(f) = \frac{q^2}{12} \frac{1}{F_s}$
In-Band Noise Level: $N_B = \int_{-f_B}^{f_B} N(f)df = \frac{q^2 f_B}{12} \frac{2}{F_s}$
Note: One R-C lowpass filter is sufficient for the anti-aliasing filter

Figure 1.8: *Anti-Aliasing filter response and noise spectrum of Oversampling A/D Converters*

The benefit of oversampling is more than an economical anti-aliasing filter. The decimation process can be used to provide increased resolution. To see how this is possible conceptually, refer to figure 1.9, which shows an example of 16 : 1 decimation process with 1-bit input samples. Although the input data resolution is only 1-bit (0 or 1), the averaging method (decimation) yields more resolution (4 bits $[2^4 = 16]$) through reducing the sampling rate by 16 : 1. Of course, the price to be paid is high speed sampling at the input speed is exchanged for resolution.

1.6 Analysis of Sigma-Delta modulation in the *Z-Transform* domain

We present now the analysis in the *Z-Transform* domain of the presented structures. The intention is to enrich the linear model presented in the previous section and translate it into the classical digital design environment. In fact when we find a z^{-1} in a block scheme we can think of it in a digital architec-

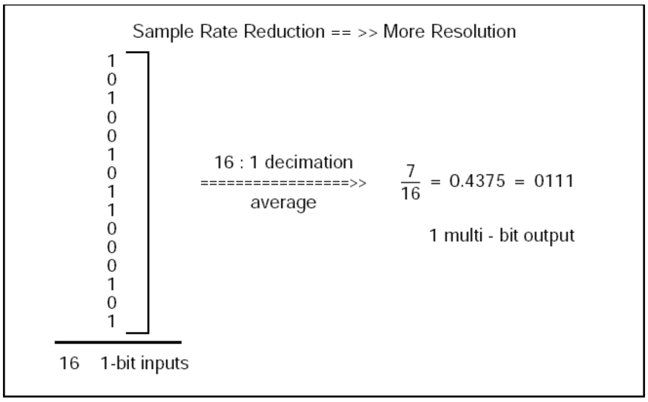


Figure 1.9: *Example of decimation process*

ture as a delay block. Consider the first-order loop shown in figure 1.10. The z-domain transfer function of an integrator is denoted by $I(z)$ and the 1-bit quantizer is modeled as an additive noise source. Standard discrete-time signal analysis yields:

$$Y(z) = Q(z) + I(z) \cdot [X(z) - z^1 \cdot Y(z)] \tag{1.6}$$

and can be solved for $Y(z)$ as:

$$Y(z) = X(z) \cdot \frac{I(z)}{1 + I(z) \cdot z^{-1}} + Q(z) \cdot \frac{1}{1 + I(z) \cdot z^{-1}} \tag{1.7}$$

where we defined the ideal integrator as:

$$I(z) = \frac{1}{1 - z^{-1}} \tag{1.8}$$

Then we can simplify and obtain:

$$Y(z) = X(z) + (1 - z^{-1}) \cdot Q(z) \tag{1.9}$$

Since the quantization noise is assumed to be white, the differentiator $I(z) = (1 - z^{-1})$ shown in equation 1.9 doubles the power of quantized noise. However, the error has been pushed towards high frequencies due to the differentiator factor $(1-z^{-1})$. Therefore, provided that the analog input signal to the modulator, $x(t)$, is oversampled, the high-frequency quantization noise can be removed by digital lowpass filters without affecting the input signal characteristics residing in baseband. This lowpass filtering is part of the decimation process. That is, after the digital decimation filtering processes, the output signal has only the frequency components from $0Hz$ to f_B.

In figure 1.11 we presented the spectrum of a first-order Σ-Δ noise shaper described in figure 1.10. The baseband (up to f_B) noise of the SDM appears to be much smaller than Nyquist samplers or delta modulators. However, for

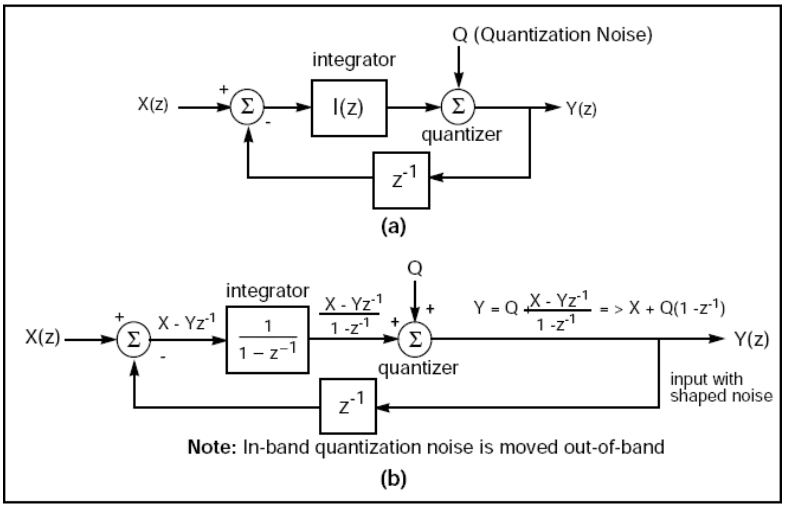


Figure 1.10: *Z-Domain analysis of first-order noise-shaper*

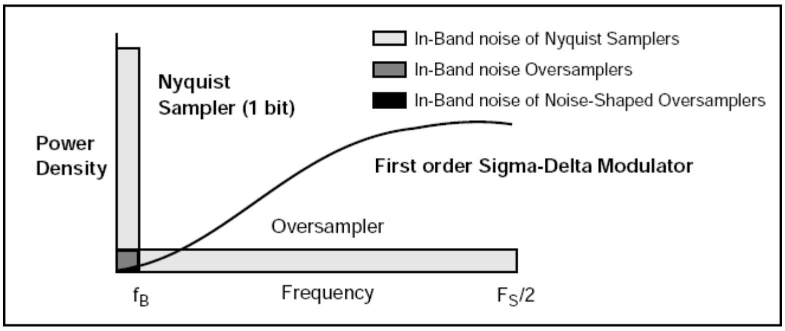


Figure 1.11: *Spectrum of a first-order* Σ-Δ *noise shaper*

the first order modulator discussed, the baseband noise can not reach below the $-96dB$ signal-to-noise ratio needed for 16-bit A/D converters we want to use, for example in a CD format audio system.

So higher order cascaded (feed-forward) Σ-Δ have been introduced and implemented [12]. The block diagrams of second and third order SDMs are shown in figure 1.12. Since these cascaded structures use a noise feed-forward scheme, the system is always stable and the analysis is simpler compared to the second-order feedback SDMs [13, 14, 15] and higher order interpolating coders with feedback loops [16, 17]. When multiple first-order Σ-Δ loops are cascaded to obtain higher order modulators, the signal that is passed to the successive loop is the error term from the current loop. This error is the difference between the integrator output and the quantization output. In these schemes the "Bit manipulation node is a functional block that gives a 1-bit output stream.

If the input signals to the second and third stage Σ-Δ loops are Q_1 and Q_2,

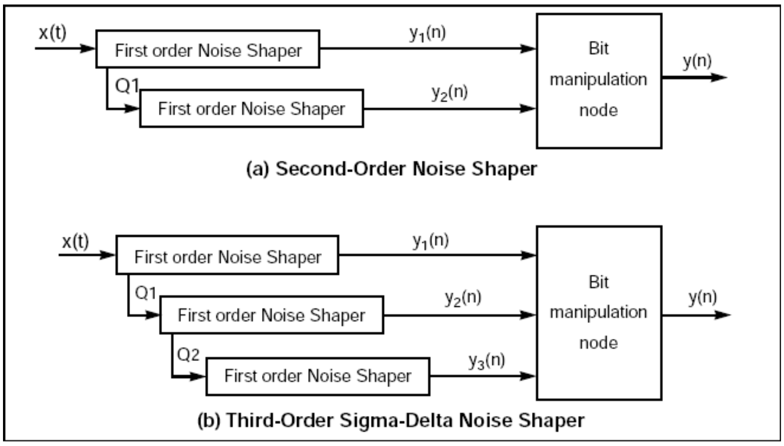


Figure 1.12: *Second-order and third-order* Σ-Δ *noise shapers: the "Bit manipulation node" is functional block that gives a 1-bit output and is again implemented as a* Σ-Δ

respectively, the quantization output for the second order SDM is given by:

$$Y_2(z) = Q_1(z) + (1 - z^{-1}) \cdot Q_2(z) \tag{1.10}$$

which, for the second order output depicted in figure 1.12, yields:

$$Y(z) = X(z) + (1 - z^{-1})^2 \cdot Q_2(z) \tag{1.11}$$

Similarly for the third-order SDM we will obtain:

$$Y(z) = X(z) + (1 - z^{-1})^3 \cdot Q_3(z) \tag{1.12}$$

where Q_3 is the quantization noise from the third Σ-Δ loop. Essentially, the noise shaping function in a Σ-Δ is the inverse of the transfer function of the filter $(1 - z^{-1})$ in the forward path of the modulator. A filter with higher gain at low frequencies is expected to provide better baseband attenuation for the noise signal. Therefore, modulators with more than one Σ-Δ loop such as the third-order system shown in figure 1.12(b), perform a higher order difference operation of the error produced by the quantizer and thus stronger attenuation at low frequencies for the quantization noise signal. The noise shaping functions of second-order and third-order modulators are compared to that of a first-order system in figure 1.13. The baseband quantization error power for the third-order system is clearly smaller than for the first-order modulator.

The above analysis can be extended to yield quantitative results for the resolution of Σ-Δ, provided that the spectral distribution of the quantization error $e(n)$ is known. It has been shown that the error generated by a scalar quantizer with quantization levels equally spaced by q is uncorrelated, assuming that the number of quantization levels is large and the quantized signal is active [18, 19] (i.e. the signal is not DC). This result is not rigorously applicable to

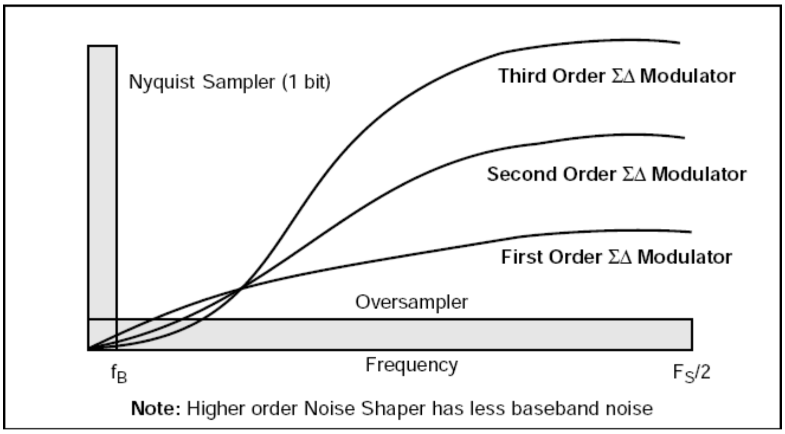


Figure 1.13: *Multi-order* Σ-Δ *noise shapers*

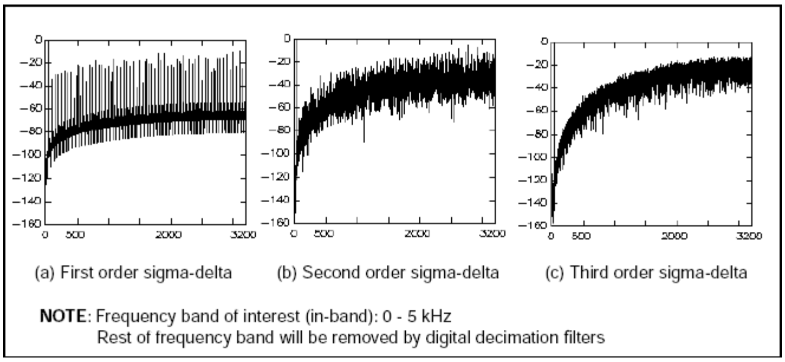


Figure 1.14: *Spectra of three* Σ-Δ *noise shapers*

SDMs, however, because Σ-Δ quantizers only have two levels. Hence, the noise and signal are somewhat correlated and signal dependent. Nevertheless, analysis based on these assumptions yields correct results in many cases. Often these analytical results provide a more intuitive interpretation for the operation of the modulator than those obtained from computer simulations. Anyway the latter must always be used to demonstrate the correctness of the analytical result in a particular case. Normally simulations provide useful spectra that can show several limits and properties of the converter, as depicted in figure 1.14.

Chapter 2

Authoring audio

While in the previous chapter we presented the general signal flow used in audio production, in the following we will focus our attention on the authoring of digital audio content. In the production chain that transforms an artistic idea into a commercial good we can define this step as a substantial format conversion for the information content.

Indeed, we don't have to confuse authoring with the mastering of music or audio; this second is still an artistically demanding part of the business and it has as source the mixed material and as output a digital or analog format [20]. The authoring of audio content, instead, has as source the audio master in both its analog or digital format, and it has as output a fixed standard format, depending on the market where the music should be employed. Now-a-days there is the common CD-format and also other two high-resolution format, that are DVD-Audio and SuperAudio CD: none of the two is still affirmed as the new high-quality standard for music and none is as widely spread as the CD. Anyway, while the DVD-Audio uses the same representation (LPCM) used in the CD and it does not need any different approach from it, the authoring of a Super Audio CD is capable of exploiting the superior characteristics of the Sigma Delta conversion that we already introduced. In this chapter we will present the state of the art as developed by Philips from 2001 and 2003 concerning the authoring algorithms to convert a classical LPCM music stream into a 1-bit stream. The fundamental idea is to store the 1-bit stream directly on the optical support; this stream will be then amplified as is by the power unit and then converted only in the end by the mean of a simple analog filter, thus requiring a simpler amplification system.

So we have to think that our master is or in analog form (an analog tape) or in digital form (a 24 bit $192kHz$ LPCM); we have to convert it into a 1-bit stream using a hardware, in the case of an analog tape, or software, in the case of LPCM, Sigma Delta modulator. The technique presented here apply for software converters, so it can be grouped under the name of Digital-to-Digital conversion (**D/D**). A note should be added: while until 5-6 years ago music was mostly sold in CD-format (together with older format as Long Playing and Cassettes) in the recent years the presence of on-line music resellers has totally changed the market landscape. In this changed field we cannot forget to mention the **mp3** format. Obviously the mp3 format is a compressed format and it is directly produced from an LPCM file with a lossy compression algorithm;

anyway if we had to produce an mp3 format for our output we can generally present the compression as a form of D/D conversion that respects its standard.

2.1 Trellis-based SDM

We can think of a Digital-to-Digital Sigma Delta modulator as the same device presented in the previous chapter, where instead of physical circuits and electric signals there are software functions representing different building blocks and real valued variables. In this abstract model, for example, a delay line is translated to a memory cell containing a state, a real value approximated with a finite precision, i.e. a *Double*, that is updated with periodic frequency. So here again we can present a block scheme for a SDM and from that we can start presenting the different functions needed to implement the system.

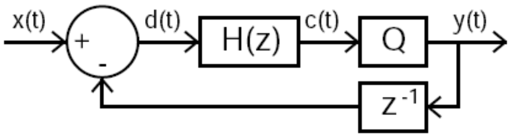


Figure 2.1: General block scheme for a software SDM

Figure 2.1 shows a typical model for a 1-bit SDM. The input signal $x(n)$ is converted to a single-bit output signal $y(n)$. In the model already introduced signal $d(n)$ is the error signal, including noise and harmonic distortion and $c(n)$ is the frequency weighted error signal [21]. The lowpass transfer function $H(z)$ is responsible for the noise-shaping effect. Signals here are presented as time continuous as if we had to convert an analog master tape. In fact our input signal is a digital multi-bit signal that we can assimilate to an analog one due to the extremely high resolution (24 bits) and sampling frequency ($F_s = 192kHz$). For simulations we can even synthesize sinusoids in *double* (32 bits) precision. The Q-block is the decision-making unit for the output sequence. In case of a conventional SDM, Q is a 1-bit quantizer (a comparator) with the following definition:

$$y(t) = \begin{cases} +1 & \text{if } c(n) \geq 0 \\ -1 & \text{if } c(n) < 0 \end{cases}$$

According to this implementation, a delay z^{-1} block in the feedback path of the loop is necessary, otherwise the adder has to know the value of $y(n)$ before $y(n)$ is actually determined. As a result of this definition, the SDM calculates the output as a function of previous input and output values only. As depicted in fig. 2.1 $d(n)$ is the error signal and $c(n)$ is the filtered version of $c(n)$. The weight function in a normal Σ-Δ is a low pass filter used for the noise shaping. The new output is chosen such that the filtered error $c(n)$ is small, but using a single bit quantizer to make this decision yields to a coarse approximation, because the output can only be $c(n) = +1$ if $c(n)$ is larger than 0 or $c(n) = -1$ if $c(n)$ is below the 0 trashold. However, minimizing the filtered error this way at each time independently is not necessarily the best solution. This is the case, for example, when a SDM starts to oscillate. Although the error is minimized at every point in time, the total weighted noise power in the output signal is far from minimized and the global output is no more describing correctly the input.

Ideal Case: an exponentially growing computational problem

So we presented a software converter that instantaneously takes a decision about the next encoded (converted) symbol evaluating the previous input and output. Indeed we can think of the problem of encoding an information message with a stream of $[+1, -1]$ symbols from another point of view: we have as input a signal $x(n)$ and we have to translate it into a signal $y(n)$ constrained to have only two possibles values, minimizing the total error. Ideally we should wait till the end of the message and then evaluate all the possible sequences of $[+1, -1]$, our alphabet symbols, that differs for a single bit. This means that we should evaluate with a previously defined cost function 2^N possible sequences, where the message needs N symbols to be encoded; unfortunately in the case of musical content this means an incredibly high value for N, due to the high sampling frequency required (sometimes we even don't know in advance the length of the recorded material!). The problem as stated before is a typical example of an exponentially growing computational problem because it grows exponentially with the length of the message.

So, instead of using a coarse comparator or the ideal decisor now described, we can implement a third decision-making unit. Figure 2.2 shows the general relation between the signals in a Noise Shaping Device (**NSD**), without showing the decision-making unit. The ideal NSD that lead to the NP-Hard problem is defined to be:

Definition (Ideal NSD) 1 *The bitstream generator that creates output stream y(n) such that the power in the frequency-weighted error signal c(n) is minimal.*

In other words, defining the power of the representation error P_E as:

$$P_E = \lim_{k\to\infty} \frac{1}{2k+1} \sum_{n=-k}^{n=k} [c(n)]^2 \tag{2.1}$$

the ideal NSD has to minimize P_E, defined below in equation 2.1, by choosing the correct symbols $y(n)$.

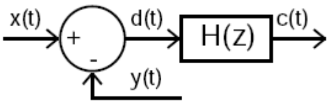


Figure 2.2: General block scheme for a Noise Shaping Device

However, as we said before, we have to evaluate the 2.1 for all the 2^N possible output sequences to globally minimize this function. Besides, the function is defined over an infinite time domain. An intermediate solution of this problem is given by the Viterbi algorithm, that could be implemented in a Trellis architecture [22]; we should stress that this is only an approximation of the ideal NSD, but solves the two inherent problems: infinite time domain and innumerable possible sequences.

The infinite time domain is reduced to the time span between 0 (starting time) and t (current time). A new measure, approximating the original power function (equation 2.1), is a cost function, defined as:

Definition (Cost Function) 1

$$C_{\omega N}(t) = \sum_{\tau=0}^{\tau=t} [c(\tau)]^2 \tag{2.2}$$

The second problem of the ideal NSD, calculating an infinite number of output sequences, is solved by considering a certain amount of *candidate output sequences* only (referred also as *output paths* or simply *paths*). The actual number of candidates is determined by an independent parameter N, that is called *Trellis order*. A candidate output sequence is a sequence of possible output symbols $\in \{-1, +1\}$ defined for time window of length t, where τ with $0 \leq \tau \leq t$ is the current time. The last N symbols are different for all candidates. Consequently, there are exactly 2^N paths. For all of these, the cost function (equation 2.2) is evaluated, where N is the sequence of the last N bits. A smaller value of the cost function indicates a better approximation to the input signal $x(n)$ meaning a smaller (local) value for the power of representation error. Thus, minimizing the cost function over a time window of length t coincides with finding the (sub)optimal output stream.

Real Case: an approximated approach with the Viterbi algorithm

Suppose that the information about all (2^N) candidates at time $t-1$ is known. [1] The information for each path consists of the history of the output sequence, the current state inside the loop-filter ($H(z)$) and the value for the cost function 2.2. Based on this situation, the Viterbi algorithm is adopted to minimize the costs for the candidates at the next time instant t. Indeed in our new approximation we have a time window of length t in which, for each path, we collected the possible output symbols; at every instant a new possible output symbol σ is "carried in" our time window, while an "old" symbol ς is chosen to be the possible output for that single candidate. Figure 2.3 (left) shows the state diagram for Trellis order $N = 2$. There are $2^N = 4$ paths, indicated by the last two output symbols. The origination of the new possible paths from the old candidates is dependent on the new output symbol, as indicated by the paths between the states. The right side of the picture shows how the new candidates at time t are chosen from the old ones at time $t - 1$. The following definitions were used:

[1] In this section the time index will be written as t for an easier reading, due to the presence of several other subscript notation, while the system is still a discrete-time one

σ or ς	$\in \{-1,+1\}$, single output bit
ω_{N-1}	sequence of N-1 output bits
$c_{\varsigma\omega N-1\sigma}(t)$	filter output after processing of σ when candidate $\varsigma\omega_{N-1}$ at $t-1$ is used.

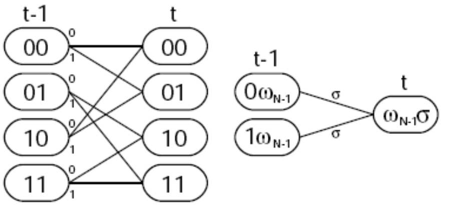


Figure 2.3: Origination of new candidates (time t) from old candidates (time $t-1$). Complete state diagram for $2^N = 4$ candidates (left), and the general case (right). For clarity, the signal level -1 is represented by the symbol "0" inside all figures.

Out of the two possible paths for every new candidate, the path with the smallest total costs is chosen:

$$C_{Path1} = c_{0\omega N-1\sigma}(t-1) + [c_{0\omega N-1\sigma}(t)]^2 \tag{2.3}$$

$$C_{Path2} = c_{1\omega N-1\sigma}(t-1) + [c_{1\omega N-1\sigma}(t)]^2 \tag{2.4}$$

$$C_{\varsigma\omega N-1\sigma}(t) = \begin{cases} C_{Path1} & \text{if } C_{Path1} \leq C_{Path2} \\ C_{Path2} & \text{if } C_{Path2} \leq C_{Path1} \end{cases} \tag{2.5}$$

After the determination of the cheapest path for each candidate, the new value for the cost function, the new filter state and the new output symbol are known, again for each candidate. The total output sequence for a new candidate is given by the output sequence of the previous state followed by σ, that is the next output symbol. The definition of the Trellis structure is that the output sequences for all 2^N candidates terminate with different symbols, thus there are 2^N different output sequences as well. Fortunately, although the sequences diverge at time t, the output sequences tend to converge to a single solution for $t \rightarrow -\infty$. An example is depicted in figure 2.4: the bold red lines show the paths chosen by the Viterbi algorithm. Between time $t-1$ and t, the four candidates have a different path, but before $t-3$ the paths are equal for all the actual candidates.

After some time, the paths between $t-1$ and t will converge to a unique solution as well. An easy example explains how this is possible: suppose that

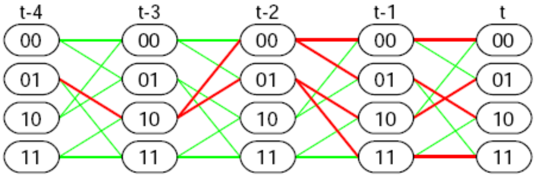


Figure 2.4: Convergence of paths: the bold red lines show the origination of the four candidates. The different candidates terminate with different output symbols, but in history ($t \rightarrow -\infty$) the output sequences converge to a single solution.

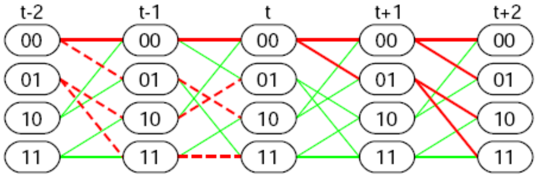


Figure 2.5: Convergence of paths: example when candidate "00" at time t is cheaper compared to the other candidates at time t. Two sample periods later, all paths converged up to time t.

the value of the cost function for the first candidate ("00") is much lower than for the other candidates. In that case, it is likely that the future paths start from candidate "00". Figure 2.5 shows the result two sample periods later. Assuming that all paths converge in the past, output symbol $y(t - t_{lat})$ can be determined at time t unambiguously when the value of t_{lat}, also indicated with *Latency*, is large enough. However, the correct value for the Latency parameter has to be determined experimentally [23]. The influence of choosing Latency too small will be discussed in detail in the following sections.

2.2 Simulations results

There are some interesting results achievable using a Trellis Σ-Δ; the more important ones are a better linearity and an increased capability of handling larger amplitude input signal. The simulation results presented here are obtained using a sinusoidal test tone at $1kHz$ with amplitude of $0dB$; we must notice that the definition of $0dB$ in the SACD standard is different from the definition we gave in the first chapter. SACD sets the $0dB$ level at $6dB$ below the theoretical full-scale DSD signal, and prohibits peaks above $+3dB$. This is done because the classic Σ-Δ ADCs can become unstable when the input amplitude rises above $0.5Volt$ and depending on the single internal architecture this phenomena can happen at different amplitude; so the standard has been related to the maximum amplitude where almost every commercial product is still strongly stable. Pictures 2.6 and 2.7 show respectively a zoom in the audio band from $2kHz$ to $10KHz$ to show the noise floor above the sinusoidal test tone and the power of the first seven harmonic components. In both pictures we can appreciate the increased linearity of the Trellis structure applied to SDMs.

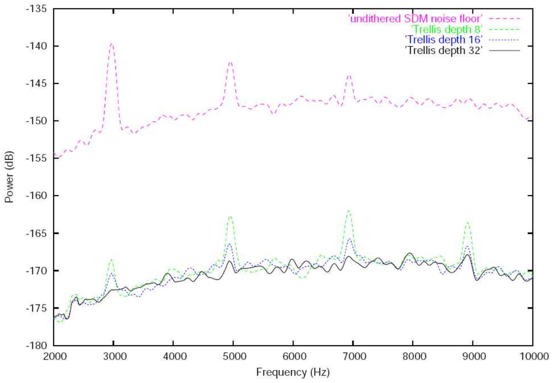


Figure 2.6: Zoom-in on 2-10 kHz range of power spectra for $0dB$ SACD $1kHz$ input. Spectra have been coherently averaged 128 times and power averaged 64 times. Also shown is the noise floor of an equivalent *undithered* SDM. Loop filter is 5^{Th} order, sample-rate $2.8MHz$, corner frequency $105kHz$, notches at 15 and $20kHz$.

Figure 2.8 displays the maximum input amplitude that can be applied to a Trellis SDM (5^{Th} order, sample-rate $2.8MHz$, corner frequency $105kHz$, notches at 15 and $20kHz$) that still results in stable operation. Going from 1 to 128 paths (i.e. *Trellis Order N* going from 0 to 7) results in an increase of maximum input amplitude of more than 30%. Going to 8 paths increases the maximum input amplitude from 0.66 to 0.81, an increase of almost 23% already. Consequently this increased stability can also be used to perform a more aggressive noise-shaping.

To create Figure 2.9 the input has been kept constant at 0 dB SACD, while the maximum stable corner frequency of the loop filter is examined as function of the number of Trellis paths. Again the 5^{Th} order loop filter contains two notches at 15 and $20kHz$. For 1 Trellis path the maximum corner frequency that results in stable operation is $137kHz$. Going to 8 paths increases this limit to $217kHz$, while 128 paths results in a maximum stable corner frequency of $445kHz$. This feature is interesting because of the much larger audio band of the resulting systems or, fixed the usual audio band, the larger transition band for the anti-alias filter that could be implemented.

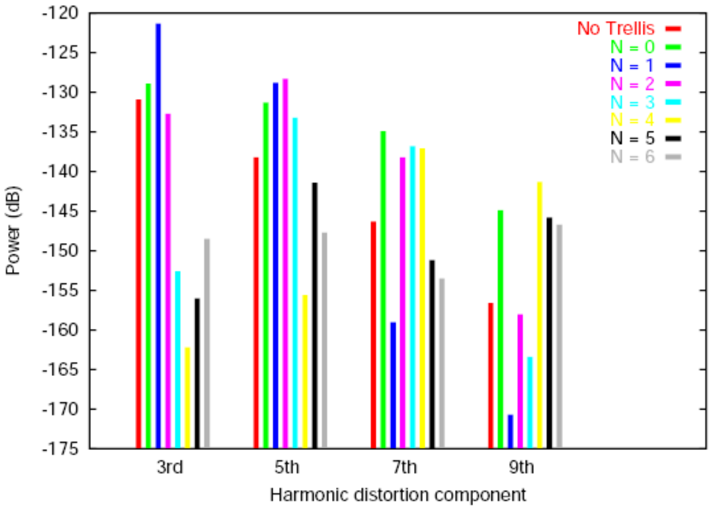


Figure 2.7: Power in the 3^{rd}, 5^{Th}, 7^{Th} and 9^{Th} harmonic distortion component as a function of the Trellis order. The power is measured relative to the signal power, which is 0 dB SACD

2.3 The Latency problem

Finally, the influence of the chosen Latency is discussed. In the second section of this chapter, it was stated that a certain delay is necessary before the output symbol is determined: t_lat or *Latency*. When Latency is chosen large enough, the determination of the output symbol is unambiguous, otherwise, there is a chance of producing the wrong output symbol.

Figure 2.4 is used as an example. If $t_{lat} = 3$, the output symbol for $t - 3$ is determined at time t. For all candidates at time t, the path value at time $t - 3$ is equal to "0", thus the output symbol can be determined unambiguously. When $t_{lat} = 2$ is used, the path value at time $t - 2$ is different for the different candidates. Thus, depending on the candidate that is used to determine the output symbol, a different output symbol can be produced. If candidate "00" or "10" is used at time t to choose the output symbol for time $t - 2$, a "0" will be produced. When candidate "01" or "11" is used, the output symbol will become "1". The influence of the possible misjudgment is visible in figure 2.10.

Here the output power spectrum of a 5^{Th} order Butterworth filter with $0dB$ SACD input signal in combination with a 5^{Th} order Trellis is shown. The spectrum is plotted for four different values of Latency: 70, 80, 85 and 90 samples delay. In case of Latency ≥ 90, the delay is long enough to produce the correct (unique) output. When Latency < 90 is used, a flat spectrum in the base band appears. This is the effect of choosing an output symbol when the paths for the different candidates did not yet converge to the correct suboptimal solution. This signal component is usually described with the term *truncation noise*.

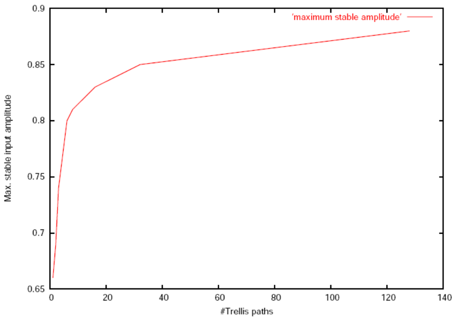


Figure 2.8: Maximum stable input amplitude versus number of Trellis paths. Loop filter is 5^{Th} order, sample-rate $2.8MHz$, corner frequency $105kHz$, notches at 15 and $20kHz$.

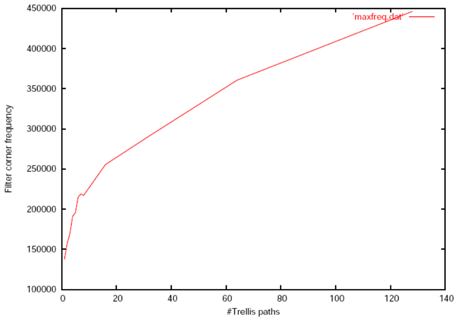


Figure 2.9: Maximum stable filter corner frequency versus number of Trellis paths. Test tone is a 0 dB SACD sine wave. Loop filter is 5^{Th} order, sample-rate $2.8MHz$, corner frequency $105kHz$, notches at 15 and $20kHz$.

The power spectral density of the truncation noise in the base band is almost frequency independent. Consequently, it can be modelled with a constant value: PSD_{TN}. Small values of Latency result in an increasing PSD_{TN}. Figure 2.11 shows PSD_{TN} as a function of *Latency*. It affirms that the power density of the truncation noise is relatively stable for $50 \leq$ Latency ≤ 85, but when Latency increases from 85 to 90, the noise suddenly disappears. The source behind truncation noise is *path truncation* (or simply *truncation*): it arises, because the output symbol is determined when not all paths converged to the final solution.

The output symbol is determined using one of the different paths that still exist after the Latency delay. Sometimes, the output symbol determination switches to another path. This jump to another path (or truncation of the previously used path) gives an instantaneous error, resulting in truncation noise. This instantaneous error should be visible as short clicks in the time domain, and is verified with another simulation.

To investigate the influence of the truncation noise in the time domain, a $1kHz$ sine wave was synthesized in a normal audio file (16 bits quantization, $F_s = 44.1kHz$). This file was up-sampled to $64F_s$, the sample frequency of

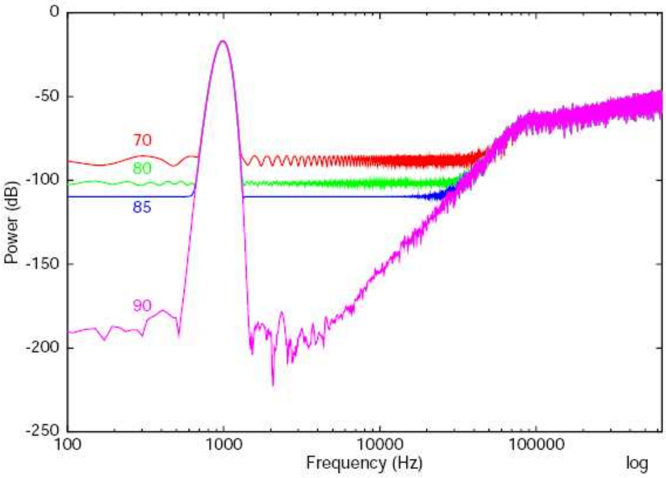


Figure 2.10: Output power spectrum for different values of Latency (70, 80, 85, 90).

SACD. The data after up-sampling is still multi-bit, in this case 32 bit. Then, this data is processed through a Trellis SDM (of which the Latency is chosen too small on purpose), and downsampled to the original audio format (16 bit, $44.1kHz$). Figure 2.12 plots a part of the audio file. Besides the original $1kHz$ signal, three short peaks are visible in this picture, due to truncation.

The time and frequency domain simulations are in accordance with each other, as the Fourier transform of an impulse signal in the time domain, is a flat power density spectrum in the frequency domain. When Latency is increasing, the peaks in the time domain will occur less frequently, reducing the noise level in the power spectrum. All peaks (even the small ones) are very audible, and should be definitely avoided. In practice, we would like to choose t_{Lat} such that there is no problem in the determination of output symbols. Simulations are necessary to choose the minimum needed value, checking the output power spectrum to verify whether a certain Latency is sufficient. Some problems are displayed in figure 2.13. The uppermost picture shows that the minimum value for Latency is not only dependent on the Trellis order, but also on the used input signal. The lower figure shows that for very small input signals, the Latency increases significantly.

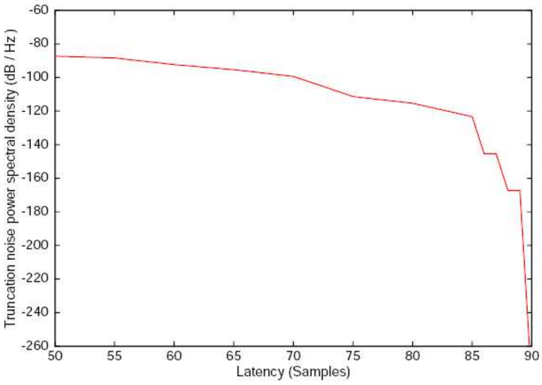


Figure 2.11: Power spectral density of the truncation noise in the output signal due to too small value of Latency.

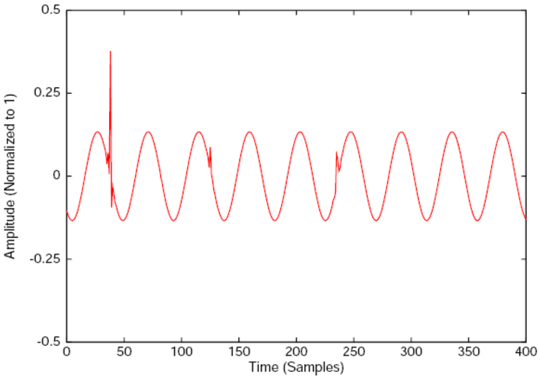


Figure 2.12: Time domain signal after downsampling to $44.1kHz$. The short peaks are the effect of truncation noise.

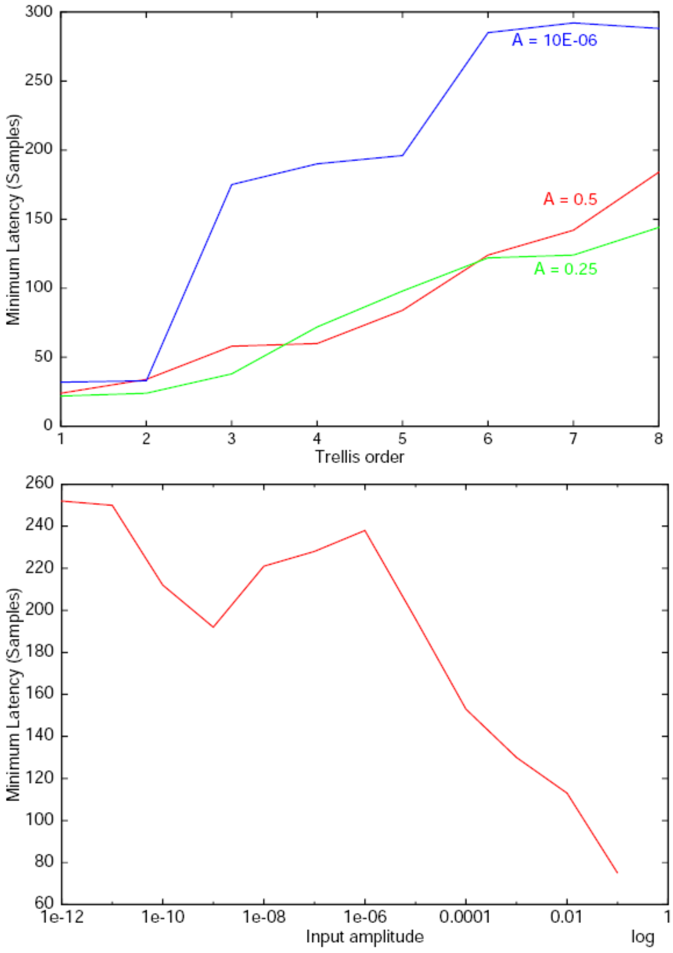


Figure 2.13: Minimum needed value for Latency for unambiguous determination of the output stream: as a function of the used Trellis order N for three different values of the input amplitude A (top), and as a function of the input amplitude for Trellis order $N = 5$ (bottom).

Chapter 3

Power loss causes in a Class-D amplifier architecture

In this chapter we will present the user front-end aimed at reproducing digital audio and music. Since the introduction of digital audio this role in the signal chain is covered by a D/A converter and an analog amplifier. The amplifier is actually driving the actuators, i.e. the loudspeakers; those are bringing back trough the air the information content previously stored in a digital form. So this part of the production chain is responsible of converting back (or decoding) the digital signal into an analog one and then amplify it. Even in this field there are several market consideration that could be done, first of all the amount of amplification needed by the final user, and then of course the quality of the system in terms of Signal-to-Noise Ratio (**SNR**) and Total Harmonic Distortion (**THD**).

Talking about amplification is broad and complex and like many other engineering fields there are several constraints and trade-off we have to deal with. Again the new technology that we presented since now had and has a strong impact on the designs of amplifiers and viceversa new amplifier architectures can suggest new ideas for digital encoding systems. It is usefull to compare the already presented systems tailored on Nyquist converters or upon Σ-Δ converters. A system using LPCM encoding, and thus Nyquist converters, produces for example a 24 bit $192kHz$ stream that must be converted again into an analog signal and then amplified trough a high quality analog amp.

Instead a system using the DSD presents a choice to the designer: we can convert the digital signal into an analog one and treat it exactly like the previous case; in fact we can think of amplifying directly the square wave made by the flow of the $[-1, +1]$ symbols representing the signal in the digital domain. This way we are amplifying both the signal *and* the noise, that is a high frequency component of this large band stream, but then we can filter it with a passive component net and get back the original analog signal already amplified. This second choice is a hybrid between an amplifier and a D/A converter. From an economical point of view we can see that while the first case still needs a high quality D/A process in order to fully exploit the 24 bits of dynamic resolution used by the LPCM encoding, the DSD needs a simple low pass filter to separate the audio signal from the quantization noise that is pushed far above the audio band by the noise shaper itself. Then the first question is if we actually have such a high frequency amplifier that can work at almost $3MHz$ as the DSD stream is sampled. And is this amplifier a good amplifier, meaning has it a good linearity and a good SNR? To answer completely this question it is interesting to investigate a class of audio amplifiers that has received a large attention in the last 30 years, that are Class-D amplifiers.

3.1 Class-D amplifier model

The design of audio amplifiers has evolved in the last century and produced several interesting architectures. The main objective is to reproduce audio signal that is to translate small electric signals representing audio information into large signals able to move actuators. These actuators are, surprisingly enough, loudspeakers approximately made with the same technology and shapes since the last eight decades. Their goal is to translate electric power into air vibration and thus acoustic power. Also here there are some important figure of merit that are used to compare and evaluate the performance of audio amplifiers; the more important ones are:

- Signal-to-Noise Ratio
- Band in which the amplifier can be modelled as a linear device
- Total Harmonic Distortion
- Maximum output power
- Electric efficiency

These parameters are dependent, like for example the maximum output power and the electric efficiency, or the SNR and the T.H.D. because they are used to evaluate different aspects of the same characteristic of the amp. Obviously a different output configuration is yielding a peculiar SNR at a specific output power, reaching a consequent efficiency. The first division we can apply to this large field of choices is the amount of amplification needed by the end user: from milliwatts used in heaphones amp or mobile media players, to a few watts in PC audio systems, to tens of watt for small "home-stereos" or automotive applications, till hundreds or thousands of watts necessary in Public Address systems to fill theaters and concert halls. Here is important to start giving a general definition of efficiency, underlining that where not explicitly stated this definition will only apply to the output stage of the amplifier.

$$E_{ff} = \frac{P_{LOAD}}{P_{LOAD} + P_{DIS}} \tag{3.1}$$

Where P_{LOAD} is the power provided to the load and the P_{DIS} is the total amount of power dissipated by the output MOS, due to several mechanisms that we will investigate in a specific paragraph.

A straightforward analog implementation of an audio amplifier is depicted in fig. 3.1; it uses transistors in linear mode to create an output voltage that is a scaled copy of the input one. The forward voltage gain is usually large (at least

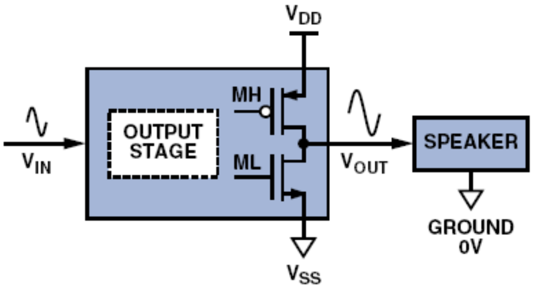


Figure 3.1: Block scheme of a linear CMOS output stage

$40dB$). If the forward gain is part of a feedback loop, the *loop gain* will also be high; feedback is often used because high loop gain improves performance suppressing distortion caused by unavoidable non-linearities in the forward path and also reducing the supply noise by increasing the power-supply rejection. The lack of this simple solution is that the output stage contains transistors that are used as DC current source, capable of supplying the maximum audio current required by the speaker. This causes a power dissipation that is often excessive because a large DC bias current usually flows in the output-stage transistors (which present a finite R_{ON}), without being delivered to the speaker, where we definitely do want it. This way a consistent power is waisted as heat because of the $I_{DS} \cdot V_{DS}$ product.

This solution has been named *Class A* due to the pristine audio quality that can offer, no matter the large power dissipation. Many other solutions (like Class B or Class AB [25]) have been developed trough the years to improve the electric efficiency, with always the intent to preserve the same audio quality. The more interesting of these alternative solutions is the *Class D* amplifier. A block scheme for a Class D amp is presented in fig. 3.2 and it is made of three major blocks.

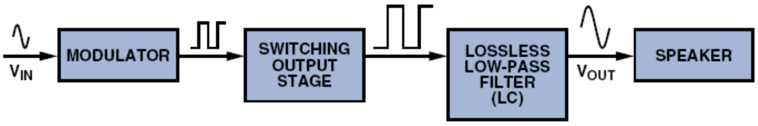


Figure 3.2: Block scheme of an open loop Class D amplifier

The first part of the amplifier is made by a modulator, whose aim is to convert the input audio signal to be amplified into a different electric signal capable of driving the output stage in a more efficient way. The second block is an inverter stage, where the transistor are biased in saturation mode, in order to be always or *fully on* or *fully off*. The last block is a lossless low-pass filter made by a resonant LC net. This filter is normally inserted between the output stage and the speaker to minimize electromagnetic interference (**EMI**) and avoid driving the load with too much high frequency energy. The filter need to be as much lossless as possible in order to retain the power dissipation advantage of the switching output stage; consequently this net uses capacitors and inductors, with the only intentionally dissipative element being the loudspeaker. Before presenting the modulation block that is the more interesting and crucial component of this chain, we will investigate a little further the structure of the switching stage to underline an important trade-off between performance and complexity.

3.2 Half-Bridge and Full-Bridge architectures

Both the structures we will now introduce are only depicting the power output stage and are always following a modulation block. Troughout this paragraph it's implied that there is a modulator of some kind providing a square wave driving the MOS transistor with a train of pulses with a $[0, 1]$ alphabet. The simplest inverter stage we can design is made by a single couple of transistor, for example a pair made by an n-mos and a p-mos like in fig 3.3

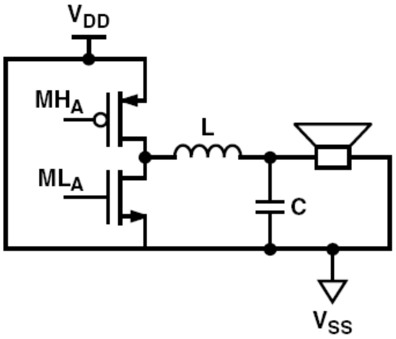


Figure 3.3: Half-bridge output scheme and LC low-pass filter

This configuration is called **Half-bridge** and it can be powered from bipolar power supplies or a single supply, but the single-supply version imposes a potentially harmful DC bias voltage, $V_{DD}/2$, across the speaker unless a blocking capacitor is added. This configuration is actually extremely similar to the linear amplifier we previously showed, but the substantial difference is the signal driving the output MOS transistor. The circuit can supply to the load, depending on the phase driving the n-MOS or the p-MOS a positive impulse $+1$ or discharging the load, meaning imposing on the load a 0 electric symbol. Thus we can think of this amplifier as capable of imposing on the load a $[0, 1]$ binary alphabet.

This solution is in the same time extremely simple and light, because we just need a pair of transistors for each signal channel. Furthermore, the need of power dissipation within the transistor is small; in fact the contribution mentioned before that is a *conduction loss* ($I_{DS} \cdot V_{DS}$) is in this case drastically reduced because neither I_{DS} nor V_{DS} are both large in the same time [24]; instead now there is a contribution to power loss related to the total capacitance C of the output stage that is switching at a frequency f_{Switch}, that is $C \cdot V^2 \cdot f_{Switch}$. This contribution is called *switching loss*. On the other side this solution has a substantial drawback linked to the poor power-supply rejection. Indeed the power supply voltage buses of a half-bridge circuit can be "pumped" beyond their nominal values by large inductors currents from the LC filter. The dV/dt of the pumping transient can be limited only by adding large decoupling capacitors between V_{DD} and V_{SS} that is limiting the band of the amp and it's requiring additional components.

Figure 3.4 shows a differential implementation of the output transistors and LC filter. This configuration is called **H-bridge** or **Full-bridge** and it is made of two half-bridge circuits that supply pulses of opposite polarity to the filter. This larger circuit is capable to impose on the load (the speaker) two different electric symbols of same amplitude but opposite sign, that is a $[-1, +1]$ alphabet. Full-bridge circuits do not suffer from bus pumping as the half-bridge ones, because inductor current flowing into one of the half-bridges flows out of the other one, creating a local current loop that minimally disturbs the power supply [24]. Also, for a given V_{DD} and V_{SS} the differential nature of the bridge means that it can deliver twice the output swing and for times the output power of single-ended implementations.

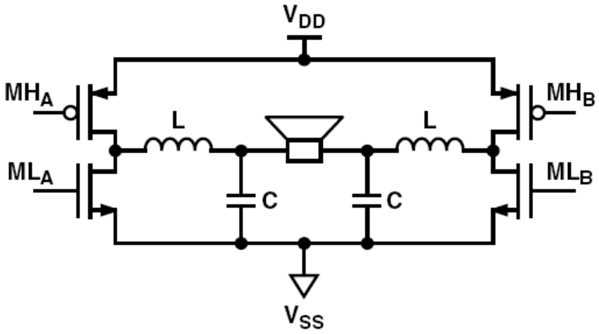


Figure 3.4: Full-bridge output scheme and LC low-pass filter

Another great advantages of Full-bridge architecture is that it can use "3-levels" modulations. Until now we intentionally did not talk about the signals driving the output stage and we simply said that our inverters are driven by tailored phases can impose on the load a $[0, 1]$ or a $[-1, +1]$ symbol alphabet. With a Full-bridge we have the chance to shortcut the load turning on both the high sides or the low sides of the bridge. Why should we desire our output stage to behave this way? Because with this "third state" that we can impose on the load we can expand the alphabet of symbols that our amplifiers can convey to the speaker. Then we will have a three state alphabet made by $[-1, 0, +1]$ symbols. In fact, this feature is offerend *for free* from the structure that we used to arrange the output MOS (the cost lies in the more complex driving circuit that we need to control the inverters). Furthermore when we switch from a -1 or a $+1$ symbol to a 0 symbol we have to turn on (and off) only 1 transistor, meaning that we are halving the capacitance C_{Gate} for that symbol transition. This will imply a significant power saving.

Before exploring in detail the different modulations that can exploit (or not) the two different alphabets we will now summarize all the power loss components and we will make some comments on the possible hardware implementation. Then, in the last part of the chapter, we will present many possible modulation schemes with detailed information about the efficiency they can achieve.

3.3 Power loss components

We already presented two major components in power dissipation for Class D amps, that are conduction loss and switching loss. The first is strictly related to the area of the transistor that implies a finite R_{ON} and then a finite V_{DS} when the MOS is turned on, i.e. an amount of dissipation that is:

$$P_{Conduction} = I_{DS}^2 \cdot R_{ON} \qquad \text{(or equivalently } I_{DS} \cdot V_{DS}). \tag{3.2}$$

Then we presented the switching loss as a power dissipated from the driving circuit to turn on and off the output MOS, thus related to the gate capacitance C_{GATE}. Actually this is only a part of a more complex switching loss contribution that we can write as:

$$P_{SwitchTOT} = P_{Switch} + P_{Gate}. \tag{3.3}$$

where

$$P_{Switch} = E_{Switch} \cdot f_{Switch}. \tag{3.4}$$

and

$$P_{Gate} \approx 2 \cdot Q_{Gate} \cdot V_{Ref} \cdot f_{Switch}. \tag{3.5}$$

In equation 3.5 V_{Ref} is the Reference Supply Voltage, Q_{Gate} is the total charge on the capacitance of the MOS and of course F_{Switch} is the switching frequency.

The term E_{Switch} is a complex term that can be expressed as:

$$E_{Switch} = \int_0^t V_{DS}(t) \cdot I_{DS}(t)dt \tag{3.6}$$

where t is the length of the switching pulse.

We have consequently a trade of because if we would like to make the MOS larger to decrease the R_{ON} and decrease the $P_{Conduction}$ term we are enlarging the gate capacitance C_{Gate} and increasing the P_{Switch} accordingly. Then numeric evaluation could be used and also graphical solutions are useful to find a minimum for a cost function comprehending both terms. Fig 3.5 depicts an example for this kind of analysis.

After presenting this first order analysis we have to notice that there are several others parameters that can influence the power losses of a Class D amp [26], but these are the more important ones, mainly because their link with the possible modulation schemes is evident. To conclude we can present a

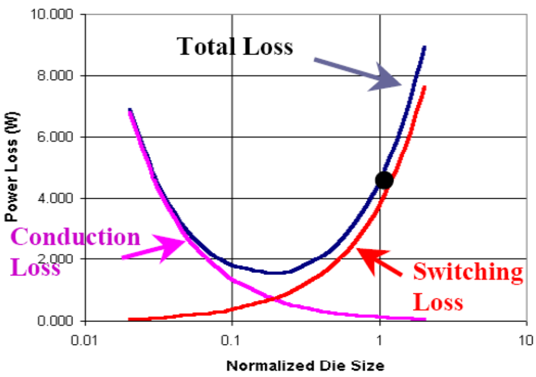


Figure 3.5: Trade-off between conduction losses and switching losses. Normalized area is used to fix the R_{ON}/C_{Gate} ratio

picture of global efficiency for an ideal Class A, ideal Class B and a measured Class D amplifier. This an example is for an audio amplifier with $10W$ P_{Load} max, meaning an average realistic listening level P_{Load} of $1W$ [25]. Under this condition the power dissipated inside a Class D output stage is $282mW$ vs. $2.53W$ for a Class B and 30.2 for a Class A. Then the Class D efficiency is varying from 78% to 90% that is reached at the maximum power, that is always much greater than Class B and Class A efficiencies that are 28% and 3% respectively.

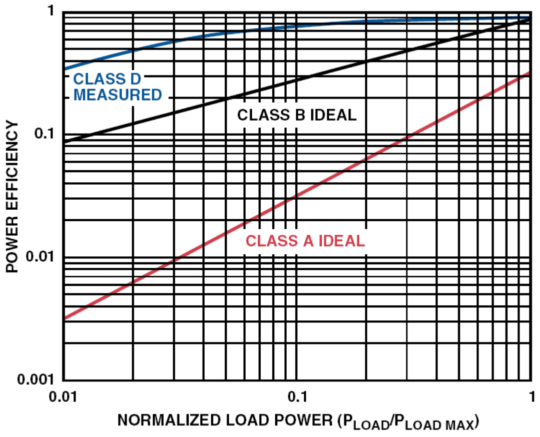


Figure 3.6: Comparison of power efficiency for Class A, Class B and Class D output stages

3.4 PWM and PDM

In this section we will present the some aspects of the most important modulation used to drive Class D amplifiers that is Pulse Width Modulation **PWM**.

Another technique that is gaining attention in the last years is the Pulse Density Modulation **PDM**, the most important example of which is Σ-Δ modulation that we already presented in the first chapter. PWM has been developed since the early fifties [27] and it is a modulating process that encodes information about an audio signal into a stream of pulses. The basic idea is that the time amplitude of these pulses is linked to the instantaneous amplitude of the input. Conceptually, PWM compares the input audio signal to a triangular or ramping waveforms that runs at a fixed (much higher) carrier frequency. This creates a stream of pulses at the carrier frequency; within each period of the carrier the duty ratio of the PWM pulse is proportional to the amplitude of the audio signal. A scheme of this process is presented in fig. 3.7 and 3.8.

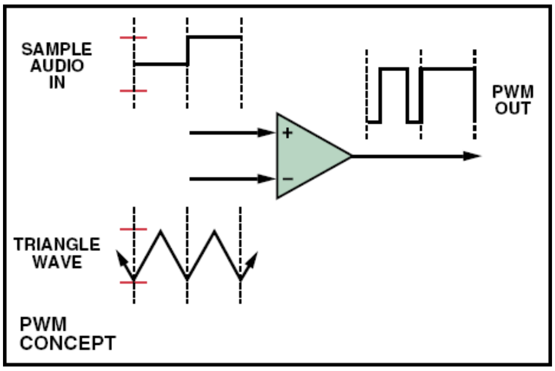


Figure 3.7: Pule Width Modulation concept

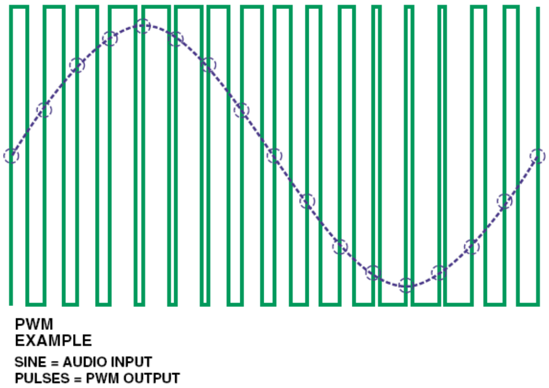


Figure 3.8: Pule Width Modulation example

In the example of figure 3.8 the audio input and the triangular wave are both centered (like in the many practical cases) around $0Volt$, so in case of a 0 input the duty cycle of the output pulses is 50%. For a large positive input the duty cycle increases towards 100% while for large negative input it goes down to 0%. If the audio amplitude exceeds that of the triangle wave we have the so called *full modulation*, where the pulse train stops switching and the duty ratio within the individual period of the carrier is or 0 or 1. This modulation technique is attractive because it can allow good performance in terms of SNR with a relatively low frequency carrier of few hundreds kHz. Also, it is stable up to nearly 100% modulation, permitting thus high output power up to the point of clipping or overload. Depending on the form of the carrier itself and on the method used to sample the audio input signal we have a broad choice of possibilities with noticeably different performances.

An alternative to PWM is, as we said before, PDM and in specific Σ-Δ modulation; this method is interesting because of the fixed minimum amplitude of the pulses that the sampling gives as output; while PWM pulses length varies significantly from large to small amplitude input signals and it can cause problems in most switching output-stage-driver circuits that have a limited drive capability (meaning that they cannot switch properly at the excessive speeds needed to reproduce short pulses of few nanoseconds!) this is not the case for Σ-Δ modulation. Here we have a fixed pulse length because of the presence of a fixed clock that control every switching instant. Analyzing the spectral content of these two different modulations we will see a large difference that consists in the presence of discrete spectral components due to the modulation process in the PWM and a continuous power spectral density for the Σ-Δ.

Here we will present several different examples of PWM modulation methods [28] and then we will try to apply some knowledge learned from this analysis/comparison to the Σ-Δ modulation in order to create a whole new signal flow, from the authoring step till the final user.

PWM methods and comparison

We intuitively presented how PWM works, now we want to present a plethora of different practical implementation and give some details on the possible performance achievable with each of them. First of all it's useful to list some desired pulse modulation characteristics, that are:

- A high linearity
- A minimal switching frequency to bandwidth ratio (f_s/B)
- A minimum high frequency spectral content at all modulation index ($M \in [0, 1]$). The ideal modulation would generate *no* HF-components.
- A low modulation complexity for easy implementation

Traditionally PWM is categorized in two major classes by the sampling method: natural sample PWM (**NPWM**) and uniform sampled PWM (**UPWM**). There are other alternative sampling methods that can be seen as hybrid methods between the two. There is a standard terminology used to classify several variants of modulation and if follows the subsequent order:

Terminology 1 {*Sampling Method*}{*Switching*}{*Edge*}

Where *Sampling Method* stands for Natural or Uniform, *Switching* stands for 2-levels or 3-levels and *Edge* stands for single sided (sawtooth carrier or ramp) or double sided (triangular carrier) [28]. A list for the resulting variants is in table 3.1.

Sampling Method	Edge	Levels	Abbreviation
Natural Sampling	Single sided	Two (AD)	NADS
(NPWM)		Three (BD)	NBDS
	Double sided	Two (AD)	NADD
		Three (BD)	NBDD
Uniform Sampling	Single sided	Two (AD)	UADS
(UPWM)		Three (BD)	UBDS
	Double sided	Two (AD)	UADD
		Three (BD)	UBDD

Table 3.1: Considered variants of PWM

The analysis and comparison of the different modulation schemes is normally based on PWM responses to single tone input. Analytical treatment of modulated multitones and noise signal is complex and investigations have therefore been carried out by computer simulation. However the response to single tone input provides nearly all interesting information about the modulation process and the normally the multitone responses can be in general well predicted from these the single tone ones. The behavior has been analyzed with double Fourier series [27] and also a very interesting graphical analysis tool has been adopted [28]. For sake of brevity we demand on reference the complete analytical derivation while we discuss the main results and present some key aspects and comparisons with the aid of some pictures. Our final goal is to find which PWM method is the best one in terms of suppressing the high frequency component linked to the presence of the carrier and, if possible, reducing power losses.

Starting a detailed investigation of the high frequency spectral characteristics we define the frequency ratio between the signal angular frequency ω and the carrier angular frequency ω_c:

$$q = \frac{\omega}{\omega_c} \tag{3.7}$$

The theoretical maximum for q is not determined by the Nyquist criteria, as with normal amplitude sampling. In fact it is determined by considering the carrier and signal input slew-rates; the resulting limitations depend on the choice of the Edge for the modulation and so we obtain:

$$SR_S = \frac{\omega_c}{2\pi} \Rightarrow \omega \leq \frac{\omega_c}{2\pi} \text{ (for Single sided modulation)} \tag{3.8}$$

$$SR_D = 2 \cdot \frac{\omega_c}{2\pi} = \frac{\omega_c}{\pi} \Rightarrow \omega \leq \frac{\omega_c}{\pi} \text{ (for Double sided modulation)} \tag{3.9}$$

In general, the high frequency components in the spectra will have an important influence on the maximal frequency ratio q allowed. Furthermore, there are also practical consideration and cost constraints that could put simpler or tighter limits to the possible frequency ratios defined in 3.8 and 3.9. Since the high frequency spectral characteristics depend on several parameters like the modulation index **M**, ω, ω_c in a complex way, it's easier to introduce a graphical analysis tool called Harmonic Envelope Surface (**HES**) that provides many details in a single figure [28]. The idea is to plot the amplitude spectrum versus modulation index M and normalized frequency ω/ω_c. Instead of a $3D$ visualization the HES uses a $2D$ plot with color (or grayscale) defining the amplitude. Cutting the HES with a surface perpendicular to the plane of the figure we will obtain a normal amplitude spectrum versus frequency graph. In the examples the parameters are define as:

- M $\in -100, 0$
- $q = 1/16$
- $f \in 0, 4$ where f is the normalized frequency (ω/ω_c)

The choice of the frequency ratio q is not important when making a comparison among different modulation schemes. In fact this choice is close to a worst case scenario where the input signal is not too low-frequency compared to the sampling carrier.

3.4.1 Natural Sampled -AD- Single Sided Modulation (NADS)

This is a 2-levels scheme, wish a single sided carrier. In this modulation there are no common-mode components over the bridge phases and actually this is a general for 2-levels modulation process. Here is the Double Fourier Series (**DFS**) for this modulation:

$$\begin{aligned} F_{NADS}(t) = {} & M\cos(y) \\ & + 2\sum_{m=1}^{\infty} \frac{1 - J_0(m\pi M)cos(m\pi)}{m\pi}\sin(mx) \\ & - 2\sum_{m=1}^{\infty}\sum_{n=\pm 1}^{\infty} \frac{J_n(m\pi M}{m\pi}\sin(mx + ny - m\pi - n\pi/2) \end{aligned} \tag{3.10}$$

where

- M is the Modulation index ($M \in [0, 1]$)
- $x = \omega_c t$ is the carrier signal angle frequency
- $y = \omega t$ is the audio signal angle frequency
- J_n is the Bessel function of nth order
- n is the index for the audio signal harmonics
- m is the index for the carrier signal harmonics

One of the most important conclusion is that with NADS the modulation signal is left unchanged, i.e. there is no forward harmonics. This means that the modulation process followed by an appropriate filtering can be considered *ideal* in terms of distortion. This is a very interesting feature of natural sampling.

The intermodulation components (**IM-components**) are very pronounced at $mx \pm ny$ and they depend strongly on the modulation index M. In figure 3.9 this can be seen as "skirts" surrounding the carrier harmonics. The components related to even harmonics of the carrier reduce with M and they are totally eliminated at idle.

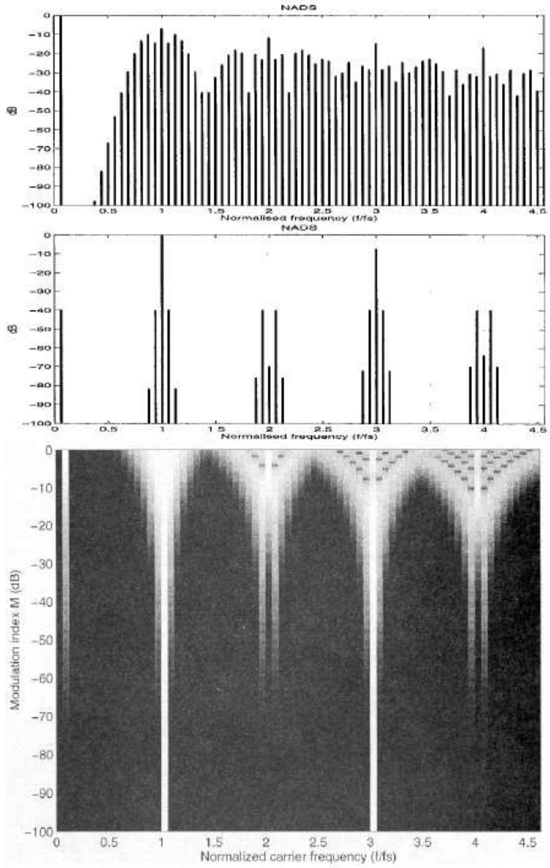


Figure 3.9: Spectrum characteristics for NADS and HES-plot, with M varying from $-100dB$ to $0dB$ (full modulation), $q = 1/16$.

3.4.2 Natural sampled -BD- Single sided modulation (NBDS)

In this modulation there is a high frequency commond mode-mode signal at the bridge phases, so there is the necessity for a common-mode filtering. The DFS for this scheme is:

$$\begin{aligned} F_{NBDS}(t) &= M\cos(y) \\ &- 2\sum_{m=1}^{\infty}\sum_{n\pm 1}^{\infty}\frac{J_n(m\pi M)}{m\pi}cos(mx+ny-m\pi)sin(n\pi/2) \end{aligned} \tag{3.11}$$

Here the HES-plot 3.10 shows very pleasant spectral characteristics at lower output leaves and all high frequency components disappear at idle. This is a clear advantage of NBDS and, in general, of all 3-levels modulation schemes. Whereas all harmonics of the carrier were present in NADS (especially the odd harmonics), they are not present in NBDS at all. All the IM-components $mx \pm ny$ with even n are *eliminated*, meaning that the spectrum only contains half the components compared to NADS. Furthermore the maximal IM-components are lower than in NADS thanks to the 3-levels switching. Based on this theoretical performance, NBDS is to be considered superior to NADS in terms of modulation spectra.

3.4.3 Natural sampled -AD- Double sided modulation (NADD)

This modulation operates with both leading and trailing edge of the pulses by using a triangle shaped carrier reference. The DFS is

$$\begin{aligned} F_{NADD}(t) &= M\cos(y) \\ &+ 2\sum_{m=1}^{\infty}\frac{J_0(m\pi M/2)}{m\pi/2}\sin(m\pi/2)\cos(mx) \\ &+ 2\sum_{m=1}^{\infty}\sum_{n=\pm 1}^{\infty}\frac{J_n(m\pi M/2)}{m\pi/2}\sin((m+n)\pi/2)\cos(mx+ny) \end{aligned} \tag{3.12}$$

and the related HES-plot is in fig. 3.11. We can notice that the argument of the Bessel functions is halved in comparison with both NADS and NBDS. This is important since the ratio by which the IM-components reduces thereby is increased, especially around the first harmonic of the carrier. There are only odd IM-components but, similar to NADS also NADD does not have a pleasant spectrum for low M, since the odd harmonics of the carrier are present with maximal amplitude. This means that NADD is superior to NADS, but not as attractive as NBDS.

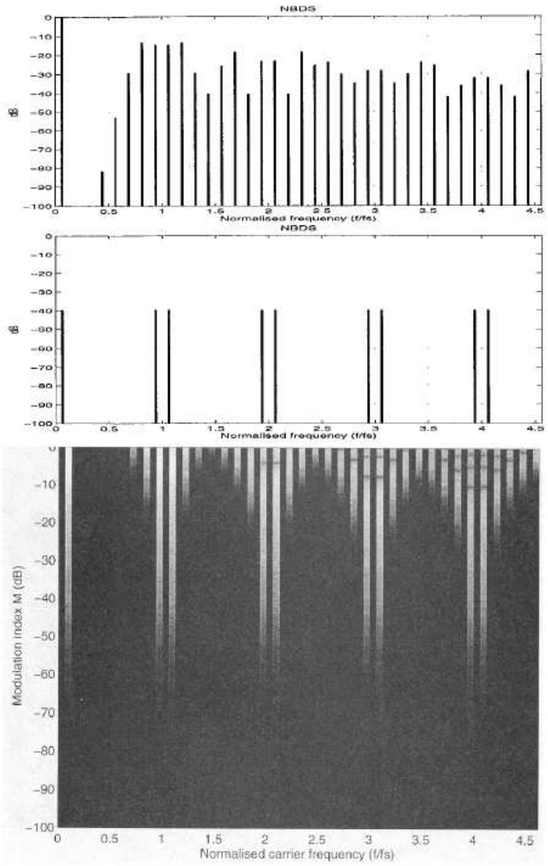


Figure 3.10: Spectrum characteristics for NBDS and HES-plot, with M varying from $-100dB$ to $0dB$ (full modulation), $q = 1/16$.

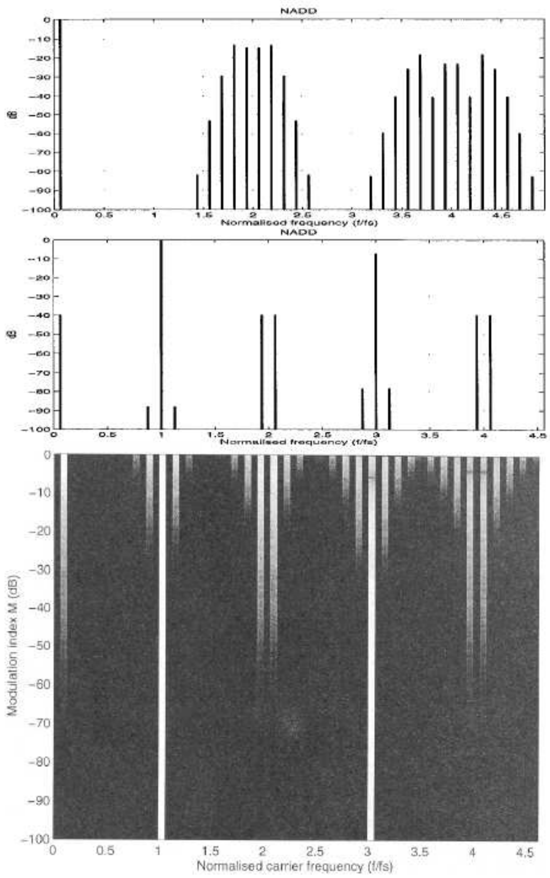


Figure 3.11: Spectrum characteristics for NADD and HES-plot, with M varying from $-100dB$ to $0dB$ (full modulation), $q = 1/16$.

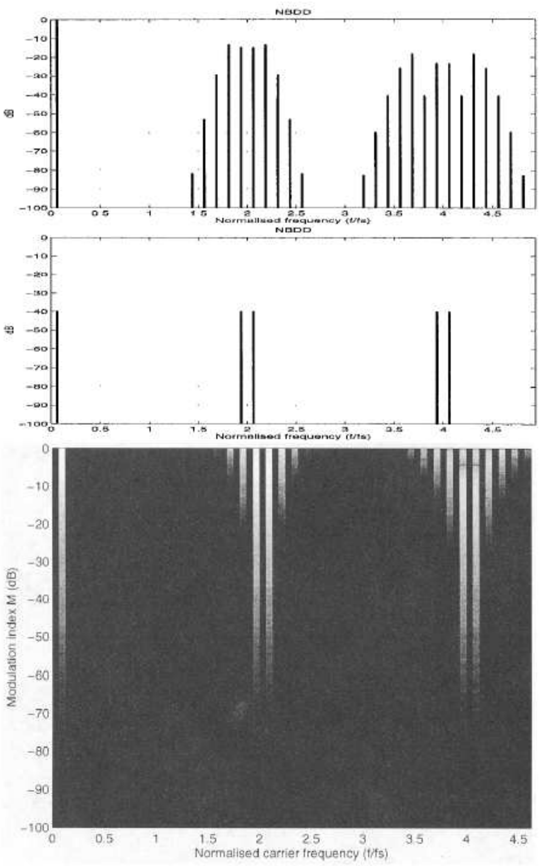


Figure 3.12: Spectrum characteristics for NBDD and HES-plot, with M varying from $-100dB$ to $0dB$ (full modulation), $q = 1/16$.

3.4.4 Natural sampled -BD- Double sided modulation (NBDD)

Like for the NADD case also in this double sided modulation we have two samples per switching period and we can write the spectrum as:

$$\begin{aligned} F_{NBDD}(t) &= M\cos(y) \\ &- 4\sum_{m=1}^{\infty}\sum)n \pm 1^{\infty}\frac{J_n(m\pi M/2)}{m\pi}\cdot \\ &\cdot \sin((m+n)\pi)/2)\sin(n\pi/2)\sin((mx+ny)-n\pi/2) \end{aligned} \tag{3.13}$$

Also here the effective sampling frequency is *doubled*, while the carrier frequency and thereby the switching losses are retained. Looking at the HES-plot in fig. 3.12 we can see that NBDD has by far the most attractive spectral characteristics. The only drawback is the need for filtering the common-mode content at the output bridge terminals.

Up to now, the most important conclusion of this analysis is that natural sampled PWM is totally free from "forward" harmonic distortion and, in terms of modulation quality, it is possible to rank the four modulation variants from "best" to "worst" this way:

1 NBDD
2 NBDS
3 NADD
4 NADS

The high frequency characteristics of NDBD combines three attractive features:

- An effective doubling of the sampling frequency, which is beneficial for demodulation and power losses control.
- A total elimination of *all* components related to the carrier.
- A near linear relationship between M and IM-component amplitude ad lower modulation indexes, which causes the idle spectrum to be free from components (almost zero switching losses at idle).

For uniform sampling we refer to reference [28] for the same Fourier and HES analysis. Here is important to say that also in this case we can rank from "best" to "worst" the different variants and not surprisingly we obtain:

1 UBDD
2 UBDS
3 UADD
4 UADS

Again the 3-levels modulations are the best in class also for the uniform sampling. After reviewing all these possible modulation schemes we can reflect for a while on the different technologies that we presented till now. We started observing that high quality audio needs for a specific kind of A/D converters, namely Σ-Δ modulators, that provide a high frequency train of pulses with, normally, a $[-1, +1]$ alphabet. Then we presented a authoring software implementation of these converters that exploits the Viterbi algorithm to increase the performance to a even higher level. Now we showed that there is the practical chance to implement an audio amplifier using a Class D architecture that *needs for* a 2-or-3-levels modulation in order to convert the audio to be amplified into a stream of pulses that drives efficiently the power output stage; so here we have a novel idea that can shorten the signal processing needed to go from the audio source to the final user front-end.

We can actually design a software Trellis converter using a *three-levels* alphabet knowing the fact that the resulting stream can be directly used to drive a high efficiency Class D full-bridge amplifier. This way we can increase the SNR performance using a Viterbi software converter and in the same time we can reduce the power consumption losses using a 3-levels based full-bridge architecture; in the following we will show that both these improvements can be achieved and this idea is not trivial. We will show, for example, that we can encode from 45% till 85% of 0 symbols in the stream converting audio signal that, like music and speech, contains part of silence or fading codas. So, this approach to software A/D conversion is traduced in a drastically lower power consumption that could be extremely useful in portable applications like mp3 readers.

Chapter 4

Three-levels Trellis Σ-Δ Modulation

In this chapter we will introduce a new signal flow that starts from the authoring step of the normal music production and is able to render an alternative file format. It is capable of increasing both quality and power saving performance. As we showed in the previous chapters the sum of two different technologies that are a Trellis Σ-Δ modulator (or converter) and three levels modulations can achieve this result.

4.1 Efficient signal representation

We illustrated in chapter 3 that a three levels modulation can sensibly reduce power losses due to switching components, especially when the amplifier is in idle state. We can now extend this thinking that audio signal can always be approximated with a finite sum of sinusoidal functions, i.e. functions that present always zero crossings. Furthermore fading codas or reverberation tails also have typical slopes that dump the whole amplitude of the signal towards zero. So to say that music and speech signals are filled with instants, short or long depending on the information content, in which the signal is sensibly smaller than the peak value. This way we can think of applying a three-levels modulation in order to exploit this signal feature and drive the class D amp more efficiently. We can implement a three-levels Σ-Δ Trellis converter.

First, we can start writing a reference code implementing a software A/D converter and, in our case, a software Σ-Δ converter; the algorithm of the converter is simple and we can write it in a pseudocode, like in table 4.1.

Instead, the Trellis Σ-Δ converter that we introduced in chapter 2 acts in a different way. While a normal Σ-Δ gives an output that is an immediate function of the previous encoded sample and the actual input, the Trellis Σ-Δ first collects, for all the possible path, the weight function cost and then, after that a proper latency time has expired, it emits the symbol to which the trellis has converged to [21]. The elementary step of this algorithm can be written in pseudocode as in table 4.2.

Of course this algorithm can easily be expanded to offer a three-levels modulation. The change to be done is to increase the alphabet of the modulation,

```
$Y :=$ StandardΣ-Δ $(X)$
$X \leftarrow$ INPUT

FilterInput $:= X-$ Previous$Y$
FilterOutput $:=$ Filter (FilterInput)
$Y :=$ Quantize (FilterOutput)

$Y \rightarrow$ OUTPUT
```

Table 4.1: Pseudocode for a standard Σ-Δ converter

```
$Y :=$ TrellisΣ-Δ $(X)$
$X \leftarrow$ INPUT

For (All NewCandidates)
      $C_1 :=$ CalculateTotalCost (Path$_1$, $X$)
      $C_2 :=$ CalculateTotalCost (Path$_2$, $X$)
      if ($C_1 < C_2$)
            Select (Path$_1$)
      else
            Select (Path$_2$)
      Save (NewCosts, NewFilterState, NewPath)
For (All NewCandidates)
      TotalCosts $:=$ TotalCosts $-$ MinimumCosts
$Y :=$ PathValue $(t_{current} - t_{lat})$

$Y \rightarrow$ OUTPUT
```

Table 4.2: Pseudocode for a Trellis Σ-Δ converter

meaning the number of possible encoding symbols to choose from that it can evaluate at each step. While in the two-levels Σ-Δ converter the Path$_1$ and Path$_2$ correspond to the $[-1, +1]$ symbols to be encoded, in a three-levels algorithm we have to evaluate *three* possible new paths at each new time instant; so we will have Path$_1$, Path$_2$ and Path$_3$ corresponding now to the $[-1, 0, +1]$ alphabet to be encoded. All the remaining part of the algorithm can be preserved.

We now desire to show how this signal representation is definitely efficient, meaning that we can drastically reduce the power of the encoded signal and thus reduce the switching losses (and in general power consumption) needed to amplify it. The basic idea is to compare the power of an reference signal when encoded with both two-levels and three-levels Trellis Σ-Δ.

We have to star with an assumption, i.e. that the *reconstructed signal* obtained after the low-pass filter, that in our class D amplifier model is the passive L-C net followed by the loudspeaker, is the same. This means that we want to evaluate the power of the encoded signals once sure that they convey the same information. To prove this a complete set of simulation has been done, with sinusoidal test signal spanning over all the band if interest, from $20Hz$ up to $20kHz$, and over different amplitudes. Then the encoded signal has been collected and filtered by a digital implementation of a reconstruction low-pass filter designed with Matlab. The encoded signal is, obviously a two-three-levels signal formed only by integer values, while the output of the low-pass filter is a double valued stream of reconstructed samples, with the same sample frequency of the original input. So a sinusoidal reference signal of the same frequency was synthesized with Matlab and compared to the two reconstructed signals, first with the one obtained from the two-levels stream and then with the one reconstructed from the three-levels one. The differences obtained where finally stored.

The result was that in all the band of interest (and even with more complex signals obtained as sums of a finite number of sinusoids) both the reconstructed signals discrepancies where all in the order of 10^{-3}-10^{-4} for the magnitude, with no phase alterations. Once proved that both encoding systems where conveying the same amount of information then, we can proceed with our comparison of the encoded signal power.

We define a **representation efficiency** η as

$$\eta_{sig} = \frac{P_{Reconstructed}}{P_{Sig}} \tag{4.1}$$

Where $P_{Reconstructed}$ is the power after the reconstruction filter and P_{sig} is the power of the encoded signal, that we can evaluate in the time domain. P_{Sig} has different values depending if it represents a two-levels modulated signal or a three-levels one.

$$P_{Sig} = \begin{cases} P_{2-lev} & \text{if the signal is encoded with a 2-levels Trellis-SDM} \\ P_{3-lev} & \text{if the signal is encoded with a 3-levels Trellis-SDM} \end{cases}$$

Furthermore, analyzing sinusoidal signals we can remember that the power of the reconstructed signal can be written in closed form. This could be done after having simulated that the reconstruction noise is negligible, as we have proved, because the power of the reconstruction noise is around $-80dB$ referred to the amplitude of the encoded signal. So $P_{Reconstructed}$ can be written as:

$$P_{Reconstructed} = 1/2A^2 \tag{4.2}$$

where A is the amplitude of the sinusoidal signal. Then, to evaluate the power of the encoded signal we can write the equation that represent the power of a signal in the time domain, that is:

$$P_{Sig} = \frac{1}{N}\sum_{i=0}^{N} x(i)^2 \tag{4.3}$$

where $x(i)$ is the i^{Th} sample of the encoded signal and N is the total amount of samples. Looking at equation 4.3 it is intuitive to affirm that P_{2-lev} is identically 1, because the only symbols we have in the two-levels alphabet are $[-1, 1]$ that squared always get 1. Indeed, P_{3-lev} can be substantially lower than 1 due to the presence of a variable amount of 0 symbols in the stream. So we can write an **Efficiency Gain** in order to evaluate the performance gain of a three-levels Trellis modulation as:

$$EffGain = \frac{\eta_{2-lev}}{\eta_{3-lev}} = \frac{\frac{A^2}{2 \cdot P_{3-lev}}}{\frac{A^2}{2 \cdot P_{2-lev}}} = \frac{P_{2-lev}}{P_{3-lev}} = \frac{1}{P_{3-lev}} \tag{4.4}$$

It's clear that a lower P_{3-lev} produces a higher efficiency gain, thus a larger energy saving. Here we present a table in which we collected several power values P_{3-lev} evaluated for decreasing amplitudes of sinusoidal test input. The last column shows the corresponding values of the Efficiency Gain.

Amplitude	P_{3-lev}	**Efficiency Gain**
8.5 10^{-1}	0.5581	1.80
6.8 10^{-1}	0.4552	2.20
5.0 10^{-1}	0.3497	2.86
2.5 10^{-1}	0.2269	4.40
1.25 10^{-1}	0.2210	4.52
6.25 10^{-2}	0.2523	3.96
3.125 10^{-2}	0.2972	3.37
1.5625 10^{-2}	0.2926	3.42
7.8125 10^{-3}	0.3033	3.30
3.9062 10^{-3}	0.3001	3.33
1.9531 10^{-3}	0.2882	3.47
9.7650 10^{-4}	0.2707	3.69
4.8820 10^{-4}	0.2957	3.38
2.4410 10^{-4}	0.2315	4.31
1.2205 10^{-4}	0.1810	5.52

Table 4.3: Power of the encoded signal using the 3-levels Trellis Σ-Δ modulation. The second column is the efficiency gain

It's interesting to notice that for a signal amplitude of 5.0 10^{-1}, that is actually the $0dB$ reference for SACD, we already have a large amount of zeros in the encoded stream, providing an encoded signal power P_{3-lev} equal to 0.3497. This is only a fraction of the two-levels modulation power that is approximately three times bigger.

In picture 4.1 we illustrate the same data and showing clearly that the efficiency for a two-levels modulation only depends on the signal amplitude, while efficiency for a three levels modulation depends on the amount of zero symbols encoded.

4.2 Performance improvement

The first desired improvement of the suggested three-levels Trellis modulation is power saving and the presented data has already cleared that this technology is achieving the prospected goal. On the other side there is another fair secondary effect that is strictly linked with the enlarged symbol alphabet for our encoding algorithm, i.e. a better Signal-to-Noise ratio. As suggested in literature [12]

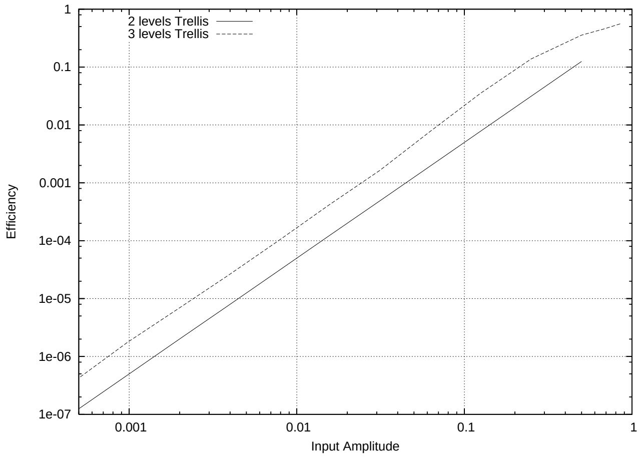


Figure 4.1: Efficiency comparison for two-and-three-levels modulation. The efficiency gain corresponds to the vertical distance of the two curves

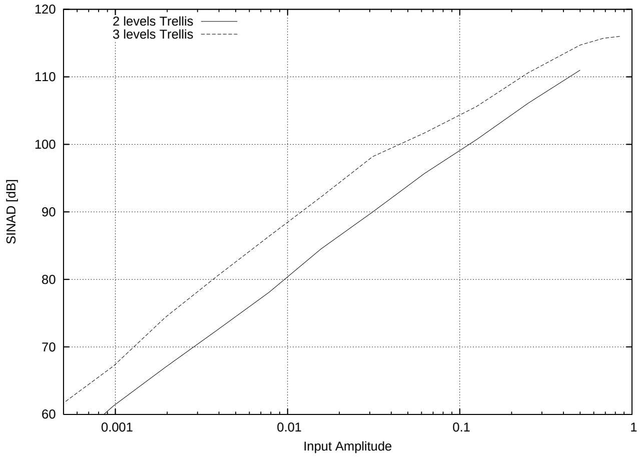


Figure 4.2: Sinad improves up to 8.2 dB.

increasing the number of representation states when converting a signal has the beneficial effect of reducing the quantization step q, thus reducing consequently the quantization noise and increasing the SNR. In figure 4.2 we can clearly see that the SNR or SINAD (that stands for Signal-to-Noise-And-Distortion ratio) has improved up to 8.2 dB.

4.3 Dynamics range extension

Looking at table 4.3 we can also notice another interesting data: the maximum allowed input amplitude before the modulator starts oscillating has increased from 0.5 to 0.85. This features (approx. +25%) is actually one face of a complex dart. In fact these data suggest two simultaneous approach to the designer of a

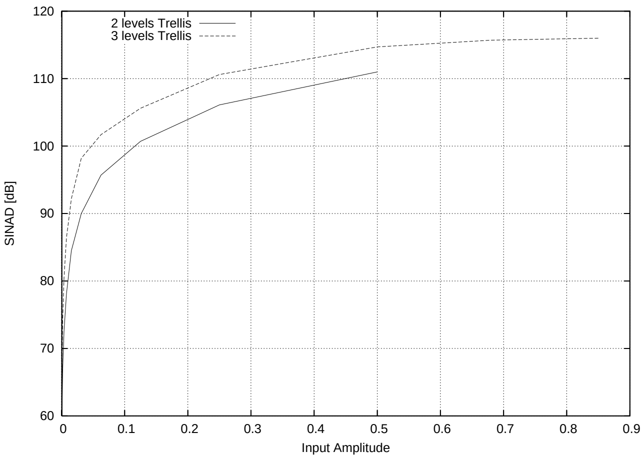


Figure 4.3: Input swing handling capability increases up to +25%

Trellis Σ-Δ converter; at first we can think of using the same noise-shaping filter and achieve power saving and, in the same time, a more powerfull rejection towards instability. At second glance we can think of imposing the same stability constraints (that are guided by the SACD standard that fixes the maximum allowed input signal amplitude at 0.5) and then obtaining a more aggressive noise shaping, thus a better Signal-to-Noise ratio [21]. In this work, starting from an already quite aggressive 5^{Th} order Σ-Δ we preferred to exploit this feature to increase stability. This will be extremely useful especially when applying the frequency-warping theory to the Σ-Δ as presented in a following chapter. In order to underline the large amount of now exploitable input swing we presented in figure 4.3 the same data of picture 4.2 expressed in linear scale.

4.4 Some reflections

Although all these positive effects on power saving, SNR and dynamic range handling capabilities the three-levels modulation has drawbacks: enlarging the alphabet from two to three symbols asks for a larger computational effort compared to the standard Trellis modulation. Furthermore, and this is perhaps a bigger issue compared to the previous one, the presence of three symbols in the encoded stream asks for more storage space on the final support. While in the two-levels stream we had only $[-1, +1]$ symbols requiring at most a single bit per symbol to be physically stored, now we have three symbols $[-1, 0, +1]$ that need a more complex solution to fit the same space.

In fact, this consideration enjoins the designer to reflect on the real capabilities of optical media that are used as physical supports for SACD and asks for a dedicated form of compression in order to preserve the positive results obtained *and* in same time being comparable to the normal two-levels Trellis modulation on the required storage space needed. We will investigate further this aspects in a following section fully dedicated to lossless audio compression.

Chapter 5

Heuristic efficient Trellis algorithm

A disadvantage of the Trellis architecture respect to the standard Σ-Δ is that for all the possibile encoding candidates we have to store several informations: first of all the output sequence that includes both the time window corresponding to the depth of the Trellis *and* the latency, that could be sensibly longer as explained in chapter [**?**]; then there are the filter states and the value of the cost function.

Furthermore, the Viterbi Algorithm has to calculate two or three possible paths for each candidate at every new time instant, depending on the two or three levels implementation. Both the computational complexity and the amount of memory increase exponentially with the depth of the Trellis, **N**. On the other side, as we presented before, the performance increase sensibly with a larger value for N, so we would like to overcome this limits for the Trellis order to relative low values. So, in this chapter a more efficient implementation of the Trellis is introduced, based on a heuristic algorithm.

5.1 Computational efficiency

The next goal for our three-levels encoding systems is to achieve the same performance in terms of SNR and Total Harmonic Distortion of the presented Trellis algorithm while, in the same time, saving computational power. The systems resources needed comprehend both CPU time and memory usage; the idea is to use an heuristic approach often used in linear integer programming, that is to extract a subset of states from our domain, i.e. a subset of promising path, and evaluate only the consequent cost function values and future candidates. So one way to increase the efficiency of our Trellis structure, is to calculate only a fraction of all the 2^N candidates.

After all, our interest is to find *the* single candidate with minimum global costs that represents our suboptimal solution. The candidates most likely to lead to the optimum final solution are taken into account, the other candidates are not used.

The following heuristic criteria are used:

- *Cheap* candidates have higher probability to converge to an optimal sequence.
- *Expensive* candidates have lower probability to converge to the optimal sequence.

To verify this hypothesis, a 10^{th} order Trellis in combination with a 5^{th} order low-pass filter was simulated with sinusoidal input signal with amplitude 0.5 and frequency $1kHz$. The candidate output sequences (that are 1024) are ordered with increasing cost function value (eq. 2.2). After that, the candidates are numbered with a cost index i; the M cheapest alternatives are the candidates with $1 \leq i \leq M$. When $t \rightarrow \infty$, the final output sequence is determined and $\forall t$ the corresponding candidate with its index i was traced back. Figure 5.1 shows the chance for a candidate with a certain cost index to become the final suboptimum solution. It is clear that the best solution is mainly determined by a small amount of all candidates (< 10 out of 1024).

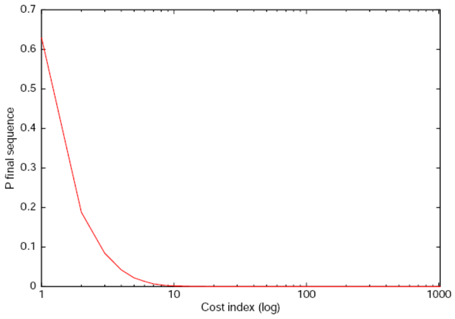


Figure 5.1: Chance for a candidate with a certain cost index to become the final solution. The cost index ranks the candidates on increasing cost function.

Taking advantage of this knowledge then the original Trellis algorithm is adapted in the following way: based on the cost index only $M \ll 3^N$ candidates are selected and used for further processing, instead of using all 3^N candidates. In brief, the new algorithm can be described as follows:

Given the M cheapest candidates at time $t-1$, calculate the M cheapest candidates at time t.

Going a little in detail, the algorithm starts with the M cheapest candidates at time $t-1$ These paths are ordered with increasing cost function values. Every candidate at $t-1$ leads to *three* new alternatives at time t, see figure 5.2. Because of this, the number of new alternatives is equal to $3 \cdot M$.

It is possible that three different candidates at time $t-1$ lead to the same new path at time t (see figure 5.3). To maintain the Trellis structure (last N encoded symbols are different for all paths), only one of the three alternatives (the cheapest) is used. In the most extreme case, the $3M$ new alternatives consist of M different candidates only. After elimination of double candidates,

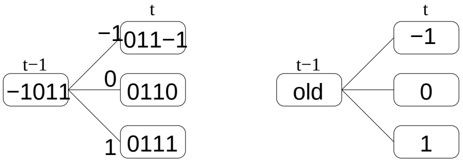


Figure 5.2: Every old candidate at time $t-1$ leads to three new candidates at time t. An example (left) and the general case (right).

the number L of new candidates is bounded by $M < L < 3M$. Then, the L candidates are sorted on cost function, and the cheapest M out of L are selected.

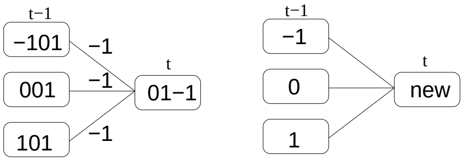


Figure 5.3: Three old candidates at time $t-1$ lead to the same new candidate at time t. To maintain the Trellis structure, it is not allowed to use both paths. An example (left) and the general case (right).

The process of the elimination of the triple candidates is within the structure itself of the implementation code of the algorithm. The data structure of choice for this task are list objects, that can offer easily a function of inserction after a check. In this particular case a new candidate is inserted in the list data structure *if and only if* it passes a check-function that preserves the new list of candidates to present identical output encoding sequences.

In pseudo-code, the heuristi algorithm could be synthetized as:

1 Calculate $3M$ new paths at time t.
2 Sort on increasing cost function.
3 When a path encoding output sequence occurs twice, remove most expensive path.
4 Store first M candidates at time t only.

An advantage of this new architecture is that we are free to chose M within the range $1 < M < 2^N$ and we can consequently increase the computational needs in a linear way. A smaller M provides a more efficient solution, while a larger M gives a better approximation to the original Trellis converter. Table 5.1 compares the required resources for both architectures. The amount of work increases linearly in Minstead of exponentially in N. Increasing N while M is constant has little effect on the required computational power. The memory usage is also linear in M, except for the memory needed to store the output symbol history because we have to remember that the Latency is increasing as a function of N. However, the gain is still significant.

	Trellis SDM	**Efficient Trellis SDM**
Computational usage at each sample step	3^{N+1} path calculations	$3M$ path calculations
Memory usage	3^N filter states 3^N cost function values $3^N \cdot \textit{Latency}$ output symbols	M filter states M cost function values $M \cdot \textit{Latency}$ output symbols

Table 5.1: Needed computational resources for the Trellis SDM and for an heuristic-efficient Trellis SDM.

5.2 Full vs. Heuristic: Results

In this section we will present the results from the comparison between the complete (or Full) Trellis algorithm and the heuristic efficient one; our primary goal is to verify that the improved SNR performance of the Trellis A/D converter is fully preserved, thus permitting to use a large value for the Trellis order N and a much smaller value for M, increasing the computational efficiency. Comparative simulations have been run using, thanks to the faster efficient algorithm, real music sampled at 24 bits and $48kHz$. This music signals have been properly oversampled ($OSR = 64$) and then converted using both encoding systems. The typical sample lenght was approx 1 sec, thus containing more than 3 millions samples.

5.2.1 SNR: Full vs. Heuristic

The first comparative table is 5.2 and contains the SNR for both encoding algorithms; it show clearly that the quality is fully preserved and it suggests the designer to increase the Trellis order N to fully exploit the large advantage of using a small value for M, i.e. the bound restricting the number of encoding paths.

Trellis Order	**SNR[dB] Full Trellis**	**SNR[dB] Efficient Trellis**
2	112.8	111.2
3	114.0	113.2
4	114.2	113.9
5	113.9	114.0
6	114.4	114.1
7	114.2	107.8
8	114.3	114.3

Table 5.2: SNR comparison for the complete Trellis algorithm and the heuristic efficient one, depending on Trellis order N.

In all the simulations presented here the bound M to the number of encoding Path was choosen following this criteria:

- for $N = 1, M = 2$;
- for $2 < N \leq 4, M = 4$;
- for $N > 4, M = 24$;

so, we are really exploiting the computational efficiency of the heuristic approximation only for larger values of N.

5.2.2 Power: Full vs. Heuristic

The second original goal of our Trellis converter is to minimize the power of the encoded stream, using a three-levels algorithm. This is again preserved in the heuristic efficient implementation, and it is presented in table 5.3. As we can see the total power of the encoded stream is slightly larger than in the complete Trellis architecture, but the gap tends to decrease when increasing the Trellis order N. Again this results suggest using large values for N.

Trellis order	**Stream Power** **Full Trellis**	**Stream Power** **Efficient Trellis**
2	0.3565	0.6743
3	0.3529	0.4900
4	0.3501	0.4322
5	0.3465	0.4043
6	0.3446	0.3891
7	0.3436	0.3798
8	0.3421	0.3736

Table 5.3: Power of the encoded stream evaluated using equation 4.3 for the complete Trellis algorithm and the heuristic efficient one, depending on Trellis order N.

5.2.3 CPU saving with same quality

Once clarified that both SNR and power saving are preserved by the efficient algorithm we can investigate further the time saving performance indicator. We choosed the CPU time as figure of merit and we evaluated, with different Trellis orders, a fixed musical input signal.

Trellis order	**CPU Time [s]** **Full Trellis**	**CPU Time[s]** **Efficient Trellis**
2	7.08	18.46
3	19.61	22.36
4	58.32	24.33
5	208.24	25.96
6	849.53	27.40
7	2565.09	28.62
8	7749.16	29.86

Table 5.4: CPU time needed for encoding a fixed length of musical signal, depending on Trellis order N.

It's interesting to notice that the software implementation of the efficient algorithm has a little overhead that results in a slightly poorer performance for very small Trellis dimensions ($N = 2, 3$). This is due to the use of lists as data structure of choice; keeping the list of path ordered by their value of the weight function requires an "ordered insertion" step at every time shift of the algorithm. This is a fixed cost, because of the fixed length of the list itsel, thus reducing its impact on the overall performance when the Trellis order increases. Indeed, the computational saving for $N > 3$ is drammatically evident, because the time needed buy the complete Trellis explodes in exponential way, translating into a seriously demanding task the encoding for 1 second of music. We can plot the data contained in table 5.4 and we obtain the figure 5.4.

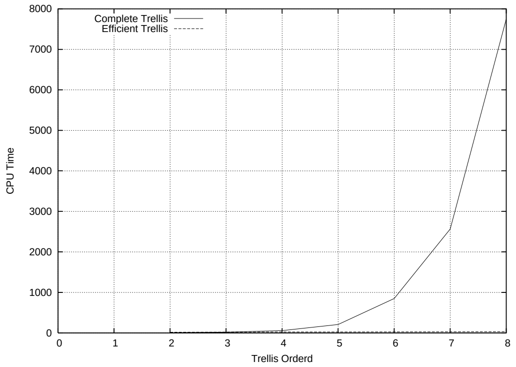


Figure 5.4: CPU time needed for encoding a fixed length of musical signal, depending on Trellis order N.

The line for the efficient algorithm is "buried" in the X-axis, so we can expand the vertical span using a logarithmic scale, as shown in figure 5.5.

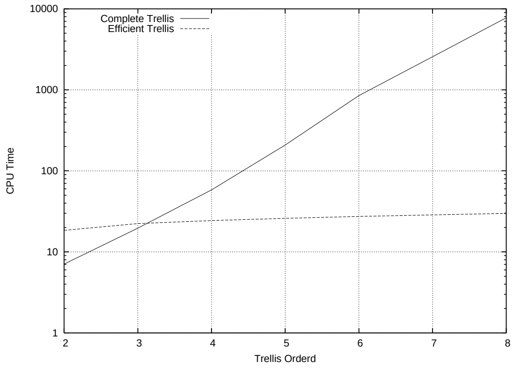


Figure 5.5: CPU time needed for encoding a fixed length of musical signal, depending on Trellis order N. Time expressed in Log scale.

So, we showd that this efficient Trellis algorithm is capable of keeping the same accuracy in the three levels modulation encoding, while requiring an almost constant time to run; exploiting this new tool we can now cope with longer simulations or we can design a complete music encoder that can convert long sections of musical content, namely complete songs. So the first investigation that is interesting because it can add another figure of merit to the three levels Trellis modulation is the one fucused on *idle tones*.

5.3 Idle tones

Idle tones are spurious tones typical of Σ-Δ conversion that are caused by the structure of the converter itself. When feeded with a continous input (a D.C value) at every sample instant the incoming input is compared with a fixed trashold and rounded with $[-1, 1]$ symbols. This operation is mathematically equivalent to a division by two, thus creating a periodic residual "tail" in the encoded signal if the input value is not in irrational proportion with the number of the quantization steps (in the classical Σ-Δ design two) [10]. Normally we don't have DC components in music signals, but of course we do have low frequency content that goes down till few Hertz. So it's a figure of merit of Σ-Δ's performance the evaluation of the presence (or not) of idle tones. The problem of evaluating idle tones is that very long simulations are required in order to avarage the output sequence power spectra and be able to identify small peaks often buried into the noise floor.

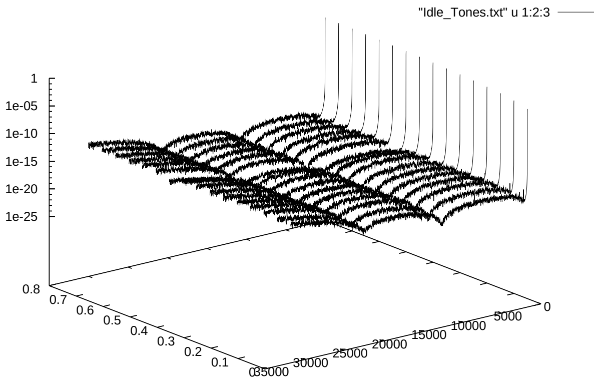


Figure 5.6: The heuristic efficient Trellis Σ-Δ does not present traces of idle tones for DC amplitudes varying from 0.05 to 0.85. The converter is also stable for all these DC inputs.

Figure 5.6 and 5.7 clearly show that there are no spurious components deriving from any direct component, no matter the intensity of the input. The power spectrum is indeed completely harmonic free even for the highest value of the amplitude ($A = 0.85$). This is another positive feature of three levels

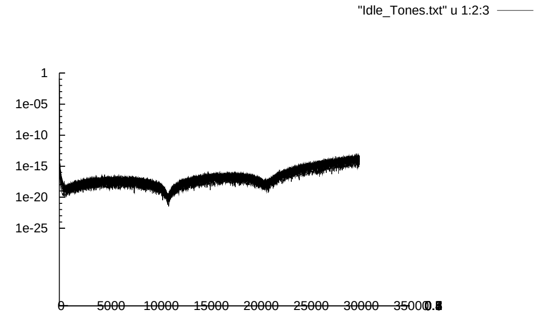


Figure 5.7: A different perspective that shows the typical power spectra profile of Σ-Δ modulation. Again there is no presence of idle tones.

Trellis modulation and this result was really requiring the efficient algorithm to be verified.

5.4 A complete music encoder

As we said in a previous section we can conclude saying that a complete music encoder able to converts all the possible WAVE file with different resolutions and samplig frequency into a three levels high quality stream has been developed and tested. This encoder has been proved to be superior to the normal SACD encoding system both on quality of the encoded signal, expressed by SNR, and power saving capability, expressed by the evaluation of the power of encoded stream. An heuristic efficient algorithm has been tailored to increase the computational efficiency.

Chapter 6

Frequency Warped Σ-Δ Modulator

After presenting an efficient algorithm that reduces the computational needs of the Trellis three-levels Σ-Δ modulation, in this chapter, a novel Σ-Δ Modulator design procedure is presented; our interest is to develop a new general model using a frequency-warped modulator filter structure. The proposed design procedure is based on warping the SDM noise transfer function (NTF) using the bi-linear all-pass transformation and it allows for some flexibility in the way the signal transfer function (STF) is affected by the warping. Both the analytical representation and a practical architecture implementation are described for first-order and high-order frequency-warped modulator filters. Topology examples are included to illustrate the effect of frequency warping on the STF and NTF frequency responses. For the high-order case, a realizable warped filter structure without algebraic signal loops is derived. Simulation results using sinusoidal test signals indicate a superior performance of the frequency-warped SDMs as compared to the topologies presented in the previous chapters.

6.1 Frequency-Warping in the noise shaping filter

As we described in the first chapter, we can think of the whole Σ-Δ as a device in which the feedback signal represents the quantization noise, while the low-pass filter in the feedforward path is designed to cancel it, more specifically to process its frequency content, moving it to a part of the spectrum that is not important for the specific task. In the high quality audio application case, the quantization noise is shifted to a frequency interval that is higher than the audio band. The role of the low-pass modulator filter is, consequently, of fundamental importance to achieve the desired performance in terms of SNR and signal-to-noise-and-distortion ratio (SINAD). With the intent of increasing the performance of a SDM without affecting the complexity of the low-pass modulator filter, a new design procedure for the modulator filter will be introduced in the next paragraph. It is based on the theory of frequency warping [39]. The resulting SDM is expected to achieve a better SNR and SINAD compared to a

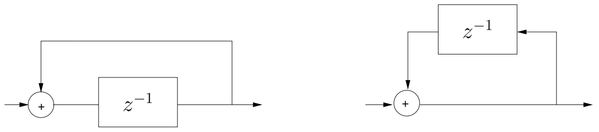


Figure 6.1: SDM low-pass filter building blocks: first-order integrator (left) and resonator (right).

traditional topology of the same order, due to an increased frequency selectivity of the warped modulator filter.

The two traditional building blocks for a SDM low-pass filter are a delay block surrounded by a local feedback loop (i.e., a first-order integrator) and a direct path with a delay in the local feedback loop (i.e., a first-order resonator), see Fig. 6.1. As stated in [35] we can immediately embed a frequency warping effect into a normal FIR or IIR filter by replacing every delay block with a bi-linear all-pass block, with warping parameter $\lambda \in (-1, 1)$:

$$z^{-1} \rightarrow \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}}. \tag{6.1}$$

This procedure is often used in audio filter design and leads to a peculiar effect: while leaving the periodicity unchanged it maps the frequency axis onto itself, resulting in a part of the frequency axis compressed and another one dilated, with

$$z^{-1} = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} \quad \forall f = k/T, \, where \; z = e^{j2\pi fT}, \; k = 0, 1, \dots \tag{6.2}$$

For $\lambda > 0$, the part of the spectrum lower than the so-called turning point frequency f_{TP} is expanded and the higher part is compressed (and viceversa, for $\lambda < 0$), with f_{TP} defined as [35]

$$f_{\mathrm{TP}} = \frac{f_S}{2\pi} \arccos(\lambda) \tag{6.3}$$

and f_S the sampling frequency.

As already proposed in lossy audio coding techniques [37], a non-uniform representation of the frequency axis can be a powerful tool for increasing the frequency resolution in the low part of the spectrum, to emulate human hearing, which indeed approximately has a constant-Q analysis capability [42]. By embedding this feature directly into the SDM, we can expect a better performance because of the more selective noise shaping. In our audio applications, we showed that it's preferred to have a high symbol rate (i.e., a high oversampling ratio (OSR)) and relatively simple low-pass filters, made up of only a few integrator stages. The usage of frequency warping in a first-order SDM architecture was described in [32] and yields an SDM design with a lower noise transfer function (NTF) Nyquist gain (also called "out-of-band gain"), which is important for stability reasons [29]. Moreover, the signal transfer function (STF) is shaped to be more frequency selective. These advantages can be achieved without significantly changing the complexity of the filter. On the other hand, there are other applications, such as telecommunication applications, in which the applicable OSR is limited due to the large signal bandwidth, hence the use of higher-order modulator topologies is required [31].

Anyway, replacing directly the integrator in the traditional first-order SDM filter offers some cues to reflect on. First of all, we are warping the NTF and STF simultaneously and, more subtly, we are now introducing a direct path from the input to the quantizer and consequently to the output. This direct path is not present in the original implementation and may introduce an implementation problem if there is no delay block in the feedback path such that an algebraic loop is constructed. We will therefore propose a design procedure in which first the NTF is warped and subsequently scaled to avoid a direct path from the input to the quantizer. In a second step, a decision has to be made on how the STF will be affected by the frequency warping. In particular, we will propose two first-order warped SDM topologies sharing the same NTF, but having a different STF. We will also mention the predicted SNR results obtained in [32], for both white noise and sinusoidal input signals. In section 6.3, the design procedure proposed in section 6.2 is generalized to high-order SDM filters. This is illustrated for the cascade-of-integrators feedforward (CIFF) SDM topology [41, Ch. 4]. Again, we start by warping and scaling the NTF, and then propose two different ways in which the warping may alter the STF, leading to two high-order frequency-warped SDM topologies. Since the local resonator feedback loops, which may be present in the traditional CIFF topology, produce algebraic loops after frequency warping, we also describe a method, inspired by [38], for transforming an unrealizable delay-free local resonator feedback loop into a realizable structure. Finally, in section 6.4, we present some simulation results of first-order and fifth-order frequency-warped SDMs and compare them to traditional SDMs in terms of SNR.

Notation

In this section, an analytical model is presented which can be used for both first-order and high-order SDMs. The notation used is adopted from [29]. We can represent a SDM as a two-input, single-output device as depicted in Fig. 6.2.

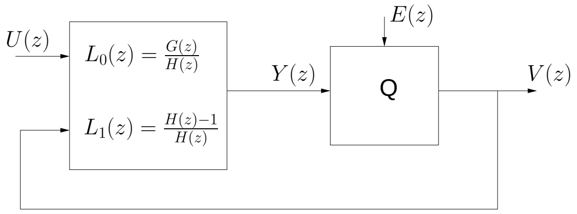


Figure 6.2: General block diagram of a single-quantizer SDM.

Here, $G(z)$ denotes the STF and $H(z)$ the NTF. $U(z)$ is the input signal and $V(z)$ is the encoded output. If we define the quantization error signal as $E(z) = V(z) - Y(z)$, we can present the classical input-output relation for the modulator, that is

$$V(z) = G(z)U(z) + H(z)E(z) \tag{6.4}$$

and the so-called loop filters are defined as

$$L_0(z) = \frac{G(z)}{H(z)} \tag{6.5}$$

and

$$L_1(z) = \frac{H(z) - 1}{H(z)}. \tag{6.6}$$

It is important to remember that in this simplified model the noise $E(z)$ is actually signal-dependent; anyway in our SNR prediction model we still use the white noise approximation and the i.i.d. assumption as stated in chapter 1.

6.2 First-Order Frequency-Warped SDM Design

6.2.1 Design Procedure

To illustrate the concept of designing a frequency-warped SDM, we start by considering the first-order case [32]. The noise shaping function is then a first-order pure differentiator, with a transfer function

$$H^{(1)}(z) = 1 - z^{-1} \tag{6.7}$$

in which the superscript is included to distinguish between the first-order and high-order case (see section 6.3). The NTF can be warped by replacing the complex variable z with the bi-linear all-pass function $\frac{z-\lambda}{1-\lambda z}$ (or, equivalently, by replacing z^{-1} with $\frac{z^{-1}-\lambda}{1-\lambda z^{-1}}$), i.e.,

$$H^{(1)}(z, \lambda) = 1 - \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} = \frac{(1+\lambda)(1 - z^{-1})}{1 - \lambda z^{-1}}. \tag{6.8}$$

As pointed out in [29], the NTF should always be scaled such that the first tap of its impulse response is 1. This is necessary to assure that the L_1 loop filter contains at least one pure time delay, hence avoiding an algebraic loop in the closed-loop SDM scheme. The first tap of the impulse response of $H^{(1)}(z, \lambda)$ equals $(1+\lambda)$, hence the desired warped and scaled NTF is obtained by dividing the transfer function in (6.8) by $(1 + \lambda)$, i.e.,

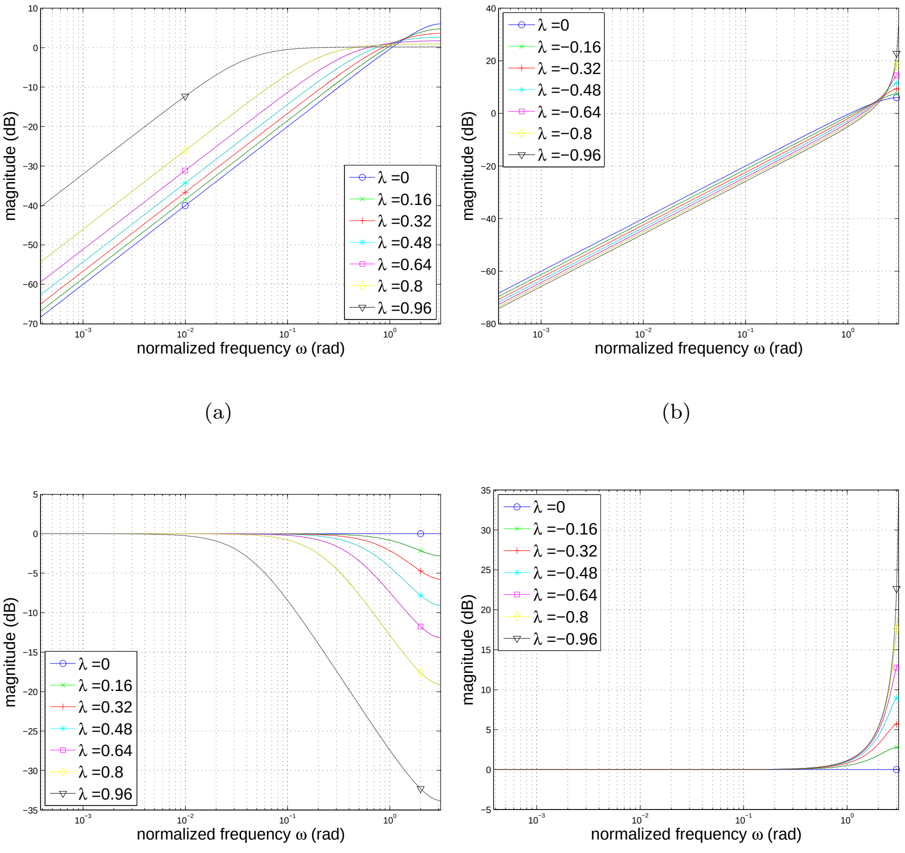
$$\tilde{H}^{(1)}(z, \lambda) = \frac{1 - z^{-1}}{1 - \lambda z^{-1}}. \tag{6.9}$$

The magnitude response of $\tilde{H}^{(1)}(z, \lambda)$ is shown in Figs. 6.3(a) and 6.3(b) for positive and negative warping parameters, respectively.

The corresponding warped loop filter $\tilde{L}_1^{(1)}(z, \lambda)$ can now be calculated as follows:

$$\tilde{L}_1^{(1)}(z, \lambda) = \frac{\tilde{H}^{(1)}(z, \lambda) - 1}{\tilde{H}^{(1)}(z, \lambda)} = -(1 - \lambda)\frac{z^{-1}}{1 - z^{-1}}. \tag{6.10}$$

At this point the warped SDM design allows for some freedom in deciding how the frequency warping will affect the STF $G^{(1)}(z)$. The effect on $G^{(1)}(z)$ follows directly from the choice of the loop filter $\tilde{L}_0^{(1)}(z, \lambda)$. There is complete freedom for choosing the loop filter L_0 and here we present a very simple one, that follows



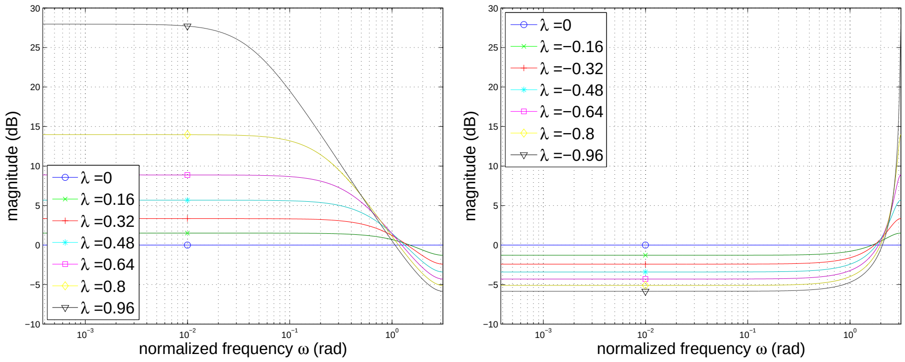
(a)

(b)

(c)

(d)

Figure 6.3: First-order frequency-warped SDM: (a) NTF magnitude response ($\lambda \geq 0$), (b) NTF magnitude response ($\lambda \leq 0$), (c) STF magnitude response, (topology I, $\lambda \geq 0$), (d) STF magnitude response (topology I, $\lambda \leq 0$).



(a)

(b)

Figure 6.4: First-order frequency-warped SDM: (a) STF magnitude response (topology II, $\lambda \geq 0$), (b) STF magnitude response (topology II, $\lambda \leq 0$).

a cost criteria; we wanted the filter to be as similar as possible to the original filter loop embedded in a SDM, reducing any additional complexity.

So, a *first possibility* is to constrain the warped loop filters $\tilde{L}_0^{(1)}(z,\lambda)$ and $\tilde{L}_1^{(1)}(z,\lambda)$ to obey the same relationship as they do in the non-warped case, i.e.,

$$L_0^{(1)}(z) = -L_1^{(1)}(z) \Rightarrow \tilde{L}_0^{(1)}(z,\lambda) = -\tilde{L}_1^{(1)}(z,\lambda). \tag{6.11}$$

This choice leads to

$$\tilde{L}_{0,\mathrm{I}}^{(1)}(z,\lambda) = (1-\lambda)\frac{z^{-1}}{1-z^{-1}} \tag{6.12}$$

which corresponds to the first-order SDM topology shown in Fig. 6.5, denoted as topology I. It can be seen that the warping merely comes down to adding a gain factor $(1-\lambda)$ in the SDM forward path. The warped STF $\tilde{G}_{\mathrm{I}}^{(1)}(z,\lambda)$ is in this case given by

$$\tilde{G}_{\mathrm{I}}^{(1)}(z,\lambda) = \tilde{L}_{0,\mathrm{I}}^{(1)}(z,\lambda)\tilde{H}^{(1)}(z,\lambda) = (1-\lambda)\frac{z^{-1}}{1-\lambda z^{-1}} \tag{6.13}$$

which is a high-pass filter if $\lambda < 0$, a low-pass filter if $\lambda > 0$, and a pure time delay if $\lambda = 0$. The STF magnitude response is plotted in Figs. 6.3(c) and 6.3(d) for positive and negative warping parameters, respectively. For sufficiently large values of OSR and not too extreme choice of λ (i.e. not to close to -1 or 1) the topology I STF response is flat and equal to $0dB$. It's interesting to notice that a positive value of λ yields a low-pass STF, even in the first-order case (where the STF usually has a flat magnitude response).

A *second possibility* for completing the design of a first-order warped SDM is to constrain $\tilde{L}_0^{(1)}(z,\lambda)$ to be equal to the loop filter $L_0^{(1)}(z)$ in the non-warped first-order SDM, i.e.,

$$\tilde{L}_{0,\mathrm{II}}^{(1)}(z,\lambda) = \frac{z^{-1}}{1-z^{-1}}. \tag{6.14}$$

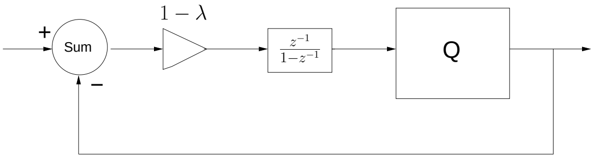


Figure 6.5: First-order frequency-warped SDM topology I ($\tilde{L}_0^{(1)}(z,\lambda) = -\tilde{L}_1^{(1)}(z,\lambda)$).

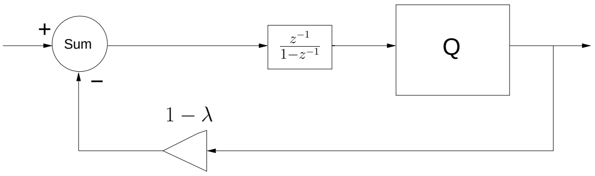


Figure 6.6: First-order frequency-warped SDM topology II ($\tilde{L}_0^{(1)}(z,\lambda) = L_0^{(1)}(z)$).

which leads to

$$\tilde{G}_{\mathrm{II}}^{(1)}(z,\lambda) = \frac{z^{-1}}{1-\lambda z^{-1}}. \tag{6.15}$$

The resulting topology, denoted as topology II, is shown in Fig. 6.6. It only differs from topology I in the position of the gain factor $(1-\lambda)$, which is now in the feedback path. Obviously, the NTF $\tilde{H}^{(1)}(z,\lambda)$ is the same for both topologies, whereas the STF $\tilde{G}_{\mathrm{II}}^{(1)}(z,\lambda)$ is a scaled version of the STF $\tilde{G}_{\mathrm{I}}^{(1)}(z,\lambda)$ in (6.13). The scaling is such that the gain at dc increases for increasing λ, whereas for topology I it is equal to 0 dB for all values of λ. This can be seen from the topology II STF magnitude response, plotted in Figs. 6.4(a) and 6.4(b) for positive and negative values of λ, respectively. For sufficiently large values of OSR and not too extreme choice of λ (i.e. not to close to -1 or 1), the topology II STF response is flat but, in contrast to topology I, it is *not* equal to $0dB$. Instead the in-band signal power increases for $\lambda > 0$, (thereby compensating for the observed noise power increase for $\lambda > 0$, see fig. 6.3(c)), and it decreases for $\lambda < 0$.

We should mention that, while the warped and scaled first-order NTF in (6.9) can be considered as a special case of the NTF of the so-called enhanced first-order SDM [31], the STFs proposed in (6.13) and (6.15) are not special cases of the STF in [31], due to the absence of a delay in the SDM feedback path. As a consequence, the results on SNR and the stability analysis presented in [31], cannot be applied to the first-order frequency-warped SDM topologies considered here.

The implementation of the first-order frequency-warped SDM topologies only requires one additional multiplier, compared to the traditional implementation, either in the forward path or in the feedback path, hence the increase in computational complexity or hardware cost is marginal.To conclude, let us summarize the steps taken in the above design procedure:

1. choose a non-warped NTF $H(z)$,

2. calculate the warped NTF $H(z,\lambda)$ by replacing z with the bi-linear all-pass function $\frac{z-\lambda}{1-\lambda z}$,
3. scale $H(z,\lambda)$ such that a scaled and warped NTF $\tilde{H}(z,\lambda)$ is obtained, which has an impulse response with the first coefficient equal to 1,
4. decide in which way the warping should affect the STF by constraining the design of the loop filter $\tilde{L}_0(z,\lambda)$, e.g.,
   - (a) constrain the new loop filter $\tilde{L}_0(z,\lambda)$ to be equal to the non-warped loop filter $L_0(z)$,
   - (b) constrain the relationship between the new loop filters $\tilde{L}_0(z,\lambda)$ and $\tilde{L}_1(z,\lambda)$ to be the same as the relationship between the old loop filters, i.e., $L_0(z) = f\{L_1(z)\} \Rightarrow \tilde{L}_0(z,\lambda) = f\{\tilde{L}_1(z,\lambda)\}$, with $f\{\cdot\}$ a linear function,
5. evaluate the warped NTF $\tilde{H}(z,\lambda)$ and the warped STF $\tilde{G}(z,\lambda) = \tilde{L}_0(z,\lambda)\tilde{H}(z,\lambda)$, such as to properly choose the warping parameter λ.

6.2.2 SNR Prediction

In [32], the SNR was predicted using the analytical model for the first-order frequency-warped SDM, and given by (for $\lambda \neq 0$)); in this calculus we have or white noise or sinusoidal signal as input, and σ_{wn}^2 or $1/2A^2$ are the signal power (A is the amplitude of the sinusoid). The power of the noise is indicated as σ_e^2. The band of interest is $f_B = 24\,kHz$ and in the case of sinusoidal input the frequency of the signal is $f_0 = 1\,kHz$. We define also $\omega_0 = (f_0/f_B)(\pi/\mathrm{OSR})$ thus becoming $\omega_0 = 2.0453e-3$ rad.

$$\mathrm{SNR_I(WN)} = 10\log_{10}\frac{\sigma_{wn}^2}{\sigma_e^2}\lambda\frac{1-\lambda}{1+\lambda}\frac{\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)}{\frac{\pi}{2\mathrm{OSR}}-\frac{1-\lambda}{1+\lambda}\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)} \tag{6.16}$$

$$\mathrm{SNR_{II}(WN)} = 10\log_{10}\frac{\sigma_{wn}^2}{\sigma_e^2}\frac{\lambda}{1-\lambda^2}\frac{\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)}{\frac{\pi}{2\mathrm{OSR}}-\frac{1-\lambda}{1+\lambda}\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)} \tag{6.17}$$

$$\mathrm{SNR_I(SIN)} = 10\log_{10}\frac{1/2\,A^2}{\sigma_e^2}\lambda\frac{(1-\lambda)^2}{1-2\lambda\cos\omega_0+\lambda^2}\frac{1}{1-\frac{2\mathrm{OSR}}{\pi}\frac{1-\lambda}{1+\lambda}\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)} \tag{6.18}$$

$$\mathrm{SNR_{II}(SIN)} = 10\log_{10}\frac{1/2\,A^2}{\sigma_e^2}\frac{\lambda}{1-2\lambda\cos\omega_0+\lambda^2}\frac{1}{1-\frac{2\mathrm{OSR}}{\pi}\frac{1-\lambda}{1+\lambda}\arctan\left(\frac{1+\lambda}{1-\lambda}\tan\frac{\pi}{2\mathrm{OSR}}\right)} \tag{6.19}$$

for topologies I and II, and in case of white noise (WN) and sinusoidal (SIN) input signals (with radial frequency $\omega_0 \in [0,\pi/\mathrm{OSR}]$), respectively.

The predicted SNR is plotted versus the warping parameter $\lambda \in (-1,1)$ in Fig. 6.7 for $\sigma_x^2/\sigma_e^2 = 1$, OSR = 64. For both signal types, we observe that the SNR of topology I decreases as $\lambda \to 1$, while it increases somewhat as $\lambda \to -1$. On the other hand, the SNR of topology II is predicted to increase strongly for $\lambda \to 1$, while it remains more or less constant for other values of λ. This behavior could be expected by examining the NTF and STF magnitude responses in Figs. 6.3 and 6.4. The warped NTF magnitude response rises significantly in the signal band at high OSRs when $\lambda \to 1$ (e.g., for $\lambda = 0.8$, the NTF magnitude response is 10 dB larger than the non-warped NTF response if $\omega < 3 \cdot 10^{-1}$ rad, corresponding to the entire signal band for OSRs larger than 10, see Fig. 6.3(a)). On the other hand, the STF magnitude response remains constant or even decreases (for positive λ) within the signal band for

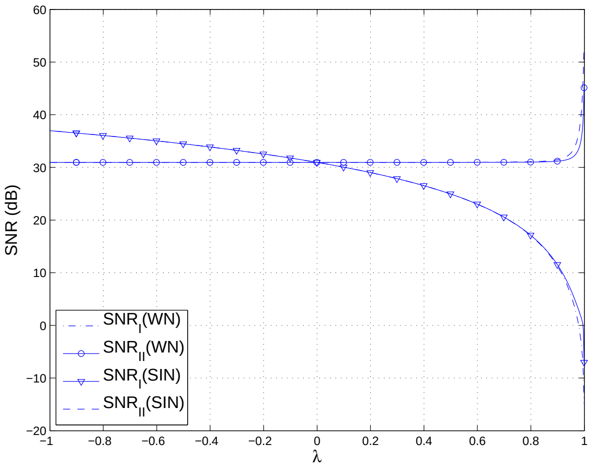


Figure 6.7: Predicted SNR vs. warping parameter for first-order frequency-warped SDM topologies I and II (WN = white noise input, SIN = sinusoidal input).

the warped SDM topology I, see Fig. 6.3(c), which obviously leads to a loss of SNR. In contrast, the STF magnitude response of topology II shows a boost in the signal band, which increases as $\lambda \rightarrow 1$, and which (over)compensates for the increase in the NTF response, see Fig. 6.4(a). From this analysis, topology II with the positive range of the warping parameter is expected to be most promising in terms of SNR improvement. This observation will be confirmed by simulation results in section 6.4.1.

6.3 High-Order Frequency-Warped SDM Design

6.3.1 Design Procedure

We want now to generalize the design procedure outlined in section 6.2 to the case of high-order SDMs. The high-order frequency-warped SDM design is illustrated for a commonly used SDM topology, namely the cascade-of-integrators feedforward (CIFF) form [41, Ch. 4]. It should however be stressed that the design procedure is also applicable to other topologies.

High-order SDM filters generally consist of two types of building blocks: first-order integrators and first-order resonators [41, Ch. 4]. Warping the NTF of a high-order SDM can be achieved by replacing these building blocks by their warped counterparts, i.e., the first-order warped integrator

$$\frac{z^{-1}}{1-z^{-1}} \rightarrow \frac{z^{-1}-\lambda}{(1+\lambda)(1-z^{-1})} = \frac{1}{1+\lambda}\Bigg(\underbrace{\frac{z^{-1}}{1-z^{-1}}}_{\text{integrator}} - \lambda \underbrace{\frac{1}{1-z^{-1}}}_{\text{resonator}}\Bigg) \qquad (6.20)$$

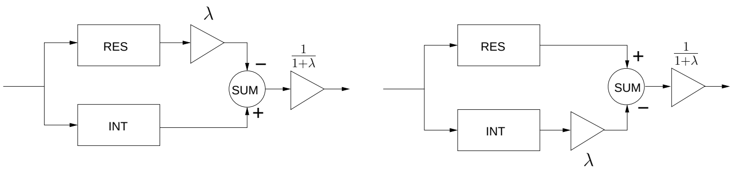


Figure 6.8: Parallel connections of a resonator and an integrator with appropriate scaling constitute the building blocks of the frequency-warped SDM filter.

and the first-order warped resonator

$$\frac{1}{1-z^{-1}} \rightarrow \frac{1-\lambda z^{-1}}{(1+\lambda)(1-z^{-1})} = \frac{1}{1+\lambda}\Big(\underbrace{\frac{1}{1-z^{-1}}}_{\text{resonator}} - \lambda \underbrace{\frac{z^{-1}}{1-z^{-1}}}_{\text{integrator}}\Big). \tag{6.21}$$

It is observed that the frequency warping of the first-order integrator and resonator functions comes down to replacing these functions by a parallel connection of a first-order integrator-resonator pair, scaled with $(1+\lambda)^{-1}$, and in which the contribution of either the resonator or the integrator is weighted with the negative warping parameter $-\lambda$ (see Fig. 6.8).

Consider the N-Th order CIFF topology, illustrated for $N = 5$ in Fig. 6.9. We will only deal with odd SDM orders, yet the design can easily be extended to the even order case. The NTF for this topology can be written as follows,

$$H^{(N)}(z) = \frac{1}{1+\sum_{k=1}^{N} a_k \Big(\frac{z^{-1}}{1-z^{-1}}\Big)^{p(k)} \prod_{j=1}^{\frac{k-p(k)}{2}} \frac{z^{-2}}{(1-z^{-1})^2 - g_j z^{-2}}} \tag{6.22}$$

where a_k are the feedforward coefficients, g_j are the coefficients of the local resonator feedback loops, and the parity function $p(k)$ is defined as

$$p(k) = \begin{cases} 0 & \text{if } k \text{ is even} \\ 1 & \text{if } k \text{ is odd} \end{cases} \tag{6.23}$$

Substituting (6.20) in the NTF transfer function (6.22) yields the following expression for the warped NTF:

$$H^{(N)}(z,\lambda) = \frac{1}{1+\sum_{k=1}^{N} a_k \Big(\frac{z^{-1}-\lambda}{(1+\lambda)(1-z^{-1})}\Big)^{p(k)} \prod_{j=1}^{\frac{k-p(k)}{2}} \frac{(z^{-1}-\lambda)^2}{(1+\lambda)^2(1-z^{-1})^2 - g_j(z^{-1}-\lambda)^2}}. \tag{6.24}$$

The first tap of the impulse response $h^{(N)}(t,\lambda)$ of the warped NTF is

$$h^{(N)}(0,\lambda) = \frac{1}{1+\sum_{k=1}^{N} a_k \Big(\frac{-\lambda}{1+\lambda}\Big)^{p(k)} \prod_{j=1}^{\frac{k-p(k)}{2}} \frac{\lambda^2}{(1+\lambda)^2 - g_j \lambda^2}}. \tag{6.25}$$

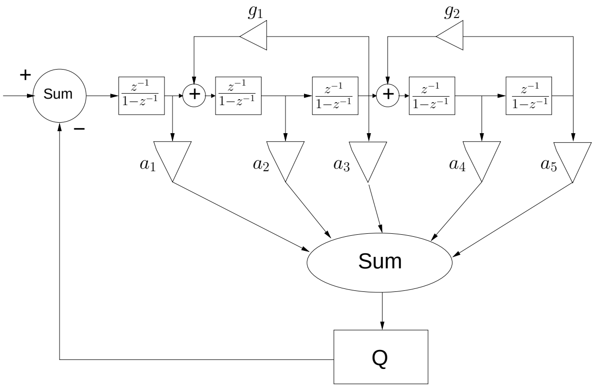


Figure 6.9: High-order ($N = 5$) SDM: CIFF topology.

such that the scaled and warped NTF can be written as follows,

$$\tilde{H}^{(N)}(z,\lambda) = \frac{1-\alpha_0}{1+\displaystyle\sum_{k=1}^{N}\alpha_k\Big(\frac{z^{-1}-\lambda}{1-z^{-1}}\Big)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{(z^{-1}-\lambda)^2}{(1-z^{-1})^2-\gamma_j(z^{-1}-\lambda)^2}} \tag{6.26}$$

with the new feedforward and local resonator feedback coefficients defined by

$$\alpha_k = \frac{a_k}{(1+\lambda)^k}, \qquad k = 1,\ldots,N \tag{6.27}$$

$$\gamma_j = \frac{g_j}{(1+\lambda)^2}, \qquad j = 1,\ldots,\frac{N-1}{2} \tag{6.28}$$

and the direct feedforward path coefficient defined as

$$\alpha_0 = -\sum_{k=1}^{N}\alpha_k(-\lambda)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}. \tag{6.29}$$

From the scaled and warped NTF in (6.26), the corresponding L_1 loop filter is calculated using the leftmost equality in (6.10), i.e.,

$$\tilde{L}_1^{(N)}(z,\lambda) = \frac{-\alpha_0-\displaystyle\sum_{k=1}^{N}\alpha_k\Big(\frac{z^{-1}-\lambda}{1-z^{-1}}\Big)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{(z^{-1}-\lambda)^2}{(1-z^{-1})^2-\gamma_j(z^{-1}-\lambda)^2}}{1-\alpha_0}. \tag{6.30}$$

The next step is to decide in which way the warping should affect the high-order STF, by constraining the L_0 loop filter. A straightforward choice is to put $\tilde{L}_0^{(N)}(z,\lambda) = -\tilde{L}_1^{(N)}(z,\lambda)$, which is similar to the constraint that resulted in the first-order frequency-warped SDM topology I, see (6.11). In this case, the

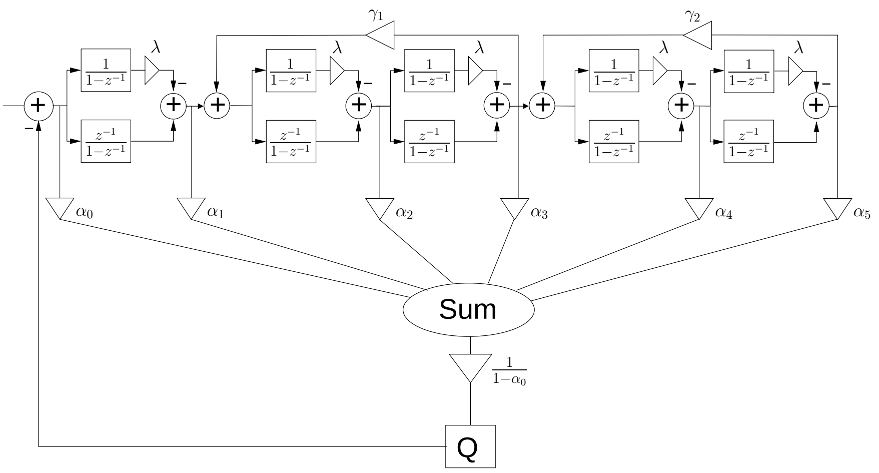


Figure 6.10: High-order ($N = 5$) frequency-warped SDM: CIFF topology I ($\tilde{L}_0^{(N)}(z,\lambda) = -\tilde{L}_1^{(N)}(z,\lambda)$).

L_0 loop filter has a transfer function

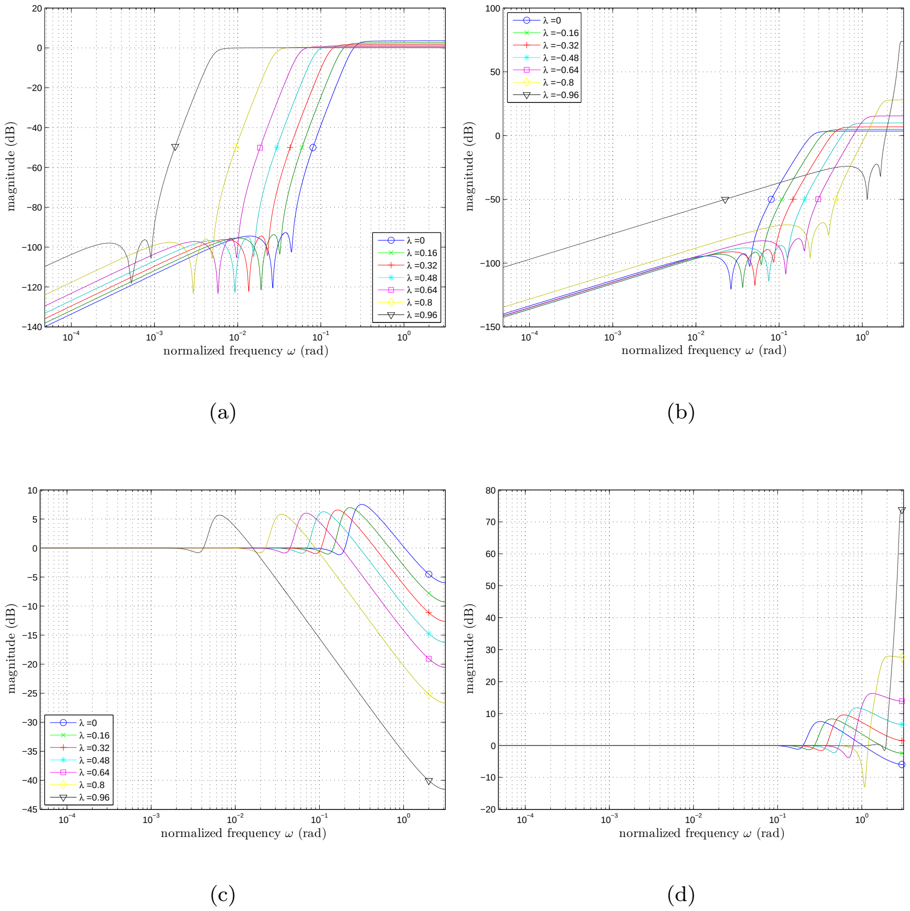
$$\tilde{L}_{0,\mathrm{I}}^{(N)}(z,\lambda) = \frac{\alpha_0 + \sum_{k=1}^{N} \alpha_k \left(\frac{z^{-1}-\lambda}{1-z^{-1}}\right)^{p(k)} \prod_{j=1}^{\frac{k-p(k)}{2}} \frac{(z^{-1}-\lambda)^2}{(1-z^{-1})^2-\gamma_j(z^{-1}-\lambda)^2}}{1-\alpha_0} \tag{6.31}$$

hence the corresponding high-order warped STF can be calculated as

$$\tilde{G}_{\mathrm{I}}^{(N)}(z,\lambda) = 1 - \tilde{H}^{(N)}(z,\lambda). \tag{6.32}$$

The resulting high-order frequency-warped SDM CIFF topology I is shown in Fig. 6.10. The magnitude response of the NTF and STF is plotted in Figs. 6.11(a)-6.11(d), for positive and negative values of the warping parameter λ. Examining Figs. 6.11(a) and 6.11(c), we notice that the NTF magnitude response rises in the signal band as $\lambda \to 1$, while in this frequency band the STF magnitude response remains constant. The same effect was observed in Section 6.2.2 for the first-order frequency-warped SDM topology I, and was presumed to be the main cause for the predicted SNR decrease as $\lambda \to 1$. Hence the warped high-order topology I described above can also be expected to perform poorly in terms of SNR for positive λ. However, in terms of SDM stability, the positive range of the warping parameter is more interesting than the negative range, since it yields a NTF with decreasing out-of-band gain [29] as $\lambda \to 1$, see Fig. 6.11(a).

It is also possible to derive a different warped high-order topology from (6.30) which has a scaling factor $1-\lambda$ in the SDM feedback path, similar to the topology II derived in the first-order case (see Section 6.2.1). However, the constraint $\tilde{L}_0(z,\lambda) = L_0(z)$, which was used to derive the first-order topology II, is not very attractive in the high-order case since the loop filters $\tilde{L}_0^{(N)}(z,\lambda)$ and $\tilde{L}_1^{(N)}(z,\lambda)$ would then strongly differ and not share many components. Nevertheless, there is a different way of interpreting the first-order topology II, which does allow



(a)

(b)

(c)

(d)

Figure 6.11: High-order ($N = 5$) frequency-warped SDM CIFF topology I: (a) NTF magnitude response ($\lambda \geq 0$), (b) NTF magnitude response ($\lambda \leq 0$), (c) STF magnitude response ($\lambda \geq 0$), (d) STF magnitude response ($\lambda \leq 0$).

for a generalization to the high-order case with preservation of the basic SDM structure (i.e., with $\tilde{L}_0^{(N)}(z,\lambda)$ and $\tilde{L}_1^{(N)}(z,\lambda)$ sharing most of their components). By comparing the first-order loop filter expressions in (6.10) and (6.14), the first-order topology II can also be derived by using the constraint

$$\tilde{L}_0(z,\lambda) = -\frac{1}{1-\lambda}\tilde{L}_1(z,\lambda). \qquad (6.33)$$

Before the constraint in (6.33) can be applied to the high-order SDM design, a factor $(1-\lambda)$ from the warped L_1 loop filter transfer function in (6.30) has to be extracted. In the Appendix, it is shown that (6.30) can be rewritten as follows:

$$\tilde{L}_1^{(N)}(z,\lambda) = \Big(\frac{\lambda-1}{1-\alpha_0}\Big)\frac{z^{-1}}{1-z^{-1}}\cdot \\ \Bigg\{\alpha_1 + \sum_{l=1}^{\frac{N-1}{2}}\Bigg(\alpha_{2l+1} + \beta_l\frac{1-z^{-1}}{z^{-1}-\lambda} - \lambda\beta_l\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2\Bigg)\cdot \\ \prod_{j=1}^{l}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\Bigg\} \tag{6.34}$$

with

$$\beta_l = \sum_{k=2l}^{N}\alpha_k\frac{(-\lambda)^{p(k)}}{\lambda^2}\prod_{j=l}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}. \tag{6.35}$$

Applying the constraint in (6.33) to (6.34) yields the desired topology II expression for the L_0 loop filter,

$$\tilde{L}_{0,\mathrm{II}}^{(N)}(z,\lambda) = \Big(\frac{1}{1-\alpha_0}\Big)\frac{z^{-1}}{1-z^{-1}}\cdot \\ \Bigg\{\alpha_1 + \sum_{l=1}^{\frac{N-1}{2}}\Bigg(\alpha_{2l+1} + \beta_l\frac{1-z^{-1}}{z^{-1}-\lambda} - \lambda\beta_l\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2\Bigg)\cdot \\ \prod_{j=1}^{l}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\Bigg\} \tag{6.36}$$

which results in the following STF:

$$\tilde{G}_{\mathrm{II}}^{(N)}(z,\lambda) = \frac{1}{1-\lambda}\big(1-\tilde{H}^{(N)}(z,\lambda)\big). \tag{6.37}$$

The high-order frequency warped SDM CIFF topology II, as defined by (6.34) and (6.36), can be realized as shown in Fig. 6.12. The following differences with topology I (Fig. 6.10) can be observed: the first stage of the SDM filter is a traditional integrator instead of a frequency-warped integrator, the odd feedforward coefficients have changed, the feedback path contains a gain factor $1-\lambda$, and finally, $\frac{N-1}{2}$ feedforward branches have been added (which merge with the feedforward branches of the odd stages in case $\gamma_j = 0$). The STF magnitude response is plotted in Figs. 6.13(a) and 6.13(b) for positive and negative values of λ, respectively.

Comparing the NTF frequency responses in Figs. 6.11(a) and 6.11(b) with the corresponding responses in the first-order case (Figs. 6.3(a) and 6.3(b)), we notice the following. Whereas warping the first-order NTF causes a wideband shift of the SNR curves (towards 0 dB SNR for $\lambda \to 1$ and towards $-\infty$ for $\lambda \to -1$), warping the high-order NTF rather results in a noticeable shift of the cut-off frequency (towards dc for $\lambda \to 1$ and towards the Nyquist frequency for $\lambda \to -1$). As a consequence, warping the high-order SDM will result in widening ($\lambda > 0$) or narrowing ($\lambda < 0$) the high-frequency band according to which the quantization noise is shaped. Since high-order SDMs are usually applied to problems featuring a large signal bandwidth [31], e.g. in telecommunications, it

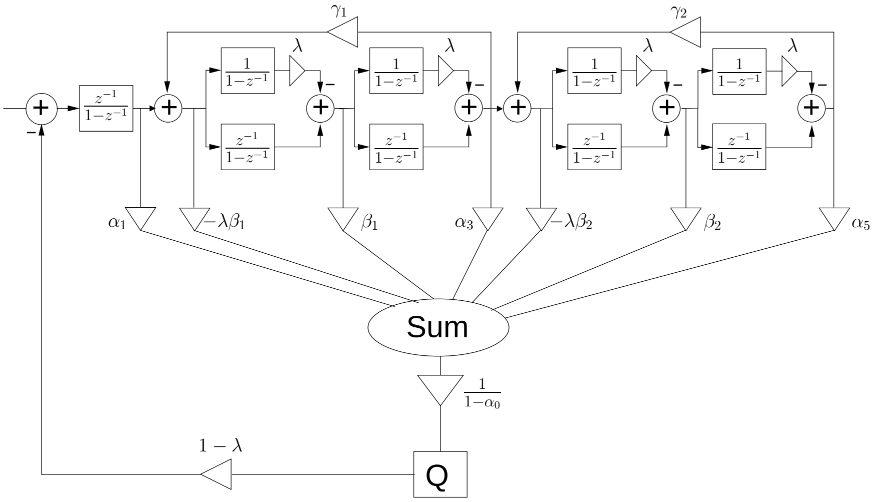


Figure 6.12: High-order ($N = 5$) frequency-warped SDM: CIFF topology II ($\tilde{L}_0^{(N)}(z,\lambda) = -\frac{1}{1-\lambda}\tilde{L}_1^{(N)}(z,\lambda)$).

is expected to be advantageous to use negative values of the warping parameter. This is confirmed by high-order SDM simulation results presented in Section 6.4.2.

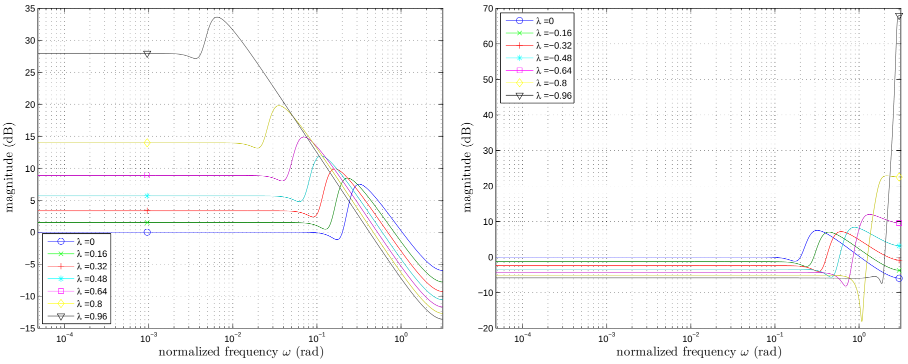
6.3.2 Implementation Issues

The high-order frequency-warped SDM CIFF topologies I and II are not realizable in the structure shown in Figs. 6.10 and 6.12, respectively, if both $\lambda \neq 0$ and $\gamma_j \neq 0$ for any $j \in \{1, \ldots, (N-1)/2\}$. This is due to the fact that the frequency warping introduces a delay-free path in each stage of the SDM filter, hence when a local feedback with coefficient γ_j is employed, an algebraic loop is created. This is a commonly encountered problem when designing warped IIR (WIIR) filters, and several solutions have been suggested [44]-[36]. The method proposed in [38] may be applied here to transform the non-realizable high-order SDM topologies described above into realizable structures.

The approach is illustrated by considering the j-Th section ($j \in \{1, \ldots, (N-1)/2\}$) of the high-order frequency warped SDM CIFF topology II, which is shown in Fig. 6.14 and consists of an odd and even SDM stage surrounded by a local feedback loop. First of all, this integrator-resonator structure is transformed into a direct-form II WIIR structure, see Fig. 6.15, which is still non-realizable. In this direct-form structure, the input and output signals of the first, second, and third unit delay element at time t are denoted as $x_0(t)$ and $y_0(t)$, $x_1(t)$ and $y_1(t)$, and $x_2(t)$ and $y_2(t)$, respectively, see Fig. 6.15. The feedback structure of the non-realizable direct-form II implementation is such that the state variable $x_0(t)$ is a function of the input $x(t)$ and the state variable $x_2(t)$, i.e.,

$$x_0(t) = x(t) + \gamma_j x_2(t). \tag{6.38}$$

If $x_0(t)$ is instead calculated using the input $x(t)$ and the state variables $y_i(t),\ i =$



(a)

(b)

Figure 6.13: High-order ($N = 5$) frequency-warped SDM CIFF topology II: (a) STF magnitude response ($\lambda \geq 0$), (b) STF magnitude response ($\lambda \leq 0$).

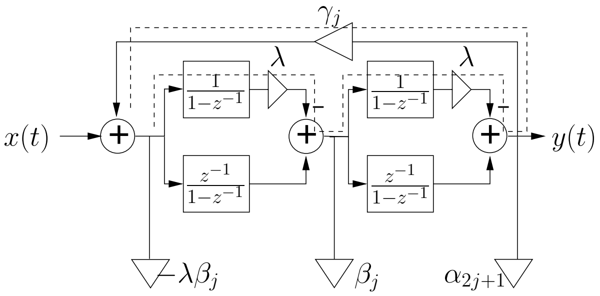


Figure 6.14: Non-realizable integrator-resonator structure of high-order frequency-warped SDM CIFF topology II section.

$0, 1, 2$, then a structure without algebraic loops can be obtained. Substituting

$$x_2(t) = y_1(t) + (y_2(t) - \lambda x_1(t)) \tag{6.39}$$

in (6.38) and

$$x_1(t) = y_0(t) + (y_1(t) - \lambda x_0(t)) \tag{6.40}$$

in (6.39),

$$x_0(t) = \frac{1}{1 - \gamma_j \lambda^2}\big(x(t) - \gamma_j \lambda y_0(t) + \gamma_j(1-\lambda)y_1(t) + \gamma_j y_2(t)\big) \tag{6.41}$$

which leads to the realizable structure shown in Fig. 6.16.

It should be noted that the feedforward branches of the transformed structure in Fig. 6.16 (with feedforward coefficients $-\lambda\beta_l, \beta_l, \alpha_{2l+1}$) are unchanged as compared to the original structure in Fig. 6.14. As a consequence, the above approach can equivalently be applied to the high-order frequency warped SDM CIFF topology I, which is made up of similar sections as the one shown in Fig. 6.14, but with different feedforward branches.

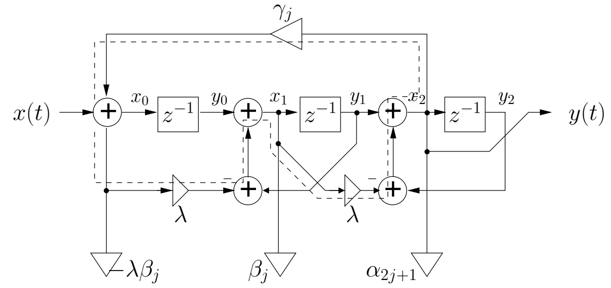


Figure 6.15: Non-realizable direct-form II WIIR structure of high-order frequency-warped SDM CIFF topology II section.

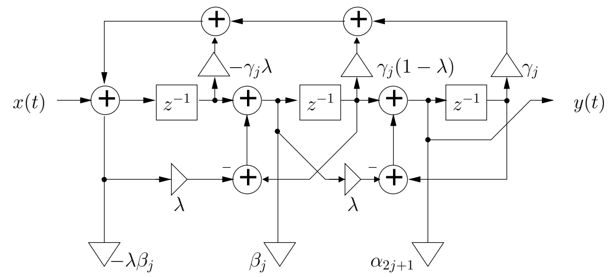


Figure 6.16: Realizable WIIR structure of high-order frequency-warped SDM CIFF topology II section.

6.4 Simulation Results

6.4.1 First-Order SDM

The positive range $[0, 1)$ of the warping parameter λ is of particular interest for first-order frequency-warped SDMs [32], since it renders the frequency warping perceptually relevant in audio applications [37],[42], and since it yields frequency-warped SDMs of which the NTF frequency response has a smaller out-of-band gain (see Figs. 6.3(a) and 6.11(a)), which is beneficial for SDM stability [29]. In section 6.2.2, it was shown that using the results derived in [32], the SNR is expected to increase for positive λ only when topology II is used.

Extensive simulations of more than $3 \cdot 10^6$ samples were run with sinusoidal input signals of variable amplitude, and with $f_0 = 1$ kHz. The signal bandwidth was $f_B = 24$ kHz and the OSR = 64, resulting in a sampling frequency $f_S = 3072$ kHz. The plot representing the SNR for both the traditional SDM and the frequency-warped SDM is shown in Fig. 6.17. For several positive values of the warping parameter λ we have found an improved SNR performance as compared to the traditional SDM, with an optimum average 6 dB gain for $\lambda = 0.45$. For larger values of λ several stability problems have aroused, so not such extensive simulation could be run. This does not mean that the whole SDM would be instable, because stability in SDM strictly depends on the time length of large signal peaks. Anyway, we prefer to refer only "fully-stable" results.

Also another set of simulation was held, with multiple sinusoidal input with

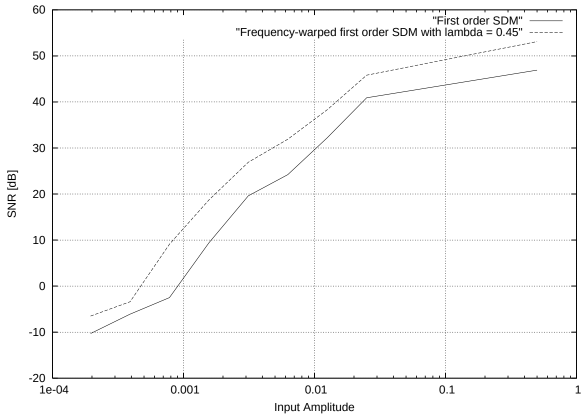


Figure 6.17: SNR comparison between a first-order traditional SDM and frequency-warped SDM (topology II) for $\lambda = 0.45$.

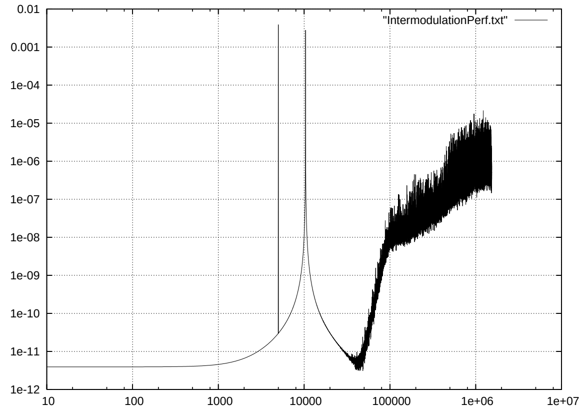


Figure 6.18: One example of simulation used in the intermodulation distortion analysis, with $f_2 = 2 \cdot f_1 + delta$, with $f_1 = 5\,kHz$, $A_1 = A_2 = -3\,dB$ SACD and $\lambda = 0.45$.

$f_2 = 2 \cdot f_1 + \delta$, with f_1 varying from $50\,Hz$ to $10\,kHz$. These simulation was done in order to evaluate eventual intermodulation distortion and possible “skirt” effects around the signal peaks in the power spectrum density. Eventually the encoding where virtually intermodulation distortion free and they did non present any difference in the noise floor, as depicted in the example shown in figure 6.18. In these simulation the total power of the input signal was equal to the power of the standard $0\,dB$ SACD test signal, meaning that each sinusoid amplitude was at $-3\,dB$.

6.4.2 High-Order SDM

As for the first-order case, a set of simulations were run for a fifth-order frequency-warped SDM CIFF topology II with sinusoidal input signals of variable amplitude and with $f_0 = 1$ kHz. Again, $f_B = 24$ kHz and OSR = 64, such that $f_S = 3072$ kHz. The plot representing the SNR for both the traditional SDM and the frequency-warped SDM is shown in Fig. 6.19. Also in the case of high-order SDM the simulation results show that frequency-warping leads to an improvement in the SNR performance, in this case for negative values of the warping parameter λ, with an optimum average 14 dB gain for $\lambda = -0.205$. Also here there were stability issues as stated before, so also in this case we refer only to simulations that offered stable behavior after more than $3 \cdot 10^6$ samples.

The SDM filter coefficients used in the simulation are obtained as follows: First, a fifth-order NTF is synthesized by placing all zeros at $z = 1$, and optimizing two complex-conjugate zero pairs for an oversampling ratio OSR $= 64$, using the method in [41, Ch. 8]. This NTF is then converted into a CIFF topology with a unit STF, which yields the feedforward and local resonator feedback coefficients: $a_1 = 0.80901, a_2 = 0.31489, a_3 = 0.072124, a_4 = 0.0096177, a_5 = 0.00051673$ and $g_1 = g_2 = 0$. Finally, the α_k and γ_j coefficients for the frequency-warped SDM CIFF topology I are calculated using (6.27) and (6.28), as follows,

$$\alpha_1 = \frac{a_1}{1+\lambda} \tag{6.42}$$

$$\alpha_2 = \frac{a_2}{(1+\lambda)^2} \tag{6.43}$$

$$\alpha_3 = \frac{a_3}{(1+\lambda)^3} \tag{6.44}$$

$$\alpha_4 = \frac{a_4}{(1+\lambda)^4} \tag{6.45}$$

$$\alpha_5 = \frac{a_5}{(1+\lambda)^5} \tag{6.46}$$

$$\gamma_1 = \frac{g_1}{(1+\lambda)^2} \tag{6.47}$$

$$\gamma_2 = \frac{g_2}{(1+\lambda)^2} \tag{6.48}$$

$$\alpha_0 = \alpha_1\lambda - \alpha_2\frac{\lambda^2}{1-\gamma_1\lambda^2} + \alpha_3\frac{\lambda^3}{1-\gamma_1\lambda^2} - \alpha_4\frac{\lambda^4}{(1-\gamma_1\lambda^2)(1-\gamma_2\lambda^2)} + \alpha_5\frac{\lambda^5}{(1-\gamma_1\lambda^2)(1-\gamma_2\lambda^2)} \tag{6.49}$$

and the β_l coefficients for topology II are obtained using (6.35):

$$\beta_1 = \alpha_2\frac{1}{1-\gamma_1\lambda^2} - \alpha_3\frac{\lambda}{1-\gamma_1\lambda^2} + \alpha_4\frac{\lambda^2}{(1-\gamma_1\lambda^2)(1-\gamma_2\lambda^2)} - \alpha_5\frac{\lambda^3}{(1-\gamma_1\lambda^2)(1-\gamma_2\lambda^2)} \tag{6.50}$$

$$\beta_2 = \alpha_4\frac{1}{1-\gamma_2\lambda^2} - \alpha_5\frac{\lambda}{1-\gamma_2\lambda^2}. \tag{6.51}$$

Also for the high order case we ran the multiple sinusoidal input simulation set, with $f_2 = 2 \cdot f_1 + \delta$, with f_1 varying from $50\,Hz$ to $10\,kHz$. Also in these simulations we preserved the total input signal power to the standard $0\,dB$ SACD, meaning that each sinusoid amplitude was at $-3\,dB$. So, also in this case we did not see any kind of intermodulation distortion nor higher noise floor due to the presence of two large input signals close to each other.

6.5 Conclusion

In this chapter we extended our previous investigation on authoring techniques using Σ-Δ modulation introducing an SDM design procedure that applies fre-

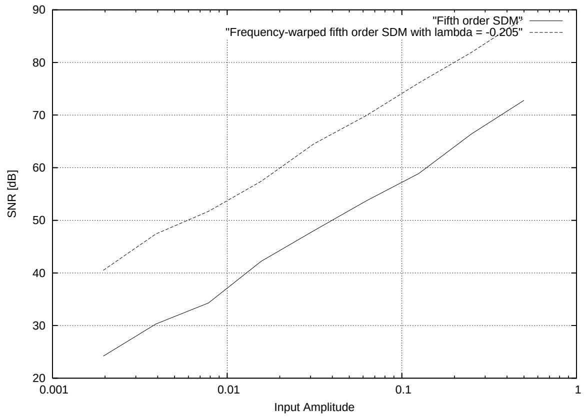


Figure 6.19: SNR comparison between a fifth-order traditional SDM and frequency-warped SDM (topology II) for $\lambda = -0.205$.

quency warping to the noise transfer function and to the signal transfer function that are embedded in the noise shaping process. We synthesized several SDM topologies that could be useful in different tasks but mainly in our audio application contest. In each case we explored both positive and negative values of the warping parameter λ, obtaining in both cases a better SNR performance. Extensive simulations were run to validate the proposed models and their results are worth noticing: In the first-order case an optimum 6 dB gain was obtained for a positive value of $\lambda = 0.45$ and for the fifth-order case an optimum 14 dB gain was obtained for a negative value of $\lambda = -0.205$. Of course this design technique is fully compatible with our previous research results and it could be integrated in the software filter representing the cost function in the Trellis three-levels authoring algorithm, thus increasing furthermore the performance.

Chapter 7

Evaluation of Entropy and Lossless Compression techniques for high bit rate audio

In this chapter we will introduce a solution to the last issue that was brought to attention when we presented the three-levels Trellis Σ-Δ modulation algorithm. In chapter 4 we showed that a side effect of introducing a further symbol to encode the information content was the need of increasing the bit rate of the encoded stream, while retaining the standard symbol rate as used by other two-levels implementations. In fact, also for the standard two-levels encoding systems there is the necessity to fit the large amount of data represented by the DSD to the physical dimension of the optical support used to store it. DVD discs are employed for their large capacity, but even they are rated to host up to 4.7 *GBytes*. If we think of offering the same playback length that we had for CDs (i.e. about 74 minutes) we will end up with:

$$\begin{aligned} SpaceNeeded &= Playback\ Length \cdot OSR \cdot F_s \cdot bit\ per\ symbol \\ &= 74\,minutes \cdot 64 \cdot 48000 \cdot 1\ (or\ 2) \\ &\approx 13.5\,GBytes\ (or\ 27\ Gbytes) \end{aligned} \tag{7.1}$$

So here we will investigate which are the actual techniques used to preserve the information content conveyed by the high rate DSD and in particular we will explain the concepts of entropy coding and lossless compression. In addition we will also show a potential lossy compression technique that is compatible with the hardware architecture we employed in our system design, namely Class D amplification.

7.1 Entropy Evaluation and Coding

Before starting with an introduction on the mathematical instruments used to investigate the possibilities of compressing a data set of information, we can easily share the concept of lossless compression. Actually this is quite related to redundancy of information as seen in many of human communication systems.

Languages in general are redundant and, when used to encode messages, they convey their content giving in the same time the possibility of correcting eventual transmission errors. The musical representation is much less redundant then other languages; indeed in a simple two dimensional plane a musical score can represent both the time and frequency evolutions of the signal. The understated approximation is to represent with the fundamental harmonic (a note) the much more complex timbre of the peculiar instrument part that is written in the musical staff. This to say that in general the information content can be represent by symbols and the set of symbols used can be minimal or redundant.

If a set is minimal trying to reduce it furthermore yields a partial loss of the information content. Otherway we can always try to reduce the symbol alphabet decreasing the space to carry the same amount of information without losing the capability of recreating the exact message that was originally encoded: this approach is lossless compression. In our specific application we have a very high frequency stream of few symbols that are part of two sets: $[-1, +1]$ is the first and $[-1, 0, +1]$ the alternative one. These symbols are representing, in the digital domain, the time and frequency evolution of the musical information. Their related streams are, as we calculated previously, too expensive to be represented without any kind of organization on the physical support.

So we must find a mechanism for compressing them without altering their information content. In information theory there is a metric that describes the amount of redundancy (or the amount or real information) present in a specific language. This metric is called entropy and it is mathematically represented by:

$$H(x) = -\sum_{x} p(x) \cdot log_2 \, p(x) \tag{7.2}$$

where $p(x)$ is the probability mass function of the random variable X, the information content of interest. Entropy is measured in bit [45]. This measure has been used in order to provide some insight on the statistical characteristics of encoded signals and to evaluate the resulting achievable compression. What we were looking for was the chance to use an entropy coder to map the more probable sequences of symbols *before* eventually using any other compression algorithm.

So we can evaluate the entropy of several two or three levels encoded streams of music. Before this we need the first order statistic of these signals ($p(x)$). So we have to collect, for this examples of encoded music, the global probability of each alphabet symbol of the two encoding systems. An immediate result is that for two levels signals $p(-1) = p(+1) = 0.5$. This was already know as our random process is not polarized, thus needing one bit per symbol. Instead the three levels case is different; in fact we already presented that the amount of zero symbols depends on the amplitude of the signal to be encoded. So we can monitor the evolution of the $p(x)$ collecting it for different musical styles samples. We can write a simple algorithm that counts for every symbol that is found in the encoded stream and after reading a complete musical file divide the number of -1,0 and $+1$ for the total number of symbols. This is the first order statistics for a single symbol outcome; then we can collect $p(x)$ also for two consequent symbols and then for three ones. This is done in order to appreciate if there are some sequences of symbols that are more probable than others. In figures 7.1, 7.2 and 7.3 there are the results for the collection of $1, 2$ and 3 symbols given a fixed music sample. To read these plots we have to translate the *numbers* of the $x-$axis into the appropriate ternary sequence representing it, agreeing to the following:

$$n = \sum_k (S_k + 1) \cdot 3^k \tag{7.3}$$

so, for example number $n = -5$ equals to the sequence of $\{-1, 0, -1\}$ symbols.

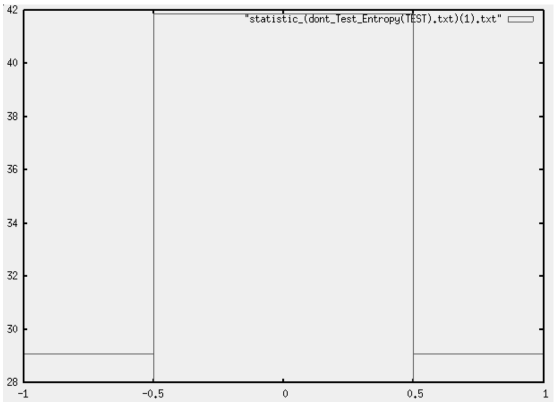


Figure 7.1: Probability percentage of encoding a single symbol from the $[-1, 0, +1]$ alphabet

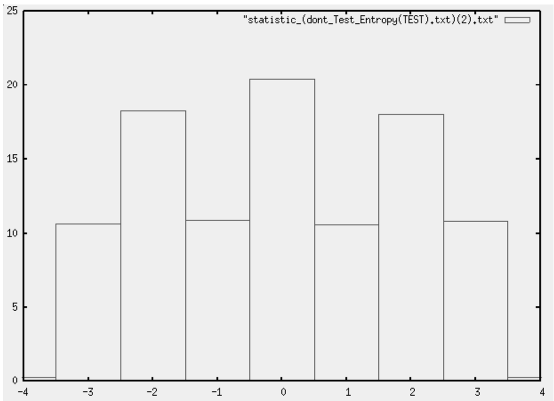


Figure 7.2: Probability percentage of encoding a sequence of two symbols from the $[-1, 0, +1]$ alphabet

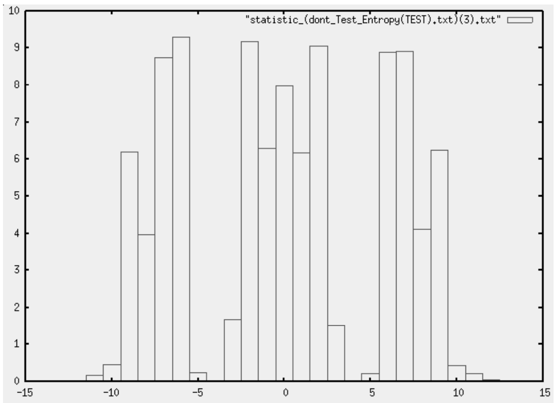


Figure 7.3: Probability percentage of encoding a sequence of three symbols from the $[-1, 0, +1]$ alphabet

We can clearly see that the Trellis algorithm is providing a stream where some sequences are almost forbidden so we aspect to see that the entropy value decreases when evaluating a longer symbol sequence evaluated. This has resulted to be verified and some entropy values extracted from some music excerpt are summarized in table 7.1.

Sample	Outcome Length	Entropy
Music A	1	H(x) = 1.56
Music A	2	H(x) = 1.45
Music A	3	H(x) = 1.30
Music B	1	H(x) = 1.48
Music B	2	H(x) = 1.46
Music C	1	H(x) = 1.52
Music C	2	H(x) = 1.40
Multiple Sin	1	H(x) = 1.28
Multiple Sin	2	H(x) = 1.21

Table 7.1: Entropy evaluated for different excerpt of music and for different length of possible symbols sequences

This procedure also underlines another important feature of the encoded stream. Depending on the kind of music converted there are differences among the possible sequences probabilities. This is clear if we compare, for example, entropy for Music A,B and C samples when evaluated on sequences of length 2. Only a different probability distribution on the possible sequences outcomes can yield a different value for H. So, this suggested to implement a *symbol-entropy* coding. If the sequences are longer than one symbol their outcome probabilities vary depending on the musical content, so we can not choose a fixed map (or look-up table) meaning a fixed entropy coding. This would be winning in some situations, but not in all of them. So the simplest entropy coding was chosen, i.e. to encode only single symbols.

The alphabet has then been mapped following this ratio:

- 0 symbol mapped with bit 0;

- -1 symbol mapped with bits 1 1;
- $+1$ symbol mapped with bits 1 0;

this lead to an symbol-entropy encoded stream that is smaller than the previous three levels one, but still larger than the two levels DSD that requires only one bit per symbol (in our examples the mean value for H was approx. 1.5 bit per symbol). Now there are two other techniques we can implement and use to try to reduce the bitrate needed to convey our information content: the first one is called Pulse Group Modulation, the latter is Linear Prediction Coding.

7.2 Pulse Group Modulation

We already said that in a time window of N samples there are 2^N possible configurations of $[0, 1]$ symbols or 3^N configuration of $[-1, 0, +1]$ symbols. We also presented that in class D amplifier it is possible to use both these alphabets to drive the output power stage. Here we present Pulse Group Modulation (**PGM**) that is a technique sometimes used to reduce the switching frequency of the output stage. The basic concept of this modulation is to use a small time window of length k that slides on the encoded signal and reorder the symbols within it, in order to obtain only k words.

So the sequences [0 0 1 0 1] and [1 0 0 1 0] will be mapped onto the same word: [0 0 0 1 1].

The mean information is preserved and a smaller set of symbol is representing the original signal. The goal is to pre-process the stream and to achieve a better compression ratio using a linear prediction compression afterward. The problem is that there could be an information loss. Authors of a previous work [46] claim that the loss is outside the audio band and the possible compression gain is large, but we showed that actually this transformation is detrimental. We ran this algorithm on a sinusoidal test input with $f_0 = 1\,kHz$. We set the time windows k in the range $[2-8]$ and we evaluated the power spectrum density of the resulting streams. In figure 7.4 it's easy to see that the power spectrum the original sinusoidal signal ($k = 0$) has a much lower noise floor than the other ones that have been pre-processed with the PGM.

So this technique is not usefull to our application but in case we could accept to use a lossy compression algorithm this mechanism is promising, achieving a large amount of compression with almost identical performance in terms of SNR among once the length of the time window is larger than 2 (see the noise floor in figure 7.4). A rough example of possible compression performances is shown in table 7.2.

Once we have seen that the only pre-processing we can use is entropy coding, we can now present the most popular technique, used also in the Super Audio CD standard, that we will tailor for our three levels alphabet modulation, namely linear prediction coding compression.

7.3 LPC Lossless compression

In this section we don't present the whole range of applications of linear prediction coding to engineering, but more simply we'd like to introduce the concept

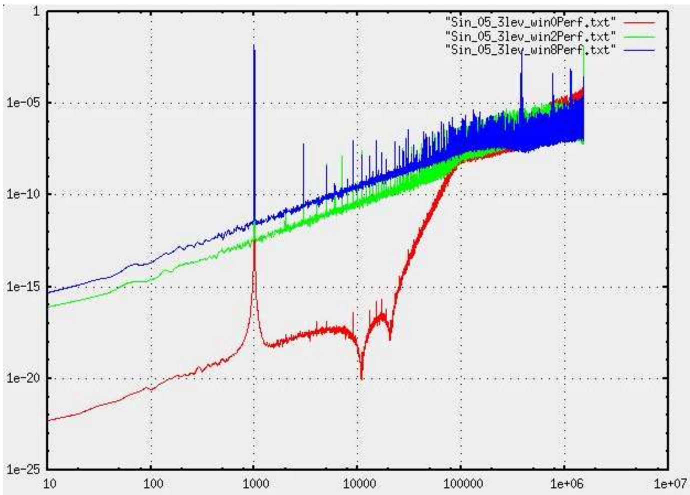


Figure 7.4: Power spectrum density for sinusoidal input pre-processed with PGM, $K = [0, 2, 8]$

Window Size k	Compressed Dimension
0	281.9 kB
2	198.7 kB
3	189.0 kB
4	134.4 kB
8	92.5 kB

Table 7.2: Possible compressed dimension for a music sample (original size 281.9 $kBytes$) pre-processed with PGM and then compressed with a general linear prediction coding algorithm.

behind it and the techniques that are decisive in solving our stated problem. We refer to reference for an introduction on the topic [47] and also for the detailed description of the standard employing this mathematical tool [48] [49].

Here it is important to say that in general, when we deal with stochastic process (that is our model for a flux of information, litterally a stream of bits), whereas it is a message, a spoken language or a song, we can try to analyze it in order to extract some features and then use them to predict its time evolution. These features could be, for example, some patterns that repeat themselves. Obviously if there is some true information content (what a reviewer could define “novelty”) in the process that we are evaluating there should be a limit to the capability of predicting it. So, music is predictable, but not only in the cultural sense that we can immediately think of; we are not talking about the almost always foreseen keynote after the fifth chord in the endings of a symphony movement that leaves to the listener that magic relief, but of the truly predictable stochastic process that actually is music for an *un-educated* listener that is a compression algorithm.

Indeed if we could ever train an algorithm with our cultural knowledge we will probably end up with a machine capable of giving us the best compressed version of a musical passage: the score. Luckily enough software are not so powerful yet, leaving musicians still some pleasure in listening to music trying to deconvolve its beauty.

So we can start presenting the classical approach that in the last decade has evolved resulting in both the Super Audio CD standard for lossless compression, namely Direct Stream Transfer (**DST**), developed by Philips [50] and the MPEG-4 Layer II, developed as international standard by Chiariglione and others [51].

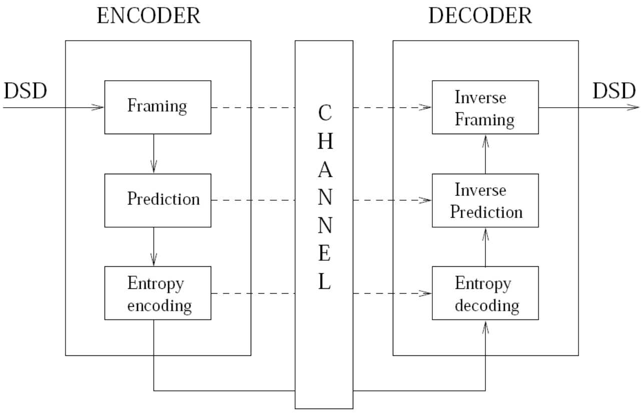


Figure 7.5: Block scheme for the DST lossless compression

In figure 7.5 there is the typical block representation common to all linear prediction coding based compression algorithm. The bit stream (DSD) is preprocessed, i.e. it is framed into specific time blocks, then it is processed by the linear predictor and then compressed using an entropy coder. We don't have to confuse the entropy coder we presented in the previous section with this one. The entropy coder we showed before was dealing with *symbols* and not with bits. Here symbols are already coded with bits and what we want to achieve is to reduce the number of bits required to convey the same information content, preserving the capability of reconstructing the original stream without errors.

We will focus here only in the linear predictor, seeing which are the basics of its functioning and which are the design parameters we can tune to achieve our goal. The basic assumption for a two levels alphabet LPC encoder is to process the stream in order to change the probability of one of the two symbols. Indeed if we still had, after the filtering, a stream where both -1 and $+1$ symbols have the same $p(x)$ we would be still unable to exploit any entropy encoding. For our three levels alphabet, instead, we already shown that the $p(x)$ for each symbol are not equal. Anyway we want to increase the $p(x)$ of one symbol towards the other two.

In figure 7.6 there are some details of the LPC encoder for a two levels stream: the goal is to filter the bitstream b with a realizable filter (note the presence of a z^{-1} delay in the filter) which output is then quantized. The

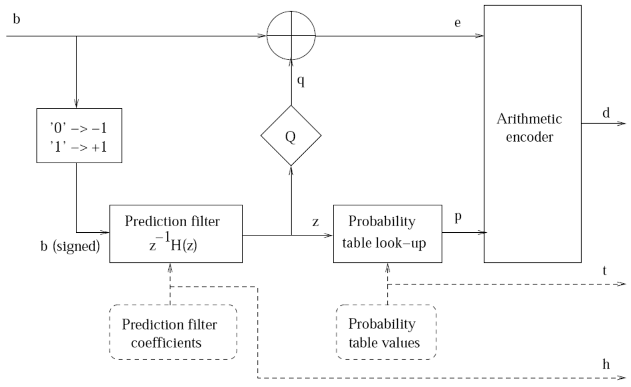


Figure 7.6: Detailed view of the linear predictor block

resulting quantized output q is subtracted to the original stream b, resulting in a stream e whose samples are called *residuals*. The more accurate the prediction, the larger presence of 0 in the resulting stream e, i.e. a symbol probability that is more and more polarized. The stream e exhibit a very large percentage of zeros that could be exploited by the subsequent entropy encoder. Furthermore, the bit stream d, which is is the product of entropy coding of e, together with the prediction filter coefficients h, carry the same information as b. The delay is mandatory to create an encoder that can be time-reversed (thus creating the decoder).

This scheme works with a two symbols alphabet $[-1, +1]$, as seen in fig. 7.6 where the bits contained in b are remapped onto the original symbols. This is done in order to have a simpler quantization step within the predictor; the output of the filter $z^{-1} \cdot H(z)$ is simply truncated to $[-1, +1]$ before being subtracted to b. In our three symbols modulated stream, we have a $[-1, 0, +1]$ alphabet thus requiring a dedicated quantizer; a different threshold setting could reveal a different performance of the encoder, that actually tries to polarize the original stream producing a larger $p(x)$ for the zero symbol as before.

So the real difference between a two and a three levels LPC encoder is the behavior of the quantizer block; after that the resulting stream e could be treated in the same way. For a two levels signal we have:

$$q(n) = \begin{cases} +1 & \text{if } z(n) \geq 0 \\ 0 & \text{if } z(n) < 0 \end{cases}$$

where for a three levels signal we have:

$$q(n)) = \begin{cases} +1 & \text{if } z(n) \geq T_h \\ 0 & \text{if } -T_h < z(n) < T_h \\ -1 & \text{if } z(n) < -T_h \end{cases}$$

while for both of the systems is valid:

$$e(n) = \begin{cases} 0 & \text{if } b(n) = q(n) \\ 1 & \text{if } b(n) \neq q(n) \end{cases}$$

A matlab implementation of this block scheme has been made; a music file, once encoded with two and three levels modulations has been framed and then, chosen a proper threshold T_h, for each frame, the residuals have been enumerated. In figure 7.7 we present the amount of residuals as function of the frames and the length of the prediction filter for the two levels case: Frames go from 0 to 30, while the taps of the filter go up to a factor 8.

We can see that the length of the prediction filter is an important parameter to tune our algorithm; the resulting number of residuals can be almost halved if we increase the prediction taps from 2 to 16 (see figure 7.7).

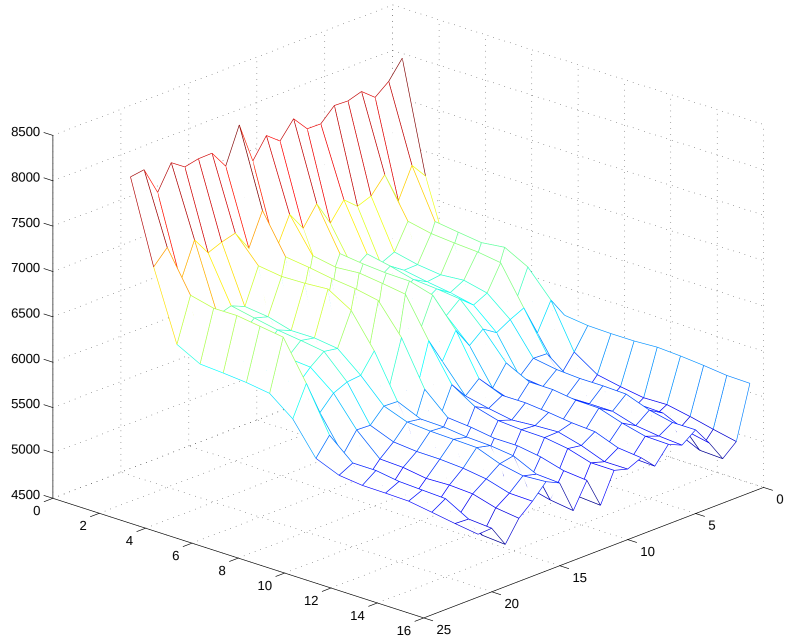


Figure 7.7: Evaluation of the behavior of the residuals varying the predictor filter length; with 16 taps the number of residuals can be almost halved.

It's also interesting to observe that there are plateaus in the plot; as stated in literature [50] this as a good reason to stop increasing the length of the prediction filter beyond a reasonable limit imposed by computational constraints.

In figure 7.8 we present the amount of the residuals as function of the frames and the length of the prediction filter for the three levels case. Here the residuals count is larger than in the two levels case, thus requiring a longer filter. Also the plateau behavior is not so much evident as before.

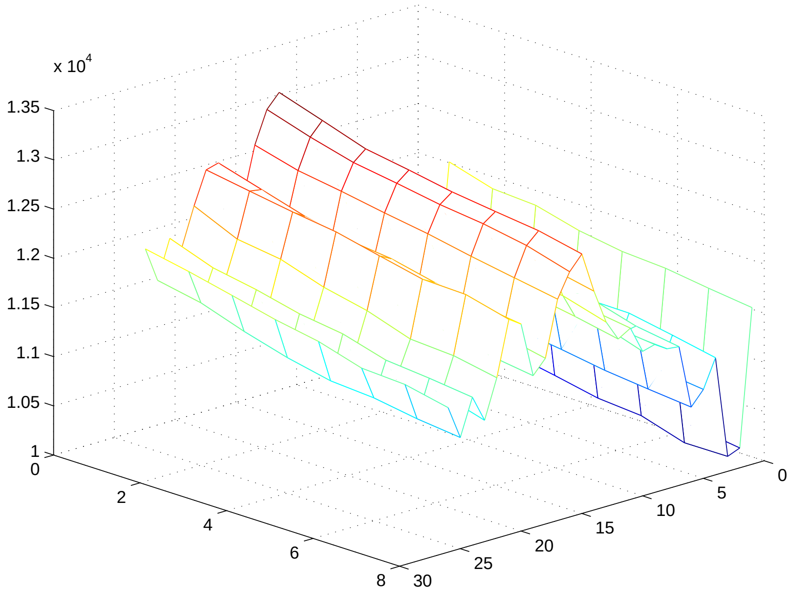


Figure 7.8: Residual for a three levels LPC predictor, with linear predictor filter length varying from 2 to 8.

The immediate conclusion is that we need a longer filter, thus requiring a larger computational effort. This should not be worrying for two different reasons: first the computational power of new PCs is so larger at every technology step that now-a-days these filtering processes are not expensive as they were a decade ago. Second, the computational effort is done only once when preparing the recorded material to be written on the physical support, meaning that the hard part in the encoding/decoding process is intentionally unbalanced in the encoding phase. Decoding is not so demanding and this is way this kind of compression techniques have been used since the introduction of CDs. Indeed also Compact Disc has a lossless coding included in its own standard.

Now we will proceed in presenting the final comparison results for our investigation.

7.4 Compression results comparison

First of all we want to describe the scenario of our comparison: we said that the three levels Trellis modulation achieves a better performance than the two levels one, up to $9\,dB$ better. So if we want to compare the two systems in a fair way we can asks for both of them to have the same quality, requiring an increase of the OSR of the two levels encoding to match the quality of the three levels one. Simulations with sinusoids have been made and they proved that to achieve the same SNR it was necessary to increase the OSR of about 30%, that is an OSR= 84.

According to eq. 7.1 this requires a bit rate equal to $4032000\,bit/sec$ while for the three levels modulation the bitrate is still $6144000\,bit/sec$. So now we can compare this two levels stream with OSR= 84 with:

- a 3 level stream with $OSR = 64$ + a custom 3 levels LPC, indicated as [**3lev**]
- a 3 level stream with $OSR = 64$ + entropy coding + a custom 2 levels LPC, indicated as [**3lev+E.C.**].

Defining the compression ratio as the dimension of the compressed stream divided by the dimension of the original one (thus smaller is better) we can present the results in table 7.3.

Signal	Original dim.	Compressed dim.	Comp. ratio	Bench. with 7zip
Sinusoid [1sec]				
2 levels	452.2 kB	254.1 kB	[0.56]	305.2 kB
3 levels	689.1 kB	227.1 kB	[0.33]	290.7 kB
Music A [1sec]				
2 levels	904.8 kB	527.5 kB	[0.58]	596.3 kB
3 levels	1300 kB	536.9 kB	[0.41]	645.7 kB
3 levels + E.C.	1080 kB	528.2 kB	[0.49]	645.7 kB
Music B [1sec]				
2 levels	906.6 kB	514.4 kB	[0.57]	566.2 kB
3 levels	1300 kB	545.4 kB	[0.42]	639.9 kB
3 levels + E.C.	1000 kB	511.8 kB	[0.51]	639.9 kB

Table 7.3: Compression results comparison table

We can now appreciate the pre-processing with entropy coding applied to the three levels modulation. In fact using this technique we can reach the same performance of the two levels LPC encoder. The final result is that we can achieve the same compression ratio achievable with the standard DST using our three levels modulation followed by entropy coding and by a custom two levels LPC encoding.

Furthermore in table 7.3 there is also a benchmark with a standard general purpose compression software that achieves always worst results when compared to our dedicated LPC encoder. This is due to the fact that in general these general purpose compressors don't have any previous knowledge of the statistical behavior of the stream to be compressed.

Conclusion

In this thesis an innovative step in high quality audio has been introduced and illustrated in detail. The original goal was to investigate new technologies in order to achieve a better quality for audio signals. A novel three-levels digital-to-digital Trellis encoding algorithm has been derived from state of the art SuperAudio CD standard and developed in order to increase its quality even further, while being capable of a drastic reduction of the power consumption needs for the final amplifier stage.

The performance improvement has been measured as a reduction of the Total Harmonic Distortion and a sensible increase of the Signal-to-Noise ratio: in a benchmark with a 5^{Th} order two-levels Trellis Σ-Δ converter the new three level encoder showed a SNR up to 8.2 dB higher. Furthermore a heuristic approach to the encoding algorithm has been developed to reduce the computational complexity: now the final encoding software shows a CPU usage that depends linearly to the extension of the signal state space, while previous algorithms showed an exponential dependence. To complete the new prosed technology a novel lossless compression process has been tailored to produce a final digital stream that fits the same storage requirements of the standard 1-Bit high quality audio. This lossless encoder is the combination of an entropy coder followed by a properly designed Linear Prediction Coder that exploits the statistical knowledge of the three-levels stream. Again in a benchmark with the SuperAudio CD standard lossless compression we showed a significant increase of the compression ratio up to 18% better.

So the final product of this thesis is an encoding algorithm that produces a higher audio quality stream while retaining a lower CPU usage to produce it and a smaller compressed file to store it.

In this thesis we also presented another innovative technology. The application of frequency-warping theory to Σ-Δ filter design: starting from psychoacoustic remarks we investigated frequency-warping and we presented a new filter design approach that yields a Σ-Δ loop filter able to reduce furthermore the noise in the audio band. We illustrated two different topologies for both low-order and high-order Σ-Δ filters, showing in both cases a sensible performance increase. This culminated with an optimal +14 dB in Signal-to-Noise ratio in a benchmark with a standard 5^{Th} order Σ-Δ modulator.

Appendix A

Calculus for high-order frequency-warped SDM CIFF topology II

The aim is to extract a factor $(1-\lambda)$ from the high-order warped L_1 loop filter transfer function given in (6.30). By substituting (6.29) in (6.30),

$$\begin{aligned}\tilde{L}_1^{(N)}(z,\lambda) &= \frac{\displaystyle\sum_{k=1}^{N}\alpha_k(-\lambda)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}-\sum_{k=1}^{N}\alpha_k\Big(\frac{z^{-1}-\lambda}{1-z^{-1}}\Big)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{(z^{-1}-\lambda)^2}{(1-z^{-1})^2-\gamma_j(z^{-1}-\lambda)^2}}{1-\alpha_0}\\ &= \frac{1}{1-\alpha_0}\sum_{k=1}^{N}\alpha_k\Bigg[(-\lambda)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}-\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^{p(k)}}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\Bigg].\end{aligned} \tag{7.4}$$

The two terms between square brackets in (7.4) can be rewritten with a common denominator as follows:

$$\tilde{L}_1^{(N)}(z,\lambda) = \frac{1}{1-\alpha_0}\sum_{k=1}^{N}\alpha_k\frac{\Big(-\lambda\dfrac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}\displaystyle\prod_{j=1}^{\frac{k-p(k)}{2}}\lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)-\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2)}{\Big(\dfrac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}\displaystyle\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2)\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)}. \tag{7.5}$$

The leftmost product term in the numerator can be expanded as follows:

$$\prod_{j=1}^{\frac{k-p(k)}{2}}\lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big) = \prod_{j=1}^{\frac{k-p(k)}{2}}\Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-1+1-\gamma_j\lambda^2\Big) \tag{7.6}$$

$$= \Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-1\Big)\prod_{j=2}^{\frac{k-p(k)}{2}}\lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)+(1-\gamma_1\lambda^2)\prod_{j=2}^{\frac{k-p(k)}{2}}\Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\lambda^2\Big).$$

The rightmost product term in the latter equation can be expanded in a similar way, and this process can be repeated to obtain

$$\prod_{j=1}^{\frac{k-p(k)}{2}} \lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big) \tag{7.7}$$

$$=\Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-1\Big)\sum_{l=1}^{\frac{k-p(k)}{2}}\Big(\prod_{j=1}^{l-1}(1-\gamma_j\lambda^2)\prod_{j=l+1}^{\frac{k-p(k)}{2}}\lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)\Big)+\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2).$$

which after substitution in (7.5) leads to

$$\tilde{L}_1^{(N)}(z,\lambda)=\frac{1}{1-\alpha_0}\sum_{k=1}^{N}\alpha_k\left\{\frac{\Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}-1\Big)\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2)}{\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2)\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)}\right. \tag{7.8}$$

$$\left.+\frac{\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}\Big(\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-1\Big)\sum_{l=1}^{\frac{k-p(k)}{2}}\Big(\prod_{j=1}^{l-1}(1-\gamma_j\lambda^2)\prod_{j=l+1}^{\frac{k-p(k)}{2}}\lambda^2\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)\Big)}{\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}\prod_{j=1}^{\frac{k-p(k)}{2}}(1-\gamma_j\lambda^2)\Big(\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-\gamma_j\Big)}\right\}.$$

Recognizing that

$$\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2-1=\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}-1\Big)\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}+1\Big), \tag{7.9}$$

$$\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^{p(k)}-1=p(k)\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}-1\Big) \tag{7.10}$$

and

$$-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}-1=(\lambda-1)\frac{z^{-1}}{z^{-1}-\lambda} \tag{7.11}$$

(7.8) can be rewritten as follows:

$$\tilde{L}_1^{(N)}(z,\lambda)=\Big(\frac{\lambda-1}{1-\alpha_0}\Big)\frac{z^{-1}}{z^{-1}-\lambda}\sum_{k=1}^{N}\alpha_k\left\{\frac{p(k)}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^{p(k)}}\prod_{j=1}^{\frac{k-p(k)}{2}}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\right. \tag{7.12}$$

$$\left.+(-\lambda)^{p(k)}\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}+1\Big)\sum_{l=1}^{\frac{k-p(k)}{2}}\prod_{j=l}^{\frac{k-p(k)}{2}}\frac{1}{1-\gamma_j\lambda^2}\prod_{j=1}^{l}\frac{\lambda^{2(\frac{k-p(k)}{2}-l)}}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\right\}.$$

The expression in (7.12) can be organized in a different way by changing the order of the summations and by grouping the product terms involving $\big(\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j\big)^{-1}$ that have equal upper limits, i.e.,

$$\tilde{L}_1^{(N)}(z,\lambda)=\Big(\frac{\lambda-1}{1-\alpha_0}\Big)\frac{z^{-1}}{z^{-1}-\lambda}\left\{\alpha_1\frac{z^{-1}-\lambda}{1-z^{-1}}\right. \tag{7.13}$$

$$\left.+\sum_{l=1}^{\frac{N-1}{2}}\Big(\alpha_{2l+1}\frac{z^{-1}-\lambda}{1-z^{-1}}+\sum_{k=2l}^{N}\alpha_k\Big(-\lambda\frac{1-z^{-1}}{z^{-1}-\lambda}+1\Big)\frac{(-\lambda)^{p(k)}}{\lambda^2}\prod_{j=l}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}\Big)\prod_{j=1}^{l}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\right\}.$$

Finally, a factor $\frac{z^{-1}-\lambda}{1-z^{-1}}$ can be extracted from the braces in (7.13) to obtain

$$\tilde{L}_1^{(N)}(z,\lambda) = \Big(\frac{\lambda-1}{1-\alpha_0}\Big)\frac{z^{-1}}{1-z^{-1}}\left\{\alpha_1+\sum_{l=1}^{\frac{N-1}{2}}\left(\alpha_{2l+1}+\beta_l\frac{1-z^{-1}}{z^{-1}-\lambda}-\lambda\beta_l\Big(\frac{1-z^{-1}}{z^{-1}-\lambda}\Big)^2\right)\prod_{j=1}^{l}\frac{1}{\big(\frac{1-z^{-1}}{z^{-1}-\lambda}\big)^2-\gamma_j}\right\} \tag{7.14}$$

with

$$\beta_l = \sum_{k=2l}^{N}\alpha_k\frac{(-\lambda)^{p(k)}}{\lambda^2}\prod_{j=l}^{\frac{k-p(k)}{2}}\frac{\lambda^2}{1-\gamma_j\lambda^2}. \tag{7.15}$$

Bibliography

[1] S.P. Lipshitz, J. Vanderkooy: *Pulse-Code Modulation:An Overview*, Journal of Audio Engineering Society, Volume 52, Number 3, March 2004.

[2] J.R. Stuart: *Coding for High-Resolution Audio Systems*, Journal of Audio Engineering Society, Volume 52, Number 3, March 2004.

[3] D. Reefman, E. Janssen: *1-Bit audio:An Overview*, Journal of Audio Engineering Society, Volume 52, Number 3, March 2004.

[4] E. Gaalaas: *Class D Audio Amplifiers: What, Why, and How*, Volume 40, June 2006, http://www.analog.com/library/analogDialogue/archives/40-06/class_d.html.

[5] E. Knapen, D. Reefman, E. Janssen, F. Bruekers: *Lossless Compression of One-Bit Audio*, Journal of Audio Engineering Society, Volume 52, Number 3, March 2004.

[6] M.A. Gerzon, P.G. Craven, J.R. Stuart, M.J. Law, R.J. Wilson: *The MLP Lossless Compression System for PCM Audio*, Journal of Audio Engineering Society, Volume 52, Number 3, March 2004.

[7] H. Inose, Y. Yasuda, J. Marakami: *A telemetering system by code modulation, delta-sigma modulation* IRE Trans. on Space, Electronics and Telemetry, SET-8, pp. 204-209, Sept. 1962.

[8] R. Steele: *Delta Modulation Systems* Pentech Press, London, England, 1975.

[9] N. Scheinberg, D. Schilling: *Techniques for correcting transmission error in video adaptive delta-modulation channels* IEEE Trans. Commun.,pp. 1064-1070, Sept. 1977.

[10] S.R. Norsworthy, R. Schreier, G.C. Temes: *Delta-Sigma Data Converters - Theory, Design and Simulation* IEEE Wiley Interscience Press, 2001.

[11] N. Ahmed, T. Natarajan: *Discrete-Time Signals and Systems* Prentice-Hall, Englewood Cliffs, NJ, 1983.

[12] R. Schreier: *An empirical study of high-order single-bit Delta-Sigma modulators* IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, Vol.40, No.8, August 1993.

[13] J.C. Candy: *A use of double integration in Sigma-Delta modulation* IEEE Trans. on Communications, Vol. COM-33, No. 3, pp.249-258, March 1985.

[14] R. Koch, et al: *A 12-bit Sigma-Delta analogto-digital converter with a 15-MHz clock rate* IEEE J. of Solid-State Circuits, Vol. SC-21, No.6, pp. 1003-1010, Dec. 1986.

[15] B. Boser, B. Wooley: *Quantization error spectrum of Sigma-Delta modulators* Proc. International Symposium on Circuits and Systems, pp. 2331-2334, June, 1988.

[16] W.L. Lee, C.G. Sodini: *A topology for higher order interpolative coders* Proc. International Symposium on Circuits and Systems, pp. 459-462, May 1987.

[17] D.R. Welland, et al: *A stereo 16-bit delta-sigma A/D converter for digital audio* Proc. the 85th convention of Audio Engineering Society, Vol. 2724 (H-12), Los Angeles, CA, Nov. 3-6, 1988.

[18] W. Bennett: *Spectra of quantized signals* Bell Syst. Tech. J., Vol. BSTJ-27, pp. 446-472, July 1948.

[19] B. Widrow: *A study of rough amplitude quantization by means of Nyquist sampling theory* IRE Trans. Circuit Theory, Vol. CT-3, pp. 266-276, Dec. 1956.

[20] B. Katz: *Mastering Audio: The Art and the Science* Focal Press

[21] E. Janssen, D. Reefman: *Advances in Trellis based SDM structures* AES 115th Convention Paper, 2003 October 10-13 N.Y.

[22] A.J. Viterbi: *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm* IEEE Transactions on information theory, vol.IT-13, No. 2, April 1967.

[23] H. Kato: *Trellis Noise-Shaping Converters and 1-bit digital audio* AES 112th Convention Paper 5615, 2002 May 10-13, Munich.

[24] M.K. Kazimierczuk, W.Szaraniec: *Class-D zero-voltage-switching inverter with only one shunt capacitor* IEE Proceedings-B, Vol. 139, No.5, September 1992.

[25] E. Gaalaas: *Class D audio amplifiers: what, why and how* Analog Dialogue 40-06, June 2006, http://www.analog.com/analogdialogue.

[26] J. Cerezo: *How to choose MOSFET for Class D amplifiers* International Rectifier, Audio DesignLine.

[27] H.S. Black: *Modulation Theory* Van Nostrand Company 1953.

[28] K. Nielsen: *A review and comparison of Pulse Width Modulation (PWM) methods for analog and digital input switching power amplifiers* AES 102nd Convention, 1997 March 22-25, Munich.

[29] R. W. Adams and R. Schreier: *Delta-Sigma Data Converters: Theory, Design, and Simulation - Ch.4 Stability Theory for* $\Delta\Sigma$ *Modulators.* IEEE Press,Piscataway, NJ, 1997.

[30] R. W. Adams: *Delta-Sigma Data Converters: Theory, Design, and Simulation - Ch.5 The Design of High-Order Single-Bit* $\Delta\Sigma$ *ADCs* IEEE Press, Piscataway, NJ, 1997.

[31] M. A. Al-Alaoui and R. Ferzli: *An enhanced first-order sigma-delta modulator with a controllable signal-to-noise ratio* Transaction on Circuit and Systems, Vol 53, No.3, pp. 634-643, March 2006.

[32] L. Ausiello and T. van Waterschoot and M. Moonen: *A first-order frequency-warped sigma delta modulator with improved signal-to-noise ratio* EUSIPCO 2008, Lausanne, Switzerland, August 2008.

[33] P. M. Aziz and H. V. Sorensen and J. van der Spiegel: *An overview of sigma-delta converters* Signal Processing Magazine, Vol.13, No.1, pp. 61-84, January 1996.

[34] J. Candy and G. Temes: *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation* IEEE Press, Piscataway, NJ, 1992.

[35] A. Härmä and M. Karjalainen and L. Savioja and V. Välimäki and U. K. Laine and J. Huopaniemi: *Frequency-warped signal processing for audio applications* Journal of Audio Engineering Society, Vol.48, No.11, pp. 1011-1031, November 2000.

[36] A. Härmä: *Implementation of frequency-warped recursive filters* Signal Processing, Vol.80, No.3, pp. 543-548, March 2000.

[37] A. Härmä and U. K. Laine: *A comparison of warped and conventional linear predictive coding* IEEE Transaction on Speach and Audio Processing, Vol.9, No.5, pp. 579-588, July 2001.

[38] M. Karjalainen and A. Härmä and U. K. Laine: *Realizable warped IIR filters and their properties* ICASSP97, Vol.3, pp. 2005-2008, Munich Germany, 1997.

[39] A. V. Oppenheim and D. H. Johnson and K. Steiglitz: *Computation of spectra with unequal resolution using the fast Fourier transform* ProcIEEE, Vol.59, No.2, pp. 299-301, February 1971.

[40] R. Schreier: *An empirical study of high-order single-bit delta-sigma modulators* TransCSIIADSP, Vol40, No.8, 461-466, August 1993.

[41] R. Schreier and G. C. Temes: *Understanding Delta-Sigma Data Converters* Wiley-IEEE Press, New York, 2005.

[42] J. O. Smith and J. S. Abel: *Bark and ERB bilinear transforms* IEEE Transaction on Speach and Audio Processing, Vol.7, No.6, pp. 697-708, November 1999.

[43] K. Steiglitz: *A note on variable recursive digital filters* IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-28, pp. 111-112, February 1980.

[44] H. W. Strube: *Linear prediction on a warped frequency scale* Journal of Acoustical Society of America, Vol.68, No.4, pp. 1071-1076, October 1980.

[45] C.E. Shannon: *A mathematical theory of communication* Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[46] J.D. Reiss, M.B. Sandler: *Efficient compression of oversampled 1-bit audio signals* AES 111th Convention, 2001 September 2124 New York, US.

[47] J. Makhoul: *Linear Prediction: A Tutorial Review* IEEE Proc., VOL. 63, N0. 4, April 1975.

[48] E. Knapen, D. Reefman, E. Janssen, and F. Bruekers: *Lossless compression of one-bit audio* Journal of Audio Engineering Society, No.52, pp. 190-199, 2004.

[49] L. Chiariglione: *ISO/IEC JTC1/SC29/WG11 N7707* http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm, MPEG2005.

[50] E. Janssen, D. Reefman: *DSD compression for recent ultra high quality 1-bit coders* AES 118th Convention, 2005 May 28-31 Barcelona, Spain.

[51] L. Chiariglione: *ISO/IEC JTC1/SC29/WG11 N4668* http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.

Contents